

Titre: Segmentation mutuelle d'objets d'intérêt dans des séquences
Title: d'images stéréo multispectrales

Auteur: Pierre-Luc St-Charles
Author:

Date: 2018

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: St-Charles, P.-L. (2018). Segmentation mutuelle d'objets d'intérêt dans des
Citation: séquences d'images stéréo multispectrales [Thèse de doctorat, École
Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/3070/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/3070/>
PolyPublie URL:

**Directeurs de
recherche:** Guillaume-Alexandre Bilodeau, & Robert Bergevin
Advisors:

Programme: Génie informatique
Program:

UNIVERSITÉ DE MONTRÉAL

SEGMENTATION MUTUELLE D'OBJETS D'INTÉRÊT DANS DES SÉQUENCES
D'IMAGES STÉRÉO MULTISPECTRALES

PIERRE-LUC ST-CHARLES
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE INFORMATIQUE)
AVRIL 2018

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

SEGMENTATION MUTUELLE D'OBJETS D'INTÉRÊT DANS DES SÉQUENCES
D'IMAGES STÉRÉO MULTISPECTRALES

présentée par : ST-CHARLES Pierre-Luc

en vue de l'obtention du diplôme de : Philosophiæ Doctor

a été dûment acceptée par le jury d'examen constitué de :

M. DAGENAIS Michel, Ph. D., président

M. BILODEAU Guillaume-Alexandre, Ph. D., membre et directeur de recherche

M. BERGEVIN Robert, Ph. D., membre et codirecteur de recherche

M. ALOISE Daniel, Ph. D., membre

M. PEDERSOLI Marco, Ph. D., membre externe

REMERCIEMENTS

Je tiens d’abord à exprimer ma reconnaissance et ma profonde gratitude envers mon directeur de recherche, Guillaume-Alexandre Bilodeau, qui a su me guider à travers de trop nombreuses impasses, et qui a toujours été présent pour discuter autant de méthodes que de météo. De plus, je remercie profondément mon codirecteur de recherche, Robert Bergevin, pour son immense aide lors de la rédaction de nos nombreuses publications, et pour ses encouragements répétés au fil des ans.

Je remercie le Conseil de Recherches en Sciences Naturelles et en Génie du Canada (CRNSG), le Fonds de Recherche du Québec – Nature et Technologies (FRQ-NT), et la Fondation de Polytechnique pour leur soutien financier lors de mes études. Merci aussi au CRIM qui m’a permis de travailler à temps partiel sur de nombreux projets de vision depuis 2011.

Par ailleurs, je souhaite remercier de façon globale tous les chercheurs qui publient leur code source : ce travail n’aurait jamais pu être réalisé sans la contribution de vos idées et de votre temps libre.

Sur une note plus personnelle, je remercie tous mes collègues du laboratoire (actuels, anciens, et très anciens!) qui ont transformé toutes ces années en pur plaisir. Un gros merci aussi à ma famille, qui a toléré mon travail excessif lors de certaines de nos fins de semaines ensemble, et plus particulièrement à mon frère et colocataire Jean-Christophe, qui a dû écoper de mon humeur massacrant dans les moments plus sombres.

Je remercie finalement les membres de mon jury, qui ont eu l’amabilité d’examiner ce travail au printemps 2018.

RÉSUMÉ

Les systèmes de vidéosurveillance automatisés actuellement déployés dans le monde sont encore bien loin de ceux qui sont représentés depuis des années dans les œuvres de science-fiction. Une des raisons derrière ce retard de développement est le manque d'outils de bas niveau permettant de traiter les données brutes captées sur le terrain. Le pré-traitement de ces données sert à réduire la quantité d'information qui transige vers des serveurs centralisés, qui eux effectuent l'interprétation complète du contenu visuel capté. L'identification d'objets d'intérêt dans les images brutes à partir de leur mouvement est un exemple de pré-traitement qui peut être réalisé. Toutefois, dans un contexte de vidéosurveillance, une méthode de pré-traitement ne peut généralement pas se fier à un modèle d'apparence ou de forme qui caractérise ces objets, car leur nature exacte n'est pas connue d'avance. Cela complique donc l'élaboration des méthodes de traitement de bas niveau.

Dans cette thèse, nous présentons différentes méthodes permettant de détecter et de segmenter des objets d'intérêt à partir de séquences vidéo de manière complètement automatisée. Nous explorons d'abord les approches de segmentation vidéo monoculaire par soustraction d'arrière-plan. Ces approches se basent sur l'idée que l'arrière-plan d'une scène peut être modélisé au fil du temps, et que toute variation importante d'apparence non prédite par le modèle dévoile en fait la présence d'un objet en intrusion. Le principal défi devant être relevé par ce type de méthode est que leur modèle d'arrière-plan doit pouvoir s'adapter aux changements dynamiques des conditions d'observation de la scène. La méthode conçue doit aussi pouvoir rester sensible à l'apparition de nouveaux objets d'intérêt, malgré cette robustesse accrue aux comportements dynamiques prévisibles. Nous proposons deux méthodes introduisant différentes techniques de modélisation qui permettent de mieux caractériser l'apparence de l'arrière-plan sans que le modèle soit affecté par les changements d'illumination, et qui analysent la persistance locale de l'arrière-plan afin de mieux détecter les objets d'intérêt temporairement immobilisés. Nous introduisons aussi de nouveaux mécanismes de rétroaction servant à ajuster les hyperparamètres de nos méthodes en fonction du dynamisme observé de la scène et de la qualité des résultats produits.

Par la suite, nous étudions les manières de recalibrer des données provenant de paires d'images multispectrales afin d'améliorer la qualité des masques de segmentation générés. L'utilisation de différentes modalités d'imagerie permet essentiellement d'augmenter le contraste dans les régions d'images où l'arrière-plan de la scène est trop similaire aux objets d'intérêt pour que ces derniers soient mieux identifiés. Par contre, le recalage d'images multispectrales n'est pas

une opération triviale, car la mise en correspondance de points entre deux images qui ne se ressemblent pas visuellement est un problème paradoxal. Nous abordons d’abord ce problème à l’aide d’une approche de mise en correspondance de contours d’objets provenant des résultats obtenus par segmentation vidéo. Une première méthode est alors formulée permettant de recalibrer le contenu d’images multispectrales pour des scènes planaires. Cette méthode est ensuite généralisée aux scènes non-planaires, et combinée à une méthode de segmentation mutuelle d’images basée sur l’intégration des données multispectrales. La solution obtenue permet ainsi d’optimiser itérativement le recalage des données multispectrales à l’aide de points de contours, et la segmentation des objets d’intérêt à l’aide des données recalées. Les résultats finaux recueillis après la convergence du processus d’optimisation sont alors meilleurs que ceux qui peuvent être obtenus par une approche de segmentation directe ou en cascade.

Nous évaluons la performance des différentes méthodes proposées dans ce travail sur de nombreux ensembles de données, et nous démontrons que celles-ci surpassent largement l’état de l’art en segmentation vidéo sans supervision. Il est aussi démontré que ces méthodes sont assez robustes pour être utilisées dans des cas réels où les conditions d’acquisition ne sont pas du tout adéquates, et lorsque l’initialisation des modèles doit être effectuée malgré la présence de bruit dans les données. Le code source de toutes les méthodes développées ainsi que les données utilisées dans nos expériences ont été publiés en ligne dans le but de stimuler le développement de nouvelles méthodes et pour encourager la recherche reproductible.

ABSTRACT

The automated video surveillance systems currently deployed around the world are still quite far in terms of capabilities from the ones that have inspired countless science fiction works over the past few years. One of the reasons behind this lag in development is the lack of low-level tools that allow raw image data to be processed directly in the field. This preprocessing is used to reduce the amount of information transferred to centralized servers that have to interpret the captured visual content for further use. The identification of objects of interest in raw images based on motion is an example of a preprocessing step that might be required by a large system. However, in a surveillance context, the preprocessing method can seldom rely on an appearance or shape model to recognize these objects since their exact nature cannot be known exactly in advance. This complicates the elaboration of low-level image processing methods.

In this thesis, we present different methods that detect and segment objects of interest from video sequences in a fully unsupervised fashion. We first explore monocular video segmentation approaches based on background subtraction. These approaches are based on the idea that the background of an observed scene can be modeled over time, and that any drastic variation in appearance that is not predicted by the model actually reveals the presence of an intruding object. The main challenge that must be met by background subtraction methods is that their model should be able to adapt to dynamic changes in scene conditions. The designed methods must also remain sensitive to the emergence of new objects of interest despite this increased robustness to predictable dynamic scene behaviors. We propose two methods that introduce different modeling techniques to improve background appearance description in an illumination-invariant way, and that analyze local background persistence to improve the detection of temporarily stationary objects. We also introduce new feedback mechanisms used to adjust the hyperparameters of our methods based on the observed dynamics of the scene and the quality of the generated output.

Next, we study how the registration of multispectral data sources can improve the quality of video segmentation results. In short, using different imaging modalities allows us to essentially increase the contrast in image regions where the observed background is too similar to the objects of interest, which then helps us identify these objects more easily. However, the registration of multispectral images is not a trivial task, as the matching of points between two images that are visually dissimilar creates a paradox. We first address this issue using a registration approach that relies on the matching of object contour points obtained via video

segmentation. A method is then proposed to tackle the registration of multispectral image data in planar scenes based on this strategy. After that, we generalize this method to non-planar scenes, and combine it with a mutual segmentation method based on multispectral data integration. This combined solution allows us to iteratively optimize multispectral data registration using object contour points and object segmentation using the registered multispectral data. The final results obtained after convergence then outperform those obtained via direct or cascaded approaches.

We evaluate the performance of the different methods proposed in this work on numerous datasets, and show that these methods largely outperform the state of the art in unsupervised video segmentation. We also show that these methods are robust enough to be used in real scenarios where capture conditions are not ideal, and where model initialization may need to be done despite the presence of noise in the input data. The source code for all developed methods as well as the datasets used for our evaluations have been published online to stimulate the development of new methods and to encourage reproducible research.

TABLE DES MATIÈRES

REMERCIEMENTS	iii
RÉSUMÉ	iv
ABSTRACT	vi
TABLE DES MATIÈRES	viii
LISTE DES TABLEAUX	xii
LISTE DES FIGURES	xiii
LISTE DES SIGLES ET ABRÉVIATIONS	xx
CHAPITRE 1 INTRODUCTION	1
1.1 Définitions et concepts de base	2
1.2 Éléments de la problématique	4
1.2.1 Détection et segmentation d’objets par modélisation d’arrière-plan . .	4
1.2.2 Recalage de paires d’images multispectrales	5
1.2.3 Segmentation mutuelle de paires d’images	6
1.2.4 Paradoxe : recalage et segmentation mutuelle	7
1.3 Objectifs de recherche	7
1.4 Contributions	8
1.5 Plan du mémoire	10
CHAPITRE 2 REVUE DE LITTÉRATURE	11
2.1 Segmentation vidéo	11
2.1.1 Soustraction d’arrière-plan par modélisation paramétrique	13
2.1.2 Soustraction d’arrière-plan par modélisation non-paramétrique	13
2.1.3 Mécanismes de contrôle et de rétroaction	16
2.1.4 Segmentation par décomposition globale	16
2.1.5 Segmentation par apprentissage profond	17
2.2 Recalage d’images multispectrales	17
2.2.1 Mise en correspondance par mesures de similarité locales	19
2.2.2 Mise en correspondance par représentation de haut niveau	20

2.2.3	Modélisation de la transformation de recalage	21
2.3	Segmentation mutuelle et cosegmentation d’images	22
2.3.1	Segmentation multi-instance (cosegmentation)	22
2.3.2	Segmentation instance unique (segmentation mutuelle)	24
2.3.3	Applications dans le domaine multispectral	25
CHAPITRE 3	SURVOL DES ARTICLES	27
3.1	Exploration des principes de modélisation et d’ajustements dynamiques . . .	27
3.2	Soustraction d’arrière-plan par modélisation de persistance	28
3.3	Exploration du recalage par mise en correspondance de contours	28
3.4	Recalage et segmentation mutuelle de paires d’images multispectrales	29
CHAPITRE 4	ARTICLE 1 : SUBSENSE : A UNIVERSAL CHANGE DETECTION METHOD WITH LOCAL ADAPTIVE SENSITIVITY	30
4.1	Introduction	30
4.2	Related Work	33
4.3	Methodology	35
4.3.1	Pixel-level modeling	36
4.3.2	Change Detection via Sample Consensus	39
4.3.3	Background Monitoring and Feedback Scheme	43
4.3.4	Further Details	48
4.4	Experiments	48
4.4.1	CDnet 2012	50
4.4.2	CDnet 2014	54
4.4.3	Processing speed	56
4.5	Conclusion	57
CHAPITRE 5	ARTICLE 2: UNIVERSAL BACKGROUND SUBTRACTION USING WORD CONSENSUS MODELS	59
5.1	Introduction	59
5.2	Related Work	62
5.3	Methodology	65
5.3.1	Word consensus for pixel-level modeling	65
5.3.2	Word consensus for frame-level modeling	71
5.3.3	Pixel-level feedback	73
5.3.4	Model adaptability	76
5.4	Experiments	77

5.4.1	CDnet 2012	78
5.4.2	CDnet 2014	82
5.4.3	Processing speed and memory footprint	84
5.5	Conclusion	86
CHAPITRE 6 ARTICLE 3: ONLINE MULTIMODAL VIDEO REGISTRATION BASED		
	ON SHAPE MATCHING	87
6.1	Introduction	87
6.2	Related Work	89
6.3	Proposed method	90
6.3.1	Shape extraction	91
6.3.2	Contour points description and matching	91
6.3.3	Correspondence reservoir and voting	92
6.3.4	Homography smoothing	94
6.4	Evaluation	95
6.5	Conclusion	100
CHAPITRE 7 ARTICLE 4: ONLINE MUTUAL FOREGROUND SEGMENTATION		
	FOR MULTISPECTRAL STEREO VIDEOS	101
7.1	Introduction	101
7.2	Previous Work	104
7.3	Proposed Approach	108
7.3.1	Stereo Registration Model	109
7.3.2	Segmentation Model	113
7.3.3	Inference and Implementation Details	117
7.4	Experiments	118
7.4.1	Evaluation Methodology	118
7.4.2	VAP 2016 Dataset	120
7.4.3	Bilodeau <i>et al.</i> 2014 Dataset	121
7.4.4	Parameters and Ablation Study	122
7.4.5	LITIV 2018 Dataset	124
7.5	Conclusion	126
CHAPITRE 8 DISCUSSION GÉNÉRALE		
8.1	Segmentation vidéo par soustraction d'arrière-plan	130
8.2	Recalage de paires d'images multispectrales	131
8.3	Segmentation mutuelle d'images multispectrales	132

8.4	Efficacité du pipeline de traitement pour usage temps-réel	133
CHAPITRE 9 CONCLUSION		135
9.1	Recommandations pour travaux futurs	136
RÉFÉRENCES		138

LISTE DES TABLEAUX

Tableau 2.1	Taxonomie des approches de segmentation vidéo par soustraction d'arrière-plan	12
Table 4.1	Average performance comparison of different model configurations on the 2012 CDnet dataset	51
Table 4.2	Complete results for SuBSENSE on the 2012 CDnet dataset	51
Table 4.3	Overall and per-category F-Measure comparisons, CDnet 2012 dataset	52
Table 4.4	Complete results for SuBSENSE on the 2014 CDnet dataset	53
Table 4.5	Overall and per-category F-Measure comparisons, CDnet 2014 dataset	53
Table 4.6	Average performance comparison of different methods on the 2012 CDnet dataset	55
Table 4.7	Average performance comparison of different methods on the 2014 CDnet dataset	55
Table 5.1	Segmentation results of PAWCS on CDnet2012	79
Table 5.2	Average per-category and overall scores on CDnet2012	80
Table 5.3	Average recall, precision and F-Measure scores on the intermittent object motion category of CDnet2012	82
Table 5.4	Segmentation results of PAWCS on CDnet2014	83
Table 5.5	Average per-category and overall scores on CDnet2014	84
Table 6.1	Minimum overlap errors achieved for all video sequence pairs of the LITIV dataset (bold entries indicate the best result).	99
Table 7.1	Evaluation results on the multispectral video segmentation dataset of Palmero et al. (2016). Bold results are the best in that category across all methods.	119
Table 7.2	Evaluation results on the multispectral video registration dataset of Bilodeau et al. (2014). Bold results are the best in that category across all methods.	122
Table 7.3	Overall performance for various configurations of the proposed method on the datasets of Palmero et al. (2016); Bilodeau et al. (2014). . . .	123
Table 7.4	Overall segmentation performance for various temporal pipeline depths on the dataset of Palmero et al. (2016).	123
Table 7.5	Evaluation results for the proposed method on our newly captured multispectral video dataset.	125

LISTE DES FIGURES

Figure 1.1	Exemple de système de vidéo surveillance avec traitement en cascade. Les couches de “bas niveau” (à gauche) sont responsables du pré-traitement des données brutes, et les couches de “haut niveau” (à droite) utilisent leurs résultats afin d’analyser et d’interpréter le contenu de la scène. Ici, les couches de bas niveau effectuent une détection et une segmentation de certains objets, qui sont ensuite reconnus et identifiés par les couches de haut niveau. La section encadrée en pointillé identifie les couches d’intérêt étudiées dans cette thèse.	2
Figure 1.2	Exemples de paires d’images visible-infrarouge dans un environnement de faible contraste. À gauche, une paire provenant du jeu de données de Palmero et al. (2016), et à droite, de notre propre jeu de données. La deuxième ligne illustre le résultat idéal qui devrait être obtenu par segmentation mutuelle des objets d’intérêt (dans ce cas, des personnes) dans ces paires d’images.	5
Figure 1.3	Exemples de problèmes pouvant affecter une méthode de détection et de segmentation d’objets d’intérêt tirés du jeu de données de Goyette et al. (2012). À gauche, l’eau présente une surface d’arrière-plan d’apparence dynamique difficile à modéliser. À droite, les changements d’illumination causés par les reflets de phares sont difficiles à ignorer. . . .	6
Figure 1.4	Paire stéréo visible-infrarouge utilisée pour la capture de séquences vidéo synchronisées dans notre laboratoire, constituée d’une caméra FLIR A40 et d’une Kinect v2 (a); et représentation schématique de la géométrie épipolaire liant ces deux caméras (b). La projection sur le plan Image 1 d’un des points réels X peut correspondre à plusieurs points dans le plan Image 2 tous situés sur l’épipole dessinée en rouge. Cette image provient de Arne Nordmann et est distribuée sous license CC BY-SA.	7
Figure 2.1	Exemple d’image typiquement analysée par des méthodes de segmentation vidéo génériques (a) tirée du jeu de données SegTrack v2 (Li et al., 2013), et exemple d’image typiquement analysée dans un contexte de vidéosurveillance (b), tirée du dataset de Wang et al. (2014a).	12

Figure 2.2	Exemple de modélisation paramétrique d'un pixel d'arrière-plan (illustré par le point rouge dans l'image à gauche) à l'aide d'un amalgame de gaussiennes avec quatre composantes pour les canaux Cb et Cr. Deux composantes sont utilisées pour représenter l'eau, et deux composantes pour représenter les feuilles de l'arbre.	14
Figure 2.3	Illustration du fonctionnement de l'approche de modélisation et de détection d'objets d'intérêt non-paramétrique de ViBe pour le pixel " x " encerclé en rouge dans l'image de gauche. La valeur observée pour x , notée $I(x)$, est comparée aux échantillons déjà présents dans le modèle du pixel x , noté $B(x)$. Si au moins deux échantillons de $B(x)$ sont intersectés dans un rayon R de l'observation $I(x)$, alors le pixel est classifié comme arrière-plan (par $S(x) = 0$). Dans le cas contraire (vrai ici), il est classifié comme avant-plan ($S(x) = 1$).	15
Figure 2.4	Exemple de recalage par paramétrisation d'homographie permettant de mettre en correspondance tous les points de l'image du bas dans le référentiel de l'image du haut. Ce recalage est uniquement possible puisqu'il s'agit d'une scène planaire, i.e. où les objets d'intérêt sont observés de loin par rapport à la distance entre les deux caméras. . .	18
Figure 2.5	Exemple de recalage par mise en correspondance dense; la carte en niveau de gris obtenue dans ce cas est une carte de disparités servant au recalage de chaque pixel. Celle-ci peut aussi servir à calculer la profondeur de tous les pixels de la scène.	19
Figure 2.6	Exemple de problème de cosegmentation typique; ici, un même type d'objet est observé dans des conditions variées, à des moments différents, et sous des points de vue uniques. La deuxième ligne illustre les résultats de segmentation attendus. Les images sont tirées du jeu de données de Winn et al. (2005).	23
Figure 2.7	Exemple de problème de segmentation mutuelle; ici, un objet unique est observé dans des conditions similaires, à des moments quasi-identiques, et sous des points de vue semblables. La deuxième ligne illustre les résultats de segmentation attendus. Les images sont tirées du jeu de données de Kowdle et al. (2011).	25

Figure 4.1	Block diagram of SuBSENSE; dotted lines indicate feedback relations. The role of each block and variable is detailed in Sections 4.3.1 through 4.3.4. In our case, post-processing used to generate the output segmentation maps from raw labeling data is based on typical median filtering and blob smoothing operations. This component is an integral part of our method since it provides segmentation noise levels to our feedback process.	36
Figure 4.2	Simplified description and comparison of LBSP features using frames picked from the copyMachine sequence of CDnet. In row (a), an intra-LBSP feature is computed to serve as the basis for other comparisons; in rows (b) and (c), inter-LBSP features are computed using (a)’s reference intensity (the green x), while soft shadow and foreground are respectively visible. Row (b)’s feature can be matched with (a)’s since textures are similar and the relative intensity threshold is correctly scaled in bright-to-dark transitions; this is not the case in row (c). Note that in fact, LBSP features are computed and compared on each color channel.	37
Figure 4.3	Examples of gradient magnitude maps obtained in different CDnet sequences by computing the Hamming weight of dense intra-LBSP descriptors with $T_r = 0.3$ (central column) and an automatically set T_r (right-hand column); bright areas indicate highly textured regions and sharp edges, while dark areas indicate flat regions.	40
Figure 4.4	Average F-Measure scores obtained on the 2012 CDnet dataset for different numbers of background samples, using <i>color-only</i> and <i>color-LBSP</i> configurations of our method (without feedback).	42
Figure 4.5	Typical 2D distributions for our monitoring variables (D_{min} and v), local distance thresholds (R) and local update rates (T) on a baseline sequence of CDnet (“highway”, row a), and on a sequence with important dynamic background elements (“fall”, row b). In R ’s 2D map, bright areas indicate high distance thresholds (thus easier sample-observation matches), while in T , they indicate low update probabilities (meaning the pixel models stay mostly unchanged over time). We can notice in both cases that trees carry high R and low T values, and vice-versa for road; this means that foreground objects will more easily be detected on the road, and are less likely to corrupt the model over time.	45

Figure 4.6	Segmentation results obtained with our proposed model on a baseline sequence of CDnet (“highway”, row a), and on a sequence with important dynamic background elements (“fall”, row b), where (i) used no feedback or post-processing, (ii) used feedback but without post-processing, (iii) used post-processing but without feedback, and (iv) used both. While false classifications are present in all variations, we can note that the results of (iv) are the most consistent with object boundaries.	47
Figure 4.7	By-products of the proposed frame-level analysis component at various times of CDnet sequences; column (i) shows the downscaled input frame used to update the moving averages, and columns (ii) and (iii) respectively show the sequence’s short-term and long-term moving averages. While the difference between these two is negligible for the entire highway (row a) and fall (row b) sequences, it quickly becomes significant in the twoPositionPTZCam sequence (row c) right after the camera rotates. The pixel-wise analysis of discrepancies between these two moving averages allows the detection of such drastic events. . . .	49
Figure 4.8	Typical segmentation results for various sequences of the 2012 version of the CDnet dataset; column a) shows groundtruth maps, b) shows our segmentation results, c) Spectral-360’s results and d) GMM’s results. From top to bottom, the sequences are highway (from the baseline category), fall (dynamic background), traffic (camera jitter), and copyMachine (shadow). Note that gray areas are not evaluated. . . .	56
Figure 4.9	Typical segmentation results for various sequences of the 2014 version of the CDnet dataset; column a) shows groundtruth maps, b) shows our segmentation results, c) FTSG’s results and d) GMM’s results. From top to bottom, the sequences are snowFall (from the bad weather category), streetCornerAtNight (night videos), twoPositionPTZCam (PTZ), and turbulence1 (turbulence). Note that gray areas are not unevaluated.	57
Figure 5.1	Block diagram of the Pixel-based Adaptive Word Consensus Segmenter. Each block represent a major component detailed in Section 5.3. . . .	63

Figure 5.2	Illustrations of possible local dictionary content for dynamic and static background regions. The bars next to each word represents their relative importance in the dictionary based on persistence. In (a), different words are kept active simultaneously while new words are inserted. In (b), the dictionary is shown right after model initialization, and many overlapping words are present due to local neighborhood sampling. In (c), the same dictionary as (b) is presented but at a later time in the sequence, showing a reduced number of active words.	67
Figure 5.3	Snapshots of the actual global word persistence maps used in the CDnet2012 fountain02 sequence at frame 500. The top-left corners show the pixel color description of each word; texture is omitted for illustration purposes. Brighter spots indicate where matches occurred and the map was updated; those points are the “seeds” from which the persistence is diffused. In total, 24 global words were active (the 12 with highest total persistence are shown here), covering over 95% of the image space with non-zero persistence values.	72
Figure 5.4	Qualitative comparison of our segmentation results with the groundtruth on various sequences of CDnet2012. Gray regions in the groundtruth are not evaluated.	81
Figure 5.5	Qualitative comparison of our segmentation results with the groundtruth on various sequences of CDnet2014. Gray regions in the groundtruth are not evaluated. An obvious false positive blob is visible in the last row’s segmentation map; this is a temporary “ghost” artifact caused by a van that left the scene after being parked there.	85
Figure 6.1	Overview of our proposed method’s principal processing stages. . . .	91
Figure 6.2	Example of shape context description on a human shape contour point using 5 radius bins and 8 angle bins.	93
Figure 6.3	Homography smoothing algorithm used for each frame.	95
Figure 6.4	Example of the polygons used for quantitative evaluation formed with manually identified keypoints in the ninth sequence pair of the LITIV dataset.	96

Figure 6.5	Registration results obtained at various moments of the first, second and fourth sequence pairs of the LITIV dataset using our proposed method. The left image in each pair shows the estimated frame-wide registration, and the right shows foreground shape registration at the same moment. The red dashed polygon shows the estimated transformation applied to the infrared image boundary, and the green one shows the ground truth transformation applied to this same boundary.	97
Figure 6.6	Polygon overlap errors obtained using our method (solid red), the method of Sonn et al. (2013) (dashed blue), and the ground truth homography (dotted gray) for the full lengths of all sequence pairs of the LITIV dataset.	98
Figure 6.7	Polygon vertices Euclidean distance errors (in pixels) obtained using our method (solid red), the method of Sonn et al. (2013) (dashed blue), and the ground truth homography (dotted gray) for all sequences of the LITIV dataset. Note that the Y axis has been cropped similarly for all graphs.	100
Figure 7.1	Examples of mutual foreground segmentation in low contrast conditions for RGB-LWIR image pairs. On the left, the person is only partly perceptible in the LWIR spectrum due to a winter coat, but is clearly perceptible in the visible spectrum. The opposite is true on the right, where legs are hard to perceive in the visible spectrum, but easy to perceive in the LWIR spectrum.	104
Figure 7.2	Flowchart of the proposed method. A monocular video segmentation method is first used to initialize segmentation masks for both cameras individually. Then, the energies of the stereo and segmentation models (described in Sections 7.3.1 and 7.3.2, respectively) are alternately minimized until a proper global solution is reached. The output of our method then consists of the refined segmentation masks of the input frames, and of the reciprocal disparity labelings computed for both cameras.	107
Figure 7.3	Simplified case of saliency evaluation during a correspondence search on an epipolar line. On the left, for the “A” pair, low contrast in one image leads to roughly uniform affinity scores and matching costs, which translate into a low local saliency value. On the right, for the “B” pair, good contrast leads to varied affinity scores and matching costs, and a high local saliency value.	111

Figure 7.4	Illustration of the simplified frame layering used in our segmentation model for temporal labeling refinement. In green, the first-order cliques that form E^{smooth^2} are used to enforce spatial coherence in every layer. In blue, the higher order cliques that form E^{temp} are used to enforce temporal coherence across layers. Note that due to foreground motion, these cliques would not all be linked to the same underlying nodes; in reality, the links are dictated by image realignment based on optical flow.	115
Figure 7.5	Examples of typical segmentation results from the VAP dataset of Palmero et al. (2016); the left two columns show the segmentation masks obtained via the method of St-Charles et al. (2016a) and used to initialize our method, and the right two columns show our final segmentation masks. Image regions properly classified as foreground are highlighted in green over the original images, while regions highlighted in orange and magenta show false positives and false negatives, respectively. Images have been cropped to show more details.	127
Figure 7.6	Overall performance for various parameter values of the proposed method on the datasets of Palmero et al. (2016); Bilodeau et al. (2014). The default configuration of each parameter is shown with the dashed line. Remember that for F_1 , higher is better, and for $\bar{d}_{\text{err}}$, lower is better.	128
Figure 7.7	Examples of typical segmentation results from our newly captured dataset; the left two columns show the segmentation masks obtained via St-Charles et al. (2016a) and used to initialize our method, and the right two columns show our final segmentation masks. Image regions properly classified as foreground are highlighted in green over the original images, while regions highlighted in orange and magenta show false positives and false negatives, respectively. Images have been cropped to show more details.	129
Figure 8.1	Exemple de modélisation graphique où chaque pixel d’une image constitue un nœud (a), et où les superpixels de l’image sont calculés et utilisés comme nœuds (b). La deuxième approche permet de réduire le nombre total de nœuds et de liens dans le graphe, ce qui accélère le temps de traitement, en plus d’offrir un meilleur lissage des données.	133

LISTE DES SIGLES ET ABRÉVIATIONS

CCTV	<i>Closed-Circuit Television</i>
CDnet	<i>ChangeDetection.net</i>
DASC	<i>Dense Adaptive Self-Correlation</i>
FIFO	<i>First In, First Out</i>
FN	<i>False Negative</i>
FP	<i>False Positive</i>
FPS	<i>Frames Per Second</i>
GMM	<i>Gaussian Mixture Model</i>
IJCV	<i>International Journal of Computer Vision</i>
KDE	<i>Kernel Density Estimation</i>
LBP	<i>Local Binary Pattern</i>
LBSP	<i>Local Binary Similarity Pattern</i>
LSS	<i>Local Self-Similarity</i>
LWIR	<i>Long-Wavelength Infrared</i>
MI	<i>Mutual Information</i>
MCC	<i>Matthew's Correlation Coefficient</i>
NCC	<i>Normalized Cross-Correlation</i>
NIR	<i>Near-Infrared</i>
PAWCS	<i>Pixel-based Adaptive Word Consensus</i>
PBAS	<i>Pixel-based Adaptive Segmenter</i>
PCA	<i>Principal Component Analysis</i>
PTZ	<i>Pan-Tilt-Zoom</i>
PWC	<i>Percentage of Wrong Classifications</i>
RANSAC	<i>Random Sample Consensus</i>
SC	<i>Shape Context</i>
SIFT	<i>Scale-Invariant Feature Transform</i>
SSD	<i>Sum of Squared Differences</i>
SuBSENSE	<i>Self-Balanced Sensitivity Segmenter</i>
TN	<i>True Negative</i>
TP	<i>True Positive</i>
ViBe	<i>Video Background Extractor</i>

CHAPITRE 1 INTRODUCTION

La vidéosurveillance appliquée à des fins de sécurité publique, d’analyse de trafic routier, ou encore de contrôle de procédés industriels ne date pas d’hier. Déjà dans les années 1970 et 1980, de grands systèmes centralisés de caméras étaient mis en place dans des lieux publics. Toutefois, l’analyse automatique des données générées par de tels systèmes était alors impossible, principalement à cause du manque de puissance de calcul des ordinateurs de l’époque. Aujourd’hui, les méthodes permettant de traiter de telles quantités d’information en temps réel (ou en différé) de façon complètement automatisée sont en pratique encore inexistantes. Tel qu’identifié par Gorodnichy et al. (2016), cela est attribuable en partie au manque de méthodes de pré-traitement avancées.

Bien que la reconnaissance de gestes ou de comportements à partir d’une séquence vidéo peut paraître simple à nos yeux, il ne s’agit pas d’une tâche aussi simple à traduire au niveau d’une application logicielle. En pratique, une séquence vidéo doit être traitée à plusieurs niveaux afin d’y extraire différents éléments de contenu qui deviennent de plus en plus facile à analyser et interpréter. C’est au fil de ces nombreuses transformations de données qu’un ordinateur peut éventuellement reconnaître des objets ou des actions parmi toutes les informations peu pertinentes contenues dans une vidéo. Les méthodes de pré-traitement mentionnées plus haut sont typiquement celles qui effectuent la première transformation de données, par exemple l’identification et le suivi de régions d’intérêt dans des images brutes. Ces régions d’intérêt peuvent ensuite être transmises à une méthode de plus haut niveau, qui pourra par exemple y détecter des individus et localiser leur visage. Une méthode d’encore plus haut niveau pourra finalement tenter de reconnaître et d’associer ces visages à ceux contenus dans une base de données déjà existante. Pour arriver aujourd’hui à un système de vidéosurveillance intelligent et complètement automatisé, chaque couche de transformation de données doit être revisitée et modernisée.

Dans le cadre de cette thèse, nous nous intéressons aux tâches dites de bas niveau, ou de pré-traitement. Plus spécifiquement, nous étudions principalement la détection et la segmentation d’objets d’intérêt à travers des séquences d’images provenant d’une ou de plusieurs caméras synchronisées. Un exemple de ces traitements est présenté à la figure 1.1.

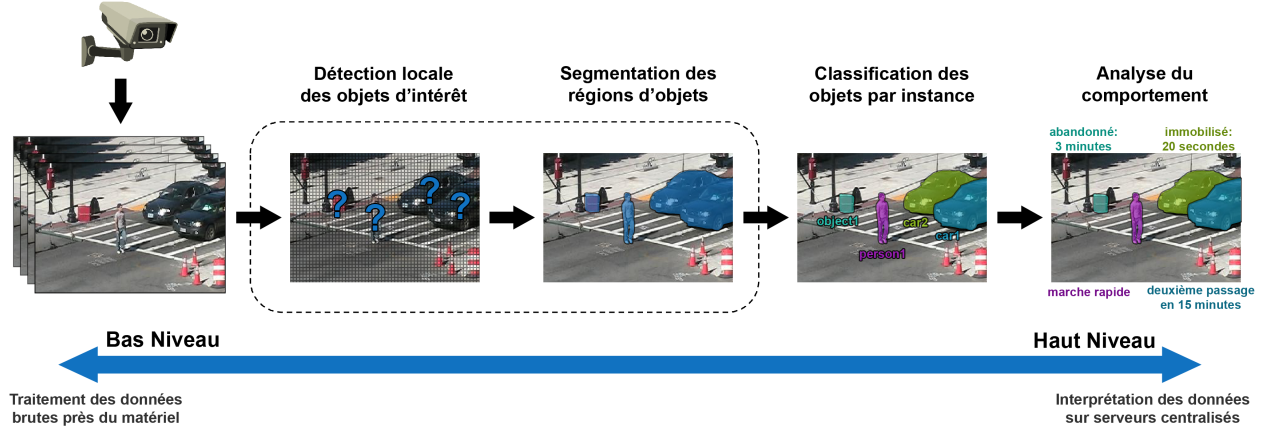


Figure 1.1 Exemple de système de vidéo surveillance avec traitement en cascade. Les couches de “bas niveau” (à gauche) sont responsables du pré-traitement des données brutes, et les couches de “haut niveau” (à droite) utilisent leurs résultats afin d’analyser et d’interpréter le contenu de la scène. Ici, les couches de bas niveau effectuent une détection et une segmentation de certains objets, qui sont ensuite reconnus et identifiés par les couches de haut niveau. La section encadrée en pointillé identifie les couches d’intérêt étudiées dans cette thèse.

1.1 Définitions et concepts de base

Notons d’abord que nous ne contraignons pas la nature des objets qualifiés comme étant “d’intérêt” (ou “ciblés”) dans notre définition du problème. En pratique, ceux-ci sont généralement des individus ou des véhicules, mais aucune connaissance *a priori* de leur apparence, de leur forme ou de leur taille n’est nécessaire pour assurer le fonctionnement des méthodes que nous proposons. Cela nous permet donc, en théorie, de cibler n’importe quel type d’objet à travers une séquence vidéo, ce qui est crucial dans un contexte de vidéosurveillance sans supervision humaine. De plus, cela permet à nos méthodes d’être réutilisées dans n’importe quel contexte sans nécessiter de réajustement par rapport au contenu vidéo analysé.

La première tâche visée, soit la détection d’objets d’intérêt, se base sur la supposition qu’un objet est considéré comme intéressant seulement s’il perturbe l’état naturel de la scène observée. En d’autres termes, dans un contexte de vidéosurveillance, si un objet apparaît subitement ou se déplace à travers le champ de vue d’une caméra, celui-ci devrait être localisé et classifié comme cible pour de futurs traitements. Puisque cette hypothèse est basée uniquement sur le comportement des objets, notons qu’il est possible que deux objets visiblement identiques puissent être classifiés différemment, ou encore qu’un même objet puisse être classifié différemment à deux moments de la séquence analysée (par exemple, s’il est manipulé par quelqu’un). Notons aussi ici que le nombre d’objets d’intérêt présent dans une scène n’est

pas contrôlé, et que ces objets peuvent être de taille et d'apparence variées.

La deuxième tâche visée, soit la segmentation d'objets d'intérêt, est couplée au problème de détection de ces objets et permet d'identifier les régions d'image qui correspondent à ceux-ci. Cette tâche est compliquée par le fait que nous ne disposons au préalable d'aucune information sur l'apparence des objets ciblés, mais elle peut toujours être accomplie à l'aide de deux approches. D'abord, puisque nous traitons des séquences d'images, il est possible de synthétiser un modèle d'apparence pour ces objets au fur et à mesure que les données sont traitées. Ce modèle peut alors servir à correctement identifier toutes les régions d'image appartenant aux objets ciblés lorsque ceux-ci gardent une apparence assez uniforme au fil du temps. Cette approche de modélisation directe des objets d'intérêt (ou de "l'avant-plan") est liée au problème classique de suivi (*tracking*) en vision par ordinateur (voir Yilmaz et al., 2006). La deuxième approche se base sur l'hypothèse qu'au contraire, un modèle d'apparence de la scène sous observation peut être construit avant l'arrivée des objets perturbateurs, et que ceux-ci peuvent être détectés et segmentés par l'analyse de leur différence d'apparence avec ce modèle. Cette deuxième approche englobe toutes les méthodes de segmentation par modélisation et soustraction d'arrière-plan (Bouwmans, 2014), et constitue en fait l'approche préconisée ici pour la détection initiale des objets d'intérêt. Dans les deux cas, le résultat de la segmentation est un masque de classification binaire séparant les pixels de l'image appartenant à un objet d'intérêt (i.e. à l'avant-plan) de ceux appartenant à l'arrière-plan.

Enfin, notons que si nous disposons d'une paire de caméras synchronisées pour observer une même scène sous deux points de vue différents, le recalage et l'intégration des données captées peut nous permettre d'éliminer certaines ambiguïtés présentes lors de la détection ou de la segmentation des objets d'intérêt (voir Zitová and Flusser, 2003). Ces ambiguïtés peuvent être causées par une occultation de l'objet ciblé par un autre élément d'avant-plan dans une des images, ou plus communément par un manque de contraste dans la scène observée (voir l'exemple de la figure 1.2). L'utilisation d'une paire de caméras multispectrale peut d'autant plus diminuer la redondance entre les représentations visuelles provenant des différentes sources de données, minimisant ainsi les chances de rencontrer une paire de régions de faible contraste. En ce qui concerne l'étape d'intégration des données lors de la segmentation, une fusion naïve des images brutes par une moyenne ou une concaténation de leurs canaux peut être suffisante. Toutefois, il est préférable d'utiliser une méthode capable de choisir la source de données à utiliser en fonction des conditions d'observation propres à chaque scène, car la pertinence des données peut varier de façon transitoire. Pour notre problème de segmentation d'objets d'intérêt présents à l'intérieur de plusieurs images, on parle alors de méthodes de co-segmentation (Rother et al., 2006), ou de segmentation mutuelle (Riklin-Raviv et al., 2008). Notons ici que ces méthodes sont généralement basées sur la modélisation de la forme ou

de l'apparence des objets d'intérêt eux-mêmes, contrairement aux méthodes de segmentation par soustraction d'arrière-plan mentionnées précédemment.

1.2 Éléments de la problématique

La détection et la segmentation d'objets d'intérêt dans des séquences vidéo synchronisées regroupent de nombreuses sous-tâches étudiées depuis longtemps en vision par ordinateur. Toutefois, celles-ci sont encore loin d'être considérées comme étant résolues, étant donné la vaste complexité des cas d'utilisation possibles dans des scénarios sans supervision. Dans cette section, nous détaillons les principaux enjeux qui compliquent la conception et le développement des méthodes visées dans le cadre de cette thèse.

1.2.1 Détection et segmentation d'objets par modélisation d'arrière-plan

La détection et la segmentation d'objets dans des séquences vidéo monoculaires selon une approche de modélisation d'arrière-plan est relativement simple lorsque l'environnement de capture est bien contrôlé. Un exemple parfait d'application dans de telles conditions est l'utilisation des écrans verts pour l'insertion d'effets spéciaux en cinématographie — l'illumination de la scène est gardée bien stable, l'arrière-plan est parfaitement uniforme, et le contraste avec les objets d'avant-plan est optimisé pour pouvoir bien les détecter en tout temps. Cela nous permet donc d'identifier facilement toutes les régions d'image qui n'appartiennent pas à l'arrière-plan, faisant indirectement ressortir tous les objets d'intérêt visibles par la caméra. Dans la réalité, il est presque impossible d'exercer un contrôle semblable sur une scène sous surveillance. Dans un lieu public, l'éclairage peut varier très rapidement, l'arrière-plan peut être constitué d'une multitude d'objets d'apparences variées, et certains éléments de la scène peuvent posséder un comportement dynamique à court ou à long terme (e.g. fontaines, arbres bougeant au vent). De plus, l'apparence des objets d'intérêt peut varier en fonction du temps (e.g. une personne qui change de vêtements), et ceux-ci peuvent affecter la disposition de leur environnement en y déplaçant des objets. Quelques exemples de ces défis dans des séquences réelles sont illustrés à la figure 1.3. Afin d'éviter de détecter tout changement d'apparence de l'arrière-plan d'une scène comme étant l'apparition d'un intrus potentiel, une méthode moderne doit donc pouvoir adapter son modèle d'arrière-plan à ces changements en continu. Le contrôle dynamique de cette adaptation est donc aussi un enjeu important, car dans le cadre de l'analyse d'images en temps réel, il est impossible de revenir sur des décisions déjà prises. La méthode doit donc déterminer automatiquement quel comportement adopter afin de faire évoluer adéquatement son modèle en fonction de la dynamique de la scène observée.

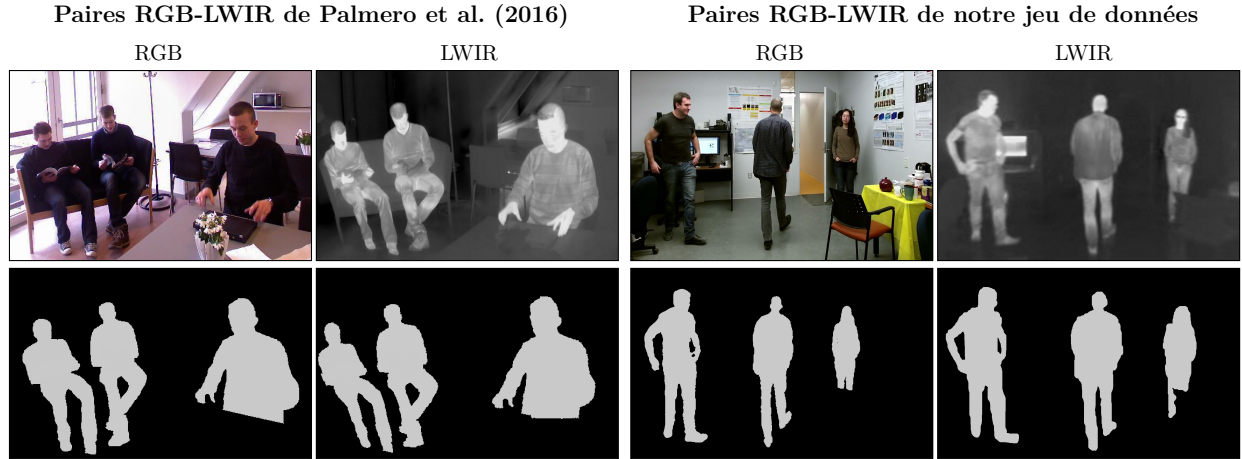


Figure 1.2 Exemples de paires d’images visible-infrarouge dans un environnement de faible contraste. À gauche, une paire provenant du jeu de données de Palmero et al. (2016), et à droite, de notre propre jeu de données. La deuxième ligne illustre le résultat idéal qui devrait être obtenu par segmentation mutuelle des objets d’intérêt (dans ce cas, des personnes) dans ces paires d’images.

1.2.2 Recalage de paires d’images multispectrales

Le recalage d’images provenant d’une paire stéréo (telle que celle illustrée à la figure 1.4.a) peut être simplifié comme étant un problème de mise en correspondance de points dans un domaine unidimensionnel grâce aux contraintes de géométrie épipolaire (voir Hartley and Zisserman, 2003). En résumé, si on dispose des paramètres de calibration intrinsèques et extrinsèques des deux caméras, il est possible de rectifier les images produites par celles-ci. Cette rectification force alors la projection d’un point 3D visible par une des caméras à s’établir dans l’image de l’autre caméra sur une droite dont l’équation est connue d’avance (i.e. une épipole) — voir l’exemple illustré par la figure 1.4.b. Cela nous permet donc de réduire l’espace de recherche des correspondances de façon considérable et de recaler des objets dans une scène 3D, même lorsque la parallaxe est non-négligeable. La recherche de correspondances en 1D sur les épipoles est en soit peu compliquée lorsque les images captées proviennent du même type de caméra ; pour un point donné, nous n’avons qu’à trouver son complément dans l’autre image qui maximise leur affinité visuelle commune. Toutefois, dans le cadre du recalage d’images multispectrales, les données visuelles brutes ne possèdent plus de similarités directes, ce qui complique la mise en correspondance de points. Par exemple, si on considère une caméra qui opère dans le spectre visible et une autre dans le spectre infrarouge “thermique” (*Long-Wavelength Infrared*), on peut comprendre que le lien entre l’apparence visible d’un objet et sa température est quasi inexistant. Il devient alors nécessaire d’utiliser



Figure 1.3 Exemples de problèmes pouvant affecter une méthode de détection et de segmentation d’objets d’intérêt tirés du jeu de données de Goyette et al. (2012). À gauche, l’eau présente une surface d’arrière-plan d’apparence dynamique difficile à modéliser. À droite, les changements d’illumination causés par les reflets de phares sont difficiles à ignorer.

une représentation de plus haut niveau des objets (e.g. contours, trajectoires) afin de pouvoir trouver des correspondances adéquates entre les deux images. Ces représentations ne sont toutefois pas toujours faciles à calculer, et elles peuvent entraîner une perte de précision lors du recalage de petits objets.

1.2.3 Segmentation mutuelle de paires d’images

Pour ce qui est de la segmentation mutuelle de paires d’images, celle-ci nous permet de mieux identifier les régions de chaque image qui appartiennent à un objet commun en fusionnant les données captées simultanément. Puisque les caméras utilisées n’ont pas le même point de vue de la scène, il se peut que la parallaxe ainsi créée rende impossible la mise en correspondance parfaite de toutes les parties d’un objet d’intérêt. La méthode de segmentation mutuelle développée doit donc déterminer dans quels cas un manque d’information visuelle issu du recalage découle de la dissimulation d’une région par une autre partie de l’avant-plan, ou découle d’un trop faible contraste entre les différentes régions d’image. Par ailleurs, si les caméras utilisées sont uniquement synchronisées au niveau logiciel, il est possible que les images captées soient légèrement décalées dans le temps. Cela peut alors entraîner des distorsions incorrigibles autour des objets en mouvement dans une des deux images. La méthode développée doit donc être assez robuste pour accommoder ces distorsions sans affecter la forme réelle de l’objet ciblé dans l’autre image. Finalement, notons que dans notre contexte de vidéosurveillance automatisée, la méthode de segmentation mutuelle ne dispose pas d’une référence initiale pour modéliser parfaitement la forme ou l’apparence des objets ciblés. En effet, si le modèle d’avant-plan de la méthode de segmentation mutuelle est initialisé grâce à

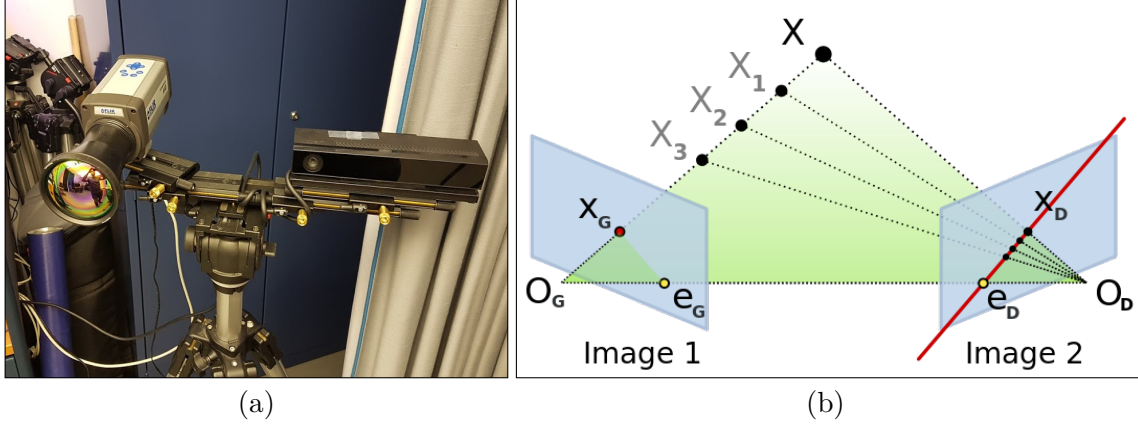


Figure 1.4 Paire stéréo visible-infrarouge utilisée pour la capture de séquences vidéo synchronisées dans notre laboratoire, constituée d’une caméra FLIR A40 et d’une Kinect v2 (a) ; et représentation schématique de la géométrie épipolaire liant ces deux caméras (b). La projection sur le plan Image 1 d’un des points réels X peut correspondre à plusieurs points dans le plan Image 2 tous situés sur l’épipole dessinée en rouge. Cette image provient de Arne Nordmann et est distribuée sous license CC BY-SA.

une méthode de segmentation monoculaire sans supervision (e.g. par soustraction d’arrière-plan), celui-ci pourrait être partiellement erroné. La méthode développée doit donc raffiner itérativement son propre modèle de référence au fur et à mesure que la segmentation des objets d’intérêt s’améliore.

1.2.4 Paradoxe : recalage et segmentation mutuelle

L’utilisation de modalités d’imagerie très différentes est avantageuse pour l’élimination des ambiguïtés liées au manque de contraste dans les images produites, mais complique beaucoup la mise en correspondance des données, tel que mentionné précédemment. Une solution permettant d’améliorer cette mise en correspondance est d’utiliser les contours de formes issus de la segmentation mutuelle des paires d’images comme représentation de haut niveau des objets d’intérêt. L’étape de segmentation mutuelle nécessite toutefois que les données soient déjà recalées afin de pouvoir les intégrer, ce qui nous entraîne vers un paradoxe d’exigences pour ces deux étapes de traitement. Il est donc clair qu’une approche d’optimisation itérative doit être développée afin de répondre aux exigences de ces méthodes simultanément.

1.3 Objectifs de recherche

L’objectif principal de cette thèse est de concevoir et de développer un pipeline de pré-traitement de séquences d’images jumelées provenant de caméras synchronisées. Ce pipeline

a pour but principal de détecter et d'isoler automatiquement les objets en mouvement dans le champ de vue de ces caméras, qualifiés comme étant “d'intérêt” dans un contexte de vidéosurveillance. La détection de ces objets doit être accomplie, peu importe les conditions d'acquisition, en tenant compte de la qualité des données provenant de chaque caméra. De manière plus spécifique, nos objectifs sont de :

- Développer une méthode monoculaire de détection et de segmentation d'objets d'intérêt pour caméra stationnaire. Cette méthode doit pouvoir détecter n'importe quel type d'objet faisant irruption dans la scène surveillée tout en s'adaptant aux variations naturelles de son arrière-plan. Elle doit aussi permettre la détection et la segmentation en continu d'objets d'intérêt temporairement immobilisés (e.g. voitures à l'arrêt, personnes assises). Finalement, elle doit être applicable à n'importe quel type de modalité d'imagerie (e.g. spectre visible, infrarouge).
- Développer une méthode de recalage d'images multispectrales basée sur l'utilisation de masques de segmentation. Cette méthode doit être en mesure de fournir des résultats partiels, même lorsque les résultats de segmentation préliminaires fournis sont imparfaits. Celle-ci doit aussi être applicable aux scènes 3D où le point de vue des caméras peut affecter notre perception de la forme des objets d'intérêt (i.e. lorsque la parallaxe n'est pas négligeable).
- Développer une méthode de segmentation mutuelle d'images multispectrales. Cette méthode doit utiliser les résultats préliminaires de segmentation afin d'initialiser son propre modèle de segmentation, et utiliser les résultats préliminaires du recalage afin d'effectuer l'intégration du contenu multispectral. Elle doit aussi utiliser ces données préliminaires sans causer une dégénérescence de son modèle, et permettre à ce dernier d'évoluer dans le temps au fur et à mesure que de nouvelles paires d'images sont traitées.
- Développer une approche d'optimisation permettant d'alterner entre la recherche de correspondances de contours pour le recalage, et l'intégration des régions d'images pour la segmentation mutuelle. Cette approche doit être conçue de manière à converger naturellement vers une solution adéquate à chaque problème tout en restant relativement rapide.

La poursuite de ces objectifs de recherche a mené à plusieurs contributions scientifiques au cours des dernières années ; celles-ci sont décrites dans la prochaine section.

1.4 Contributions

Les travaux décrits dans cette thèse ont été réalisés en trois phases entre 2013 et 2018.

La première phase de travaux, entreprise entre 2013 et 2016, a porté sur le développement de différentes méthodes de détection et de segmentation d’objets d’intérêt par modélisation d’arrière-plan. La première génération de ces méthodes a servi à étudier différentes façons d’incorporer des descripteurs binaires de texture à l’intérieur d’une approche de modélisation non-paramétrique par pixel (St-Charles and Bilodeau, 2014). Ces descripteurs permettent d’augmenter la sensibilité de la méthode aux variations de textures locales tout en ignorant certaines variations d’illumination dans la scène. L’utilisation de ces descripteurs dans un contexte de détection d’arêtes a d’ailleurs été explorée plus tardivement (St-Charles et al., 2016b). Nous avons ensuite étudié l’effet de différents mécanismes de rétroaction (“*feedback*”) pour contrôler l’adaptation dynamique des modèles non-paramétriques à petite échelle (St-Charles et al., 2014, 2015a). Un nouveau type de modèle non-paramétrique basé sur une analyse de persistance locale a enfin été proposé dans le but de mieux traiter les scènes où certains objets bougent de façon irrégulière (St-Charles et al., 2015b, 2016a).

La deuxième phase de travaux a porté sur le recalage d’images multispectrales, et elle s’est tenue entre 2014 et 2016. D’abord, une étude de performance sur l’efficacité des descripteurs d’images classiques dans la recherche de correspondances multispectrale a été réalisée en collaboration avec d’autres chercheurs de notre laboratoire (Bilodeau et al., 2014). Suite à celle-ci, l’utilisation de descripteurs de formes pour la mise en correspondance de points de contours a été étudiée pour le recalage de scènes planaires, i.e. où la parallaxe est négligeable (St-Charles et al., 2015c). Ce dernier travail a enfin été étendu aux scènes non-planaires dans le cadre du projet d’un stagiaire de notre laboratoire (Nguyen et al., 2016).

La troisième phase de travaux a porté sur les méthodes de cosegmentation et de segmentation mutuelle, et s’est déroulée entre 2017 et 2018. Lors de celle-ci, une méthode de segmentation et de recalage combinée pour des paires d’images multispectrales a d’abord été proposée (St-Charles et al., 2017). Cette méthode a ensuite été améliorée afin de pouvoir traiter des séquences de paires d’images, et le résultat de ces travaux a enfin été soumis au *International Journal of Computer Vision* (IJCV) en février 2018. Un nouvel ensemble de données multispectrales permettant l’évaluation combinée de méthodes de recalage et de segmentation a d’ailleurs été publié au même moment.

Finalement, soulignons que le code source de toutes les méthodes développées dans le cadre de cette thèse a été publié en même temps que leur article respectif¹.

1. Voir <https://github.com/plstcharles/litiv> pour plus d’informations.

1.5 Plan du mémoire

Cette thèse est structurée de la façon suivante. Dans le chapitre 2, nous présentons une revue des travaux pertinents reliés aux objectifs mentionnés dans la section 1.3. Dans le chapitre 3, nous présentons un survol du pipeline de traitement proposé et des articles qui s’y rattachent. Ces articles sont ensuite présentés individuellement dans les chapitres 4, 5, 6, et 7. Le chapitre 8 propose une discussion générale du travail accompli et quelques améliorations qui auraient pu être apportées à nos méthodes. Enfin, une conclusion est formulée dans le chapitre 9 donnant quelques pistes de recherche pour de futurs travaux.

CHAPITRE 2 REVUE DE LITTÉRATURE

Dans ce chapitre, nous survolons les techniques de base et les travaux déjà réalisés qui sont reliés aux objectifs de recherche énumérés dans la section 1.3. Nous en profitons aussi pour identifier certaines tendances des travaux récents de la littérature dans le domaine, et nous discutons des avantages et des inconvénients de certaines techniques modernes.

2.1 Segmentation vidéo

Tel que mentionné dans le chapitre 1, les méthodes de segmentation binaires (i.e. qui séparent l’avant-plan d’une scène de son arrière-plan) doivent d’entrée de jeu se baser sur au moins une supposition afin de pouvoir détecter l’avant-plan de la scène, c’est à dire les objets d’intérêt. Lorsqu’on ne dispose que d’une seule image, il est nécessaire d’utiliser des hypothèses de saillance (“*saliency*”) touchant les objets dans l’image. Par exemple, on peut supposer qu’il existe toujours un objet d’intérêt dans le champ de vue de la caméra, que cet objet possède un bon contraste par rapport au reste de la scène, ou encore qu’il est à peu près centré dans l’image (voir e.g. Arbelaez et al., 2011). La généralisation de cette approche aux séquences vidéo sans point de vue fixe permet d’exploiter la redondance temporelle d’apparence de l’avant-plan, sous l’hypothèse que les objets d’intérêt restent visibles au fil du temps. L’utilisation de modèles graphiques hautement connectés permet alors d’effectuer un suivi spatiotemporel très précis des objets d’intérêt tout en ignorant les variations de l’arrière-plan, qui lui peut être identifié par une analyse de flux optique. Les méthodes évaluées par Perazzi et al. (2016) fonctionnent à peu près toutes selon ce principe général.

L’utilisation de fortes suppositions sur la présence, la taille, le nombre ou la forme des objets d’intérêt dans une scène n’est toutefois pas idéale dans le contexte d’une application de vidéosurveillance. En pratique, les caméras utilisées sont souvent statiques et elles possèdent un champ de vue large qui n’est pas nécessairement centré sur les objets d’intérêt (un exemple comparatif est donné à la figure 2.1). Une stratégie de segmentation vidéo qui s’applique mieux à ce cas particulier est la soustraction d’arrière-plan. Cette stratégie permet de détecter et de segmenter des objets d’intérêt sous la seule supposition que le point de vue de la caméra est fixe. Cette supposition nous permet alors de modéliser l’arrière-plan de la scène en continu, et la détection d’objets en intrusion peut alors être effectuée en déterminant quelles régions de l’arrière-plan sont soudainement moins semblables à celles du modèle. La logique ici est que ces régions d’arrière-plan sont fort probablement cachées derrière un objet qui n’est pas naturellement présent dans la scène.



Figure 2.1 Exemple d'image typiquement analysée par des méthodes de segmentation vidéo génériques (a) tirée du jeu de données SegTrack v2 (Li et al., 2013), et exemple d'image typiquement analysée dans un contexte de vidéosurveillance (b), tirée du dataset de Wang et al. (2014a).

Tableau 2.1 Taxonomie des approches de segmentation vidéo par soustraction d'arrière-plan

Approche générale	Approche spécifique	Méthodes
Modélisation paramétrique	Amalgame de gaussiennes	Friedman and Russell (1997)
		Stauffer and Grimson (1999)
	Amalgame dynamique	Zivkovic (2004) Haines and Xiang (2014)
Modélisation non-paramétrique	<i>Sample Consensus</i>	ViBe (Barnich and Van Droogenbroeck, 2011)
		PBAS (Hofmann et al., 2012)
	Dictionnaires	<i>Codebook</i> (Kim et al., 2005)
		Wu and Peng (2010)
	Histogrammes	Mayer and Mundy (2014)
		Heikkila and Pietikäinen (2006)
Décomposition globale	Estimation de densité	Zhang et al. (2008)
		Elgammal et al. (2000)
	<i>Robust-PCA</i>	Zivkovic and van der Heijden (2006)
		Liao et al. (2010)
	<i>Subspace tracking</i>	DECOLOR (Zhou et al., 2013)
		Candès et al. (2011)
		GRASTA (He et al., 2012)
		pROST (Seidel et al., 2014)

Il existe trois grandes approches de modélisation pour la stratégie de soustraction d’arrière-plan ; celles-ci sont décrites dans les sous-sections 2.1.1, 2.1.2, et 2.1.4 ci-dessous. Une taxonomie de certaines des méthodes citées plus loin est présentée dans le tableau 2.1. Pour plus d’information sur d’autres approches atypiques ainsi qu’une revue des récentes méthodes de soustraction d’arrière-plan, nous redirigeons le lecteur vers les travaux de Brutzer et al. (2011) et de Bouwmans (2014).

2.1.1 Soustraction d’arrière-plan par modélisation paramétrique

L’approche de modélisation paramétrique est basée sur l’idée que l’arrière-plan peut être décrit à l’aide d’un ensemble de distributions de probabilité exprimées par leurs paramètres. L’exemple le plus simple de cette approche de modélisation a été introduit par Friedman and Russell (1997) et Stauffer and Grimson (1999) ; ceux-ci proposent d’utiliser un modèle d’amalgame de gaussiennes (“*Gaussian Mixture Model*”) dans le but de décrire l’apparence de chaque pixel de l’arrière-plan d’une scène. Tel qu’illustré à la figure 2.2, la combinaison de ces gaussiennes permet de capturer plusieurs représentations (ou “modes”) de l’arrière-plan en un point, ce qui est très avantageux lorsque certains éléments de la scène possèdent un comportement dynamique (e.g. arbres, eau). La modélisation paramétrique est une approche populaire principalement à cause de sa simplicité d’implémentation et de son efficacité en termes de temps de calcul. Celle-ci a d’ailleurs fait l’objet de nombreuses études proposant différentes façons de contrôler le nombre de composantes de chaque pixel, ou proposant une amélioration à l’adaptation des paramètres des distributions en fonction du dynamisme de l’arrière-plan (Zivkovic, 2004; Lee, 2005; Haines and Xiang, 2014).

2.1.2 Soustraction d’arrière-plan par modélisation non-paramétrique

Le but de l’approche de modélisation non-paramétrique est d’utiliser des représentations compactes basées sur les données déjà observées afin de décrire l’arrière-plan de la scène de façon plus spécifique. Cette approche est mieux adaptée que l’approche paramétrique lorsque la complexité de la scène est difficile à décrire à l’aide de distributions de probabilité. Par contre, celle-ci requiert généralement l’observation de plus de données et l’utilisation d’une plus grande quantité de mémoire pour entreposer les représentations de l’arrière-plan. Un exemple de modélisation non-paramétrique classique est la méthode de Elgammal et al. (2000). Cette méthode effectue une estimation de la densité de probabilité de l’apparence de l’arrière-plan pour chaque pixel du modèle à partir d’observations d’intensités locales par la méthode de Parzen-Rozenblatt (voir Parzen, 1962, il s’agit d’une généralisation de l’estimation par histogramme). Une simplification assez récente et très efficace de cette méthode a

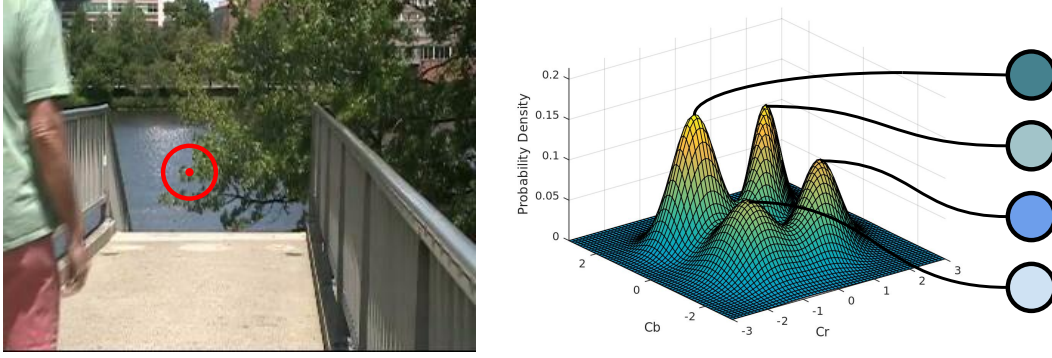


Figure 2.2 Exemple de modélisation paramétrique d'un pixel d'arrière-plan (illustré par le point rouge dans l'image à gauche) à l'aide d'un amalgame de gaussiennes avec quatre composantes pour les canaux Cb et Cr. Deux composantes sont utilisées pour représenter l'eau, et deux composantes pour représenter les feuilles de l'arbre.

été proposée par Barnich and Van Droogenbroeck (2011). ViBe, leur nouvelle méthode, modélise l'arrière-plan pour un pixel donné en gardant tout simplement en mémoire un ensemble d'observations récentes d'intensités captées dans le voisinage local du pixel (voir l'exemple de la figure 2.3). Grâce à la mise à jour aléatoire des échantillons de cet ensemble, cette nouvelle méthode est très robuste face aux comportements dynamiques de l'arrière-plan.

Afin de mieux gérer l'importance des échantillons contenus dans un modèle d'arrière-plan non-paramétrique, Kim et al. (2005) ont proposé d'associer un poids relatif à chaque échantillon. Cette valeur permet alors de déterminer quel échantillon devrait être remplacé dans un modèle lorsque celui-ci est mis à jour (importance minimale), et quel échantillon a le plus de chance d'être observé dans une nouvelle image (importance maximale). L'utilisation de ces poids est un principe clé des approches de modélisation par dictionnaire (ou par “*codebooks*”). Celles-ci permettent de réduire la quantité de mémoire nécessaire pour modéliser l'arrière-plan d'une scène de façon non-paramétrique tout en gardant la flexibilité d'une approche multimodale. De plus, cette approche permet de garder intacte la représentation intégrale de l'arrière-plan d'une scène malgré la présence à long terme d'un objet d'intérêt immobilisé dans celle-ci. Ce type de modèle a d'ailleurs été amélioré dans de nombreux travaux au fil des ans (voir e.g. Wu and Peng, 2010; Mayer and Mundy, 2014). Notons ici que l'approche de modélisation que nous avons choisi d'utiliser dans notre méthode de soustraction d'arrière-plan finale (présentée dans le chapitre 5) est une combinaison de modélisation non-paramétrique par *codebooks*, mais qui profite des avantages de l'échantillonnage stochastique de ViBe.

D'autre part, il existe de nombreuses façons d'interpréter l'apparence d'un pixel ou d'une région pour l'échantillonnage d'un modèle d'arrière-plan non-paramétrique. L'utilisation di-

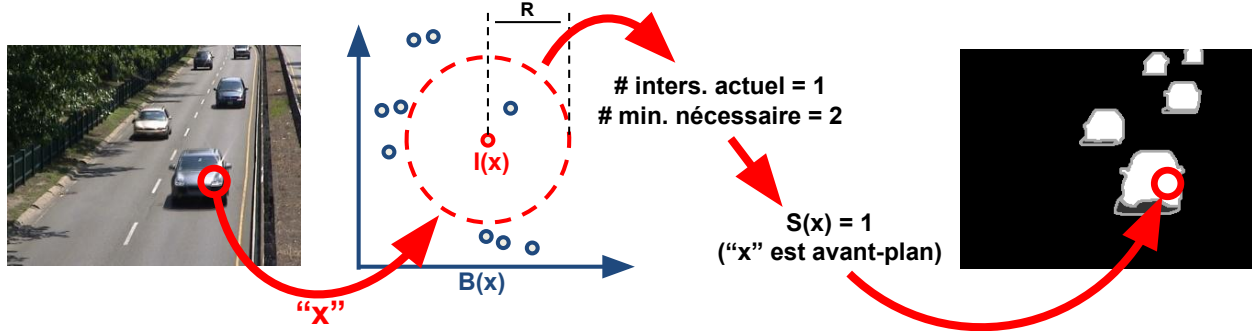


Figure 2.3 Illustration du fonctionnement de l’approche de modélisation et de détection d’objets d’intérêt non-paramétrique de ViBe pour le pixel “ x ” encerclé en rouge dans l’image de gauche. La valeur observée pour x , notée $I(x)$, est comparée aux échantillons déjà présents dans le modèle du pixel x , noté $B(x)$. Si au moins deux échantillons de $B(x)$ sont intersectés dans un rayon R de l’observation $I(x)$, alors le pixel est classifié comme arrière-plan (par $S(x) = 0$). Dans le cas contraire (vrai ici), il est classifié comme avant-plan ($S(x) = 1$).

recte de la couleur (sous un modèle RGB, HSV ou autre) reste l’approche la plus simple, mais celle-ci omet l’information sur la structure spatiale ou spatiotemporelle de la sous-région avoisinante du pixel analysé. Plusieurs auteurs se sont donc penchés sur l’utilisation de descripteurs locaux binaires dans le but d’améliorer la perceptibilité des changements de textures. Heikkila and Pietikäinen (2006) étaient les premiers à proposer un modèle basé strictement sur ce genre de descripteur (*Local Binary Pattern*) ; ils ont démontré que leur méthode était très tolérante aux variations d’illumination et robuste face aux régions d’arrière-plan dynamiques. Cette robustesse a toutefois été obtenue en s’appuyant sur le calcul et la comparaison d’histogrammes de descripteurs, ce qui rend leur approche moins sensible aux variations de texture dont l’échelle est inférieure à celle du patron binaire utilisé. Une amélioration de cette méthode a d’ailleurs été proposée par Zhang et al. (2008) ; ceux-ci ont décidé de calculer les histogrammes de descripteurs sur différentes trames et de les fusionner, exploitant ainsi une partie de l’information temporelle qui n’était pas considérée dans l’algorithme original. Une autre approche a été suggérée par Liao et al. (2010) qui ne requiert pas l’utilisation d’histogrammes ; un nouveau type de descripteur invariable aux changements d’illumination a été introduit et utilisé directement dans un modèle d’arrière-plan par échantillonnage similaire à celui de Elgammal et al. (2000). Finalement, les descripteurs de type “*Local Binary Similarity Pattern*” ont été introduits par Bilodeau et al. (2013) et il a été démontré que ceux-ci caractérisent mieux l’apparence de l’arrière-plan pour des scènes intérieures et extérieures que le modèle de couleur RGB. Notons d’ailleurs que dans ce dernier travail, les descripteurs sont comparés directement à l’aide de la distance de Hamming. Nous utilisons ces descripteurs dans nos méthodes présentées dans les chapitres 4 et 5.

2.1.3 Mécanismes de contrôle et de rétroaction

De nombreux travaux modernes ont proposé de combiner des approches de soustraction d'arrière-plan avec différents mécanismes leur permettant d'adapter leur processus de détection d'avant-plan ou d'adaptation de modèles locaux. Par exemple, dans l'amélioration de ViBe proposée par Van Droogenbroeck and Paquot (2012), le niveau de bruit observé dans la carte de segmentation est utilisé pour contrôler la stratégie de mise à jour du modèle. Pour la méthode de Hofmann et al. (2012) qui est aussi basée sur ViBe, c'est une analyse du niveau de dynamisme de l'arrière-plan ("*background dynamics*") pour chaque pixel qui dicte comment modifier les paramètres d'adaptation et de sensibilité de l'algorithme. Du côté des méthodes paramétriques, de nombreux travaux se basent sur l'analyse de la variance d'intensité locale dans le but de déterminer le nombre de distributions à utiliser dans la représentation de l'arrière-plan (voir e.g. Zivkovic and van der Heijden, 2006). Des composantes de post-traitement peuvent aussi être utilisées avec n'importe quelle approche pour déclencher des mécanismes de mise à jour du modèle d'arrière-plan ; pensons par exemple ici aux modules de détection d'objets statiques de Evangelio and Sikora (2011); Morde et al. (2012) ou bien au module d'analyse de changements au niveau des trames de Toyama et al. (1999). Plusieurs mécanismes de contrôle et de rétroaction sont d'ailleurs proposés pour les méthodes des chapitres 4 et 5, mais ceux-ci sont conçus de façon à éviter les complications provoquées par l'immobilisation à long terme d'objets d'avant-plan.

2.1.4 Segmentation par décomposition globale

La modélisation d'arrière-plan par analyse des composantes principales de la scène observée ou par l'approximation de ses matrices de bas rang ("*low-rank approximation*") permet de séparer l'avant-plan de l'arrière-plan dans une vidéo de façon globale (Zhou et al., 2013; Gao et al., 2014; Bouwmans et al., 2016). Cette approche est basée sur l'hypothèse que la vidéo analysée, prise sous la forme d'une matrice ou d'un tenseur 3D, possède un arrière-plan linéairement corrélé dans le domaine temporel. La vidéo peut alors être décomposée entièrement sous la forme d'une matrice représentant l'arrière-plan et de *outliers* qui représentent les formes d'avant-plan. L'avantage principal de ce type d'approche est que les séquences vidéo sont considérées de façon intégrale afin de détecter tous les objets d'intérêt qu'elles contiennent. En d'autres mots, il n'est pas nécessaire d'analyser les trames une par une afin d'y détecter les objets en mouvement. Cela élimine donc certaines ambiguïtés liées à la classification initiale des objets s'introduisant lentement dans une scène. Par contre, cette stratégie n'est pas idéale pour le traitement en temps réel, car elle nécessite l'utilisation de nombreuses trames consécutives pour son analyse, et elle demande beaucoup de ressources

de calcul lorsque les séquences traitées sont longues.

2.1.5 Segmentation par apprentissage profond

Notons finalement l'existence d'un nombre grandissant de méthodes de segmentation basées sur l'utilisation de techniques d'apprentissage profond. Contrairement aux méthodes basées sur la modélisation progressive et le suivi d'avant-plan et aux méthodes de soustraction d'arrière-plan, les méthodes par apprentissage profond utilisent des données externes afin d'apprendre à reconnaître la présence d'un objet d'intérêt dans une scène a priori. Cet apprentissage hors-ligne est effectué de façon supervisée à l'aide d'une série d'unités de traitement non-linéaires qui encodent la représentation des objets recherchés de façon de plus en plus abstraite. Lors de l'analyse d'une image, la couche créée au plus haut niveau de l'architecture d'encodage peut alors être interprétée pour détecter la présence d'un objet, ou encore être réinjectée dans une série d'unités de traitement inversée afin d'obtenir une carte de segmentation sémantique (voir e.g. l'architecture U-Net de Ronneberger et al., 2015). De nombreuses méthodes telle Mask-RCNN (He et al., 2017) ont été conçues pour segmenter des images individuellement, mais elles performant tout aussi bien lorsqu'elles sont appliquées sur des séquences d'images. Celles-ci sont par contre limitées par la quantité et la variété des données disponibles lors de l'entraînement, et par les classes d'objets qu'elles peuvent reconnaître lors de tests réels.

Une stratégie alternative aussi basée sur l'apprentissage profond a été étudiée par Braham and Van Droogenbroeck (2016) afin d'éliminer le besoin de connaître toutes les classes d'objets d'avance. Leur méthode est basée sur l'apprentissage et la classification des différences entre l'arrière-plan construit de manière traditionnelle et le contenu de la scène sous observation. Il est aussi possible d'entraîner un réseau de neurones à partir de données annotées provenant de la séquence traitée elle-même ; cela produit une méthode de segmentation supervisée capable de produire des résultats comparables à ceux d'un annotateur humain (voir e.g. Wang et al., 2017). Notons finalement que ces méthodes ne gèrent jusqu'à maintenant pas très bien les cas où certains objets s'immobilisent temporairement, car leur approche de modélisation ne comporte généralement pas de composante temporelle.

2.2 Recalage d'images multispectrales

Il existe de nombreuses approches de recalage permettant de combiner le contenu de deux images à l'échelle de leurs pixels, mais ces approches nécessitent toutes d'obtenir en premier lieu un certain nombre de correspondances entre ces deux images. Notons que cette étape

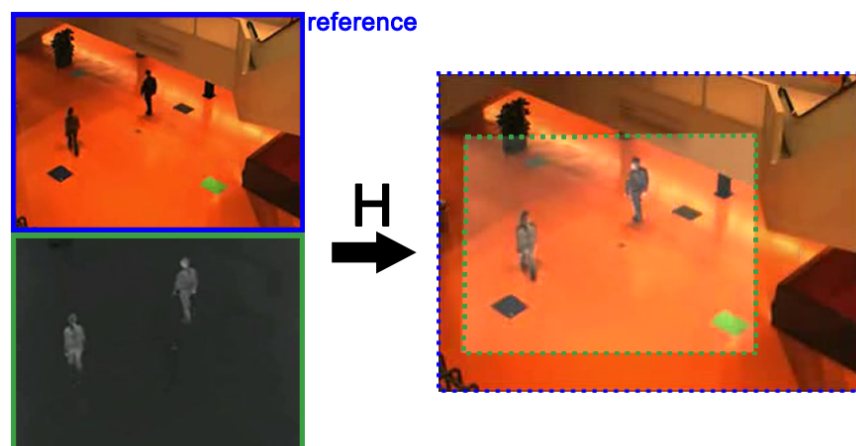


Figure 2.4 Exemple de recalage par paramétrisation d’homographie permettant de mettre en correspondance tous les points de l’image du bas dans le référentiel de l’image du haut. Ce recalage est uniquement possible puisqu’il s’agit d’une scène plane, i.e. où les objets d’intérêt sont observés de loin par rapport à la distance entre les deux caméras.

de mise en correspondance peut être simplifiée par certaines contraintes géométriques, tel qu’expliqué dans la section 1.2.2. Une fois les correspondances trouvées, celles-ci peuvent être utilisées afin d’estimer un modèle de transformation paramétrique permettant de recaler tous les points des deux images (voir l’exemple de la figure 2.4). Par contre, dans les cas où la parallaxe causée par la profondeur des objets de la scène n’est pas négligeable (i.e. pour les scènes dites “non-planaires”), les modèles paramétriques ne peuvent plus être appliqués, car le recalage n’est plus global. Il faut donc utiliser une approche de modélisation non-paramétrique basée sur le calcul de correspondances denses (i.e. pour chaque pixel), tel qu’illustré dans l’exemple de la figure 2.5. La mise en correspondance dense produit alors une carte de disparités encodant le vecteur de déplacement requis pour trouver le complément de n’importe quel point dans une des deux images. Dans le cas des paires d’images rectifiées, il existe un lien direct entre la disparité estimée de chaque pixel et sa profondeur réelle dans la scène (Hartley and Zisserman, 2003). La carte de disparités peut donc être considérée comme une carte de profondeur, à un facteur d’échelle près.

Dans le domaine multispectral, le problème de mise en correspondance est beaucoup plus compliqué, car la combinaison de régions d’images qui ne se ressemblent visuellement pas est paradoxale. Par exemple, dans le cas du spectre visible et de l’infrarouge thermique (*Long-Wavelength Infrared*, LWIR), un pic d’intensité dans l’infrarouge indique la présence d’un objet “chaud”, mais cela ne donne à peu près pas d’information sur son apparence visible. Il faut donc faire abstraction du contenu visuel brut des images à l’aide de mesures comparatives

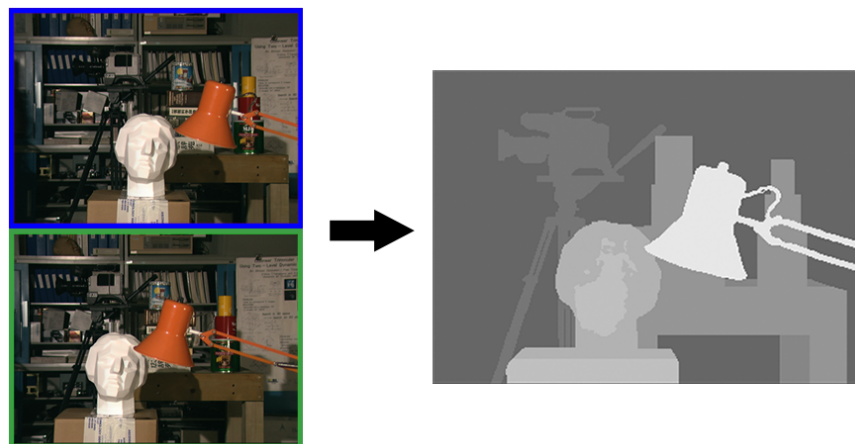


Figure 2.5 Exemple de recalage par mise en correspondance dense ; la carte en niveau de gris obtenue dans ce cas est une carte de disparités servant au recalage de chaque pixel. Celle-ci peut aussi servir à calculer la profondeur de tous les pixels de la scène.

d'auto-similarité, ou encore utiliser des représentations de plus haut niveau des objets de la scène. Ces deux approches sont décrites dans les sections 2.2.1 et 2.2.2, respectivement. La section 2.2.3 présente ensuite quelques approches de modélisation paramétrique utilisées dans des travaux de recalage multispectral, en plus de certains exemples de recalage par estimation de cartes de disparités.

Notons enfin qu'il existe certains moyens physiques d'effectuer le recalage d'images de différents spectres au moment même de l'acquisition ; l'approche de (St-Laurent et al., 2010) qui repose sur un miroir semi-réfléchissant en est un exemple. Cette stratégie ne permet toutefois pas d'exploiter l'information 3D de la scène puisque les capteurs sont virtuellement superposés, ce qui brise la perception de profondeur. De plus, ces systèmes sont généralement fragiles, et ils peuvent altérer les couleurs captées par les deux capteurs.

2.2.1 Mise en correspondance par mesures de similarité locales

Pour la mise en correspondance de points dans des images de même spectre, des mesures de similarité basées directement sur l'intensité des pixels telles *Normalized Cross-Correlation* (NCC) et *Sum of Squared Differences* (SSD) ont souvent été utilisées dans la littérature (Brown, 1992; Zitová and Flusser, 2003). Ces mesures sont toutefois rarement utilisées dans le cas de paires d'images multispectrales, car les liens d'apparence entre les objets des deux images sont souvent trop faibles. En supposant que les régions d'intérêt sont assez contrastées et qu'elles possèdent des formes similaires dans les deux spectres analysés, il est possible d'utiliser le critère d'information mutuelle (*Mutual Information*, MI) comme mesure de si-

milarité (Egnal and Daniilidis, 2000; Pluim et al., 2003; Krotosky and Trivedi, 2007). Une autre alternative est de calculer la distance entre descripteurs *Local Self-Similarity* (LSS) afin d’obtenir une mesure de similarité pour chaque paire de pixels des deux images analysées, tel que démontré par Torabi and Bilodeau (2013). Ce descripteur est basé sur l’idée que la structure locale des éléments d’une image peut être encodée à l’aide de comparaisons d’apparence dans l’image même. Une évaluation de la fiabilité de différentes mesures de similarité et de différents descripteurs pour le problème de recalage visible-infrarouge a d’ailleurs été proposée par Bilodeau et al. (2014). Notons ici que l’approche de recalage proposée dans le chapitre 7 de cette thèse est basée sur la mise en correspondance de points à l’aide un descripteur de similarités locales similaire à LSS, nommé DASC (*Dense Adaptive Self-Correlating descriptor*; Kim et al., 2015).

D’autre part, notons qu’il est possible de remplacer l’approche de recherche de correspondances selon une mesure de similarité locale par l’utilisation d’un détecteur-descripteur de points-clés (e.g. SIFT; Lowe, 2004) combiné avec une approche d’association de points-clés. Il est toutefois difficile de trouver une stratégie de détection et de description convenable au problème multispectral (voir e.g. Aguilera et al., 2012; Ricaurte et al., 2014).

2.2.2 Mise en correspondance par représentation de haut niveau

Au lieu de se limiter à l’information de bas niveau (e.g. intensités, gradients) pour la recherche de correspondances, il est possible de construire des représentations de plus haut niveau du contenu visuel des images à recaler (e.g. arêtes, contours) et d’utiliser ces nouvelles représentations dans le but d’établir des correspondances entre les images. Cette stratégie est intéressante dans le cas du recalage multispectral puisque, tel que mentionné dans la section précédente, les paires d’images analysées ont souvent peu d’affinités visuelles directes. Le traitement de séquences de paires d’images est encore plus intéressant, car la redondance temporelle entre les images consécutives peut aussi être exploitée dans le but de construire d’autres types de représentations de haut niveau (e.g. trajectoires).

D’abord, parmi les types de représentations qui peuvent être construites à partir d’une seule image, notons que les silhouettes d’objets et les cartes d’arêtes ont souvent été utilisées pour la recherche de correspondances multispectrales. Coiras et al. (2000) ont proposé une méthode de recalage basée sur l’extraction de lignes droites des cartes d’arêtes générées à partir d’images provenant d’environnements urbains (i.e. où les scènes contiennent de nombreux repères rectilignes). Pistarelli et al. (2013) ont proposé d’utiliser l’espace de Hough comme domaine de recherche pour trouver des correspondances entre segments de droites dans des conditions similaires. Tian et al. (2015) ont utilisé des cartes d’arêtes tirées de gradients pour

le recalage de visages dans les spectres visible et infrarouge, mais ils ont décrit leurs ensembles de points en utilisant la technique de *Shape Context* de Belongie et al. (2002). Cette technique de description permet de représenter de façon compacte le voisinage d'un point de contour en fonction de la disposition des autres points autour de lui. Notons que la méthode que nous proposons dans le chapitre 6 utilise le descripteur de Belongie et al. (2002) pour son approche de mise en correspondance de contours d'objets. Ce descripteur est réutilisé une deuxième fois dans la méthode du chapitre 7 afin de calculer une carte d'affinités dense entre deux masques de segmentation.

Pour ce qui est des méthodes exploitant l'aspect temporel du recalage de vidéos, les solutions de Bilodeau et al. (2011a); Sonn et al. (2013); Zhao and Sen-ching (2014) se basent toutes sur la mise en correspondance de points de contours tirés de masques de segmentation obtenus par soustraction d'arrière-plan. Les méthodes de Caspi et al. (2002); Bilodeau et al. (2011b); Torabi et al. (2012) se basent aussi sur ce type de segmentation, mais elles utilisent plutôt les trajectoires des objets d'avant-plan comme source de correspondances. Ces deux approches ont l'avantage d'être robustes et efficaces pour des cas simples, mais elles ne peuvent pas gérer les cas où un objet d'avant-plan est occulté par un autre, ce qui combine les contours des deux objets. De tels cas sont communs lorsque les scènes traitées sont non-planaires. De plus, il est supposé que l'avant-plan peut toujours être adéquatement séparé de l'arrière-plan dans les deux spectres analysés, et que la forme des régions obtenues est à peu près équivalente, ce qui n'est pas toujours vrai. Notre méthode présentée dans le chapitre 7 s'attaque à ce problème en optimisant les résultats de recalage et de segmentation de manière itérative.

2.2.3 Modélisation de la transformation de recalage

En ce qui concerne enfin la modélisation de la transformation de recalage entre deux images, nous décrivons deux grands types d'approches, soit encore une fois les approches paramétriques et non-paramétriques.

Les approches paramétriques sont employées ici par les méthodes qui considèrent peu de correspondances par paire d'images analysées. Selon le modèle utilisé, celles-ci ajustent une équation qui approxime la transformation de recalage globale recherchée à partir de seulement quelques paires de points. Ces approches utilisent généralement une méthode d'ajustement telle la transformée linéaire directe (*Direct Linear Transform*; Hartley and Zisserman, 2003) ou le *Random Sample Consensus* (RANSAC; Fischler and Bolles, 1981) pour déduire les paramètres de cette équation. Les modèles de transformation paramétriques les plus simples sont basés sur des transformations rigides ou projectives (i.e. homographiques, voir la figure 2.4 pour un exemple) et de nombreuses méthodes en font directement usage pour assurer

la rapidité du calcul de leurs résultats (Sonn et al., 2013; Caspi et al., 2002; Torabi et al., 2012). Ces méthodes sont toutefois contraintes au recalage de scènes planaires puisque de tels modèles ne tiennent pas compte de l’effet de parallaxe. Par scènes planaires, il est question ici de scènes où les capteurs sont proches et où les cibles sont très loin de ces derniers (recalage homographique à l’infini), ou encore de scènes où toutes les cibles sont localisées sur un même plan (recalage par *planar ground*), tel que décrit par Krotosky and Trivedi (2007). Goshtasby (2005) énumèrent enfin de nombreux types de modèles paramétriques et non-linéaires qui peuvent être utilisés pour le recalage d’images.

Les approches non-paramétriques sont quant à elles plus souvent utilisées par les méthodes de recalage basées sur la mise en correspondance 1D par mesure de similarité locale, car elles nécessitent un très grand nombre de correspondances. Dans le cas où les images à recaler sont rectifiées (i.e. lorsque les plans d’image des deux capteurs coïncident), le problème revient en fait à estimer une carte de disparités, telle qu’illustrée à la figure 2.5. Cette carte représente alors le modèle de transformation lui-même et elle peut être lissée en 3D afin de mieux respecter la géométrie des différentes surfaces la scène (Kohli et al., 2009; Zhang et al., 2015). Pour un traitement plus rapide dans un contexte de vidéosurveillance en ligne, il est aussi possible d’effectuer un lissage par morceau des régions d’intérêt (*piecewise planar*; Gallup et al., 2010; Barrera et al., 2013) ou bien par colonnes de pixels (Torabi and Bilodeau, 2013).

2.3 Segmentation mutuelle et cosegmentation d’images

Tel que mentionné précédemment, le problème entourant la segmentation combinée de l’avant-plan de plusieurs images sans information *a priori* peut être abordé de deux façons différentes. L’approche par cosegmentation est celle qui est la plus générique, car elle ne nécessite pas le recalage des images analysées, mais elle suppose certaines propriétés des objets ciblés. Elle est décrite dans la section 2.3.1 ci-dessous. Le deuxième type d’approche que nous distinguons ici est la segmentation mutuelle, qui elle nécessite le recalage des images, mais qui offre aussi en général une meilleure précision avec moins de suppositions. Celle-ci est décrite dans la section 2.3.2. Enfin, nous recensons différents travaux déjà publiés étudiant la segmentation combinée d’images multispectrales dans la section 2.3.3.

2.3.1 Segmentation multi-instance (cosegmentation)

L’idée de partager les données provenant de plusieurs images afin de mieux segmenter les objets d’intérêt qu’elles ont en commun provient originalement de Rother et al. (2006). Ceux-ci ont d’ailleurs introduit le terme “cosegmentation” en même temps que leur méthode basée



Figure 2.6 Exemple de problème de cosegmentation typique ; ici, un même type d’objet est observé dans des conditions variées, à des moments différents, et sous des points de vue uniques. La deuxième ligne illustre les résultats de segmentation attendus. Les images sont tirées du jeu de données de Winn et al. (2005).

sur l’élaboration d’un modèle graphique de cohérence visuelle. Ici, le principe de maximisation de cohérence entre l’apparence de régions d’avant-plan vient combler le besoin en information nécessaire pour l’identification des objets saillants qui avait été discuté dans la section 2.1. Fait intéressant, la méthode de cosegmentation de Rother et al. (2006) ne nécessite pas qu’un même objet soit présent dans toutes les images ; différentes instances d’un même type d’objet peuvent aussi être proprement segmentées, tant que ces instances sont visuellement similaires (voir l’exemple de la figure 2.6). Aucune contrainte de forme n’est d’ailleurs imposée aux objets segmentés afin de garder la méthode robuste face aux objets articulés, et invariante par rapport à la pose des caméras. Par contre, cette approche suppose qu’un même modèle d’apparence peut être partagé parmi tous les objets d’intérêt et que leur taille par rapport à la surface de l’arrière-plan est à peu près connue (ce ratio est utilisé comme hyperparamètre). Pour améliorer leurs résultats, Vicente et al. (2011) ont proposé d’incorporer la notion d’*objectness* (“objectivité” ; Carreira and Sminchisescu, 2010) déjà popularisée ailleurs à l’intérieur de leur modèle d’avant-plan. Dans cette nouvelle méthode, de nombreuses propositions de sous-régions d’avant-plan sont générées puis évaluées par un modèle entraîné sur une base de données de régions d’images possédant les caractéristiques d’*objectness* recherchées (e.g. contraste aux frontières, uniformité visuelle). Les sous-régions les plus saillantes sont alors conservées pour la segmentation finale. L’approche de Faktor and Irani (2013) poursuit un objectif similaire en considérant que les “bonnes” sous-régions d’avant-plan partagent certaines particularités qui sont plus rares dans le reste de la scène,

leur permettant d’être assemblées tel un casse-tête.

Par ailleurs, une extension de la cosegmentation au domaine temporel a été proposée par Rubio et al. (2012) en se basant sur la propagation de caractéristiques de saillance et d’*objectness* à travers des “tubes” de superpixels connectés dans un modèle graphique. Cette approche a été généralisée par Chiu and Fritz (2013), qui se sont intéressés à la segmentation simultanée d’objets provenant de plusieurs classes, et par Guo et al. (2013), qui ont travaillé sur le problème de reconnaissance d’actions à partir d’un processus d’appariement de trajectoires.

Notons que de telles méthodes de cosegmentation génériques n’offrent pas de bons résultats lorsque les objets d’intérêt dans les images à segmenter sont de nature trop variée, lorsqu’ils sont difficiles à distinguer de l’arrière-plan, ou encore lorsqu’ils sont cachés par d’autres objets moins intéressants. Les méthodes de cosegmentation interactives (i.e. utilisant une intervention humaine) peuvent résoudre certains de ces problèmes (e.g. Batra et al., 2010), mais elles ne sont pas applicables dans un contexte d’application entièrement automatisée.

2.3.2 Segmentation instance unique (segmentation mutuelle)

Contrairement à la cosegmentation générique, la tâche de segmentation mutuelle définie dans le cadre de cette thèse considère que les images analysées contiennent les mêmes objets à l’avant-plan et qu’elles sont prises à peu près en même temps (voir l’exemple de la figure 2.7). Cela permet donc l’application de certaines contraintes qui ne peuvent pas être utilisées dans le cas générique, par exemple sur la géométrie commune des surfaces d’objets ciblés, et sur leurs mouvements dans le domaine temporel (s’il s’agit de séquences d’images). Par contre, afin de pouvoir bien exploiter l’information provenant de chaque source simultanément, les images analysées doivent être recalées, ce qui n’était pas le cas pour les méthodes de cosegmentation de la section 2.3.1.

Lorsqu’un sous-ensemble des images analysées provient de capteurs de profondeur, la détection d’objets d’intérêt et le recalage de ceux-ci peuvent être accomplis beaucoup plus facilement. Par exemple, Djelouah et al. (2015) ont proposé un modèle probabiliste pour le traitement mixte d’images du spectre visible et de cartes de profondeur dans le but d’obtenir la silhouette 2D des objets d’avant-plan. Leur solution permet de combiner des indices de géométrie et d’apparence des objets d’intérêt afin de déterminer quel partitionnement avant-plan/arrière-plan utiliser dans chaque image, sans nécessiter de reconstruction 3D de la scène.

Le cas de segmentation mutuelle qui nous intéresse plus particulièrement dans le cadre de cette thèse est celui où seules les images de deux caméras sont analysées, et où nous ne

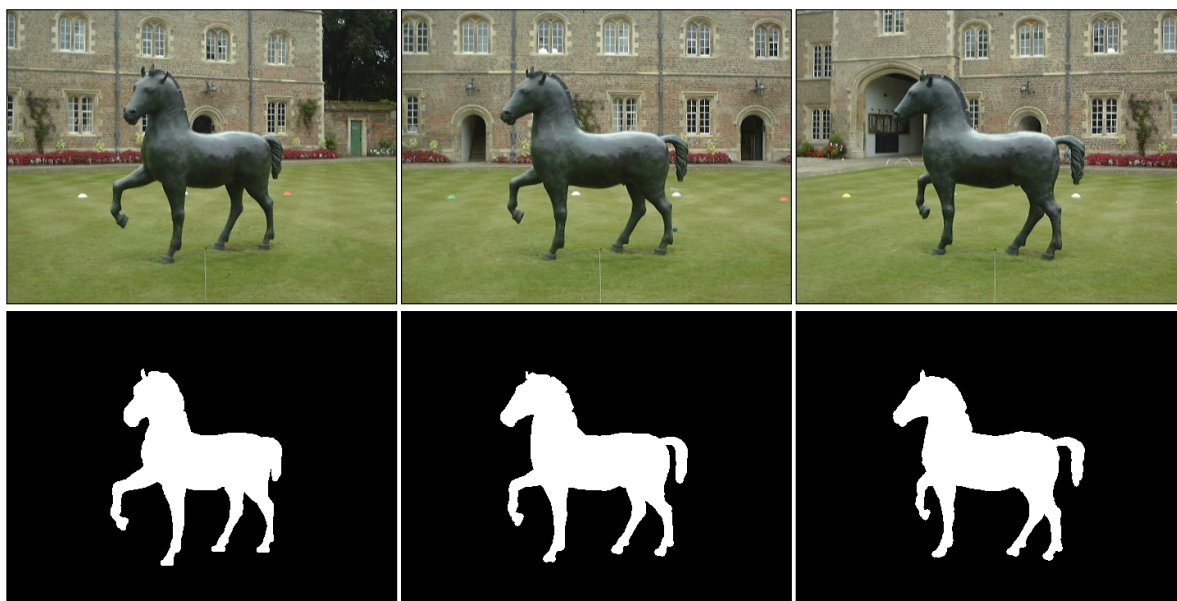


Figure 2.7 Exemple de problème de segmentation mutuelle ; ici, un objet unique est observé dans des conditions similaires, à des moments quasi-identiques, et sous des points de vue semblables. La deuxième ligne illustre les résultats de segmentation attendus. Les images sont tirées du jeu de données de Kowdle et al. (2011).

disposons pas d'information simplifiant le recalage des objets de la scène. Les données nécessaires au recalage doivent donc être déduites des images en même temps que les objets d'intérêt sont détectés et segmentés. Dans ce contexte, le travail le plus pertinent est celui de Riklin-Raviv et al. (2008). Ces auteurs proposent une approche par laquelle les contours d'objets provenant d'une image sont utilisés comme indices dynamiques pour l'optimisation itérative des contours dans l'autre image (et vice-versa). Ils modélisent la transformation de recalage par homographie entre les points de contours extraits, et ils sont capables de correctement segmenter les d'objets d'intérêt même en cas d'occultation et de faible contraste. Leur méthode doit toutefois disposer d'une identification grossière des régions d'intérêt dans chaque paire d'images, et elle nécessite l'utilisation d'un hyperparamètre pour contrôler la force du lissage de formes dans les cas ambigus de segmentation. Notre travail présenté dans le chapitre 7 est en fait une généralisation de cette approche qui traite de multiples objets dans des scènes non-planaires de façon complètement automatisée.

2.3.3 Applications dans le domaine multispectral

Nous discutons finalement de quelques travaux qui ont porté spécifiquement sur les problèmes de recalage et de segmentation de paires d'images multispectrales en simultané, qui est le but ultime du pipeline de traitement proposé dans cette thèse. Notons que les bandes spectrales

utilisées dans ces travaux sont le spectre visible (RGB) et l’infrarouge thermique (LWIR).

D’abord, Torabi et al. (2012) proposent une solution basée sur le recalage par points de trajectoire des objets d’intérêt dans des scènes planaires, et qui fusionne des cartes de segmentation à l’aide d’une approche d’intégration probabiliste. La solution de Zhao and Sen-ching (2014) est relativement semblable ; ceux-ci utilisent une approche de recalage basée sur la mise en correspondance de contours pour des scènes semi-planaires, et utilisent une approche de segmentation considérant les images fusionnées afin de produire une carte de segmentation finale. Puisque l’approche de recalage de ces deux solutions est basé sur la forme des objets d’avant-plan, il leur est impossible de gérer les cas où des objets d’intérêt se croisent, ou lorsqu’ils sont occultés par d’autres éléments de la scène dans une des images. De plus, la segmentation des objets d’intérêt finale produite pour une paire d’images n’est pas considérée dans le but d’améliorer le recalage de ces objets dans une deuxième passe d’optimisation. En d’autres termes, ces solutions ne résolvent pas la problématique mentionnée dans la section 1.2.4 de manière itérative.

D’un autre côté, Davis and Sharma (2007) proposent de combiner le résultat d’une méthode de soustraction d’arrière-plan à une carte d’arêtes saillantes dans le but d’améliorer la fusion d’objets d’intérêt par contours. Leur segmentation finale est alors obtenue en remplissant les cartes de contours combinées par une approche de type *watershed*. Li et al. (2017) proposent quant à eux d’utiliser une méthode de soustraction d’arrière-plan par décomposition globale traitant des paires d’images RGB-LWIR, tel qu’expliqué dans la section 2.1.4. Dans ce cas-ci, le principal défaut des solutions proposées est qu’elles ne peuvent que considérer des scènes parfaitement planaires (i.e. où la parallaxe est négligeable).

Au meilleur de nos connaissances, le travail présenté dans cette thèse est le premier qui s’attaque au problème de recalage et de segmentation de paires d’images multispectrales pour des scènes non-planaires.

CHAPITRE 3 SURVOL DES ARTICLES

Cette thèse propose une structure par articles pour les quatre prochains chapitres. Les articles recensés dans ces chapitres sont directement reliés aux objectifs spécifiques énoncés dans la section 1.3. La combinaison de deux des méthodes proposées dans ces articles (chapitres 5 et 7) constitue une solution globale au problème de segmentation de paires d’images multispectrales sous la forme d’un pipeline de traitement automatisé. Les deux autres articles (chapitres 4 et 6) constituent des travaux préliminaires ayant servi à l’exploration de solutions pour la version finale de notre pipeline de traitement. Les articles choisis et les liens entre ceux-ci sont résumés brièvement ci-dessous.

3.1 Exploration des principes de modélisation et d’ajustements dynamiques

L’article présenté dans le chapitre 4 propose une méthode de segmentation vidéo monoculaire par soustraction d’arrière-plan. Cette méthode se base sur une approche de modélisation non-paramétrique par échantillonnage similaire à ViBe (Barnich and Van Droogenbroeck, 2011), et sur l’usage de mécanismes de rétroaction au niveau de chaque pixel. Ces mécanismes permettent d’ajuster les hyperparamètres d’adaptation du modèle et de détection d’intrus de manière automatique en fonction des données observées. La première contribution de cet article est une étude de l’efficacité des descripteurs locaux binaires (*Local Binary Similarity Pattern*, LBSP) dans la représentation des régions de l’arrière-plan. Combinés à l’information d’intensité locale provenant de l’image, ces descripteurs permettent de mieux caractériser les régions observées dans la scène. Cela se traduit en une meilleure performance de détection des objets d’intérêt, et en une meilleure robustesse du modèle d’arrière-plan face aux changements d’illumination. La deuxième contribution de cet article est l’introduction de deux nouveaux indices utilisés pour l’analyse du niveau de dynamisme de l’arrière-plan de chaque pixel. Le premier indice est basé sur les distances calculées entre les observations et les prédictions du modèle, et le deuxième indice est basé sur la quantité de pixels bruités présents dans les cartes de segmentation produites. Ces indices sont combinés dans de nouveaux mécanismes de rétroaction permettant d’ajuster les hyperparamètres du modèle de chaque pixel. Le but de cette combinaison est d’isoler les changements locaux causés par la présence d’objets d’intérêt à l’avant-plan des changements causés par une région d’arrière-plan instable. La solution de segmentation proposée est évaluée sur un vaste ensemble de données et il est démontré que sa performance surpasse largement celle de dizaines de méthodes déjà publiées. Ce premier article a servi de base pour l’élaboration de notre méthode de segmentation vidéo monoculaire

finale, résumée dans la section 3.2.

3.2 Soustraction d’arrière-plan par modélisation de persistance

L’article présenté dans le chapitre 5 propose une méthode ayant un but similaire à celle de la section 3.1 (chapitre 4). Toutefois, cette nouvelle méthode introduit une approche de modélisation non-paramétrique basée sur l’utilisation de dictionnaires assignant une valeur de persistance à chaque représentation de l’arrière-plan. Inspirées par les poids de la technique des *codebooks* de Kim et al. (2005), ces valeurs de persistance sont calculées en fonction du degré de récurrence des représentations de l’arrière-plan. L’analyse de persistance aide alors à compresser le modèle d’arrière-plan de chaque pixel sans affecter sa complexité, mais permet aussi de raisonner sur les comportements dynamiques réguliers à long terme de l’arrière-plan. En d’autres termes, il devient possible, sur de longues périodes, de modéliser (toujours au niveau de chaque pixel) les régions d’une scène qui changent régulièrement d’apparence, e.g. la chaussée d’une route malgré la présence de voitures temporairement arrêtées. La méthode présentée est donc capable de détecter des objets d’intérêt immobilisés dans une scène même à très long terme. D’autre part, la modélisation par dictionnaire emprunte certains principes de mise à jour des modèles stochastiques afin d’améliorer sa robustesse aux comportements irréguliers de l’arrière-plan, et elle utilise aussi des mécanismes d’ajustement par rétroaction pour contrôler ses hyperparamètres. Finalement, une approche de modélisation d’arrière-plan au niveau de l’image entière est proposée et combinée avec l’analyse de persistance locale. La solution complète est encore une fois évaluée sur un vaste ensemble de données et elle surpasse les performances de l’état de l’art, particulièrement dans les séquences où les objets d’intérêt se déplacent de manière intermittente. Cette méthode est utilisée dans notre pipeline de traitement final afin de détecter les objets d’intérêt présents dans une scène ; elle fournit ses résultats sous forme de cartes de segmentation pour initialiser la méthode de la section 3.4.

3.3 Exploration du recalage par mise en correspondance de contours

L’article présenté dans le chapitre 6 propose une méthode de recalage de paires d’images multispectrales tirées de séquences vidéo synchronisées. La stratégie utilisée est basée sur la mise en correspondance de points de contours obtenus à partir d’une méthode de segmentation vidéo par soustraction d’arrière-plan — il s’agit en fait de la méthode résumée dans la section 3.2. Pour chaque paire d’images, les points de contours des objets d’avant-plan sont décrits à l’aide du descripteur *Shape Context* de Belongie et al. (2002), et assortis entre les images à l’aide de l’algorithme Hongrois (Kuhn, 1955). Les meilleures correspondances sont

alors sauvegardées et maintenues stochastiquement dans un réservoir temporel. Ce réservoir est ensuite utilisé afin de calculer par RANSAC (Fischler and Bolles, 1981) les paramètres d’une transformation de recalage projective permettant de mettre en correspondance tout le contenu des deux images. Cette transformation est réévaluée pour toutes les trames de la vidéo, permettant un recalage continu idéal même pour des scènes “presque planaires”. Ce travail exploratoire a permis de confirmer que le recalage de forme est viable même lorsque les résultats de segmentation utilisés en entrée sont bruités. Cette conclusion a mené en partie à l’élaboration de la stratégie de recalage du chapitre 7 (résumée dans la section 3.4 ci-bas), qui s’étend aux scènes non-planaires dans un processus d’optimisation itératif.

3.4 Recalage et segmentation mutuelle de paires d’images multispectrales

Enfin, l’article présenté dans le chapitre 7 propose une solution complète permettant de résoudre les problèmes de recalage et de segmentation de paires d’images multispectrales simultanément et de manière automatisée. Cette solution est basée sur un processus d’optimisation itératif; d’un côté, des indices locaux d’apparence ainsi que les contours d’objets d’intérêt obtenus par segmentation sont utilisés pour effectuer le recalage, et de l’autre, l’intégration des données multispectrales recalées est utilisée pour améliorer la précision des masques de segmentation. Ces deux objectifs sont exprimés par des fonctions d’énergie qui sont minimisées de manière alternative pour chaque paire d’images. Le modèle de chaque fonction est construit selon une approche graphique permettant de tisser des liens complexes entre tous les pixels d’une image. L’utilisation de tels liens résulte en une solution qui considère le problème d’optimisation de façon globale. Une structure à plusieurs couches permettant le lissage spatiotemporel des résultats produits est aussi proposée. D’autre part, l’initialisation du modèle de segmentation est réalisée à l’aide de masques fournis par la méthode résumée dans la section 3.2. Ces masques permettent l’identification partielle des objets d’intérêt dans la scène, tâche que la solution proposée ici ne peut pas effectuer par elle-même. Les résultats de cette nouvelle solution sont évalués sur de nombreux ensembles de données, et il est démontré que l’approche proposée améliore grandement la qualité des masques initiaux provenant de la segmentation vidéo.

CHAPITRE 4 ARTICLE 1 : SUBSENSE : A UNIVERSAL CHANGE DETECTION METHOD WITH LOCAL ADAPTIVE SENSITIVITY

St-Charles, P.-L., Bilodeau, G.-A., Bergevin, R.

IEEE Transactions on Image Processing, Vol. 24, Issue 1, 2015, pp. 359-373.

(© 2015 IEEE; Reprinted with permission.)

<https://doi.org/10.1109/TIP.2014.2378053>

Abstract

Foreground/background segmentation via change detection in video sequences is often used as a stepping stone in high-level analytics and applications. Despite the wide variety of methods that have been proposed for this problem, none has been able to fully address the complex nature of dynamic scenes in real surveillance tasks. In this paper, we present a universal pixel-level segmentation method that relies on spatiotemporal binary features as well as color information to detect changes. This allows camouflaged foreground objects to be detected more easily while most illumination variations are ignored. Besides, instead of using manually-set, frame-wide constants to dictate model sensitivity and adaptation speed, we use pixel-level feedback loops to dynamically adjust our method’s internal parameters without user intervention. These adjustments are based on the continuous monitoring of model fidelity and local segmentation noise levels. This new approach enables us to outperform all 32 previously tested state-of-the-art methods on the 2012 and 2014 versions of the ChangeDetection.net dataset in terms of overall F-Measure. The use of local binary image descriptors for pixel-level modeling also facilitates high-speed parallel implementations: our own version which used no low-level or architecture-specific instruction reached real-time processing speed on a mid-level desktop CPU. A complete C++ implementation based on OpenCV is available online.

4.1 Introduction

The use of change detection algorithms to identify regions of interest in video sequences has long been a stepping stone in high level surveillance applications. In their simplest form, they allow the subtraction of static background from scenes where relevant objects are always in motion. In most cases however, “foreground” objects may move intermittently (e.g. cars at a traffic light), they may not be focal points in a camera’s field of view, and uninteresting background regions may also exhibit dynamic behavior (e.g. swaying tree branches, water fountains). Simplistic background subtraction methods and traditional image

segmentation approaches are thus ill-suited for active region labeling in real video surveillance tasks. Modern change detection algorithms are generally split into three parts: first, a background model of the scene is created and periodically updated by analyzing frames from the video sequence. Then, preliminary foreground/background segmentation labels (or probabilities) are assigned to all pixels of every new frame based on their similarity to the model. Finally, regularization is used to combine information from neighboring pixels and to make sure uniform regions are assigned homogeneous labels.

Background modeling can be approached in many different ways: to allow high-speed implementations, most methods rely on independent pixel-level models which are assembled into a larger background model. Color intensities are typically used to characterize local pixel representations in non-parametric models or for local distribution estimation in parametric models. Due to its simplicity, this kind of approach cannot directly account for spatiotemporal coherence between pixels and instead delegates this responsibility to the regularization step. This often culminates in less-than-ideal segmentation results due to the absence of texture analysis, especially when camouflaged foreground objects are involved.

Because this is a two-class segmentation problem and because of the wide range of possible scenarios, parameters that control model sensitivity and adaptation rate are usually left to the user to define. These can be very difficult to adjust for particular problems, especially when illumination variations, dynamic background elements and camouflaged objects are all present in a scene at the same time. Additionally, good knowledge of the data and of the change detection algorithm itself is required to achieve optimal performance in any given situation. Previously, most methods have used global thresholds under the assumption that all observations would show similar behavior throughout the analyzed sequences, which is rarely the case. While it is possible to dynamically adjust such parameters based on comparisons between observations and values predicted by the model, this approach cannot be used continuously due to disparities caused by foreground objects. In all cases, a typical outcome of bad parameterization is segmentation noise: usually observed under the form of “blinking pixels” (i.e. pixels that often switch between foreground and background classification over time), such noise indicates that the model is too sensitive to change in a certain region.

In this paper, we present a *universal* method for change detection, meaning it can be directly applied on most video surveillance scenarios and still result in near-optimal performance. Our work has already been partially described before (St-Charles and Bilodeau, 2014; St-Charles et al., 2014). Here, we give new and more detailed explanations on the different parts of our method while providing in-depth analyses of new results. In short, we first use an improved spatiotemporal binary similarity descriptors along with color intensities to

characterize pixel representations in a nonparametric paradigm. This novel approach allows the detection of subtle local changes caused by camouflaged foreground objects, which are not typically visible at the pixel level. It also ignores disturbances caused by soft shadows and other types of illumination variations or noise. Pixel representations are captured and stored as “samples” in pixel-level models, which as a whole form our background model. Due to the conservative update strategy and neighbor-spread rules used to update these samples, our model is resistant to shaking cameras and intermittent object motion.

Then, in order to improve out-of-the-box flexibility in complex scenarios, we propose a new feedback scheme to dynamically control our algorithm’s sensitivity and adaptation speed. This component is based on the continuous analysis of background dynamics at the pixel level: it treats frame regions differently based on recent observations, their similarity to pixel models as well as local segmentation noise levels. While segmentation noise is often seen as detrimental in foreground/background labeling tasks, we show that it can be used to guide feedback when it is managed correctly, thus solving the dynamic adjustment problem in the presence of foreground objects. In our case, the highly sensitive nature of our pixel representations allows sparse noise to be generated very easily, also improving the efficiency of our approach. Ultimately, our feedback scheme can identify complex motion patterns and adjust our model’s parameters locally without compromising segmentation performance elsewhere.

To keep our method’s complexity minimal and its implementation simple, we avoid using preprocessing and color conversion/normalization on analyzed frames. Furthermore, our segmentation decisions do not rely on pixel-wise probabilities or on an energy function, and require no region-level or object-level processing. Our regularization step is also simplified to morphological operations and median blurring, which are able to eliminate all salt-and-pepper segmentation noise. A complete evaluation on the 2012 and 2014 versions of the ChangeDetection.net (CDnet) dataset (Goyette et al., 2012; Wang et al., 2014a) shows that we outperform all 32 previously ranked methods in terms of overall *F-Measure*, as well as in nine out of eleven categories (including baseline). These methods include ViBe (Barnich and Van Droogenbroeck, 2011) and ViBe+ (Van Droogenbroeck and Paquot, 2012), which use a similar sample-based background model that only relies on color, and PBAS (Hofmann et al., 2012), which uses a feedback scheme that does not monitor segmentation noise for its adjustments. In the end, our results demonstrate that our approach is suitable for most complex change detection challenges, but also that good generalization is not earned at the expense of performance in simple scenarios. The full C++ implementations of our method’s two stages presented in Section 4.3 (i.e. with and without feedback) are available online.

4.2 Related Work

Most methods used for change detection in video sequences are based on the idea that, when using stationary cameras, disparities between an analyzed frame and a background reference are usually indicative of foreground objects. The advantage behind this concept is that no prior knowledge is required to detect the objects, as long as their appearance differs enough from the background (i.e. they are not camouflaged). As opposed to solutions based on object detection, this approach can accurately describe the contour of moving objects instead of simply returning their bounding box. However, finding a good reference image in order to do actual “background subtraction” is almost always impossible due to the dynamic nature of real-world scenes.

Instead of relying on an existing reference image for change detection, the earliest adaptive methods used pixel-wise intensity averages and Kalman filtering to create parametric background models from which comparisons are made. This kind of approach is robust to noise and can slowly adapt to global illumination variations, but is generally inadequate against shadows and multimodal background regions. Gaussian Mixture Models (GMM, Friedman and Russell, 1997; Stauffer and Grimson, 1999) were introduced to solve the latter problem and remain to this day a very popular solution. They allow dynamic background elements to be modeled through color intensities at individual pixel locations using a mixture of Gaussian probability density functions. New adaptive and more flexible variations of GMM were also proposed over the years (Zivkovic, 2004; Zivkovic and van der Heijden, 2006; Lee, 2005; KaewTraKulPong and Bowden, 2002) to allow dynamic numbers of components for modeling as well as to improve their convergence rate.

Nonparametric models based on Kernel Density Estimation (KDE, Elgammal et al., 2000) were also introduced early on and improved in more recent state-of-the-art methods (Mittal and Paragios, 2004; Sheikh and Shah, 2005; Zivkovic and van der Heijden, 2006). Unlike parametric models, these rely directly on local intensity observations to estimate background probability density functions at individual pixel locations. Most of them however only incorporate observations on a first-in, first-out basis, and are thus unable to model both long-term and short-term periodic events without holding on to large amounts of data. The stochastic sampling approach presented by Barnich and Van Droogenbroeck (2011); Van Droogenbroeck and Barnich (2014) and improved by Van Droogenbroeck and Paquot (2012); Hofmann et al. (2012) solves this problem by using a random observation replacement policy in the model. Note that our own approach is derived from it. The codebook methods of Kim et al. (2004); Wu and Peng (2010) present another alternative to solve this problem: they cluster observations into codewords and store them in local dictionaries, allowing for a wider range of

representations to be kept in the background model. Kim et al. (2004) also proposed a normalized color distance measure that quickly gained popularity in change detection due to its relatively low cost and robustness to illumination variations. Non-parametric models are also often considered for hardware or high-speed parallel implementations due to their data-driven nature (Barnich and Van Droogenbroeck, 2011). For example, the method of Wang and Dudek (2014), which relies on pixel-level template matching, reported on the CD-net website exceeding 800 frames per second while processing 320x240 videos using a single mid-level GPU (GTX 460).

Unprecedented methods based on artificial neural networks have been proposed and achieve good results on various change detection scenarios without prior knowledge of the involved motion patterns (Maddalena and Petrosino, 2008, 2012). However, they require a training period of variable length depending on the presence of foreground objects in the early frames of the video sequences. To address this problem, weightless neural networks based on binary nodes were instead used for online learning by Gregorio and Giordano (2014). Other works have also focused on improving foreground/background label coherence using advanced regularization techniques based on connected components (Van Droogenbroeck and Paquot, 2012; Morde et al., 2012), superpixels and Markov Random Fields (Schick et al., 2012), or even static/abandoned object detection (Evangelio and Sikora, 2011; Morde et al., 2012). Some methods instead rely on region-level (Jodoin et al., 2012), frame-level (Oliver et al., 2000; Tsai and Lai, 2009) or hybrid frame/region-level (Toyama et al., 1999; Nonaka et al., 2012) comparisons to explicitly model the spatial dependencies of neighboring pixels. The recently proposed method of Wang et al. (2014b) also relies on a hybrid, multi-level system: it combines flux tensor-based motion detection and classification results from a Split Gaussian Mixture Model (SGMM), and uses object-level processing to differentiate immobilized foreground objects from “ghost” artifacts.

The use of local binary descriptors to improve the spatial awareness of change detection methods has also been studied: Heikkila and Pietikäinen (2006) were the first to propose a solution based on Local Binary Patterns (LBP). Their method was demonstrated to be tolerant to illumination variations and robust against multimodal background regions. This robustness is achieved using LBP histograms at the pixel level, but at the expense of sensitivity to subtle local texture changes. An improved version of this method was proposed by Zhang et al. (2008): by computing LBP feature histograms on consecutive frames and then merging them, they benefit from using underexploited temporal motion information. Another approach was suggested by Liao et al. (2010) that does not require histograms: Scale-Invariant Local Ternary Patterns (SILTP) can be used directly at the pixel level to detect local changes when incorporated in a modified KDE framework. Finally, Local Binary Similarity Patterns

(LBSP) were demonstrated by Bilodeau et al. (2013) to surpass traditional color comparisons when used to detect subtle changes in baseline scenarios via Hamming distance thresholding. We chose to improve upon LBSP features for our complete method due to their simplicity and effectiveness in change detection (detailed in Section 4.3.1). We also chose not to use histograms in our model, and instead we directly incorporate LBSP descriptors in a simplified density estimation framework based on a step kernel (like the one used by Barnich and Van Droogenbroeck, 2011).

As for the use of feedback mechanisms to adjust parameters on-the-fly in this context, many parametric methods (such as Stauffer and Grimson, 1999; Zivkovic, 2004; Zivkovic and van der Heijden, 2006) already used local variance analysis or comparison results to guide segmentation behavior. Moreover, post-processing components are sometimes used to trigger model update mechanisms, like in the case of static object detectors (Evangelio and Sikora, 2011; Morde et al., 2012) or frame-level components (Toyama et al., 1999). However, feedback is rarely used solely at the pixel level to control both change detection sensitivity and model update rate. A notable exception is the Pixel-Based Adaptive Segmenter (PBAS, Hofmann et al., 2012), which exploits “background dynamics” (i.e. the study of background motion patterns and model fidelity) to control local decision thresholds and update rates. In Section 4.3.3, we improve upon their design by allowing continuous background monitoring and by restraining parameter adjustments to regions with unstable segmentation behavior. These regions are identified through the detection of blinking pixels, which were also used heuristically by Van Droogenbroeck and Paquot (2012) to guide model updates.

For more information on foreground/background segmentation via change detection, note that many review papers and surveys have been published over the years on the subject (Parks and Fels, 2008; Herrero and Bescós, 2009; Benezeth et al., 2010; Brutzer et al., 2011; Jodoin et al., 2014).

4.3 Methodology

As stated in the previous section, our proposed approach is based on the adaptation and integration of Local Binary Similarity Pattern (LBSP) features in a nonparametric background model that is then automatically tuned using pixel-level feedback loops. We coined our complete method SuBSENSE, short for “Self-Balanced SENSitivity SEgmenter”. We detail how it works in three steps: first, we show in Section 4.3.1 how individual pixels are characterized using spatiotemporal information based on RGB values and LBSP features; then, we present in Section 4.3.2 how these representations can be gathered, updated and used in a stochastic, sample-based model, resulting in a fairly simple and effective change detection

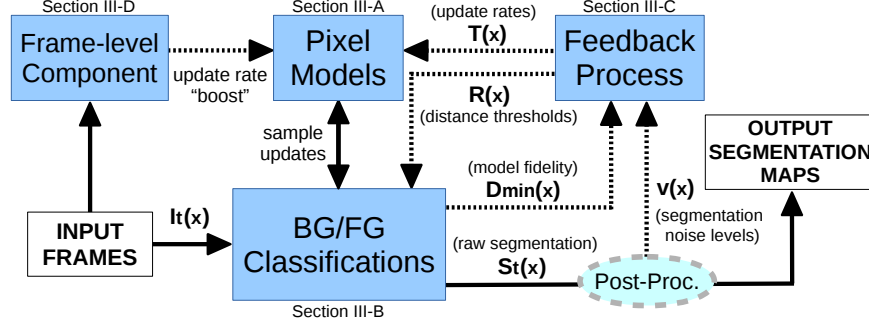


Figure 4.1 Block diagram of SuBSENSE; dotted lines indicate feedback relations. The role of each block and variable is detailed in Sections 4.3.1 through 4.3.4. In our case, post-processing used to generate the output segmentation maps from raw labeling data is based on typical median filtering and blob smoothing operations. This component is an integral part of our method since it provides segmentation noise levels to our feedback process.

method; finally, we show in Section 4.3.3 how model fidelity and segmentation noise monitoring drives the feedback scheme we used to control our algorithm’s sensitivity and adaptation speed locally. Extra implementation details about our method and its frame-level analysis component are given in Section 4.3.4.

An overview of SuBSENSE’s architecture is presented in Fig. 4.1 based on a block diagram.

4.3.1 Pixel-level modeling

Pixel-level modeling, as opposed to region-level or object-level modeling, usually allows high-speed parallel implementations to be developed with relative ease due to how the workload is already split and kept isolated at a low level. However, the absence of information sharing between such local models puts the entire burden of spatial (or spatiotemporal) labeling coherence on the method’s regularization scheme. To counter this, we characterize pixel-level representations using not only their RGB values, but Local Binary Similarity Pattern (LBSP) features, which operate in the spatiotemporal domain. This approach improves the odds of detecting camouflaged objects when their texture differs from the background’s, and can even tolerate illumination changes when all local color intensities vary equally over time. Moreover, these features have a very low computational cost, and are discriminative enough to be used directly in pixel models without relying on local histograms.

As described by Bilodeau et al. (2013) and shown in Fig. 4.2, LBSP features are computed on a predefined 5x5 grid. They can be considered a counterpart to Local Binary Pattern (LBP) and Local Ternary Pattern (LTP) features: instead of assigning binary codes based on whether a given adjoining intensity is lesser or greater than the central reference, they

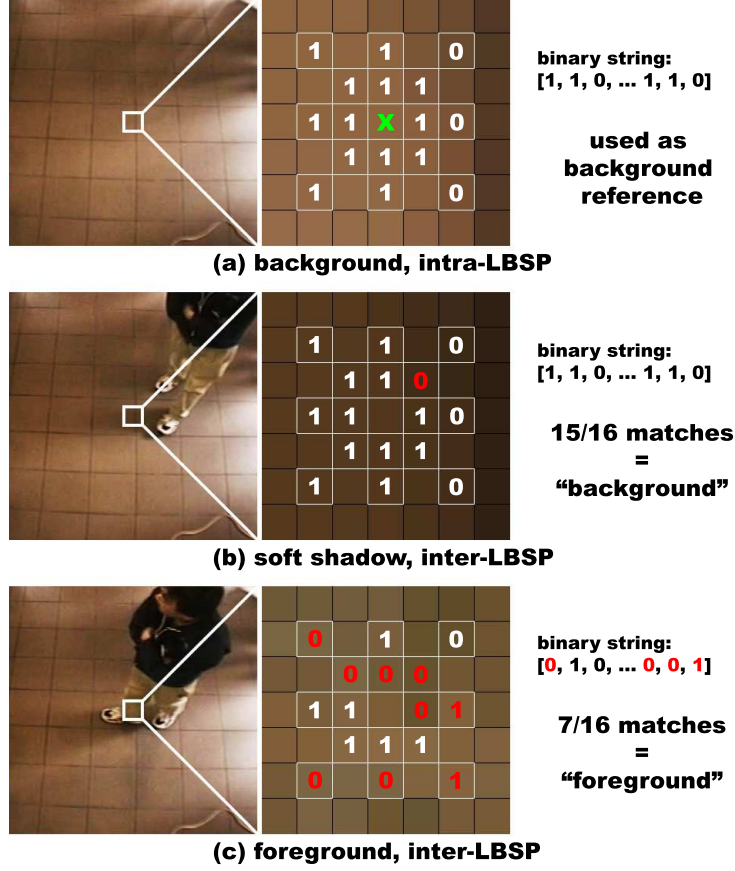


Figure 4.2 Simplified description and comparison of LBSP features using frames picked from the copyMachine sequence of CDnet. In row (a), an intra-LBSP feature is computed to serve as the basis for other comparisons; in rows (b) and (c), inter-LBSP features are computed using (a)’s reference intensity (the green x), while soft shadow and foreground are respectively visible. Row (b)’s feature can be matched with (a)’s since textures are similar and the relative intensity threshold is correctly scaled in bright-to-dark transitions; this is not the case in row (c). Note that in fact, LBSP features are computed and compared on each color channel.

assign them based on similarity (via absolute difference thresholding). More specifically, the following equation is used to compute an LBSP binary string centered at a given location x :

$$LBSP(x) = \sum_{p=0}^{P-1} d(i_p, i_x) \cdot 2^p \quad (4.1)$$

with

$$d(i_p, i_x) = \begin{cases} 1 & \text{if } |i_p - i_x| \leq T_d \\ 0 & \text{otherwise} \end{cases}, \quad (4.2)$$

where i_x is the “central reference” and corresponds to the intensity of the pixel at x , i_p corresponds to the intensity of the p th neighbor of x on the predefined pattern, and T_d is the

internal similarity threshold. In short, LBSP features require the same number of internal operations per comparison as LBP and fewer than LTP, but are more discriminative in the context of change detection (see the experimental results of Section 4.4.1). Furthermore, the pattern seen in Fig. 4.2 covers more pixels than the basic version of LBP or LTP (typically the 8-connected neighbors) without having to interpolate intensity values. LBSP features can also be made sensitive to spatiotemporal variations by picking a central reference intensity (i_x) from a previous frame. This is called inter-LBSP by the original authors, in opposition to intra-LBSP when computations are kept within a single frame. Examples of inter-LBSP computations are also presented in Fig. 4.2.

The first major improvement we propose to traditional LBSP features is related to their internal threshold, T_d . As discussed by Liao et al. (2010), the replacement of a similar threshold used in LTP by a term that is relative to the reference intensity (i_x) makes the binary descriptors much more resistant to illumination variations. In our case, due to the nature of inter-LBSP features, this modification becomes even more effective against shadows, as presented in the middle row of Fig. 4.2. Simply put, since we always use a reference intensity from a previous frame (Fig. 4.2.a, the green X mark) to compute inter-LBSP descriptors (Fig. 4.2.b and c), and since shadows are always characterized by bright-to-dark color transitions, the similarity threshold will remain high and local pixels are likely to stay classified as background. However, in the case of dark-to-bright transitions (which are less likely to be shadows, and more likely to be relevant changes), the similarity threshold will remain low, thus avoiding possible false background classifications. In short, to apply this modification to LBSP features, we only need to replace (4.2) by the following equation, where T_r is the new relative internal threshold (bound to $[0,1]$):

$$d(i_p, i_x) = \begin{cases} 1 & \text{if } |i_p - i_x| \leq T_r \cdot i_x \\ 0 & \text{otherwise} \end{cases} \quad (4.3)$$

We determined experimentally that using $T_r \approx 0.3$ resulted in noise-free, discriminative descriptors in nearly all tested video sequences. However, for optimal flexibility in our final method, this value was automatically scaled over time based on the texture content of the analyzed scenes so that all videos would bear approximately the same overall gradient magnitude sums. That way, scenes with very little background texture or few object edges would rely on much lower T_r values than scenes with cluttered backgrounds, making the former much more sensitive to local texture variations than the latter. We measure the gradient magnitude of individual frames by summing the Hamming weights (rescaled to $[0,1]$) of all its intra-LBSP descriptors. We then slowly adjust T_r if this measure is too low or too high

(i.e. outside a pre-determined interval). In Fig. 4.3, we can see that the highway sequence of CDnet (a) is not affected by these automatic adjustments, as it presents a good balance between cluttered and flat regions. On the other hand, the mostly texture-less blizzard sequence (b) automatically settles for $T_r \approx 0.1$, which accentuates foreground object contours, and the fall sequence causes T_r to increase above 0.3, reducing potential segmentation noise induced by noisy texture patterns in trees.

In summary, for our nonparametric pixel models, we define a single background pixel representation (or “sample”, as referred to in the following sections) in RGB space as a combination of color intensities (8-bit values) and intra-LBSP binary strings (16-bit vectors). We do not convert RGB values to a format with normalized brightness as we determined that the extra computational cost was not worth the improved resistance to illumination variations, to which LBSP features already provide ample flexibility. When trying to match a background sample to an actual observation on a given frame, we first compare color values using L1 distance. If a match is still possible after thresholding, we generate inter-LBSP binary strings using the color values of the sample as reference and using the 5x5 comparison grid on the current input frame. Then, inter-LBSP and intra-LBSP strings (see Fig. 4.2) are compared via Hamming distance thresholding. Therefore, to consider a background sample similar to a local observation, both color values and binary descriptors must be successfully matched. In contrast to a purely color-based approach, this two-step verification is inclined to reject more matches, thus to classify more pixels as foreground (and therefore generate more segmentation noise, as desired).

In the next section, we show how this kind of pixel-level modeling can be used in a basic sample consensus framework to achieve good change detection performance. Note that we tested our pixel models using different local binary descriptors and we present our results in Section 4.4.1.

4.3.2 Change Detection via Sample Consensus

To be properly used for change detection, our pixel-level modeling approach requires maintenance and classification rules that do not rely on the clustering or averaging of samples, as LBSP features cannot easily be combined. Thus, we opted for a basic sample consensus approach similar to ViBe’s (Barnich and Van Droogenbroeck, 2011): derived from the work of Wang and Suter (2007), it determines if a given observation should be considered foreground or background based on its similarity to recently observed samples. This sample-based framework allows our *color-LBSP* pixel representations to be used effectively to detect even the most subtle local changes while staying quite robust to irrelevant motion in complex

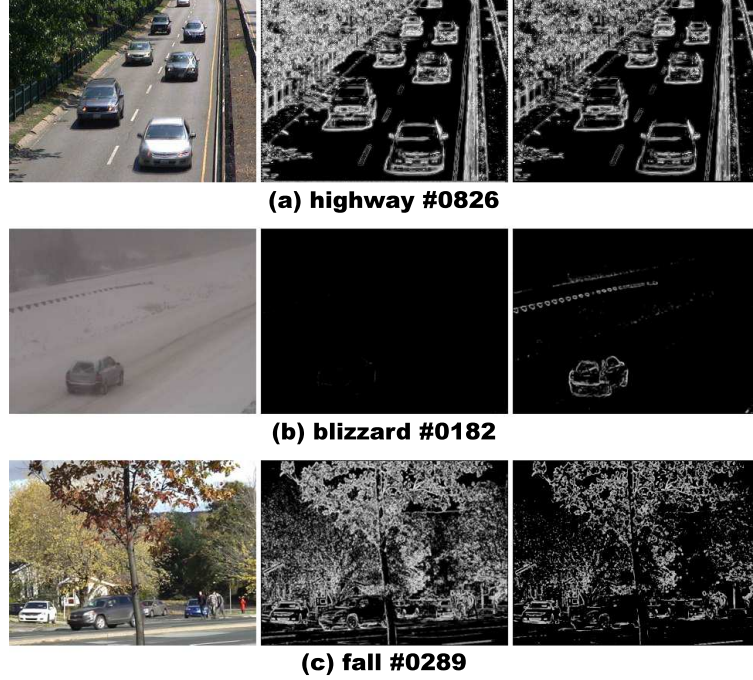


Figure 4.3 Examples of gradient magnitude maps obtained in different CDnet sequences by computing the Hamming weight of dense intra-LBSP descriptors with $T_r = 0.3$ (central column) and an automatically set T_r (right-hand column); bright areas indicate highly textured regions and sharp edges, while dark areas indicate flat regions.

scenarios.

The way it works is rather simple: first, the background model, noted B , is formed through the combination of pixel models, which each contain a set of N recent background samples:

$$B(x) = \{B_1(x), B_2(x), \dots, B_N(x)\} \quad (4.4)$$

These samples, as described in the previous section, are matched against their respective observation on the input frame at time t , noted $I_t(x)$, to classify the pixel at coordinate x as foreground (1) or background (0). A simplified version of this classification test where we ignore our two-step *color-LBSP* verification is presented in (4.5); pseudocode for the two-step approach is presented in the work of St-Charles and Bilodeau (2014).

$$S_t(x) = \begin{cases} 1 & \text{if } \#\left\{dist\left(I_t(x), B_n(x)\right) < R, \forall n\right\} < \#_{min} \\ 0 & \text{otherwise} \end{cases} \quad (4.5)$$

where S_t is the output segmentation map, $dist(I_t(x), B_n(x))$ returns the distance between the

current observation and a given background sample, R is the maximum distance threshold and $\#_{min}$ is the minimum number of matches required for a background classification. In this context, a small R value means that the model has to be very accurate in order to successfully classify pixels as background. Using a larger R leads to better resistance against irrelevant change, but also makes it harder to detect foreground objects that are very similar to the background. We further discuss how this parameter is used to obtain individual color and LBSP distance thresholds in Section 4.3.3.

We fixed $\#_{min} = 2$ for our method as it was demonstrated by Wang and Suter (2007); Barnich and Van Droogenbroeck (2011); Van Droogenbroeck and Barnich (2014); Van Droogenbroeck and Paquot (2012); Hofmann et al. (2012) to be a reasonable trade-off between noise resistance and computational complexity. However, the number of samples per pixel model (N) has to be raised above the usual $N = 20$ proposed by Barnich and Van Droogenbroeck (2011): this is due to the larger representation space induced by LBSP features, which are described using 16 bits instead of 8. Typically, N is used to balance the precision and sensitivity of sample-based methods: using fewer samples leads to more sensitive but less precise models, and vice-versa. As it can be seen in Fig. 4.4, when using only color information, the overall *F-Measure* score (which indicates the “balanced” performance of an algorithm) tends to saturate when N reaches 20. Yet, with our own pixel-level modeling approach, it stabilizes at a higher value.

Although these values depend on the complexity of the studied scenes, we determined that based on the 2012 CDnet dataset, a minimum of $N = 35$ samples was required for our method to be used universally, with $N = 50$ being preferred for better precision. Increasing N does not directly affect the average pixel classification time in stable regions, as the first $\#_{min}$ are usually enough to break off the matching process. It can however lengthen it in dynamic background regions (where matches are sparser) and when foreground is present. As a side note, the relative processing time difference between these two configurations was about 15% on the entire 2012 CDnet dataset.

We update pixel models using a conservative, stochastic, two-step approach similar to the one of Barnich and Van Droogenbroeck (2011): first, every time a pixel at x is classified as background using (4.5), a randomly picked sample of $B(x)$ has a $1/T$ probability to be replaced by the observation at $I_t(x)$, where T is a “time subsampling factor”, as defined by Barnich and Van Droogenbroeck (2011). Then, one of the neighbors of $B(x)$ also has a $1/T$ probability of seeing one of its samples replaced by this same observation. This new parameter controls the adaptation speed of our background model: small values lead to high update probabilities (and thus to the rapid evolution of the model) and vice-versa. It is

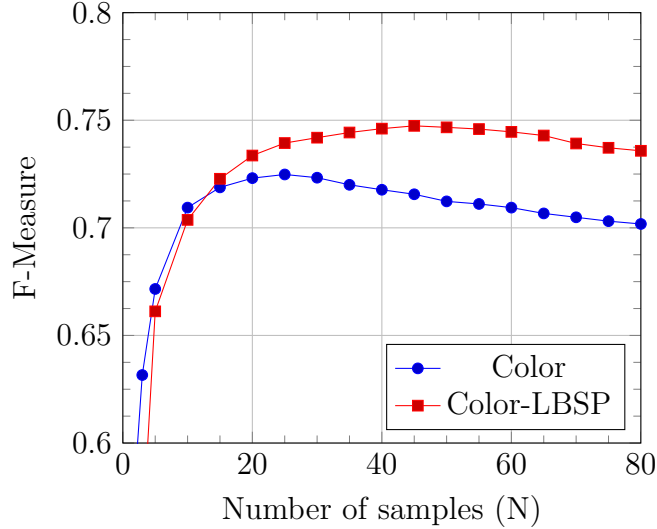


Figure 4.4 Average F-Measure scores obtained on the 2012 CDnet dataset for different numbers of background samples, using *color-only* and *color-LBSP* configurations of our method (without feedback).

adjusted automatically, as described in the next section.

The fact that samples are replaced randomly instead of based on when they were last modified insures that a solid history of long-term and short-term background representations can be kept in our pixel models. Likewise, since new samples can only be inserted when a local pixel is recognized as background, this approach prevents static foreground objects from being assimilated too fast (as is often the case for methods using “blind update” strategies). In theory, this conservative approach implies that, given enough contrast, some foreground objects will never be incorporated into the background model. In practice, noise and camouflage always cause gradual foreground erosion, meaning that all static objects will eventually be classified as background.

The second update step described earlier (i.e. the “spatial diffusion” step, as named by Barnich and Van Droogenbroeck, 2011) allows regions that are homogeneous with the background to be absorbed much faster. In other words, ghost artifacts, which are commonly defined as falsely classified background regions due to the removal of an object from the observed scene, can be eliminated rapidly since they share many similarities with other parts of the background. Moreover, this same “diffusion” step improves the spatial coherency of the background model to the point where limited camera motion can be tolerated. Besides, relying on texture information prevents the spread of samples across object boundaries. Simply put, even if a sample is wrongfully transported from one region to another, the odds that it

might be matched in the new region are much lower due to the use of LBSP features, which would detect a textural change near the border. In fact, a static foreground object with a color similar to the background may be correctly classified for a long time, given that its border is noticeable.

Overall, as presented in Section 4.4.1, this basic method is very robust against global illumination changes and soft shadows, and it can easily detect camouflaged foreground objects that are left unexposed when using a traditional color-based approach. However, using LBP-like features at the pixel level still results in false foreground classifications in most dynamic background regions, as textures are much harder to match than color values. The last part of our method, which is presented in the following section, benefits from this predicament.

4.3.3 Background Monitoring and Feedback Scheme

So far, we have seen how R , the maximum sample distance threshold, and T , the model update rate, are the two most important parameters in our method. They essentially control its precision and sensitivity to local changes and can determine how easily moving elements are integrated in the model. In the works of Barnich and Van Droogenbroeck (2011); Van Droogenbroeck and Paquot (2012), global values were determined empirically for both and used frame-wide. This kind of approach is flawed, as using a global strategy to control model maintenance and labeling decisions implies that all pixels will always present identical behavior throughout the analyzed video sequence. In reality, this assumption almost never holds since an observed scene can present background regions with different behaviors simultaneously, and these can vary over time. Moreover, even if it were possible to fix parameters frame-wide and obtain good overall performance, finding an optimal set of values for a specific application requires time as well as good knowledge of the method and dataset.

So, in our case, we consider R and T pixel-level state variables and adjust them dynamically to avoid these parameterization problems. Ideally, to increase overall robustness and flexibility, we would need to increase R in dynamic background regions to reduce the chance of generating false positives, and T wherever foreground objects are most likely to be immobilized to make sure they do not corrupt the model. However, wrongfully increasing T in dynamic regions can cause the model to stop adapting to useful background representations, and increasing R in static regions can worsen camouflage problems, leading in both cases to even more false classifications. Therefore, to properly adjust these variables, we first need to determine the nature of the region which overlies a given pixel x while avoiding region-level or object-level analyses because they are time-consuming.

A feedback approach based on the analysis of pixel-level background motion patterns (“back-

ground dynamics”) was proposed by Hofmann et al. (2012) for this purpose: it uses the results of recent comparisons between pixel models and local observations to control local distance thresholds and update rates. Their technique, albeit successful, has an important drawback: since local comparison results (i) cannot be used when x is classified as foreground, and (ii) are only used for feedback when $B(x)$ is updated, the response time of variable adjustments is rather long (especially in noisy regions). This means that intermittent dynamic background motion (e.g. swaying tree branches due to wind bursts) will still cause many false classifications.

What we propose instead is a feedback scheme based on a two-pronged background monitoring approach. Similar to the solution of Hofmann et al. (2012), we first measure background dynamics based on comparison results between our pixel models and local observations, but we do this continuously, without any regards to classification results or model updates. This ensures an optimal response time to events in all observed frame regions. Additionally, we measure local segmentation noise levels based on the detection of blinking pixels. This allows dynamic background regions to be distinguished from static regions where foreground objects might be present, essentially guiding which local adjustments should be constrained to avoid over-adaptation and camouflage problems. Two new dynamic controllers for R and T are then introduced, which both rely on local “indicators” emanating from the monitoring of background dynamics and segmentation noise.

So, first of all, the idea behind analyzing background dynamics is to measure the motion entropy of a single pixel location over a small temporal window based on model fidelity. To obtain an indicator of such behavior, we use a recursive moving average, defined as

$$D_{min}(x) = D_{min}(x) \cdot (1 - \alpha) + d_t(x) \cdot \alpha \quad (4.6)$$

where α is the learning rate, and $d_t(x)$ the minimal normalized *color-LBSP* distance between all samples in $B(x)$ and $I_t(x)$. Since $D_{min}(x)$ is bound to the $[0,1]$ interval, an entirely static background region would have $D_{min}(x) \approx 0$, and a dynamic region to which the model cannot adapt to would have $D_{min}(x) \approx 1$. Following the same logic, areas with foreground objects would also present high D_{min} values since foreground detection is defined through disparities between pixel models and local observations. This is why Hofmann et al. (2012) avoided using a similar continuous monitoring approach; in our case, it simply means that we cannot use this indicator by itself to control R and T , as both risk deteriorating when foreground objects stay in the same area for too long. This kind of behavior can be observed in Fig. 4.5: large foreground objects (in this case, cars), just like dynamic background elements, can steadily increase local $D_{min}(x)$ values.

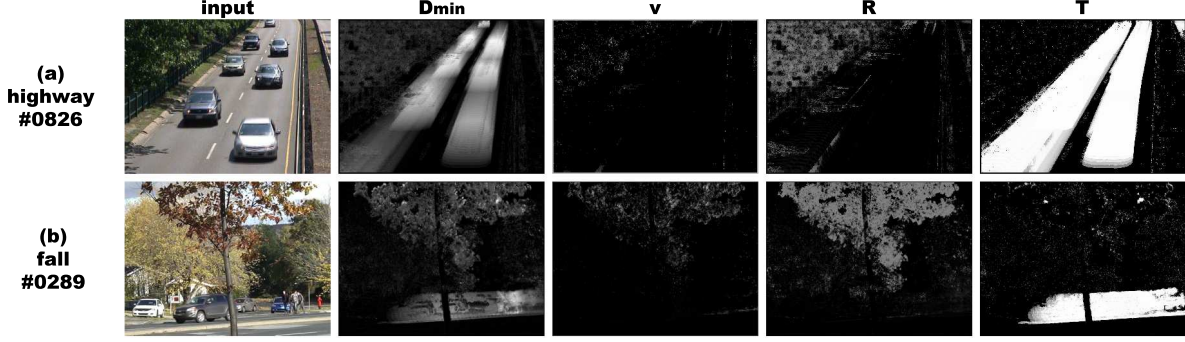


Figure 4.5 Typical 2D distributions for our monitoring variables (D_{min} and v), local distance thresholds (R) and local update rates (T) on a baseline sequence of CDnet (“highway”, row a), and on a sequence with important dynamic background elements (“fall”, row b). In R ’s 2D map, bright areas indicate high distance thresholds (thus easier sample-observation matches), while in T , they indicate low update probabilities (meaning the pixel models stay mostly unchanged over time). We can notice in both cases that trees carry high R and low T values, and vice-versa for road; this means that foreground objects will more easily be detected on the road, and are less likely to corrupt the model over time.

The monitoring of blinking pixels can help solve the complication behind our continuous approach for D_{min} . In this case, it is similar to measuring the segmentation entropy of individual pixel locations, and it allows our method to distinguish noisy regions from purely static regions. This kind of differentiation can guide adjustments so that dynamic background motion triggers feedback mechanisms that regular background or immobile foreground regions cannot. To obtain such an indicator, we first define a 2D map of pixel-level accumulators, noted v . Then, for every new segmented frame S_t , we compute the binary map of all blinking pixels at time t , noted X_t , by using an XOR operation with the previous segmentation results, S_{t-1} . Finally, we update v using

$$v(x) = \begin{cases} v(x) + v_{incr} & \text{if } X_t(x) = 1 \\ v(x) - v_{decr} & \text{otherwise} \end{cases} \quad (4.7)$$

where v_{incr} and v_{decr} are respectively 1 and 0.1, and $v(x)$ is always ≥ 0 . This formulation means that regions with little labeling noise would typically have $v(x) \approx 0$, while regions with unstable labeling would have large positive $v(x)$ values. Directly using an XOR operation to detect blinking pixels can be inconvenient since the borders of moving foreground objects would also be included in the result; this can however be dealt with by nullifying all areas in X_t which intersect with the post-processed and dilated version of S_t . In the end, as shown in Fig. 4.5, the distribution of values in v can help highlight all frame regions where dynamic background elements are truly present, as opposed to regions where change has been recently seen.

With D_{min} and v defined, we can now introduce dynamic controllers for the main parameters of our method. First, local distance thresholds can be recursively adjusted for each new frame using

$$R(x) = \begin{cases} R(x) + v(x) & \text{if } R(x) < (1 + D_{min}(x) \cdot 2)^2 \\ R(x) - \frac{1}{v(x)} & \text{otherwise} \end{cases}, \quad (4.8)$$

where $R(x)$ is a continuous value always ≥ 1 . The exponential relation between $R(x)$ and $D_{min}(x)$ is chosen over a linear relation since it favors sensitive behavior in static regions (and thus helps generate sparse segmentation noise), but also provides robust and rapidly scaling thresholds elsewhere. Here, the segmentation noise indicator $v(x)$ is used as a factor which, in dynamic regions, allows faster threshold increments and can even freeze $R(x)$ in place when $D_{min}(x)$ recedes to lower values. This is particularly helpful against intermittent dynamic background phenomena, as described earlier. However, in static regions, this same factor allows $R(x)$ to stay low even when $D_{min}(x) \approx 1$, which is especially useful when foreground objects are immobilized over x .

Besides, note that in (4.5), R is only an abstract value used to simplify notation; the actual distance thresholds for color and LBSP comparisons are obtained from $R(x)$ using

$$R_{color}(x) = R(x) \cdot R_{color}^0 \quad (4.9)$$

and

$$R_{lbsp}(x) = 2^{R(x)} + R_{lbsp}^0, \quad (4.10)$$

where R_{color}^0 and R_{lbsp}^0 respectively carry the default color and LBSP distance thresholds (30 and 3 in our case). These minima are reached when $R(x) = 1$ and they represent the smallest amount of relevant local change our model can perceive. We define the relation behind $R_{lbsp}(x)$ as nonlinear due to the binary nature of LBSP descriptors and their comparison operator (Hamming distance).

On the other hand, local update rates are recursively adjusted using

$$T(x) = \begin{cases} T(x) + \frac{1}{v(x) \cdot D_{min}(x)} & \text{if } S_t(x) = 1 \\ T(x) - \frac{v(x)}{D_{min}(x)} & \text{if } S_t(x) = 0 \end{cases}, \quad (4.11)$$

where $T(x)$ is limited to the $[T_{lower}, T_{upper}]$ interval (by default, $[2, 256]$), and $D_{min}(x)$ is used concurrently with $v(x)$ to determine the variation step size¹. In short, this relation dictates that regions with very little entropy (both in terms of segmentation and motion) will see rapid

1. Note that in practice, the indeterminate $\frac{0}{0}$ form cannot be achieved in the second right-hand statement of (4.11) as $D_{min}(x)$ never actually reaches 0 due to its infinite impulse response property.

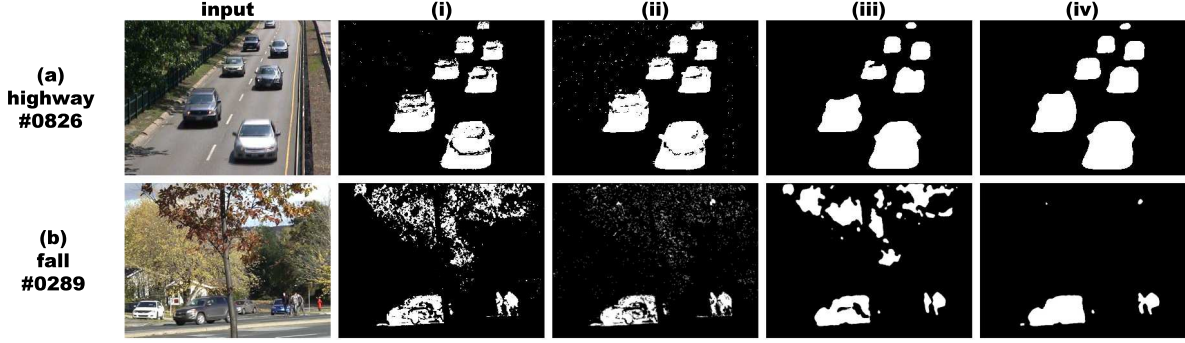


Figure 4.6 Segmentation results obtained with our proposed model on a baseline sequence of CDnet (“highway”, row a), and on a sequence with important dynamic background elements (“fall”, row b), where (i) used no feedback or post-processing, (ii) used feedback but without post-processing, (iii) used post-processing but without feedback, and (iv) used both. While false classifications are present in all variations, we can note that the results of (iv) are the most consistent with object boundaries.

update rate increases (i.e. sudden drops in model update probabilities) whenever pixels are classified as foreground (noted $S_t(x) = 1$). In other words, in regions that are static and motionless ($v(x) \approx D_{min}(x) \approx 0$), once foreground is detected, the model adaptation process will be instantly halted (due to very high $T(x)$). This process can be resumed slowly based on the value of $v(x)$, and only once the foreground is gone. As for dynamic background regions (or otherwise unstable regions), this equation simply dictates that variations will be much smoother, allowing the model to keep adapting to new observations even through continuous foreground classifications.

Overall, our feedback process depends not only on the results of internal comparisons between pixel models and local observations, but also on past labeling results. Due to how $v(x)$ works, our approach is more effective when sparse segmentation noise is present under the form of blinking pixels in most analyzed frame regions. Fortunately, due to the sensitive nature of LBSP descriptors, such noise can easily be generated when local distance thresholds are properly adjusted. Furthermore, the use of a median filter as a post-processing operation eliminates all of it, leaving the actual output of our method intact at a very low cost. We present in Fig. 4.5 the typical 2D distributions of R , T , D_{min} and v on frames showing static and dynamic background regions. In Fig. 4.6 we present segmentation results obtained with and without our feedback process (as well as with and without post-processing). In the latter, we can observe that before post-processing, the feedback-less configuration (Fig. 4.6.i) displays many bad foreground blobs, but less random salt-and-pepper (blinking pixel) noise than the configuration that used feedback (Fig. 4.6.ii). After post-processing however, the version with feedback (Fig. 4.6.iv) is much more reliable. This demonstrates how important sparse segmentation noise is in our feedback process, and how easily we can eliminate it.

4.3.4 Further Details

We stated earlier that our pixel models can handle short-term and long-term motion patterns due to the stochastic nature of their update rules. This is however not the case for our feedback process, as D_{min} is updated using a recursive moving average formula. To counter this, we keep two sets of D_{min} variables simultaneously up-to-date using different learning rates (i.e. a short-term one, $\alpha^{ST} = 25$, and a long-term one, $\alpha^{LT} = 100$). Then, when updating $R(x)$ via (4.8) and $T(x)$ via (4.11), we respectively use the current minimum and maximum $D_{min}(x)$ value between the two moving averages. This modification allows smoother distance thresholds and update rates adjustments, increasing the stability and responsiveness of our feedback scheme against most types of periodic background disturbances.

Also, to improve performance in scenarios with drastic background changes (e.g. “light switch” events) or moving cameras, we added a lightweight frame-level analysis component to our method. Its goal is to automatically scale T_{upper} and T_{lower} and trigger partial model resets in extreme cases (like the frame-level component of Toyama et al., 1999). It works by analyzing discrepancies between short-term and long-term temporal averages of downsampled input frames (using the same learning rates as D_{min}). When disparities are omnipresent and persistent, it indicates that the analysis region suffered from an important change, and that the model should allow faster adaptations frame-wide. Downscaling the input frames allows faster overall processing and helps ignore shaking cameras while temporal averages are used to blur out and reduce the effects of foreground objects on the discrepancies analysis. This solution allows our method to compose with sudden and continuous camera movements, as long as the observed scene presents enough high-level detail. It is much less computationally expensive than analyzing camera movements using optical flow or feature detection/matching techniques. By-products of this component, namely the downsampled input frames and the temporal averages for different CDnet sequences, are shown in Fig. 4.7.

For implementation details on how the default parameters are scaled in this frame-level component, in the feedback process or for grayscale images, the reader is invited to refer to the source code².

4.4 Experiments

To properly evaluate our method, we need to rely on more than a few hand-picked frames from typical surveillance videos. It is very difficult to compare state-of-the-art change detection methods, as many of them were tested on small datasets with few scenarios, and using

2. <https://github.com/plstcharles/litiv>

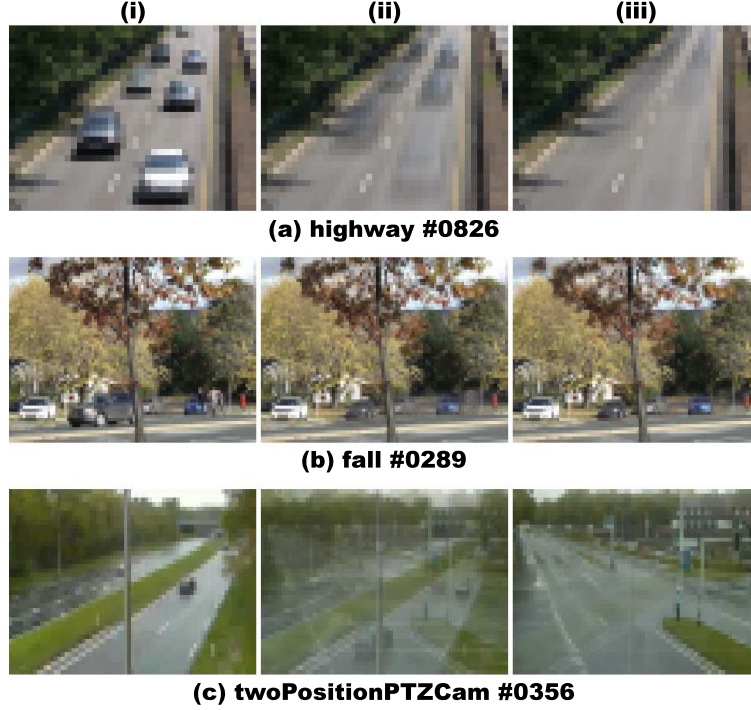


Figure 4.7 By-products of the proposed frame-level analysis component at various times of CDnet sequences; column (i) shows the downscaled input frame used to update the moving averages, and columns (ii) and (iii) respectively show the sequence’s short-term and long-term moving averages. While the difference between these two is negligible for the entire highway (row a) and fall (row b) sequences, it quickly becomes significant in the twoPositionPTZCam sequence (row c) right after the camera rotates. The pixel-wise analysis of discrepancies between these two moving averages allows the detection of such drastic events.

different ground truth sources. Fortunately, a real benchmark was introduced for the 2012 CVPR Workshop on Change Detection (Goyette et al., 2012). Unlike its predecessors, the ChangeDetection.net (CDnet) dataset offers a wide variety of segmentation scenarios set in realistic conditions along with accurate ground truth data. More specifically, $\sim 88,000$ frames obtained from 31 video sequences were manually labeled and split into six categories: baseline, camera jitter, dynamic background, intermittent object motion, shadow and thermal. It was originally tested on 19 state-of-the-art methods, but has since been used to rank dozens more on their website, thus becoming a solid reference for method comparisons. This dataset was also updated for the 2014 version of the same CVPR Workshop (Wang et al., 2014a), adding 22 videos and $\sim 70,000$ annotated frames in five new, much harder categories: bad weather, low framerate, night videos, pan-tilt-zoom and turbulence. In both versions, the official metrics used to rank methods are *Recall* (Re), *Specificity* (Sp), *False Positive Rate* (FPR), *False Negative Rate* (FNR), *Percentage of Wrong Classifications* (PWC), *Precision*

(Pr) and *F-Measure* (FM). For their specific descriptions, refer to the works of Goyette et al. (2012); Wang et al. (2014a).

We primarily use *F-Measure* to compare the performance of different methods as it was found by Goyette et al. (2012) to be closely correlated with the ranks used on the CDnet website, and is generally accepted as a good indicator of overall performance. We chose not to compare methods using their overall ranks on the CDnet website based on three reasons: 1) due to the non-linearity of overall ranks, adding or removing a method from the “comparison pool” (even when it is clearly outperformed by others) can substantially affect how top methods are ranked; 2) since the ranking system relies on both FPR and Sp, which are reciprocal ($FPR = 1 - Sp$), “precise” methods are unduly favored over “sensitive” ones; and 3) because change detection is typically an unbalanced binary classification problem (there are far more background pixels than foreground pixels in analyzed sequences), using PWC as defined by Goyette et al. (2012) once again favors “precise” methods. To provide a better assessment of the overall performance of our method compared to others, we also evaluated top-ranked methods in separate tables using Matthew’s Correlation Coefficient (MCC). This metric is designed to appraise the performance of binary classifiers in unbalanced problems, and is defined by

$$MCC = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}, \quad (4.12)$$

where TP, TN, FP and FN are defined like in the work of Goyette et al. (2012).

As required, we used a unique parameter set for all videos to determine the true flexibility of our method. Like we said earlier, the post-processing operations we use are only based on median blur and morphological operations and serve as our regularization step, eliminating irrelevant blobs/holes from segmentation maps. Note that all our segmentation results can be downloaded online via the CDnet website³.

4.4.1 CDnet 2012

To demonstrate our first key contribution (i.e. *color-LBSP* is preferable to other pixel-level characterization approaches), we present in Table 4.1 how our pixel models fare in comparison to similar alternatives when all are used on the 2012 CDnet dataset under the sample-based approach detailed in Section 4.3.2. Note that these scores are the averages obtained over six scenario categories based on the evaluation technique of CDnet. Here, we can immediately see that 3x3 LBP and SILTP features are ill-suited to this kind of pixel-level modeling. SILTP features can detect texture variations more easily, but just like LBP, they are far too sensitive

3. <http://wordpress-jodoin.dmi.usherb.ca/method/139/>

Table 4.1 Average performance comparison of different model configurations on the 2012 CDnet dataset

Configuration	Pr	Re	FM
Color	0.841	0.672	0.710
LBP	0.350	0.686	0.394
SILTP	0.402	0.628	0.410
LBSP	0.772	0.758	0.721
Color-LBP	0.727	0.782	0.713
Color-SILTP	0.712	0.801	0.715
Color-LBSP	0.743	0.821	0.745
Color-LBSP-Feedback	0.856	0.831	0.826
Color-LBSP-Feedback-LessNoise	0.790	0.877	0.816

Table 4.2 Complete results for SuBSENSE on the 2012 CDnet dataset

Category	<i>Re</i>	<i>Sp</i>	<i>FPR</i>	<i>FNR</i>	<i>PWC</i>	<i>Pr</i>	<i>FM</i>
baseline	0.9520	0.9982	0.0018	0.0480	0.3574	0.9495	0.9503
camera jitter	0.8243	0.9908	0.0092	0.1757	1.6469	0.8115	0.8152
dynamic background	0.7768	0.9994	0.0006	0.2232	0.4042	0.8915	0.8177
interm. object motion	0.6578	0.9915	0.0085	0.3422	3.8349	0.7957	0.6569
shadow	0.9419	0.9920	0.0080	0.0581	1.0120	0.8646	0.8986
thermal	0.8161	0.9908	0.0092	0.1839	2.0125	0.8328	0.8171
overall	0.8282	0.9938	0.0062	0.1718	1.5446	0.8576	0.8260

in most scenarios and cannot properly detect color changes between two frames when textures are unaffected. Nonetheless, both LBP and SILTP configurations obtain decent results when combined with color. According to *F-Measure* scores, only LBSP features are preferable to color information in this framework; this is due to their ability to detect color and texture changes simultaneously. Furthermore, the *color-LBSP* configuration offers performance on par with the best methods tested by Goyette et al. (2012). In general, we can note that configurations that used local binary descriptors were inherently more sensitive to change than the baseline color-based approach (as visible through *Recall* scores), albeit at the expense of segmentation precision. However, among all tested binary descriptor configurations, *color-LBSP* is the best choice because it combines good sensitivity and precision.

We also inserted into Table 4.1 the results obtained using our *color-LBSP* configuration with the proposed feedback process (noted *color-LBSP-Feedback*) to demonstrate our second key contribution, i.e. continuous dynamic parameter adjustments can drastically improve perfor-

Table 4.3 Overall and per-category F-Measure comparisons, CDnet 2012 dataset

Method	overall(2012)	baseline	cam.jitter	dyn.bg.	int.obj.mot.	shadow	thermal
SuBSENSE (proposed)	0.826	0.950	0.815	0.818	<i>0.657</i>	0.899	0.817
Gregorio and Giordano (2013)	<i>0.778</i>	0.908	<i>0.781</i>	0.809	0.567	0.841	0.762
Sedky et al. (2014)	0.777	0.933	0.716	0.787	0.566	<i>0.884</i>	0.776
Haines and Xiang (2014)	0.776	0.929	0.748	<i>0.814</i>	0.542	0.813	<i>0.813</i>
Evangelio and Sikora (2011)	0.766	0.921	0.672	0.688	0.715	0.865	0.735
Hofmann et al. (2012)	0.753	0.924	0.722	0.683	0.575	0.860	0.756
Schick et al. (2012)	0.737	0.929	0.750	0.696	0.565	0.791	0.693
Maddalena and Petrosino (2012)	0.728	<i>0.933</i>	0.705	0.669	0.592	0.779	0.692
Van Droogenbroeck and Paquot (2012)	0.722	0.871	0.754	0.720	0.509	0.815	0.665
Elgammal et al. (2000)	0.672	0.909	0.572	0.596	0.409	0.803	0.742
Barnich and Van Droogenbroeck (2011)	0.668	0.870	0.600	0.565	0.507	0.803	0.665
Stauffer and Grimson (1999)	0.662	0.825	0.597	0.633	0.520	0.737	0.662

^a Note that red-bold entries indicate the best result in a given column, and blue-italics the second best.

mance. We can observe that *color-LBSP* obtains worse overall *Precision* and *Recall* scores (and thus a worse *F-Measure*) than our complete method. This is due to the sensitive nature of our pixel models, which cause many false classifications when they are not properly adjusted to their overlying regions. In the case of *color-LBSP*, a compromise had to be made to reach good overall flexibility (i.e. sensitivity had to be globally lowered to accommodate for complex scenarios). This comparison demonstrates that adjusting local parameters dynamically is extremely beneficial for this kind of approach, especially in terms of *Precision*. Note that for *color-LBSP* and *color-LBSP-Feedback*, the results were obtained by independently tuning their threshold values (i.e. R_{color}^0 and R_{lbsp}^0) for optimal overall *F-Measures*. Interestingly, and as we expected, the *color-LBSP-Feedback* configuration performed much better with lower default thresholds (i.e. with a higher theoretical sensitivity) than *color-LBSP*, even in simple scenarios with completely static backgrounds; this can be explained by the affinity of our feedback scheme for noisier segmentation results. Furthermore, we present how this same feedback scheme performs when some sparse noise is removed using a 3x3 median filter before detecting blinking pixels (under the *color-LBSP-Feedback-LessNoise* configuration). From these results, we can again see that guiding local change sensitivity and update rates using segmentation noise is beneficial, as the *Precision* and *F-Measure* scores obtained for this new configuration are lower than those of *color-LBSP-Feedback*.

The complete results of our best configuration (*color-LBSP-Feedback*, noted SuBSENSE below) on this same dataset are then displayed in Table 4.2. While these numbers may not mean much by themselves, we can see that overall performance in the baseline and shadow

Table 4.4 Complete results for SuBSENSE on the 2014 CDnet dataset

Category	<i>Re</i>	<i>Sp</i>	<i>FPR</i>	<i>FNR</i>	<i>PWC</i>	<i>Pr</i>	<i>FM</i>
bad weather	0.8213	0.9989	0.0011	0.1787	0.4527	0.9091	0.8619
low framerate	0.8537	0.9938	0.0062	0.1463	0.9968	0.6035	0.6445
night videos	0.6570	0.9766	0.0234	0.3430	3.7718	0.5359	0.5599
pan-tilt-zoom	0.8306	0.9629	0.0371	0.1694	3.8159	0.2840	0.3476
turbulence	0.8050	0.9994	0.0006	0.1950	0.1527	0.7814	0.7792
overall (2014)	0.7935	0.9863	0.0136	0.2064	1.8378	0.6228	0.6386
overall (2012+2014)	0.8124	0.9904	0.0096	0.1876	1.6780	0.7509	0.7408

Table 4.5 Overall and per-category F-Measure comparisons, CDnet 2014 dataset

Method	overall(all)	overall(2014)	bad weather	low framerate	night videos	ptz	turbulence
SuBSENSE (proposed)	0.741	0.639	0.862	0.645	0.560	<i>0.348</i>	0.779
Wang et al. (2014b)	<i>0.728</i>	<i>0.600</i>	<i>0.823</i>	0.626	<i>0.513</i>	0.324	0.713
Gregorio and Giordano (2014)	0.681	0.549	0.684	0.641	0.374	0.322	0.723
Sedky et al. (2014)	0.673	0.558	0.757	<i>0.644</i>	0.483	0.365	0.543
Wang and Dudek (2014)	0.658	0.501	0.767	0.469	0.380	0.135	<i>0.756</i>
Maddalena and Petrosino (2012)	0.596	0.437	0.662	0.546	0.450	0.041	0.488
Zivkovic and van der Heijden (2006)	0.594	0.492	0.759	0.549	0.420	0.213	0.520
Elgammal et al. (2000)	0.571	0.445	0.757	0.548	0.436	0.037	0.448
Stauffer and Grimson (1999)	0.569	0.461	0.738	0.537	0.410	0.152	0.466

^a Note that red-bold entries indicate the best result in a given column, and blue-italics the second best.

categories is very good, and our method’s *Recall* is generally high. These two categories consist of simple sequences where pedestrians and cars are the main focus with various types of camouflage and illumination variation problems involved. We can also notice that the intermittent object motion category poses the biggest challenge; this is true for any change detection solution that does not focus explicitly on static object detection and segmentation. This category mostly contains videos where objects are abandoned and parked cars suddenly start moving. The camera jitter and dynamic background scenarios are well handled by our approach, as in both cases, overall *F-Measure* and *Precision* are above 80%. The same can be said for thermal-infrared videos; in this case, our automatic distance threshold adjustments allow very good *Recall* despite numerous important camouflage problems in all sequences.

We show in Tables 4.3 and 4.6 how our method compares to recent and classic state-of-the-art solutions, based on their overall 2012 CDnet results. Due to a lack of space, we only listed the classic and top-ranked methods out of the 32 that were published as of July 2014.

In Table 4.3, SuBSENSE is better in five out of six categories, with a 6.2% relative overall *F-Measure* improvement over the previous best method (CwisarD, Gregorio and Giordano, 2013). Our performance in the intermittent object motion category is well above the average of the methods tested in Goyette et al. (2012), and is only surpassed by Evangelio and Sikora (2011), who used static object detection to specifically target this kind of scenario. While it cannot be deduced from the results shown here, our per-category *Recall* scores are the main reason why our *F-Measures* are so high compared to other methods; this again demonstrates that our approach can handle most camouflage problems due to its ability to detect very subtle spatiotemporal changes and locally adjust its sensitivity to change. The smaller overview shown in Table 4.6 also support this conclusion: both our overall Recall and Precision scores are much higher than most others. We can also note that overall MCC scores are somewhat correlated with F-Measure scores, and a similar gap between our method and the second best is visible. Typical segmentation results for our proposed method as well as for the next best method (Spectral-360, Sedky et al., 2014) and a classic one (GMM, Stauffer and Grimson, 1999) are shown in Fig. 4.8.

4.4.2 CDnet 2014

For the 2014 update of the CDnet dataset, we first show in Table 4.4 the complete results of SuBSENSE on all new scenario categories. It is easy to see how these new scenarios are much harder to deal with than the original ones: only bad weather and turbulence scenarios seem to result in acceptable performance (with *F-Measures* greater than 80%). The former consists of outdoor surveillance footage taken under snowy conditions and the latter shows long distance thermal-infrared video surveillance with important air turbulence due to a high temperature environment. Results in the pan-tilt-zoom category, on the other hand, are our worse so far; in this kind of scenario, the basic assumption behind all low-level change detection methods, i.e. the camera remains static, is violated. Our frame-level analysis component allows our method to have a minimum level of functionality, but despite the apparent simplicity of most sequences, half of the segmentation results would hardly be of any use to other applications. Using a more sophisticated, high-level approach would result in better performance than a pixel-level change detection solution in this kind of scenario. Besides, even though scores in the low framerate category are generally low, in reality, our method performed well on all but one video. In this particular sequence, a marina is filmed at 0.17 frames per second under wavering global lighting conditions while many background elements (boats, water) show intense dynamic behavior. Finally, night videos also pose a significant challenge: the footage in this category is only taken from urban traffic monitoring cameras at night, meaning that photon shot noise, compression artifacts, camouflaged objects and glare effects from car

Table 4.6 Average performance comparison of different methods on the 2012 CDnet dataset

Method	Pr	Re	FM	MCC
SuBSENSE	0.858	0.828	0.826	0.827
Sedky et al. (2014)	<i>0.846</i>	0.777	<i>0.777</i>	<i>0.784</i>
Haines and Xiang (2014)	0.793	<i>0.827</i>	0.776	0.782
Evangelio and Sikora (2011)	0.834	0.770	0.766	0.776
Gregorio and Giordano (2013)	0.774	0.818	0.778	0.773
Hofmann et al. (2012)	0.816	0.784	0.753	0.766

Table 4.7 Average performance comparison of different methods on the 2014 CDnet dataset

Method	Pr	Re	FM	MCC
SuBSENSE	0.751	0.812	0.741	0.754
Wang et al. (2014b)	<i>0.770</i>	<i>0.766</i>	<i>0.728</i>	<i>0.753</i>
Gregorio and Giordano (2014)	0.773	0.661	0.681	0.709
Sedky et al. (2014)	0.705	0.735	0.673	0.703
Wang and Dudek (2014)	0.716	0.704	0.658	0.671
Maddalena and Petrosino (2012)	0.609	0.762	0.596	0.615

headlights must all be handled simultaneously.

We then compare SuBSENSE to all other top-ranked methods tested on this same dataset in Tables 4.5 and 4.7. First, in Table 4.5, our *F-Measure* scores once again stand out as well above average. In fact, even in the PTZ category, all evaluated methods obtained similar or worse scores, despite some of them using more sophisticated frame-level motion analysis approaches. Overall, we still obtain the best *F-Measure* scores in four out of five categories, and surpass the second best method with a 6.5% relative improvement in overall *F-Measure* for the 2014 dataset only. This indicates that our method is extremely flexible and can adapt even to the most difficult change detection scenarios. In Table 4.7, we can see that MCC is not as correlated with F-Measure as it was for the 2012 dataset; our method still achieves the best overall score, but only marginally. Note that since part of the 2014 dataset is withheld for online testing, all MCC scores were computed based on the publicly available dataset, which may explain this difference. Besides, the second and third best methods, FTSG (Wang et al., 2014b) and CwisarDH (Gregorio and Giordano, 2014), receive noticeably better overall Precision scores. Again, we show in Fig. 4.9 segmentation results for our proposed method as well as for the second best one (FTSG) and a classic one (GMM). Note that for all 2014

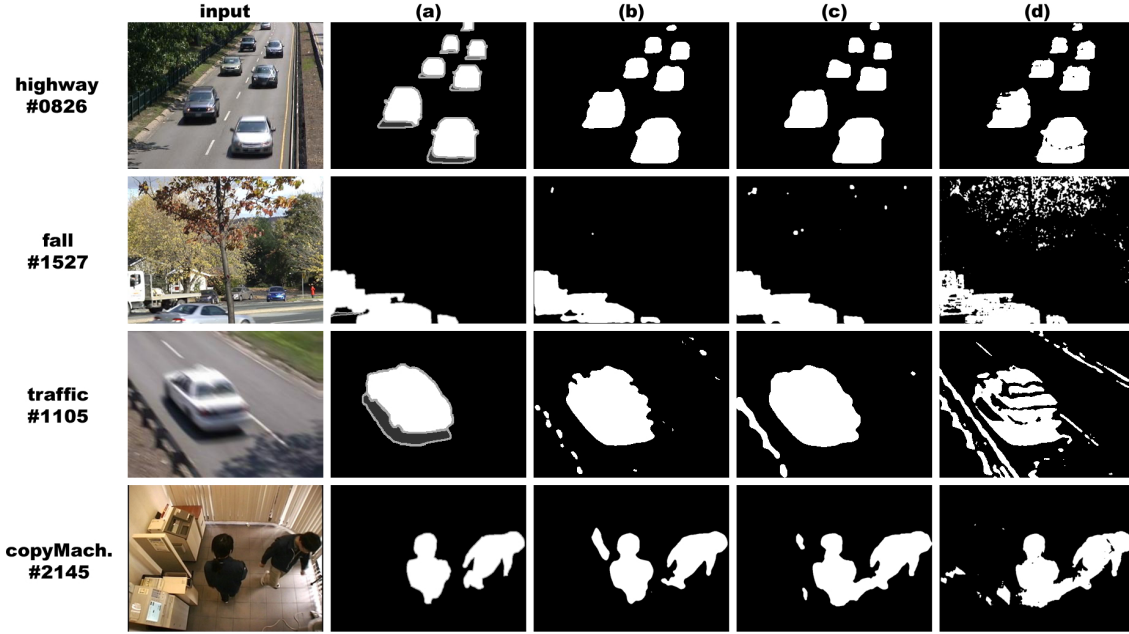


Figure 4.8 Typical segmentation results for various sequences of the 2012 version of the CDnet dataset; column a) shows groundtruth maps, b) shows our segmentation results, c) Spectral-360’s results and d) GMM’s results. From top to bottom, the sequences are highway (from the baseline category), fall (dynamic background), traffic (camera jitter), and copyMachine (shadow). Note that gray areas are not evaluated.

results, we used the same parameters as for the 2012 dataset.

4.4.3 Processing speed

We did not optimize any component of our final method; it ran on a third generation Intel i5 CPU at 3.3 GHz with no architecture-specific instruction, using OpenCV’s background subtraction interface (C++) and saving output frames on a local hard drive. Nonetheless, the feedback-less version (*color-LBSP* in Section 4.4.1) clocked an average of over 90 frames per second on the entire 2012 CDnet dataset, and the complete method (SuBSENSE) processed the same dataset at 45 (both were running one sequence per CPU core). Comparing these results with those of recent state-of-the-art methods on the same dataset is impossible due to the wide variety of platforms involved for testing, the lack of a common reference and the rarity of openly available source code.

Besides, our method can be fully initialized with a single image, requires no training and performs all dynamic adjustments on-line, for every new frame. The low-level nature of our approach also favors high-speed parallel implementations, as no region-level or object-

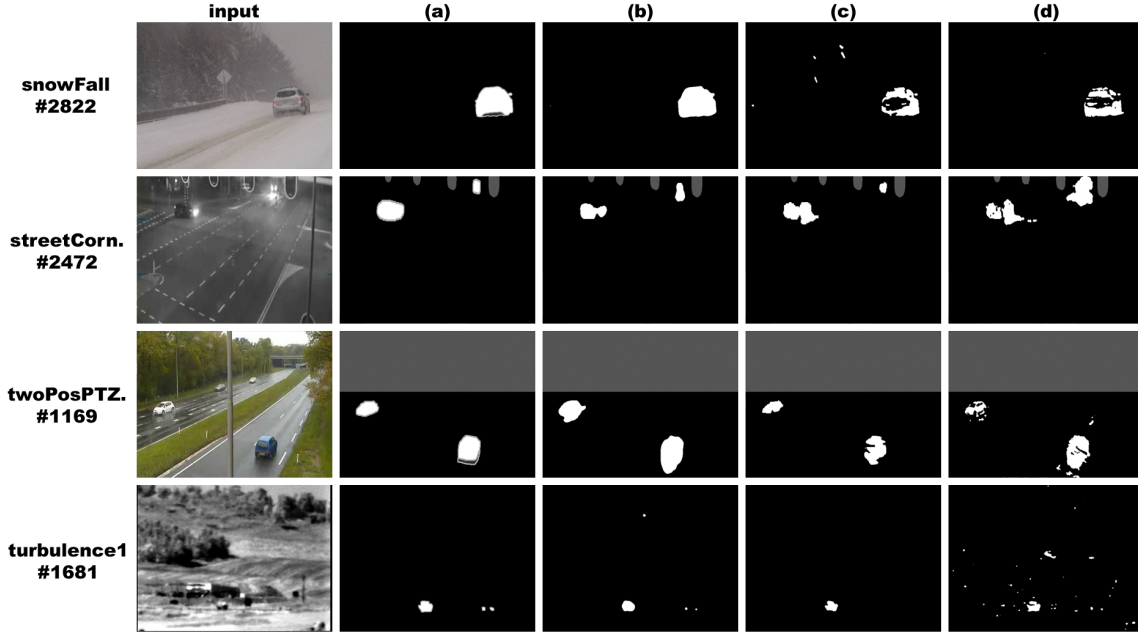


Figure 4.9 Typical segmentation results for various sequences of the 2014 version of the CDnet dataset; column a) shows groundtruth maps, b) shows our segmentation results, c) FTSG’s results and d) GMM’s results. From top to bottom, the sequences are snowFall (from the bad weather category), streetCornerAtNight (night videos), twoPositionPTZCam (PTZ), and turbulence1 (turbulence). Note that gray areas are not unevaluated.

level processing is required. Processing speed could be further improved by reducing the total number of samples in our pixel models (N), at the expense of some flexibility in complex scenarios. This same parameter could also be controlled dynamically to reduce overall computational cost since fewer samples are usually needed for good classifications in static background regions.

4.5 Conclusion

We presented a novel, highly efficient and *universal* foreground/background segmentation algorithm based on change detection in video sequences. Our method uses spatiotemporal information based on color and texture to characterize local representations in pixel-level models while staying robust to most types of illumination variations, including shadows. It also relies on a pixel-level feedback scheme that automatically adjusts internal sensitivity to change and update rates. Our approach continuously monitors both local model fidelity and segmentation noise to guide these adjustments, allowing for fast responses to intermittent dynamic background motion. As such, it can be effectively used in complex surveillance

scenarios presenting many different challenges simultaneously.

Experiments on the largest change detection dataset available yet have shown that, in terms of average *F-Measure*, we surpass all previously tested methods in nine out of eleven scenario categories as well as overall. Categories where our segmentation results were still inaccurate can be considered the next major challenge in change detection: in those cases, the assumptions that have been commonly followed since the late 1990s no longer hold (e.g. the camera no more static). These experiments have also confirmed the benefit of using LBSP features in our pixel models as well as the benefit of using our continuous parameter adjustment scheme based on model fidelity and segmentation noise.

A number of improvements can still be considered for our method; for example, region-level or object-level analyses could be used as extra regularization steps to improve the shape consistency of blobs over time. Also, more sophisticated post-processing operations based on connected components or Markov Random Fields could also help eliminate larger noise patches from our final segmentation results. Besides, since our method is relatively simple and operates at the pixel level, it has a lot of potential for hardware and high-speed parallel implementations.

CHAPITRE 5 ARTICLE 2: UNIVERSAL BACKGROUND SUBTRACTION USING WORD CONSENSUS MODELS

St-Charles, P.-L., Bilodeau, G.-A., Bergevin, R.

IEEE Transactions on Image Processing, Vol. 25, Issue 10, 2016, pp. 4768-4781.

(© 2016 IEEE; Reprinted with permission.)

<https://doi.org/10.1109/TIP.2016.2598691>

Abstract

Background subtraction is often used as the first step in video analysis and smart surveillance applications. However, the issue of inconsistent performance across different scenarios due to a lack of flexibility remains a serious concern. To address this, we propose a novel non-parametric, pixel-level background modeling approach based on word dictionaries that draws from traditional codebooks and sample consensus approaches. In this new approach, the importance of each background sample (or word) is evaluated online based on their recurrence among all local observations. This helps build smaller pixel models that are better suited for long-term foreground detection. Combining these models with a frame-level dictionary and local feedback mechanisms leads us to our proposed background subtraction method, coined “PAWCS”. Experiments on the 2012 and 2014 versions of the ChangeDetection.net dataset show that PAWCS outperforms 26 previously tested and published methods in terms of overall F-Measure as well as in most categories taken individually. Our results can be reproduced with a C++ implementation available online.

5.1 Introduction

The segmentation of foreground and background regions in video sequences based on change detection is a fundamental, yet challenging early vision task. Often simply called background subtraction, it has been well studied over the years. It generally serves as a low cost, high accuracy alternative to unconstrained binary segmentation based on spatiotemporal feature clustering. Background subtraction is typically based on a single hypothesis: all images of a sequence share a common “background”, from which discrepancies are to be considered of interest (or “foreground”). Thus, it is especially useful in applications with static cameras, or when registration between images is possible, as background modeling and foreground classification can be solved solely at the pixel level. Furthermore, this type of segmentation requires no prior knowledge of the foreground, making it ideal for online surveillance applications (or more generally, in intelligent environments).

The main challenges of background subtraction lie in adaptive background modeling and in the definition of “relevant change”, i.e. deciding how discrepancies between observations and model predictions should be classified. In nearly all applications, the background cannot be considered timeless as it may present noisy or dynamic elements (e.g. rippling water, swaying trees), and its content may change over the sequence (e.g. cars entering and leaving a parking lot). Also, while easily detectable changes caused by illumination variations may not be relevant to most applications, subtle changes caused by “camouflaged” foreground objects (i.e. similar to the background) have to be detected correctly. Classic and modern background subtraction challenges have been highlighted by Goyette et al. (2014); Bouwmans (2014); Brutzer et al. (2011); Cucchiara et al. (2003).

Research has previously focused on improving modeling and classification for selected challenges individually, but very little work has been addressing them holistically. Therefore, most background subtraction methods require significant application-specific tuning to achieve good segmentation results in complex scenarios. Few of them actually perform well across many common use cases without supervision or preprocessing. A “universal” background subtraction solution has to:

1. learn the proper balance between sensitivity and precision based on past observations and segmentation coherence to make good unsupervised decisions;
2. ignore irrelevant changes in the observed scene which concur with previously recognized patterns; and
3. determine how and when foreground objects are absorbed in the background model, and avoid model corruption when the background is altered.

Achieving these objectives is complicated by the nature of most background subtraction approaches that, by design, operate online at the pixel level for better efficiency. As such, they cannot easily analyze large-scale change patterns, and must rely on complex regularization schemes (e.g. frame-wide energy minimization with higher order potentials) to produce good results.

What we propose in this paper is a background subtraction method that can be applied to a large variety of scenarios without manual parameter readjustment, coined PAWCS (Pixel-based Adaptive Word Consensus Segmenter). More specifically, we first introduce a new persistence-based word dictionary scheme for instance-based background modeling that simultaneously addresses short-term and long-term adaptation challenges at the pixel and frame level. Unlike traditional codebook or sample consensus approaches, this novel non-parametric modeling strategy allows for the online principled learning of static and dynamic background regions at a low memory cost. This is because it dynamically maintains the

minimal number of background samples (or words) required for proper segmentation. Persistence estimation is used to gauge the importance and reliability of each background word over time based on local match counts. Persistence values then influence the rate at which each word is updated and used for classifications. The long-term retention of good words despite the presence of static foreground and the rapid suppression of irrelevant words (e.g. captured while bootstrapping) are thus assured by design. In other words, PAWCS requires no explicit training to populate its background models, and keeps them up-to-date while processing new frames.

Our novel word models adopt the primary update and maintenance principles of stochastic sample consensus models, meaning that background words: 1) do not need to be unique inside dictionaries; 2) can be built upon local image descriptors; and 3) are randomly updated in an online fashion. This strategy diverges from the traditional codebook maintenance strategy (i.e. Kim et al., 2005) as it allows words to overlap in feature space. This also means words can be shared between pixel-level dictionaries and can contribute to a frame-wide dictionary without having to solve costly unicity conflicts. As discussed by Barnich and Van Droogenbroeck (2011); Van Droogenbroeck and Barnich (2014); St-Charles et al. (2015a), spreading information between neighboring pixel models drastically improves segmentation coherence since it acts as a regularization step; the same is true for our method. The use of a frame-wide (“global”) dictionary as a complement to pixel-level (“local”) dictionaries further improves spatial coherence, and allows the capture of large-scale background change patterns.

Our proposed method also automatically adjusts its primary parameters by incorporating closed-loop controllers into each pixel-level model. That way, each background region can exhibit its own modeling and classification behavior, which can also evolve over the analyzed sequences. Parameter adjustments are regulated by monitoring: 1) segmentation noise prior to regularization; 2) similarity between background models and observations; 3) region instability based on recurring label changes; and 4) the propagation of illumination updates in local neighborhoods. These four aspects are used to guide and/or trigger various feedback mechanisms that affect model update rates, matching and classification thresholds. Unlike previous methods that could also dynamically adapt to the scene (e.g. Zivkovic and van der Heijden, 2006; Hofmann et al., 2012), our strategy is not hindered by foreground, and it does not rely on a sliding window analysis of inputs/outputs. Instead, it relies on a causal infinite impulse response filter and simple heuristics to quickly and efficiently react to short-term and intermittent disturbances at the pixel level.

We evaluate PAWCS using the 2012 and 2014 versions of the ChangeDetection.net (CDnet)

benchmark (Goyette et al., 2014, 2012; Wang et al., 2014a) and compare our results with those of 21 methods listed on its online platform as well as the self-reported results of 6 recently published methods (27 in total). Our new approach outperforms most of them in overall performance, and in most categories taken individually. To make its usage and future comparisons outside this benchmark easier, we offer the full C++ implementation of PAWCS along with its entire testing framework online.

Note that our method was previously introduced (St-Charles et al., 2015b); here, we offer an extended description of our approach, discuss new experiments on the 2014 CDnet dataset, and compare our results with the new state-of-the-art.

5.2 Related Work

Many background modeling paradigms have been introduced over the years: the earliest and most popular is pixel-level modeling, as it allows simple, scalable, high-speed implementations. This is due to the fact that this modeling strategy relies on low-level features (e.g. color intensities, gradients) to track background representations, as opposed to region-level or object-level information. Two of the three best methods listed by Goyette et al. (2012), namely PBAS (Hofmann et al., 2012) and ViBe+ (Van Droogenbroeck and Paquot, 2012), follow this paradigm as they are both based on stochastic intensity sampling (Barnich and Van Droogenbroeck, 2011). Parametric approaches based on Gaussian Mixture Models (GMM, Stauffer and Grimson, 1999) or non-parametric ones based on Kernel Density Estimation (KDE, Elgammal et al., 2000) as well as their derivatives (KaewTraKulPong and Bowden, 2002; Zivkovic and van der Heijden, 2006) can all be considered part of this family. With a goal similar to Zivkovic and van der Heijden (2006), the work of Haines and Xiang (2014) recently proposed a Dirichlet process Gaussian mixture modeling approach that automatically determines its optimal distribution parameters and component count in a data-driven fashion. Their “confidence capping” strategy allows background representations to be captured and forgotten in a principled manner, much like the “maximum negative run-length” evaluation strategy proposed by Kim et al. (2004). In this latter work, codewords are introduced, which are created by clustering reoccurring background representations during a training phase. These codewords are then amassed into codebooks at the pixel-level to create independent, low-level background models. Compared to classic non-parametric models, codebooks can accurately describe multimodal background regions and avoid corruption due to static foreground while having a small memory footprint. This modeling approach was first improved by Kim et al. (2005) to allow the online adaptation of codebooks, then by Wu and Peng (2010) by extending codewords to the spatiotemporal domain, and more recently

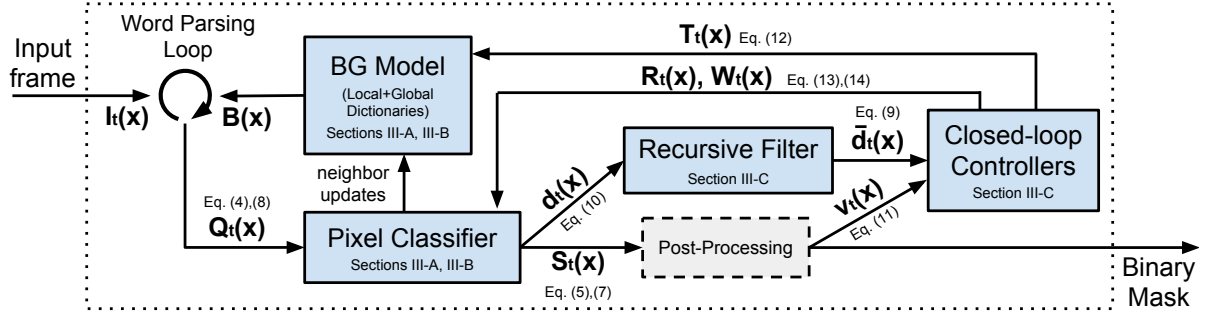


Figure 5.1 Block diagram of the Pixel-based Adaptive Word Consensus Segmenter. Each block represent a major component detailed in Section 5.3.

by Mayer and Mundy (2014) by using duration dependent hidden Markov models to identify and capture periodic background change patterns. Our proposed word-based modeling approach detailed in the following section differs from traditional codebook approaches in that it does not cluster background representations into unique codewords during a training phase. Furthermore, it allows words to be sorted within background models and replaced online based on their persistence in the observed data.

Pixel-based methods are not restricted to use intensity values for background modeling; they can also capture texture information using local descriptors. Doing so helps produce richer, “spatially aware” background samples that are critical for the detection of camouflaged foreground objects. Heikkila and Pietikäinen (2006); St-Charles and Bilodeau (2014); Yang et al. (2015); Silva et al. (2015); Lin et al. (2014) all use Local Binary Patterns (LBP) or other similar binary features for this purpose and achieve good tolerance to illumination variations. The works of Braham and Van Droogenbroeck (2015); Bouwmans (2014) have studied the proper selection and usage of low-level features in pixel-level background modeling.

Frame-level background modeling via Principal Component Analysis (PCA) and low-rank or sparse decomposition approaches is a popular alternative to pixel-level modeling (Oliver et al., 2000; Candès et al., 2011; Zhou et al., 2013; Gao et al., 2014). These approaches are however not ideal for surveillance applications as most rely on batch or offline processing or suffer from scaling problems. Sun et al. (2015) address scaling problems by reformulating principal component analysis for 2D images, and their method achieves much lower memory consumption and computational cost than traditional methods. Some online approaches have also been proposed recently (He et al., 2012; Feng et al., 2013; Seidel et al., 2014), but they are still very computationally expensive.

An early finding in the field is that background modeling should not be limited in scope to frame-wide or pixel-independent processes (Toyama et al., 1999). Doing so would restrict

their perception of change to a single spatial scale and therefore make it harder to address common background maintenance challenges. Multi-scale and “hybrid” methods such as those of Toyama et al. (1999); Chen et al. (2007); Wang et al. (2014b); Yang et al. (2015) have emerged in light of this to improve segmentation coherence at different spatial scales. Coarse-to-fine and block-based strategies can also be adopted to solve this problem. For example, Stagliano et al. (2015) recently proposed a block-based modeling approach based on sparse coding that relies on patch dictionaries learned online to improve spatial coherence. Furthermore, features extracted from High-Efficiency Video Coding (HEVC) macroblocks have been used by Dey and Kundu (2015) for low-cost block-based modeling with good spatiotemporal coherence. Alternative solutions that rely on self-organizing maps (Maddalena and Petrosino, 2012; Zhao et al., 2015; Ramírez-Alonso and Chacón-Murguía, 2016), neural networks (Gregorio and Giordano, 2014; Braham and Van Droogenbroeck, 2016), or probabilistic frameworks such as Markov Random Fields (Sheikh and Shah, 2005; Schick et al., 2012; Wu and Peng, 2010) to expose relationships between independent background models have also been studied.

A problem common to most classic methods is that they lack flexibility: even though they can provide good results on individual sequences when tuned properly, few of them can actually perform equally well across large datasets when used “out-of-the-box”. Modern methods sometimes propose ways to dynamically control either model complexity (Zivkovic and van der Heijden, 2006; Haines and Xiang, 2014), adaptation rates (Lin et al., 2011; Chen and Ellis, 2014) or classification thresholds (Van Droogenbroeck and Paquot, 2012; Wang and Dudek, 2014) online. These however often rely on complex object-level or frame-level analyses and react slowly to intermittent changes. In the case of PBAS (Hofmann et al., 2012), which simultaneously controls model adaptation rates and classification thresholds via pixel-level feedback loops, delayed and fluctuating sensitivity variations can still cause segmentation problems. Our previous work (St-Charles et al., 2015a) addressed this by proposing fast-response feedback loops based on a dual measurement approach, but did not offer a way to simultaneously control model complexity. The recent solution of Bianco et al. (2015) also addresses the flexibility problem of segmentation methods by combining them using a genetic programming approach.

The interested reader is referred to recent surveys of Brutzer et al. (2011); Bouwmans (2014); Sobral and Vacavant (2014); Bouwmans and Zahzah (2014); Goyette et al. (2014) for details on the background subtraction field.

5.3 Methodology

As shown in Figure 5.1, our proposed method can be split into five components: 1) a full “background model”, actually composed of multiple pixel-level (local) models and a frame-level (global) model; 2) a classifier, which produces “raw” segmentation decisions for each pixel based on outlier detection; 3) a post-processing (or regularization) step that relies on basic morphological operations to measure and eliminate segmentation noise; 4) a recursive measurement filter used to simulate the response of a sliding window over model-observation similarity indicators; and 5) a closed-loop controller block, responsible for adjusting the internal parameters of other components based on their state and output. The pixel- and frame-level models along with our classification strategy are presented in Sections 5.3.1 and 5.3.2. The recursive measurement filter as well as feedback mechanisms and controllers are discussed in Section 5.3.3. Finally, three heuristics adopted to further improve our method’s adaptability are presented in Section 5.3.4.

5.3.1 Word consensus for pixel-level modeling

Our novel non-parametric modeling approach is essentially a hybrid between codebook (Kim et al., 2005) and sample consensus strategies (Wang and Suter, 2006; Van Droogenbroeck and Barnich, 2014). “Word consensus” inherits the main advantages of these modeling strategies while avoiding their pitfalls, namely costly updates and high memory requirements. As in other pixel-level, non-parametric modeling approaches, the idea behind word consensus is to simultaneously build independent background models by gathering data samples from local observations. Then, new observations can be classified based on their model overlap. In the following paragraphs, we present an overview of our proposed word-based modeling approach and define some basic terms, and then discuss feature matching, classification and update mechanisms.

Overview and definitions. In contrast to the terminology of Kim et al. (2005), we consider model samples as “words” instead of “codewords” since they are not obtained via clustering, and are thus not necessarily unique. Furthermore, our word-based models are termed “dictionaries” instead of “codebooks”, as the words they contain are sorted and systematically parsed for matches during classification. We define the “local” dictionary of a given pixel x as

$$B_l(x) = \{\omega_1, \omega_2, \dots, \omega_N\} \quad (5.1)$$

where ω_n are background words, and N is the number of words in the dictionary. Each word essentially consists of a background representation (characterized using local image descrip-

tors and/or low-level features) and a transient persistence value. This value is estimated online based on the recurrence (i.e. match count) of the word among recent observations at x . In our modeling approach, persistence defines the basic criteria of word replacement and update policies: the more “persistent” a word is, the less likely it is to be forgotten or replaced by new observations. In comparison to modeling approaches with planned obsolescence policies, our approach is especially useful when segmenting intermittently moving objects.

We evaluate the persistence of a word ω at time t using

$$q_t(\omega) = \frac{n_\omega}{(t_\omega^l - t_\omega^f) + 2 \cdot (t - t_\omega^l) + t_0}, \quad (5.2)$$

where n_ω is ω ’s total occurrence count, t_ω^f and t_ω^l are, respectively, the time at which it was first and last observed, and t_0 is a fixed offset value. The principle behind this equation unique to our modeling approach is comparable to the maximum negative run-length measure of Kim et al. (2005) and the confidence capping of distribution components of Haines and Xiang (2014): it promotes the retention of recurring background words and helps forget those that have not been observed recently. The first denominator term, $(t_\omega^l - t_\omega^f)$, measures the lifespan of ω and is used to scale the persistence of words to the $[0, 1]$ range. It also reflects the “persistence inertia” of ω , i.e. how well it resists persistence value fluctuations caused by short time spans without occurrences. Besides, note that since the second term of the denominator, $(t - t_\omega^l)$, is multiplied by two, the penalty incurred by a distant last occurrence is essentially doubled. This means that while recurrence is important, good words that suddenly disappear from observations can still be eliminated quickly despite very long lifespans. The time offset t_0 is only used here to prevent new words from having important persistence values.

Given (5.2), a mandatory training phase to learn and filter words in local dictionaries is not required since the importance of all words can be continuously estimated over the analyzed sequence. Thus, the segmentation process can be fully initialized by sampling local pixel neighborhoods from a single video frame that may contain foreground. Dictionaries will then naturally stabilize over time to contain only recurring background words. This approach however requires proper feature matching and update mechanisms, which will be discussed next. For now, note that our local dictionaries adapt to permanent background changes by learning a new word when local observations cannot be matched to any existing background word. We illustrated the content of a local dictionary for a multimodal background region in Figure 5.2a.

To avoid the infinite expansion of local dictionaries, we cap the number of active words

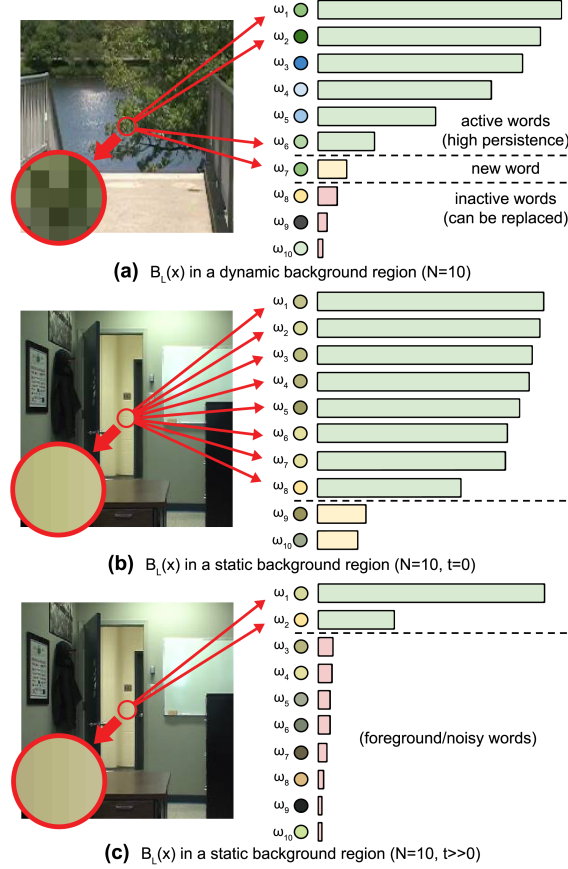


Figure 5.2 Illustrations of possible local dictionary content for dynamic and static background regions. The bars next to each word represents their relative importance in the dictionary based on persistence. In (a), different words are kept active simultaneously while new words are inserted. In (b), the dictionary is shown right after model initialization, and many overlapping words are present due to local neighborhood sampling. In (c), the same dictionary as (b) is presented but at a later time in the sequence, showing a reduced number of active words.

they can contain (N). To keep word counts low, we can ignore or remove words that have negligible persistence values. If a dictionary is full, words with the lowest persistence values are eliminated and replaced by new ones. To make these evaluations and replacements more efficient, we systematically reorder the content of local dictionaries during the matching process using a bubble sort algorithm. That way, more persistent words are checked for matches first, and they are less likely to be replaced by new ones.

Feature selection and matching. In PAWCS, we characterize pixel neighborhoods (for both background words and local observations) using RGB intensities and Local Binary Similarity Pattern descriptors (LBSP, Bilodeau et al., 2013). Local descriptors can be used like any other low-level feature in our modeling approach since, by design, we avoid merging or

clustering data samples. The goal of using LBSP here is to improve robustness to illumination variations and enhance segmentation coherence (or smoothness) through local texture description and matching. As shown by St-Charles and Bilodeau (2014), this color/LBSP description approach also boosts change detection sensitivity (which is beneficial), but induces additional false positives in regions with dynamic textures. We discuss how we solve this problem via feedback mechanisms below.

First, to determine if a word ω from a local dictionary $B_t(x)$ matches the observation of x at time t (noted $I_t(x)$), we evaluate their color resemblance (via ℓ_1 distance and color distortion, c.f. Kim et al., 2004) and LBSP intersection (via Hamming distance). If all three distances fall below given change detection thresholds, then ω is considered a match for $I_t(x)$. To shorten the equations presented in the following sections, we simply write $\|I_t(x) - \omega\| < R_t(x)$ to refer to this matching step. We define $R_t(x)$ as a distance threshold for x at time t , with $R_t(x) \geq 1$. In practice, color and LBSP distance thresholding is done in parallel, and their threshold values are respectively calculated from $R_t(x)$ using

$$R_{c,t}(x) = R_c \cdot \sqrt{R_t(x)} \quad (5.3a)$$

and

$$R_{d,t}(x) = R_d + 2^{R_t(x)}, \quad (5.3b)$$

where R_c and R_d are fixed baseline values. Unlike color thresholds, LBSP descriptor thresholds rely on an exponential relation which is better suited to their nature: small $R_{d,t}(x)$ values lead to discriminative texture matching, and larger $R_{d,t}(x)$ values are used for approximate gradient matching. As we will discuss in Section 5.3.3, $R_t(x)$ is automatically increased in regions exhibiting dynamic textures. A large enough $R_t(x)$ value can thus exclude LBSP descriptors from the matching process by inducing a $R_{d,t}(x)$ value larger than the maximum LBSP Hamming Distance. This leads to a desirable reduction of change detection sensitivity, and ultimately to the elimination of false positives in dynamic texture regions.

Pixel classification. The nature of binary segmentation methods based on change detection implies that all pixels in a video frame are considered foreground unless their description matches with the background model. As explained earlier, we do not impose a unicity constraint on words in local dictionaries. This means that finding a single match in the background model for an observation is not enough to directly proclaim a pixel as background. Our classification approach instead relies on the idea of pixel labeling via consensus, i.e. we consider all words from a local dictionary that overlap (or match) the observation of a pixel in feature space to determine its segmentation label. In classic sample consensus methods,

the number of matched background samples is the sole determinant of the classifier: if at least a given number of samples are matched, the pixel is labeled as background. This can be interpreted as one-class classification for outlier detection. In our case, rather than using a fixed match count as the background classification threshold, we calculate the persistence sum of all matched words, and compare this value with a second dynamic threshold (noted $W_t(x)$). More specifically, for a pixel x , we compute the local dictionary persistence sum as

$$Q_{l,t}(x) = \sum_{\omega \in B_l(x)} q_t(\omega) \text{ s.t. } \|I_t(x) - \omega\| < R_t(x), \quad (5.4)$$

and then classify x as foreground (1) or background (0) using

$$S_t(x) = \begin{cases} 1 & \text{if } Q_{l,t}(x) < W_t(x) \\ 0 & \text{otherwise} \end{cases}, \quad (5.5)$$

where S_t is the output (raw) segmentation map. Again, we discuss how $W_t(x)$ is automatically computed in Section 5.3.3.

Since we do not rely on foreground modeling, (5.4) and (5.5) encapsulate a one-class outlier detector. Here, $R_t(x)$ dynamically controls the reachability distance in feature space, which is similar to having an adaptive k in a k -Nearest Neighbor classifier. On the other hand, $W_t(x)$ dictates the minimum cumulative instance weight (or persistence sum) required to consider x as an inlier. Note that the persistence sum $Q_{l,t}(x)$ can be seen as a general indicator of how well $I_t(x)$ is represented by the content of $B_l(x)$. We will reuse this representativeness estimation in Section 5.3.3 to evaluate model-observation similarity.

Remember that words are sorted in local dictionaries based on their persistence values; this means the summation in (5.4) can be implemented so that it stops once the accumulated persistence is greater than $W_t(x)$. The primary ensuing advantage is faster processing, as dictionaries do not need to be fully parsed for matches. This termination criterion however also has a beneficial effect on the way word persistence values fluctuate over time, which we discuss next.

Word and dictionary updates. As stated before, the persistence value of a word can be evaluated online using (5.2) based on its occurrence count and the time of its first/last observations. In practice, these persistence parameters are only updated when a match is found in (5.4). This means that if the summation stops before reaching a potential matching word ω because similar alternatives exist with higher persistence values, then ω 's persistence will slowly decay, and it will slide down in the dictionary's word ranking. As illustrated in Figure 5.2b and 5.2c, this is actually an important update mechanism that gradually elimi-

nates background words which overlap in feature space. This mechanism is thus responsible for controlling the number of active words (i.e. words with a significant persistence value) in each local dictionary based on the complexity of its associated background region. Below, we discuss two other update mechanisms needed to make our proposed modeling approach universal.

The modeling approach described so far cannot cope with gradual background changes (e.g. the illumination variation caused by a growing cloud cover) since the background descriptions kept by words inside local dictionaries are not updated. To improve robustness in such cases, we randomly replace the color component of matched background words with observed values. This is only done when local texture variation (expressed by the distance between matched and observed LBSP descriptors) and color distortion (as presented by Kim et al., 2004) are negligible, and with probability $\rho = 1/T_t(x)$. Here, $T_t(x) \geq 1$ is a dynamic update rate further discussed in Section 5.3.3. This update mechanism is meant to only allow a small proportion of background words to be modified in response to gradual illumination changes. Therefore, words left untouched will continue modeling the previous background state when the change is only temporary. This strategy implicitly prevents incorrigible model drift and the saturation of local dictionaries by words whose descriptions differ in brightness, but not in local texture. The direct replacement of a word’s color component by an observed value also helps diversify dictionary content, and it is less costly than merging them.

In order to improve the consistency between neighboring local dictionaries, our method shares an important trait with recent sample-consensus methods (e.g. Barnich and Van Droogenbroeck, 2011; Hofmann et al., 2012; St-Charles and Bilodeau, 2014; St-Charles et al., 2015a): pixel-level background information diffusion. The diffusion process increases robustness to infrequent periodic change by sharing information between afflicted regions, and helps erode tenacious false positive segmentation blobs caused by “ghosts” in the background model (see Cucchiara et al., 2003) for an exact definition). In its original form (introduced by Barnich and Van Droogenbroeck, 2011), information diffusion grants pixel models adjacent to x a chance to have one of their samples randomly replaced by the description of $I_t(x)$, but only if x is classified as background. To achieve similar results in our novel modeling approach, when (5.5) returns $S_t(x) = 0$, we randomly select a neighbor of x and update the persistence of each word of its local dictionary that matches with $I_t(x)$. Again, the probability of doing this is $\rho = 1/T_t(x)$. Note that a beneficial effect of these updates is a persistence boost for words in static background regions, which helps detect camouflaged and immobile foreground objects. In contrast to the spatial context improvement proposed by Wu and Peng (2010), our strategy allows pixel models to resist disturbances caused by small background displacements and vibrating cameras because spatial information is shared proactively.

The three proposed update mechanisms allow our pixel models to behave like codebooks in static background regions. Once stable, each local dictionary requires only one or two highly persistent words (which are kept up to date during illumination changes) to provide good segmentation results. In dynamic background regions, our pixel models are akin to sample consensus models, where 20 to 30 different words can be active at once, all with low cohesion and persistence values. Our pixel-level modeling approach alone is however unable to recognize patterns that are too large or outside the influence of local information diffusion. For this, we introduce a frame-level modeling approach, described next.

5.3.2 Word consensus for frame-level modeling

Using a frame-wide or “global” dictionary (noted B_g) along with local dictionaries for classifications fulfills the requirement set by the final Wallflower principle (Toyama et al., 1999), i.e. background models should take into account changes at differing spatial scales. Capturing large scale background patterns is crucial when the observed scene exhibits intermittent change over large areas, or when background elements are removed from it. Besides, our pixel description approach produces highly specific words that, if tied to intermittent background patterns, may be discarded before being observed enough times to build a strong persistence value. Our proposed global dictionary model acts as a long term memory that tracks the use of common words and spreads their influence beyond the reach of the pixel-level diffusion update mechanism. Its only purpose is to provide a safeguard against false positive detections by relabeling x (i.e. overruling $S_t(x) = 1$) when a “global” word with strong persistence is matched to its pixel-level observation.

The persistence definition of (5.2) becomes problematic when considering background words in a global setting, as it implies no ownership or localization concept. Moreover, the state of a dictionary shared among all pixel models would depend on the ordering of local updates. Therefore, we define the persistence of global words by using 2D maps that “localize” the importance of these words in image space. These persistence maps can be seen as heat maps over the observed frames; some examples are shown in Figure 5.3. For each pixel x and each global word ω , a map value (noted $\Phi_\omega(x)$) is directly linked to a single pixel model $B_l(x)$. This means maps can be fully updated asynchronously. Furthermore, this one-for-one mapping in image space also means that frame-wide operations (blurring, saturation, decimation) on global word persistence maps are well defined and easy to implement.

Persistence map contents are updated in two ways. First, each time a pixel x is classified as background, it may be randomly elected to parse B_g for a matching word ω and perform a

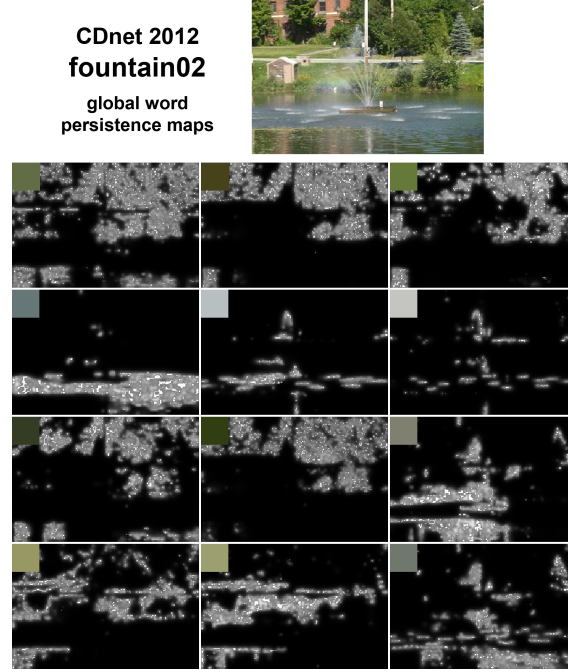


Figure 5.3 Snapshots of the actual global word persistence maps used in the CDnet2012 fountain02 sequence at frame 500. The top-left corners show the pixel color description of each word; texture is omitted for illustration purposes. Brighter spots indicate where matches occurred and the map was updated; those points are the “seeds” from which the persistence is diffused. In total, 24 global words were active (the 12 with highest total persistence are shown here), covering over 95% of the image space with non-zero persistence values.

localized update of its persistence map. We define this update as

$$\Phi_{\omega}(x) = \min \left(\Phi_{\omega}(x) + Q_{l,t}(x), Q_{l,t}(x) \right). \quad (5.6)$$

This essentially creates a seed point in Φ_{ω} used to propagate persistence values to neighboring pixels. Second, the propagation itself is achieved by periodically blurring the maps of all global words using a normalized box filter. To avoid obtaining homogeneous results over time, $\Phi_{\omega}(x)$ values are also decimated while filtering. These two steps create another word diffusion mechanism that has a much larger reach than the one presented in Section 5.3.1. Persistence map decimation also allows the global dictionary to “forget” words over time.

Finally, the foreground classification of a pixel x can be overruled if a matching global word (with a strong localized persistence) is found. More specifically, we replace (5.5) by

$$S_t(x) = \begin{cases} 1 & \text{if } Q_{l,t}(x) + Q_{g,t}(x) < W_t(x) \\ 0 & \text{otherwise} \end{cases}, \quad (5.7)$$

where

$$Q_{g,t}(x) = \begin{cases} \Phi_{\omega}(x) & \text{if } \exists \omega \in B_g \text{ s.t. } \|I_t(x) - \omega\| < R_t(x) \\ 0 & \text{otherwise (i.e. no match found)} \end{cases} \quad (5.8)$$

Since the LBSP descriptors we use are highly specific and hard to match across regions, we simplify the global word matching step in (5.8): instead of using a Hamming distance, we compare the Hamming weights of the descriptors. This is equivalent to an approximate gradient magnitude comparison. While gradient orientation information is lost, this approach preserves texture strength information, which is enough for discriminative change detection.

New words can also be inserted in global dictionaries: pixel models can be randomly selected to copy one of their highly-persistent local background words to B_g and initialize its persistence map, but only if it is missing from B_g . Again, the maximum number of active words in B_g is capped (N), and words with the lowest overall persistence are replaced first once full. The initialization is done by simply querying random pixel models for words and filling persistence maps until the global dictionary is full. Note that since (5.8) and the update mechanisms only target one word at a time (i.e. the first found match), and since global word descriptions do not change throughout their lifespan, feature space overlap between global words is highly unlikely. This means global dictionaries typically contain more unique words than local dictionaries.

5.3.3 Pixel-level feedback

Recall that the equations presented in the previous sections primarily relied on three types of pixel-independent parameters. Namely, the feature-space distance threshold used for matching in (5.4) and (5.8), $R_t(x)$; the persistence threshold used for classification in (5.5) and (5.7), $W_t(x)$; and the rate used to determine the probability of triggering all update mechanisms, $T_t(x)$. While fixed values could be used frame-wide, they would need to be determined empirically from the analyzed videos, and doing so would prevent pixel models from handling multiple segmentation challenges at once. Therefore, we dynamically adjust these internal parameters using pixel level closed-loop controllers. Subsequently, we use simple heuristics to increase the chance of triggering further model updates and parameter adjustments where needed. As stated in Section 5.1, four aspects of our method are monitored and used to control these feedback mechanisms: two of them are related to the state of the background model (model-observation similarity and illumination update propagation) and two to the output (segmentation noise and instability). These are discussed below.

Evaluating the similarity between local dictionary words and local observations captured from input frames is the first step in determining if the background is adequately modeled. We do

so for each pixel x by recursively filtering the estimations of minimal matching distances between local observations and words along with the “representativeness gap” of missed matches (defined below). This provides a temporally smooth measurement of model-observation similarity, noted $\bar{d}_t(x)$, which can be used in closed-loop controllers. The proposed recurrence relation is defined as

$$\bar{d}_t(x) = (1-\alpha) \cdot \bar{d}_{t-1}(x) + \alpha \cdot d_t(x) \quad (5.9)$$

with

$$d_t(x) = \max \left(\min_{\omega \in B_t(x)} \|I_t(x) - \omega\|, \frac{W_t(x) - Q_{l,t}(x)}{W_t(x)} \right), \quad (5.10)$$

where $\alpha \in [0, 1]$ is a fixed coefficient, and $\frac{W_t(x) - Q_{l,t}(x)}{W_t(x)}$ is the “representativeness gap”. As stated earlier, this is because $Q_{l,t}(x)$ can be interpreted as an indicator of model representativeness.

In practice, since multiple distances are evaluated in $\|I_t(x) - \omega\|$, we find the minimum distance for each feature (color, LBSP) in $B_t(x)$ independently, normalize them to the $[0, 1]$ range, and then only use the maximum one in (5.10). The “representativeness gap” is used to counterbalance the similarity score for models that have good matches but low persistence sums. In the end, $d_t(x)$ and $\bar{d}_t(x)$ can only take values in the $[0, 1]$ range. The nature of the two terms compared in (5.10) makes “ideal” background modeling (reflected by $d(x) \approx 0$) very hard to achieve in practice. This forces continuous feedback that helps diversify pixel models. The recurrence relation of (5.9) essentially models the infinite impulse response of an exponentially weighted sliding window filter, but at a very low computational cost. This approach allows fast feedback responses to intermittent and irregular changes.

Since $\bar{d}_t(x)$ is updated every frame, foreground objects that pass or stay immobile over x will inevitably increase its value over time. Therefore, this first indicator cannot be used by itself to control parameter adjustments as it does not always truly reflect the similarity between background models and observations. To address this, we rely on a second pixel-level indicator, noted $v_t(x)$, that monitors noise in the segmentation results. The assumption behind $v_t(x)$ is that inadequately modeled regions emit more noise (i.e. alternating segmentation labels) than other regions, which are instead constantly labeled as foreground or background. We define $v_t(x)$ as a segmentation noise accumulator, and update it using

$$v_t(x) = \begin{cases} v_{t-1}(x) + 1 & \text{if } (S_t(x) \oplus S_{t-1}(x)) = 1 \\ v_{t-1}(x) - 0.1 & \text{otherwise} \end{cases} \quad (5.11)$$

where $\oplus$ is the XOR operator, and $v_t(x)$ is prevented from taking negative values. This definition essentially means that only static regions will exhibit low $v_t(x)$ values.

Using $\bar{d}_t(x)$ and $v_t(x)$, we can now describe our pixel-level controllers. First, we define the adjustment mechanism for local update rates as

$$T_t(x) = \begin{cases} T_{t-1}(x) + \frac{\lambda_T}{v_t(x) \cdot \bar{d}_t(x)} & \text{if } S_t(x) = 1 \\ T_{t-1}(x) - \frac{\lambda_T \cdot v_t(x)}{2 \cdot \bar{d}_t(x)} & \text{if } S_t(x) = 0 \end{cases} \quad (5.12)$$

where $\lambda_T \in [0, 1]$ is a fixed scaling factor, and $T_t(x)$ is bound to the $[1, 256]$ interval. Like all sample-consensus methods, we use an inversely proportional relation to calculate the probability ρ of triggering an update mechanism from $T_t(x)$. This means that high $T(x)$ values lead to fewer updates, and that when foreground is detected in static regions with low segmentation noise, model updates will almost immediately stop. In other words, $T(x)$ will max out quickly due to $v_t(x) \approx \bar{d}_t(x) \approx 0$. However, dynamic and noisy background regions will keep allowing model updates for much longer, as in those cases, $v_t(x) \gg 0$ and $\bar{d}_t(x) \approx 1$, which results in smoother variations.

Having models that update frequently is usually not enough to eliminate all false foreground classifications caused by strong dynamic background change. Locally adjusting feature matching thresholds is often the fallback solution to avoid having to build and maintain very large background models. As stated in Section 5.3.1, we derive both color and LBSP matching distance thresholds from a common value, $R_t(x)$. The adjustment control loop behind this parameter based on our two pixel-level indicators can be described as

$$R_t(x) = \begin{cases} R_{t-1}(x) + \lambda_R \cdot v_t(x) & \text{if } R_{t-1}(x) < (1 + \bar{d}_t(x) \cdot 2)^2 \\ R_{t-1}(x) - \frac{\lambda_R}{v_t(x)} & \text{otherwise} \end{cases} \quad (5.13)$$

where $\lambda_R \in [0, 1]$ is a fixed scaling factor. Note that $R_t(x)$ can only take values greater or equal to 1; this lower limit reflects the baseline matching distance threshold used in perfectly static regions. We control $R_t(x)$ variations via $\bar{d}_t(x)$ based on an exponential relation since it allows much easier feature matching in highly unstable regions (i.e. when $\bar{d}_t(x) \gg 0$). Here, $v_t(x)$ directly controls the variation step size of $R_t(x)$. In static regions, it prevents $R_t(x)$ from increasing too fast (which helps against camouflage problems), and in dynamic regions, it prevents it from decreasing too fast while $\bar{d}_t(x)$ fluctuates.

Feature matching distances and local persistence sums are closely related in the feedback process as both influence $\bar{d}_t(x)$ through (5.10). Therefore, we tie the adjustment mechanism

of persistence thresholds to $R_t(x)$, and define it as

$$W_t(x) = \frac{q_t(\omega_1)}{R_{t-1}(x) \cdot 2} \quad (5.14)$$

where ω_1 is the first word of $B_l(x)$, and thus its most persistent one due to sorting. The idea here is to always have at least one local word with enough persistence to classify a pixel as background. This also means that the value of $W_t(x)$ is kept in the persistence range dictated by the words of $B_l(x)$, and each pixel model can have a unique classification behavior.

While our pixel-level controllers rely on segmentation noise to provide rapid parameter adjustments, this type of noise is an inherent characteristic of pixel-based segmentation due to shot noise in the analyzed images. Segmentation noise can also be easily eliminated afterwards using post-processing or regularization techniques. In PAWCS, we ultimately clean the raw segmentation maps S_t by using median blurring and morphological operations. This essentially mimics the effect of a more complex frame-level regularization approach at a low cost.

5.3.4 Model adaptability

The pixel-level controllers are mostly responsible for the overall flexibility of our method, but we also define three heuristics to further improve model adaptability. The first one relies on texture analysis to provoke extra updates in uniform background regions. This leads to stronger word persistence in local dictionaries, and thus better long-term word retention. In short, we cut down the value of $T_t(x)$ when the observed LBSP descriptor in $I_t(x)$ is “flat” (i.e. its Hamming weight is close to zero), which causes update mechanisms to trigger more often. The assumption here is that uniform regions are better background candidates than regions with strong gradients. Cluttered regions are largely unaffected by this criterion; this indirectly helps prevent “ghosts” from forming in highly textured background regions due to small, temporary texture displacements.

The second heuristic we use is based on segmentation instability. Large, long-term discrepancies between the “raw” output segmentation $S_t(x)$ and its post-processed equivalent are not as likely due to alternating labels than to small dynamic background regions (e.g. leaves in a tree). Our local feature description approach is very sensitive to such small variations; to help post-processing, we double the fixed R_d offset in (5.3b) when discrepancies above a fixed threshold are detected for x . This reduces the texture change detection sensitivity in the matching process and eliminates more false positive classifications in those regions.

Our last heuristic is responsible for spreading illumination updates between neighboring pixel models and causing chain reactions that allow large background surfaces to be updated rapidly. As stated in Section 5.3.1, the color component of words can sometimes be updated to account for gradual changes in the background. To propagate these updates, we keep a 2D map of where they happen; then, for a pixel x , if one of its neighbors was recently updated, we halve the value of $T_t(x)$. This approach allows our model to respond rapidly and efficiently to frame-wide lighting variations.

5.4 Experiments

The state-of-the-art presented in Section 5.2 makes it clear that traditional datasets (e.g. the work of Toyama et al., 1999; Li et al., 2004) are too small and no longer challenging to modern background subtraction methods. Moreover, classic methods (e.g. Stauffer and Grimson, 1999; Elgammal et al., 2000) have long since been surpassed and no longer offer a good performance reference. To properly evaluate our method, we rely on the 2012 and 2014 versions of the ChangeDetection.net (CDnet) benchmark and dataset (Goyette et al., 2014, 2012; Wang et al., 2014a). Unlike older alternatives, the CDnet dataset offers a wide variety of real-world sequences split into eleven categories based on the challenges they contain. Totalling nearly 160,000 manually annotated frames, it is several orders of magnitude larger than other real-world datasets, and multiple times larger than those based on synthetic data. Part of the groundtruth is also withheld and kept for online testing to prevent overtuning.

In Section 5.4.1 and 5.4.2, we respectively discuss our 2012 and 2014 CDnet results, and compare them to those of 27 methods listed online¹ or self-reported in prior publications (we report only the top performers). Note that not all methods tested on the 2012 dataset have been tested on the 2014 version, as the latter is much harder, and some authors prefer focusing on a smaller subset of segmentation challenges. Finally, we discuss memory footprint, processing speed and the possibility of a parallel implementation in Section 5.4.3.

We use the same PAWCS configuration to process all sequences of the 2012 and 2014 versions of the CDnet dataset. This is a disadvantage, as methods which were only tested on the 2012 version were specifically tuned for only those categories. We assume all self-reported results used for comparisons followed the tuning guidelines of the CDnet benchmark. To determine which parameters to use for LBSP feature description, we followed the approach of St-Charles et al. (2015a), which dynamically balances them based on the observed scene’s gradient content. We also used the frame-level component presented by St-Charles et al.

1. <http://www.changedetection.net>

(2015a) to detect drastic changes (e.g. light switch events) in the analyzed sequences, and automatically reset our model when needed. Besides, we fix the value of the metaparameters presented in the previous sections as follows:

- Word weight offset value: $t_0 = 1000$
- Maximum number of words per dictionary: $N = 50$
- Baseline color distance threshold: $R_c = 20$
- Baseline LBSP distance threshold: $R_d = 2$
- Feedback recurrence adaptation rate: $\alpha = 0.01$
- Local update rate change factor: $\lambda_T = 0.5$
- Local distance threshold change factor: $\lambda_R = 0.01$

For more details on the post-processing steps and feedback heuristics of PAWCS, the reader is invited to refer to our implementation².

We limit the presentation of qualitative results (Figures 5.4 and 5.5) due to space constraints and because the difference between state-of-the-art methods and groundtruth is sometimes hard to perceive. Our full segmentation results can be downloaded via the CDnet evaluation platform, where more granular comparisons are also possible with most methods based on seven evaluation metrics.

5.4.1 CDnet 2012

We first present in Table 5.1 the average Recall (Re), Precision (Pr), F-Measure (FM) and Matthew’s Correlation Coefficient (MCC) scores of PAWCS on CDnet2012. The definition of the first three metrics can be found in the work of Goyette et al. (2012). MCC is also used here since it provides a good assessment of overall performance in unbalanced binary classification problems, which background subtraction falls into. It is defined by

$$\text{MCC} = \frac{(\text{TP} \cdot \text{TN}) - (\text{FP} \cdot \text{FN})}{\sqrt{(\text{TP} + \text{FP}) \cdot (\text{TP} + \text{FN}) \cdot (\text{TN} + \text{FP}) \cdot (\text{TN} + \text{FN})}}, \quad (5.15)$$

where True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) are defined like in the work of Goyette et al. (2012).

Note that the description of each challenge, category and sequence is provided by Goyette et al. (2014, 2012). We can see from Table 5.1 that our method offers good balance between precision and recall in all test categories but “camera jitter” and “intermittent object motion”.

2. <https://github.com/plstcharles/litiv>

Table 5.1 Segmentation results of PAWCS on CDnet2012

Category	Re	Pr	FM	MCC
baseline	0.941	0.939	0.940	0.938
camera jitter	0.784	0.866	0.814	0.812
dynamic background	0.887	0.904	0.894	0.894
interm. obj. motion	0.749	0.839	0.776	0.774
shadow	0.917	0.871	0.891	0.887
thermal	0.850	0.828	0.832	0.828
overall	0.855	0.875	0.858	0.855

In the former, vibrating cameras cause the entire observed scenes to be perceived as dynamic. This means that pixel model updates trigger very often and that static foreground objects become part of them very fast, which leads to lower Recall scores. The second problematic category is of particular interest here: “intermittent object motion” contains four videos focused on long-term immobile object segmentation, and two on “ghost” elimination (i.e. purging the model of background objects removed from the scene). A lower Recall score in this category indicates that some foreground objects were eventually “lost” to the background model over time. In reality, this difference is not surprising, as very little time is allowed to “learn” the empty background, therefore our words never attain “strong” persistence values. This category is also by far the hardest in CDnet 2012; we discuss it in detail later in this section. The other categories shown in Table 5.1 deal with more traditional challenges of background subtraction, and our method performs well in all of them.

Next, we present in Table 5.2 a compilation of the per-category and overall F-Measure scores of the 23 top-performing methods published and available online as of July 2015. Like Haines and Xiang (2014); St-Charles et al. (2015a), we do not use the official CDnet rankings in these results, as adding and removing methods from the comparison pool (no matter their performance) can drastically affect their final ordering. The F-Measure scores used instead were found to be closely correlated with the method rankings in all three previous CDnet evaluation reports (Goyette et al., 2014, 2012; Wang et al., 2014a). F-Measure is also the most common metric used for comparison in the literature, and thus allows us to list practically all self-reported results of methods not yet on the online benchmark.

Table 5.2 shows that PAWCS outperforms all other methods based on overall Recall, Precision, and F-Measure scores for CDnet2012. It also places first or second for F-Measure in four out of six categories, with a noticeable gap to the state-of-the-art only in “intermittent object motion”. Even in the “camera jitter” category, the results of PAWCS are only slightly worse than those of the best methods, which rely on multiple configurations (Chen et al.,

Table 5.2 Average per-category and overall scores on CDnet2012

Method	Baseline	Cam. Jitt.	Dyn. Bg	Int. Mot.	Shadow	Thermal	Overall (2012)		
	FM	FM	FM	FM	FM	FM	Re	Pr	FM↓
PAWCS (proposed)	<i>0.940</i>	0.814	0.894	0.776	<i>0.891</i>	0.832	0.855	0.875	0.858
Chen et al. (2015) [†] *	0.935	<i>0.817</i>	0.867	0.798	0.813	<i>0.825</i>	-	-	<i>0.842</i>
Wang et al. (2014b)*	0.933	0.751	<i>0.879</i>	<i>0.789</i>	0.883	0.777	<i>0.838</i>	<i>0.867</i>	0.835
St-Charles et al. (2015a)*	0.950	0.815	0.818	0.657	0.899	0.817	0.828	0.858	0.826
Sajid and Cheung (2015)*	0.928	0.836	0.790	0.709	0.778	0.811	0.790	0.849	0.809
Gao et al. (2014) [†]	0.928	0.815	0.782	0.653	0.806	0.760	0.799	0.798	0.796
Wang and Dudek (2014)*	0.881	0.711	0.843	0.721	0.813	0.760	0.791	0.827	0.788
Gregorio and Giordano (2013)	0.908	0.781	0.809	0.567	0.841	0.762	0.818	0.774	0.778
Sedky et al. (2014)	0.933	0.716	0.787	0.566	0.884	0.776	0.777	0.846	0.777
Haines and Xiang (2014)	0.929	0.748	0.814	0.542	0.813	0.813	0.827	0.793	0.776
Stagliano et al. (2015) [†]	0.877	0.725	0.753	0.686	0.810	0.793	0.785	0.773	0.774
Yang et al. (2015) [†]	0.883	0.787	0.808	0.525	0.860	0.754	0.828	0.786	0.769
Dey and Kundu (2015) [†]	0.934	0.712	0.828	0.535	0.864	0.735	-	-	0.768 ^b
Evangelio et al. (2014)	0.921	0.672	0.688	0.715	0.865	0.735	0.768	0.835	0.766
Zhou et al. (2013)	0.923	0.778	0.708	0.595	0.832	0.708	0.801	0.727	0.757
Hofmann et al. (2012)	0.924	0.722	0.683	0.575	0.860	0.756	0.784	0.816	0.753
Allebosch et al. (2015)*	0.917	0.713	0.578	0.578	0.820	0.838	0.809	0.741	0.741
Schick et al. (2012)	0.929	0.750	0.696	0.565	0.791	0.693	0.804	0.751	0.737
Candès et al. (2011)	0.911	0.722	0.694	0.537	0.789	0.719	0.701	0.776	0.729
Maddalena and Petrosino (2012)	0.933	0.705	0.669	0.592	0.779	0.692	0.802	0.732	0.728
Hernandez-Lopez and Rivera (2014)	0.921	0.487	0.750	0.741	0.809	0.662	0.777	0.761	0.728
Ramírez-Alonso and Chacón-Murguía (2016)*	0.918	0.721	0.671	0.510	0.795	0.703	0.708	0.801	0.720
Zhao et al. (2015) [†]	0.927	0.634	0.675	-	-	0.793	-	-	-

^a Red-bold entries indicate the best result in a given column, and blue-italics the second best.

^b We recalculated the overall result of Dey and Kundu (2015) based on the CDnet evaluation guidelines.

* Extracted from CDnet2014 results.

[†] Self-reported.

2015), prior image realignment (Gao et al., 2014), or supervised training (Sajid and Cheung, 2015) to address the shaking camera challenge. This means our dictionary update and feedback mechanisms are already quite sufficient for this task. Besides, compared to others, PAWCS excels in the “shadow” category, offering performance similar to the performance of multiple methods in the less-challenging “baseline” category. In fact, as stated by Wang et al. (2014a), while “hard” shadows are still a challenge for all methods, soft shadows are no longer problematic to modern solutions. In our case, this is due to our illumination update mechanisms as well as our choice of features for word description. We can also note that overall, our “online” method outperforms all recent robust PCA-based methods (Candès et al., 2011; Zhou et al., 2013; Gao et al., 2014) despite the fact that we “stream” the data in an online fashion (as opposed to offline batch-processing and optimization).

The advantages of our novel word consensus modeling strategy can be easily outlined by comparing our results to those of SuBSENSE (St-Charles et al., 2015a) since both methods use similar features and feedback mechanisms. While the results of PAWCS are slightly worse in the “baseline” category and equivalent in “camera jitter” and “shadow”, they are

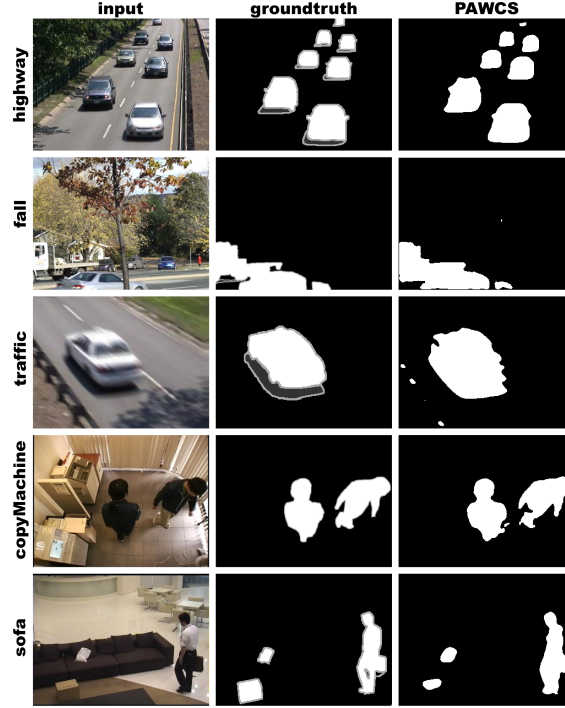


Figure 5.4 Qualitative comparison of our segmentation results with the groundtruth on various sequences of CDnet2012. Gray regions in the groundtruth are not evaluated.

clearly superior in “dynamic background” and “intermittent object motion”. This means our dictionaries can better reflect the multimodality of dynamic regions while staying sensitive to outliers, and that persistence is an ideal way to determine which words are more important for modeling based on past observations. The difference in the “baseline” category results can be explained by a slight increase in false negative classifications due to the global dictionary acting as a “fallback” model to prevent false positives elsewhere. Comparisons of some of our segmentation results with the groundtruth are shown in Figure 5.4.

Finally, we compare the results of various methods for the “intermittent object motion” category in Table 5.3. We can note that most recent methods perform poorly when static foreground object and background ghosts are involved. The top performers, FTSG (Wang et al., 2014b) and Shared-GMM (Chen et al., 2015), both rely on foreground modeling as well as region-level or object-level processing steps and heuristics to specifically tackle this category. On the other hand, our modeling approach only relies on the analysis of word persistence based on the principles detailed in Section 5.3. Thus, we keep to low-level mechanisms without involving object shape semantics. Our design is therefore in accordance with the first Wallflower principle (Toyama et al., 1999), but at a disadvantage when faced with these challenges.

Table 5.3 Average recall, precision and F-Measure scores on the intermittent object motion category of CDnet2012 ^a

Method	Recall	Precision	F-Measure↓
Chen et al. (2015) ^{†*}	-	-	0.798
Wang et al. (2014b)*	0.835	0.933	<i>0.789</i>
PAWCS (proposed)	0.749	<i>0.839</i>	0.776
Hernandez-Lopez and Rivera (2014)	<i>0.808</i>	0.762	0.741
Wang and Dudek (2014)*	0.762	0.753	0.721
Evangelio et al. (2014)	0.736	0.814	0.715
Sajid and Cheung (2015)*	0.639	0.820	0.709
Stagliano et al. (2015) [†]	0.684	0.709	0.686
Gao et al. (2014) [†]	0.653	0.715	0.683
St-Charles et al. (2015a)*	0.658	0.796	0.657
Maddalena and Petrosino (2012)	0.724	0.590	0.592
Allebosch et al. (2015)*	0.742	0.563	0.578
Hofmann et al. (2012)	0.670	0.705	0.575
Sedky et al. (2014)	0.595	0.719	0.566
Haines and Xiang (2014)	0.676	0.653	0.542

^a Red-bold entries indicate the best result in a given column, and blue-italics the second best.

* Extracted from CDnet2014 results.

[†] Self-reported.

5.4.2 CDnet 2014

As stated earlier, the 2014 dataset is much more complex than the original version. Its categories include videos captured outside during snowstorms, extremely low framerate videos with rapid illumination variations and color profile changes, highway surveillance videos captured at night with intense glare effects from car headlights, videos captured by pan-tilt-zoom (PTZ) cameras being operated, and thermal imaging videos of long-range surveillance in high temperature environments. Just like for the CDnet2012 dataset, we first present in Table 5.4 PAWCS’s performance on CDnet2014 using average Recall (Re), Precision (Pr), F-Measure (FM) and Matthew’s Correlation Coefficient (MCC) scores. Note that since MCC is not computed by the online CDnet benchmark platform, and since half of the groundtruth is withheld, the MCC scores reported here are only representative of half of the dataset.

In comparison with the metrics shown for CDnet2012, it is quite clear that the 2014 dataset is more challenging. Only the “bad weather” category has F-Measure and MCC scores above 80%, and two categories have Precision or Recall scores below 50%. These Precision and

Table 5.4 Segmentation results of PAWCS on CDnet2014

Category	Re	Pr	FM	MCC ^a
bad weather	0.718	0.947	0.815	0.812
low framerate	0.773	0.641	0.659	0.657
night videos	0.361	0.654	0.415	0.432
pan-tilt-zoom	0.698	0.473	0.462	0.491
turbulence	0.812	0.681	0.645	0.786
overall (2014 only)	0.672	0.679	0.600	0.636
overall (2012+2014)	0.772	0.786	0.740	0.756

^a Approximated based on available groundtruth.

Recall scores respectively indicate that more than half of foreground classifications were false, and more than half of all true foreground classifications were missed by our method. Surprisingly, PAWCS performed better in the “pan-tilt-zoom” category (in which the basic assumption of the static camera is violated) than in “night videos”. This can be explained by two factors: first, our global dictionary allows large uniform regions (which make up the bulk of all background areas in PTZ videos) to be properly recognized as background despite important camera motion. Second, due to the use of LBSP descriptors, and since our dictionaries are kept at minimal word counts, our method is sensitive to noise and color variations in low contrast background regions (such as the ones in “night videos”). On the other hand, the “turbulence” and “bad weather” categories also mostly contain low contrast videos with dynamic background regions, making it a combination of very challenging problems. Finally, the “low framerate” category shows decent results in all but one sequence (not shown here), filmed at one frame per six seconds. This problematic sequence presents very large color variations along with dynamic background elements, making it very hard to process.

We compare in Table 5.5 the scores obtained by 14 methods which were tested thus far on CDnet2014. We omit the 2014 dataset results of Chen et al. (2015) as their published results are significantly different from those reported online. Again, the performance of PAWCS is well above the average, ranking second in overall F-Measure (by a marginal difference) to the work of St-Charles et al. (2015a), and third in overall F-Measure for the 2014 categories only. On the other hand, on the official CDnet rankings available online based on all seven evaluation metrics (not shown here) list PAWCS as the best method by a good margin.

The F-Measure scores of PAWCS are lower than those of the state-of-the-art in the “bad weather”, “night videos” and “turbulence” categories. As mentioned before, this is due to the dynamic background/low contrast combination of challenges present in those sequences, to which our low-complexity modeling approach has difficulty adapting. Still, the overall

Table 5.5 Average per-category and overall scores on CDnet2014

Method	Bad Weath.	Low Fr.	Night Vid.	PTZ	Turbul.	Overall (2014)			Overall (2012+2014)		
	FM	FM	FM	FM	FM	Re	Pr	FM	Re	Pr	FM↓
St-Charles et al. (2015a)	0.862	0.645	<i>0.560</i>	0.348	0.779	0.794	0.623	<i>0.639</i>	0.812	0.751	0.741
PAWCS (proposed)	0.815	<i>0.659</i>	0.415	0.462	0.645	0.672	<i>0.679</i>	0.600	0.772	0.786	<i>0.740</i>
Wang et al. (2014b)	<i>0.823</i>	0.626	0.513	0.324	0.713	0.678	0.652	0.600	0.766	0.770	0.728
Sajid and Cheung (2015)	0.773	0.628	0.516	<i>0.512</i>	0.570	0.635	0.617	0.599	0.719	0.744	0.714
Allebosch et al. (2015)	0.779	0.663	0.655	0.584	0.671	<i>0.758</i>	0.701	0.670	<i>0.786</i>	0.722	0.709
Gregorio and Giordano (2014)	0.684	0.641	0.374	0.322	0.723	0.531	0.678	0.549	0.661	<i>0.773</i>	0.681
Sedky et al. (2014)	0.757	0.644	0.483	0.365	0.543	0.693	0.540	0.558	0.735	0.705	0.673
Wang and Dudek (2014)	0.767	0.469	0.380	0.135	<i>0.755</i>	0.599	0.584	0.501	0.704	0.716	0.658
Ramírez-Alonso and Chacón-Murguía (2016)	0.774	0.494	0.416	0.330	0.464	0.579	0.561	0.496	0.650	0.692	0.618
Maddalena and Petrosino (2012)	0.662	0.546	0.450	0.041	0.488	0.715	0.462	0.437	0.762	0.609	0.596
Zivkovic and van der Heijden (2006)	0.759	0.549	0.420	0.213	0.520	0.658	0.547	0.492	0.665	0.679	0.594
Varadarajan et al. (2013)	0.683	0.531	0.427	0.247	0.458	0.582	0.543	0.469	0.594	0.697	0.574
Stauffer and Grimson (1999)	0.738	0.537	0.410	0.152	0.466	0.653	0.484	0.461	0.685	0.603	0.571
Elgammal et al. (2000)	0.757	0.548	0.436	0.037	0.448	0.729	0.457	0.445	0.738	0.581	0.569

^a Red-bold entries indicate the best result in a given column, and blue-italics the second best.

2012+2014 F-Measure score of PAWCS demonstrates that it is very flexible, and that it tackles most challenges without compromising too much of its performance elsewhere. In terms of Recall and Precision, we can observe balanced scores that are higher than those of most methods in both dataset versions, further demonstrating the overall flexibility of our approach. We present some qualitative comparisons between the groundtruth and our segmentation results in Figure 5.5.

5.4.3 Processing speed and memory footprint

Our C++ implementation processes the entire CDnet2012 dataset on a third generation, quad-core Intel i5 CPU (one sequence per core) at 22 frames per second, and it processes individual QVGA sequences on a single core at 15 frames per second. This is about 50% slower than St-Charles et al. (2015a) given equal model sizes for both methods, but still much faster than most video segmentation methods. Comparing this result to others is difficult due to the lack of open-source implementations available online; we offer ours for future reference.

Out of the entire processing load for a single frame, about 75% of it is for parsing dictionaries for matches and updating them, 15% is for feedback mechanisms, 5% is for frame-wide operations on global word persistence maps, and 5% is for output regularization via morphological operations and median blurring. The complexity of PAWCS is constant with respect to the number of pixels in the input frames. Given the ample opportunities for parallelization, real-time processing of very high resolution videos appears achievable. We recently implemented a similar non-parametric sample consensus method (St-Charles and Bilodeau, 2014) on GPU, and reached a speedup of 30x over its CPU implementation (i.e. it could process several

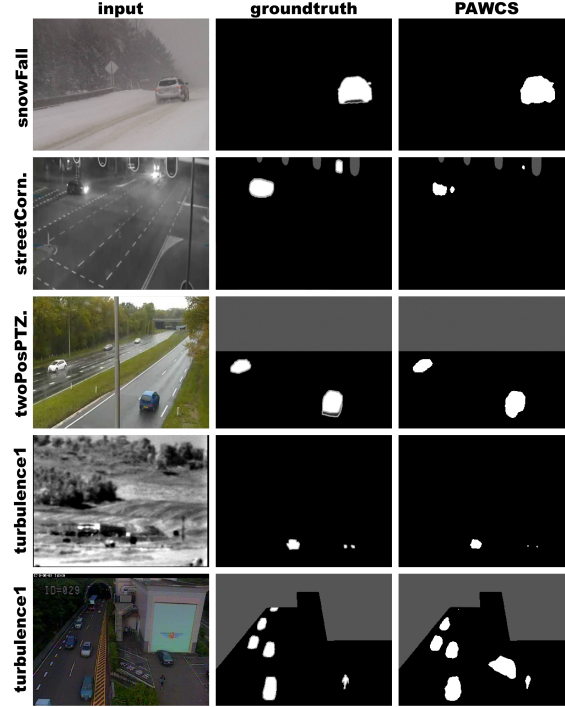


Figure 5.5 Qualitative comparison of our segmentation results with the groundtruth on various sequences of CDnet2014. Gray regions in the groundtruth are not evaluated. An obvious false positive blob is visible in the last row’s segmentation map; this is a temporary “ghost” artifact caused by a van that left the scene after being parked there.

thousand frames per second). This speed was capped only by the memory bandwidth of the hardware we used.

As for the memory footprint of our proposed method, note that a background word requires three bytes of memory per channel to store color information and LBSP binary strings (based on the 5x5 pattern of Bilodeau et al., 2013), and three integers to store its persistence parameters (four bytes each, if $t_{\max} > 2^{16}$ is expected). In a worse-case scenario where all pixel models require $N = 50$ background words (and those words are never discarded), and where the target video is RGB 1080p (≈ 2.1 megapixels), the memory requirement set by our pixel modeling approach would be slightly over 2 GB. Global words would also add less than 2 MB each to the total, given that their persistence maps are implemented using one byte per pixel. For embedded/mobile applications, given an average of $N = 5$ active words per pixel model and a QVGA resolution, the memory requirement of PAWCS would be less than 10 MB.

5.5 Conclusion

“Word consensus”, a new non-parametric pixel-level modeling approach, has been presented. It works by capturing local image samples and evaluating their recurrence among recent observations. We showed through our experiments with PAWCS that word consensus is performant when tackling segmentation challenges involving static foreground objects and multimodal background regions. Its ability to automatically deduce which model samples (or words) are the most important background components based on temporal persistence allows it to keep a low overall memory footprint. With the addition of closed-loop controllers and other feedback mechanisms, our complete method allows each targeted image pixel to behave differently in terms of classification behavior and model complexity. A frame-level word dictionary is also considered to increase spatial coherence between pixel models and prevent false classifications due to large-scale background change patterns.

Our results showed that our method is superior to most in scenarios with traditional and modern background subtraction challenges. There is still room for improvement however; using a more sophisticated output regularization step (e.g. Schick et al., 2012), or explicitly modeling the foreground appearance of objects pixel-wise are good avenues for future work.

CHAPITRE 6 ARTICLE 3: ONLINE MULTIMODAL VIDEO REGISTRATION BASED ON SHAPE MATCHING

St-Charles, P.-L., Bilodeau, G.-A., Bergevin, R.
 IEEE Conference on Computer Vision and Pattern Recognition Workshops
 Boston, MA, USA, June 7-12, 2015, pp. 26-34.
 (© 2015 IEEE; Reprinted with permission.)
<https://doi.org/10.1109/CVPRW.2015.7301293>

Abstract

The registration of video sequences captured using different types of sensors often relies on dense feature matching methods, which are very costly. In this paper, we study the problem of “almost planar” scene registration (i.e. where the planar ground assumption is almost respected) in multimodal imagery using target shape information. We introduce a new strategy for robustly aligning scene elements based on the random sampling of shape contour correspondences and on the continuous update of our transformation model’s parameters. We evaluate our solution on a public dataset and show its superiority by comparing it to a recently published method that targets the same problem. To make comparisons between such methods easier in the future, we provide our evaluation tools along with a full implementation of our solution online.

6.1 Introduction

Although automatic multimodal image and video registration has been intensively studied over the years (Zitová and Flusser, 2003; Oliveira and Tavares, 2014), most methods still rely on dense feature matching through area-based similarity measures computation (Pluim et al., 2000, 2003; Krotosky and Trivedi, 2007; Bilodeau et al., 2014). While this approach has the advantage of being able to register non-planar scenes, it is generally very computationally expensive, thus making it unsuitable for emerging mobile and distributed computer vision applications where information fusion might be required. Many multimodal surveillance systems capture images at medium/long distances from their targets, meaning that in those cases, planar models can be assumed without excessively compromising registration quality. In this context, lightweight approximate registration solutions can be adopted to replace their more complex counterparts. Instead of densely searching for local correspondences between images, lightweight solutions rely on sparse correspondences taken from common salient features, which are fitted to a parametric model in order to find a frame-wide rigid or projective transformation (homography).

The main problem behind this simplified approach is finding features that are shared between the studied image modalities and that are easy to automatically identify and match. As presented by Aguilera et al. (2012), traditional keypoint detectors and invariant descriptors are not well suited to multimodal imagery, and require important tuning to achieve decent results. Specialized detection and description methods have been proposed by Aguilera et al. (2012); Mouats and Aouf (2013); Ye and Shan (2014) to address this problem, but these methods do not work for image modalities where the relation between the appearance of objects is not easily defined. These methods are also inadequate when image resolution is too low or when there is a lack of similar textural content between the images. Therefore, more robust means of finding matches between images have to be considered.

In this paper, we propose an automatic video registration method that relies on correspondences found via shape-of-interest matching. Instead of independently analyzing each video frame to extract salient features or edge maps to use for correspondences, we rely on shape contours found using continuous foreground-background video segmentation. While this restricts our approach to applications where the targets of interest can be automatically segmented, it is not affected by the difference in pixel color characteristics of the studied image modalities. This means that our method can be used with any type of sensor, as the appearance of the targets does not matter (as long as they can be segmented).

Our first contribution is a strategy to preserve good shape contour matches throughout the analyzed sequences, which makes our transformation estimation approach robust to continuously imperfect segmentation and small static targets that do not contribute useful correspondences. Instead of temporally accumulating correspondences in a first-in, first-out buffer for RANSAC-based fitting (Fischler and Bolles, 1981), we use a buffer in which correspondences that are identified as persistent outliers, based on a voting scheme, are randomly replaced. Our second contribution is a method for smoothing transitions between transformations estimated at different times based on the approximate overlap of the analyzed foreground shapes. This prevents our solution from locking on to a global registration transformation that might not adequately reflect the nature of the studied scene (e.g. when the scene is not truly planar). Under the assumption that the foreground shapes are the real targets of interest in the scene, this smoothing approach allows improved overall registration through the continuous update of the transformation model’s parameters.

We evaluate our solution by comparing it to a recently proposed method using a public dataset, and show that our overall strategy is superior. To make future comparisons on this dataset and with our method easier, we have made our source code public, and we provide

the video segmentation masks and the evaluation tools we used¹.

6.2 Related Work

As described by (Krotosky and Trivedi, 2007), a planar model can be assumed for image or video registration in two cases: 1) when the sensors are nearly collocated and the alignment targets are far from them (infinite homographic registration), or 2) when all alignment targets lie on the same plane in the scene (planar ground registration). In both cases, the parallax effects caused by the camera baseline distance are assumed to be negligible. Registration can then be achieved by having a human expert select various keypoints in one image modality and search for their equivalent in the other, and by solving the parametric transformation model using these correspondences. Manual registration is however troublesome when large datasets containing many different sensor configurations have to be processed, as it is extremely time-consuming. Automatic video registration thus has to solve the multimodal keypoint detection and matching problem, and provide a robust way to identify the best homography within a temporal window.

Target silhouette information and edge maps have been used before to find correspondences between multimodal image sets (Coiras et al., 2000; Pistarelli et al., 2013; Mouats and Aouf, 2013; Tian et al., 2015). Coiras et al. (2000) estimated transformations in outdoor urban environments by extracting straight lines from edge maps and using them to find correspondences between sets of polygons. More recently, Pistarelli et al. (2013) proposed to use Hough space as the search domain to find multimodal correspondences between segments under similar conditions. Mouats and Aouf (2013) opted to use a keypoint detector based on phase congruency instead of edge points directly, but relied on local edge histograms to describe and match them between modalities. Tian et al. (2015) also used edge maps for thermal-visible face registration, but described point sets using shape context (Belongie et al., 2002). In their case, the infrared images were very contrasted, meaning that many strong edges were easily identifiable. Strategies based on silhouettes and edge maps are not always adequate, as contours representing region boundaries are not guaranteed to be shared between all modalities. For example, when using thermal-infrared sensors, objects with uniform temperatures might not display any edges while some are identifiable in other spectra.

The advantage of video registration over image registration is that methods can rely on target motion to find correspondences more easily. Shape contours obtained via temporal foreground-background segmentation (Han and Bhanu, 2007; Bilodeau et al., 2011a; Zhao

1. <http://www.polymtl.ca/litiv/vid/index.php>

and Sen-ching, 2014; Sonn et al., 2013) or shape trajectories (Caspi et al., 2002; Bilodeau et al., 2011b; Torabi et al., 2012) are viable strategies, as long as they can properly distinguish targets from the background. Correspondences found between shape contours or trajectories are used to find the transformation model parameters, but their quality highly depends on the accuracy of the segmentation algorithm. In the work of Han and Bhanu (2007), a hierarchical genetic algorithm is adopted to quickly find an optimal transformation while avoiding local maxima. Zhao and Sen-ching (2014) used a simplified transformation model using calibration priors based on a 1D-scan approach. The works of Sonn et al. (2013); Caspi et al. (2002); Torabi et al. (2012) all rely on a similar strategy: correspondences from a temporal window are added to a global potential match buffer (or “reservoir”), which is then analyzed using a random sample consensus (RANSAC)-based method (Fischler and Bolles, 1981) to find the parameters which best fit the transformation model. In the work of Caspi et al. (2002), correspondences from all frames are used at once, meaning that it cannot estimate the registration transformation in an online fashion. In the works of Sonn et al. (2013); Torabi et al. (2012), a small first-in, first-out (FIFO) buffer is used to accumulate and analyze a few seconds worth of correspondences. Our own strategy presented in Section 6.3.3 uses a similar RANSAC approach for online transformation estimation, but accumulates contour matches using a random sampling and persistence voting approach (meaning that these matches do not have a predefined lifetime in the buffer).

The shape contour description and matching strategy that most closely resembles ours is that of Sonn et al. (2013): they used the Discrete Curve Evolution (DCE) algorithm originally proposed by Latecki and Lakämper (2000) to prune foreground shapes into hexadecagons with visually similar boundary parts. We believe that more accurate registration can be achieved by describing and matching all shape contour points without pruning, leaving the filtering responsibility to RANSAC. Therefore, for our own contour description and matching needs, we use shape context (Belongie et al., 2002), as detailed in Section 6.3.2.

6.3 Proposed method

Our method can be split into several parts, as shown in Fig. 6.1. First, foreground-background segmentation is used on each video frame to obtain shape contours from targets present in the scene. Contour points are then described and matched using the iterative shape context approach of Belongie et al. (2002). Following that, all matches are added to a correspondence reservoir (i.e. a temporal buffer), which itself is analyzed by a RANSAC algorithm to identify inliers and outliers and to estimate ideal transformation parameters. Finally, the identified inliers are used for persistence voting in the reservoir, and the estimated model parameters

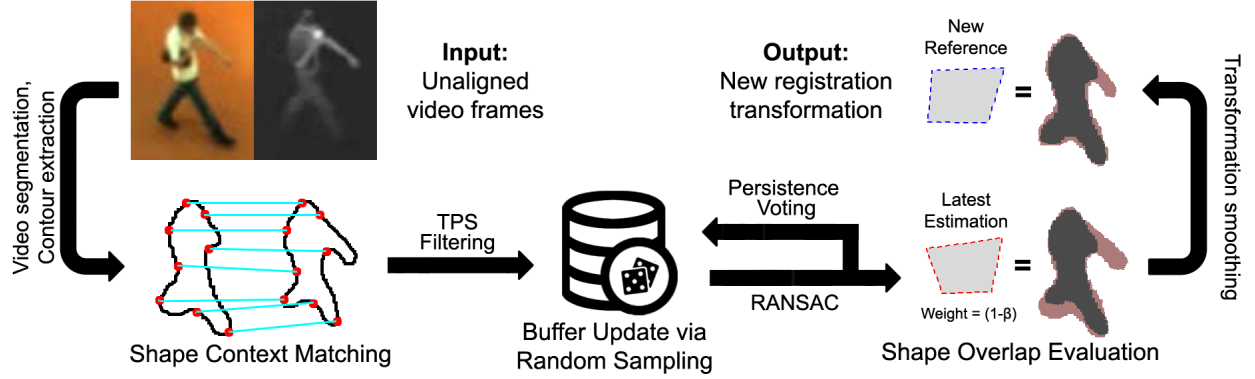


Figure 6.1 Overview of our proposed method’s principal processing stages.

are used to update the reference (or “best-so-far”) registration homography. In the following subsections, we detail each step of this process.

6.3.1 Shape extraction

The initial step in our method is identifying targets of interest in each video sequence using foreground-background segmentation. Since the dataset we use in Section 6.4 only contains sequences with static cameras, we opted for an approach based on change detection via background modeling (commonly referred to as “background subtraction”). The method we use is the one of St-Charles et al. (2015b): it builds a statistical model of the observed scene using color and binary features, and uses feedback mechanisms to dynamically adapt to changing conditions. Getting shape contours from the binary image masks provided by this method is trivial, and no pruning or extra post-processing is done to simplify these foreground shapes. We chose this segmentation method due to its ease-of-use and because it provides good segmentation results in image modalities inside and outside the visible spectrum.

6.3.2 Contour points description and matching

We address the shape contour matching problem as the task of establishing correspondences in a bipartite graph where the disjoint sets are composed of noisy polygon vertices taken from different image modalities. As mentioned earlier, we follow the approach introduced by Belongie et al. (2002) to compare foreground object contours. Our ultimate goal is to find, for each contour point in the first image modality, the contour point in the second modality that offers the best match given their respective position within their original shapes.

This approach is straightforward: first, contour points are all assigned a shape context de-

descriptor which expresses the relative disposition of other contour points in the same modality using a uniform log-polar histogram (as shown in Fig. 6.2). These descriptors are then exhaustively compared using a χ^2 test to determine similarity scores, and the correspondence problem is solved using the Hungarian method (Kuhn, 1955). These three steps are repeated multiple times for each frame in order to eliminate outlying matches from the contours, which are identified after solving the correspondence problem based on their low similarity scores. Between each iteration, a Thin Plate Spline model (TPS, Duchon, 1977) is used to determine the optimal elastic transformation that aligns the filtered contours, and new descriptors based on the transformed shapes are generated. Given a predetermined maximum number of iterations to run, this approach helps identify which correspondences will be used to find the frame-wide registration transformation detailed in Section 6.3.3. Note that the transformation estimated by the TPS method cannot be used for frame-wide image registration, as its elastically fitted solution might cause important distortions in regions far from the analyzed contour points. Therefore, it is only used as a temporary solution, and the contour point matches later added to the correspondence buffer contain their original coordinates.

Unlike the DCE approach of Sonn et al. (2013) that directly relies on Euclidean distances between multimodal contours, our approach is completely invariant to translations and scaling since all distances in shape context descriptors are relative and normalized. Furthermore, it is not restricted to a constant number of points per contour and it does not consider boundary convexity as a shape attribute, meaning it is more robust to noisy shapes caused by inadequate segmentation. The correspondence problem is also solved optimally, which is better than using the greedy matching algorithm of Sonn et al. (2013). Besides, note that due to the unknown relation between the analyzed image modalities, we do not consider local appearance when computing the similarity scores between shape contours.

6.3.3 Correspondence reservoir and voting

Using contour matches from a single frame pair for scene registration would likely result in noisy transformations that disregard large planar scene areas (which might be of interest to some applications). Therefore, the homography we are looking for has to be computed using correspondences taken from multiple frames to ensure accurate scene-wide registration. To address this problem, we use a temporal buffer (or reservoir) to accumulate enough correspondences so that a robust model fitting algorithm, RANSAC (Fischler and Bolles, 1981), can estimate a proper global homography.

In the work of Sonn et al. (2013), a first-in, first-out (FIFO) circular buffer strategy was adopted to keep 100 frames worth of contour point pairs. While easy to implement, the

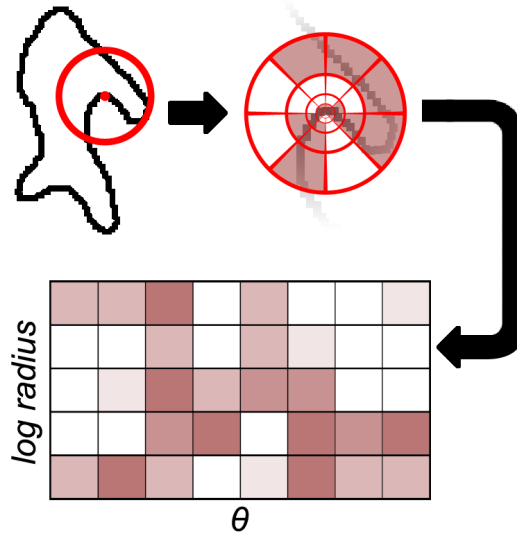


Figure 6.2 Example of shape context description on a human shape contour point using 5 radius bins and 8 angle bins.

primary disadvantage of this approach is that if the targets of interest remain static (or do not move much) during those 100 frames, or if the segmentation is continuously inaccurate, the buffer will be filled with correspondences that are not representative of the sought frame-wide transformation. To solve this problem, we use an identically sized buffer, but instead of following a FIFO rule, we replace the correspondences it contains using a random policy reminiscent of conservative sample consensus models.

Simply put, given a reservoir $R = \{p_1, p_2, \dots, p_N\}$ containing N previously found point pairs, for each new point pair p found by multimodal contour matching, we will randomly pick one of the reservoir pairs and replace it, but only if it is considered a “persistent outlier”. These persistent outliers can be identified based on the number of times they were omitted by the RANSAC algorithm during the estimation of the homography parameters for each new frame. To keep track of this, we define a voting map, noted $V = \{c_1, c_2, \dots, c_N\}$, which accumulates the inlier/outlier counts of each point pair (following RANSAC fitting) based on this logic:

$$c_i = \begin{cases} c_i + 1 & \text{if } p_i \text{ is an inlier according to RANSAC} \\ c_i - 1 & \text{otherwise} \end{cases}$$

For a new frame, all pairs p_i which have negative c_i values are considered persistent outliers. In practice, we determined that with this approach, the proportion of such outliers is always around 50%, which means that the reservoir never saturates and new correspondences can

always be swapped in. We also determined that due to the presence of our reservoir, even a small shape moving across a limited portion of the sensor’s field of view can provide enough contour matches to estimate a good global homography; this is discussed further in Section 6.4.

6.3.4 Homography smoothing

Following the planar ground assumption to simplify video registration tasks might not always faithfully reflect the reality of the observed scenes. Therefore, the goal of our method is not to simply find a good frame-wide homography for the entire analyzed sequence, but to make sure that this transformation adequately aligns the targets of interest, even when non-planar transformations are involved. Basically, we are looking for a middle ground between estimating a timeless, global homography and estimating one that only focuses on aligning currently visible targets of interest without regard to previously found homographies or to the rest of the scene. To achieve this, while processing a pair of video sequences, we continuously update the homography which results in the best registration seen thus far (determined heuristically) using the model parameters found via RANSAC for each new frame. Under the assumption that segmented foreground shapes are truly objects of interest, this allows for slightly improved contour alignment in “almost planar” scenes while minimally affecting the registration quality of other frame regions. When the analyzed scene fully respects the planar assumption, the results provided by this middle ground approach are identical to those of the global, timeless approach.

First, we describe how the registration quality of a homography is appraised at run time. The only data which can be used to assess if a transformation is appropriate are the foreground shapes obtained by the segmentation step. Thus, the appraisal metric we use is the overlap error, defined for two foreground shapes S_i and S_j (both in the same coordinate space) as

$$E(S_i, S_j) = 1 - \frac{\#(S_i \cap S_j)}{\#(S_i \cup S_j)}, \quad (6.1)$$

where $\#(S)$ returns the pixel count of foreground region S . Given perfectly segmented targets in a truly planar scene, a null overlap error would indicate an ideal registration transformation (both shapes are perfectly aligned).

As for the smoothing step itself, given a smoothing factor α (by default, $\alpha=2$), a newly estimated homography H_{new} , its calculated foreground shape overlap error E_{new} , a reference (or “best-so-far”) homography H_{ref} , its reference (or “best-so-far”) overlap error E_{ref} , and finally H_{ref} ’s current overlap error on the latest foreground shapes E_{curr} , we follow the

```

1: if  $E_{new} < E_{curr}$ 
2:   if  $E_{new} < E_{ref} \vee E_{new} < E_{curr} \cdot 2$ 
3:      $\alpha \leftarrow 2$ 
4:   else
5:      $\alpha \leftarrow \alpha + 1$ 
6:   end if
7:    $\beta \leftarrow \frac{\alpha-1}{\alpha}$ 
8:    $E_{ref} \leftarrow E_{ref} \cdot \beta + E_{new} \cdot (1 - \beta)$ 
9:    $H_{ref} \leftarrow H_{ref} \cdot \beta + H_{new} \cdot (1 - \beta)$ 
10: end if

```

Figure 6.3 Homography smoothing algorithm used for each frame.

algorithm presented in Fig. 6.3 to smooth the homography transition between two frames. In summary, if the new homography H_{new} produces an overlap error E_{new} smaller than what the reference homography H_{ref} produced on the current foreground shapes (E_{curr}), then the reference needs to be updated. In that case, if the disparity between the two errors is large enough, or if the new error is simply smaller than the reference error (E_{ref}), the reference homography (H_{ref}) and error value (E_{ref}) will be replaced by the straight average between them and their newly found counterparts. Otherwise, they will be replaced by a weighted mean which depends on the value of the smoothing factor (α).

The role of α is to control the weight given to the reference homography when it is combined with a new one. It is responsible for automatically balancing our method between directly using new homographies for each frame which focus on aligning matched contour points, and converging to a global homography which fits the entire scene more adequately. The latter case can be achieved when $\alpha \rightarrow \infty$, but in practice, this is unlikely to happen; as we will see in Section 6.4, our method never truly stops adapting to newly estimated homographies. Overall, as noted earlier, this smoothing strategy allows for better alignment of contours when considering “almost planar” scenes, and it helps quickly stabilize the registration when processing the first video frames with contour matches.

6.4 Evaluation

Evaluating how well a method behaves in terms of registration quality for “almost planar” scenes is not trivial. In the case of real planar scenes, the sought homography can be found manually, and registration quality can be evaluated by calculating the distance between points projected using this homography and the automatically estimated one. For scenes that do not fully respect the planar assumption, results have to be qualitatively evaluated, or a criterion

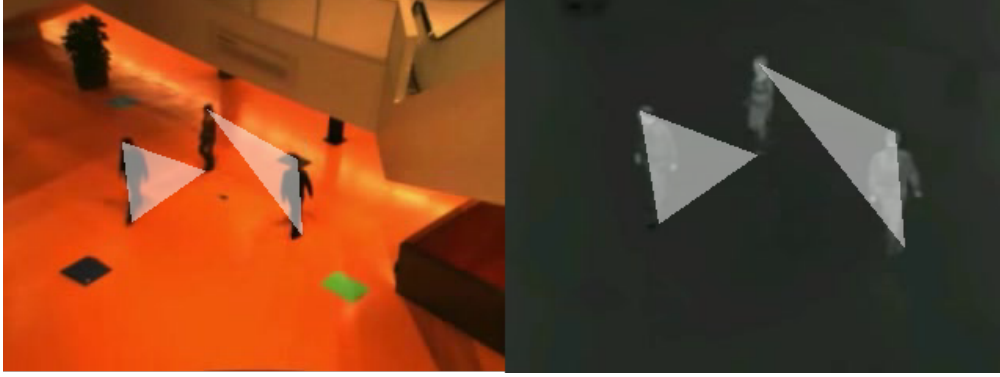


Figure 6.4 Example of the polygons used for quantitative evaluation formed with manually identified keypoints in the ninth sequence pair of the LITIV dataset.

based on the degree of overlap of manually identified scene structures has to be used. Such a criterion is proposed by Torabi et al. (2012): for their own visible-infrared registration dataset, they manually selected points throughout the frames of their sequence pairs which were easily identifiable and matchable, and connected them to create polygons sets (an example of this operation is shown in Fig. 6.4). Once a polygon set is projected into the other's coordinate space, the overlap error can be used as the criterion to judge the image registration quality. By choosing points on scene elements that do not respect the planar assumption, one can highlight part of the non-rigid transformation that needs to be modeled by the automatic approach. Using these polygons instead of the segmented foreground shapes to calculate the overlap error eliminates the uncertainty caused by inaccurate video segmentation, and it allows a better coverage of the observed scene.

For our own tests, we also use the LITIV dataset of Torabi et al. (2012). However, since the polygons they manually drew for their quantitative evaluations could not be obtained, we had to draw our own. To make future quantitative comparisons between video registration methods easier, we have made these new polygon sets, the segmentation masks we obtained from the method of St-Charles et al. (2015b) as well as our evaluation tools available online, along with a C++ implementation of our method.

In total, nine visible-infrared sequence pairs of lengths varying between 200 and 1200 frames were analyzed. These were taken with different sensor baselines at various orientations from the ground plane. Homographies found by matching manually identified points are provided in the dataset, and are used in the following figures to illustrate ground truth global registration results. Sonn et al. (2013) provided us with an implementation of their own method, which we use as our basis for comparison. For fairness, both methods rely on the same segmentation results, both are evaluated using the overlap error defined in (6.1), and both use

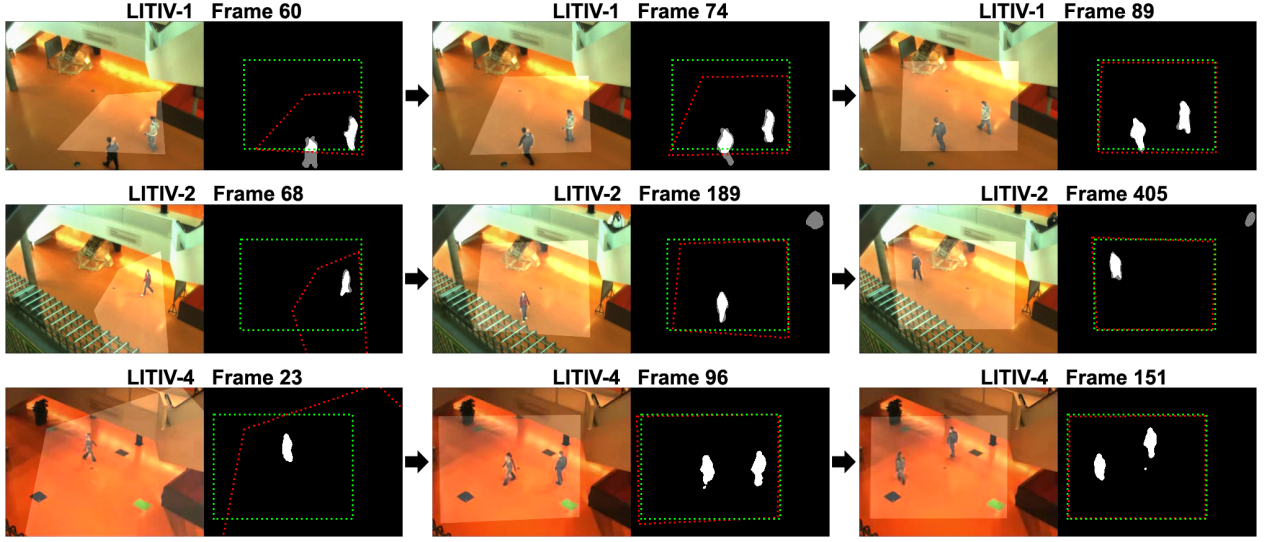


Figure 6.5 Registration results obtained at various moments of the first, second and fourth sequence pairs of the LITIV dataset using our proposed method. The left image in each pair shows the estimated frame-wide registration, and the right shows foreground shape registration at the same moment. The red dashed polygon shows the estimated transformation applied to the infrared image boundary, and the green one shows the ground truth transformation applied to this same boundary.

a single parameter set for all sequences.

We show in Fig. 6.5 how our method performs in the first, second and fourth sequence pairs of the studied dataset. We can see that for various sensor placements, an acceptable alignment of foreground shapes is found soon after a target first becomes visible (this happens at different moments in each sequence; our earliest results are shown in the left column). Over time, this alignment is refined to make the registration of other scene elements possible. For the first sequence pair (top row), even though the detected targets only travel in a small portion of the sensor’s field of view, a very good transformation is found less than 30 frames after the first appearance of foreground shapes.

For quantitative evaluation, since online video registration takes time to stabilize, it makes little sense to compare average error measures that might be affected by important aberrations present early in the analyzed sequences. Previous works (Sonn et al., 2013; Torabi et al., 2012) addressed this issue by either considering only the minimum errors achieved for each sequence pair, or by arbitrarily picking time intervals where the method is considered “stable”, and computing average metrics from those. In our case, in order to present a global view of how our method adapts to each newly estimated homography, we present error-to-time curves for

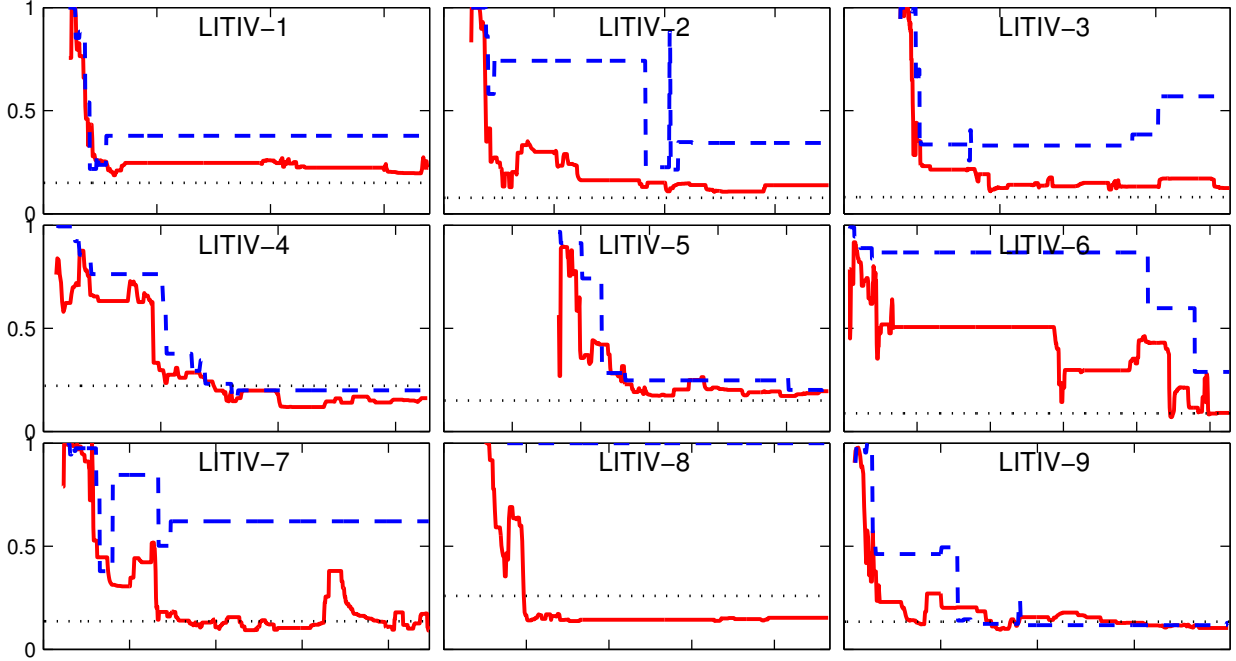


Figure 6.6 Polygon overlap errors obtained using our method (solid red), the method of Sonn et al. (2013) (dashed blue), and the ground truth homography (dotted gray) for the full lengths of all sequence pairs of the LITIV dataset.

our method and compare them to those of Sonn et al. (2013) in Figs. 6.6 and 6.7. The overlap errors of Fig. 6.6 are calculated using (6.1) with the ground truth polygonal shapes, and the Euclidean distance errors of Fig. 6.7 are calculated with the vertices of these polygons. In both cases, the transformation is applied on the infrared set, and the common coordinate space is in the visible image.

From the results shown in Fig. 6.6, we can note that our method reaches lower overlap errors faster than the method of Sonn et al. (2013), stabilizes at those levels more often, and manages to outperform the ground truth homography in five out of nine sequence pairs (LITIV-4 and LITIV-6 through LITIV-9). Outperforming the ground truth is possible because its homography reflects a global transformation only ideal for a planar scene, and the manually drawn polygons, just like the rest of the scene, do not fully respect the planar assumption. Since these polygons are partially based on points found on the targets of interest, transformations that focus on the alignment of these targets are more likely to get smaller overlap errors.

Besides, our method had no trouble estimating frame-wide registrations for LITIV-7 and LITIV-8, unlike the method of Sonn et al. (2013), which was unable to find adequate ho-

Table 6.1 Minimum overlap errors achieved for all video sequence pairs of the LITIV dataset (bold entries indicate the best result).

Sequence	Proposed	Sonn et al. (2013)
LITIV-1	0.187	0.217
LITIV-2	0.106	0.214
LITIV-3	0.108	0.258
LITIV-4	0.118	0.152
LITIV-5	0.172	0.167
LITIV-6	0.069	0.289
LITIV-7	0.091	0.379
LITIV-8	0.137	1.000
LITIV-9	0.095	0.117

homographies through the entire lengths of these sequences. In LITIV-7, we can see a strong temporary increase in overlap error near the end of the sequence: this is due to the matching of a single small shape with a strong shadow which produces outliers for a long period of time. This error quickly fades as better homographies are estimated after the target starts moving in the following frames.

As shown in Table 6.1, our method reached much lower minimum errors than the method of Sonn et al. (2013) in all but one sequence pair (LITIV-5), where the difference between the two is very small. In all cases except LITIV-4 and LITIV-7, an homography resulting in an overlap error of less than 50% was found after processing less than 30 frames (after the first appearance of foreground) containing at least one shape visible in both fields of view.

The curves illustrating polygon vertices registration errors shown in Fig. 6.7 generally depict the behavior observed in Fig. 6.6, but with a larger gap between our method and the method of Sonn et al. (2013) for LITIV-6 through LITIV-9. In three of those cases, the curves of Sonn et al. (2013) are mostly outside the 15 pixels error range of the graphs, but ours reach 2 or 3 pixels errors by the end of each sequence pair. We can also notice in the last graph of this new figure (LITIV-9) that our smoothing approach prevents our solution from locking onto a “decent” homography, and instead continuously refines one to achieve extremely small registration errors at the end of the sequence.

As for the computation time, when operating directly on the foreground shapes provided by the video segmentation algorithm, our proposed registration method processed video sequences at speeds varying between 15 and 150 frames per second, depending on the number of targets in the scene (we used C++ code on a laptop’s 4th generation Intel i7 CPU at 2.8 GHz).

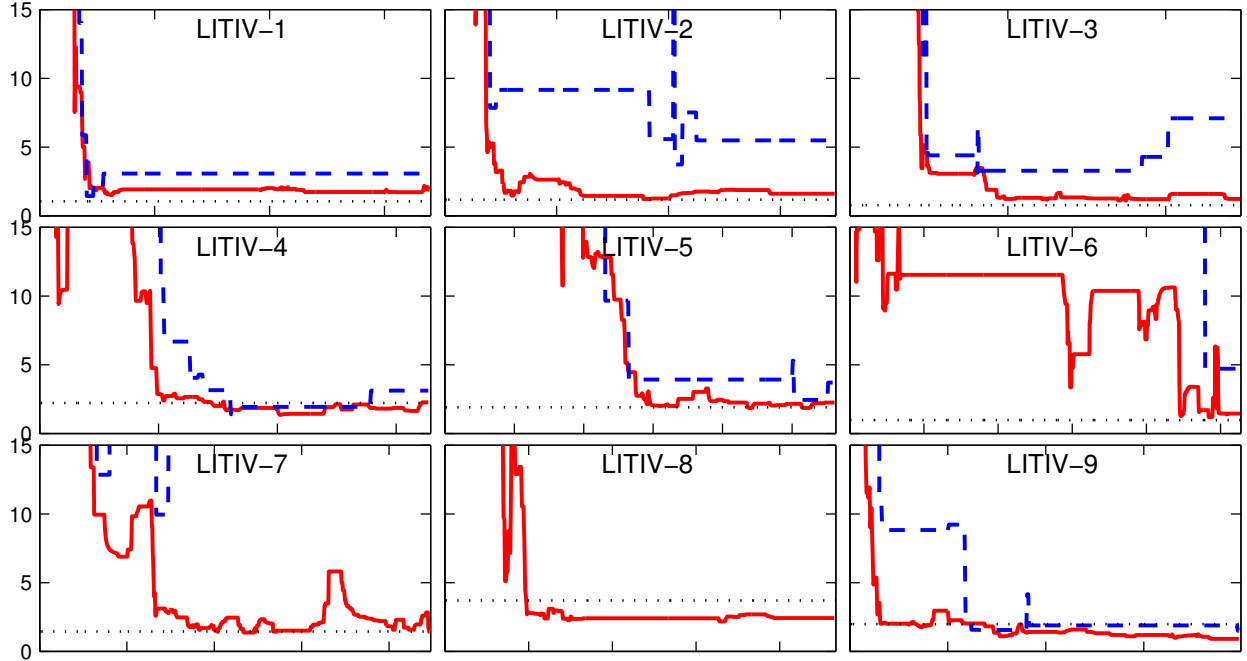


Figure 6.7 Polygon vertices Euclidean distance errors (in pixels) obtained using our method (solid red), the method of Sonn et al. (2013) (dashed blue), and the ground truth homography (dotted gray) for all sequences of the LITIV dataset. Note that the Y axis has been cropped similarly for all graphs.

6.5 Conclusion

In this paper, we presented an online multimodal video registration method that relies on the matching of shape contours to estimate the parameters of a planar transformation model. We showed that randomly sampling a correspondence buffer and adding temporal smoothing between estimated homographies can quickly lead to stable results, and even allow “almost planar” scenes to be registered adequately. Our solution outperforms a recently published method. It manages to align manually annotated polygon sets based on scene structures better than the ground truth homography could in the majority of sequences we tested. Given adequate target segmentation, this approach could be used to register image sequences from cameras which are slowly moving, but still have overlapping fields of view. It could also be generalized to non-planar registration if transformations were continuously estimated for each foreground shape.

CHAPITRE 7 ARTICLE 4: ONLINE MUTUAL FOREGROUND SEGMENTATION FOR MULTISPECTRAL STEREO VIDEOS

St-Charles, P.-L., Bilodeau, G.-A., Bergevin, R.

Article submitted to IJCV in February 2018.

(Submission ID: VISI-D-18-00108)

Abstract

The segmentation of video sequences into foreground and background regions is a low-level process commonly used in video content analysis and smart surveillance applications. Using a multispectral camera setup can improve this process by providing more diverse data to help identify objects despite adverse imaging conditions. The registration of several data sources is however not trivial if the appearance of objects produced by each sensor differs substantially. This problem is further complicated when parallax effects cannot be ignored when using close-range stereo pairs. In this work, we present a new method to simultaneously tackle multispectral segmentation and stereo registration. Using an iterative procedure, we estimate the labeling result for one problem using the provisional result of the other. Our approach is based on the alternating minimization of two energy functions that are linked through the use of dynamic priors. We rely on the integration of shape and appearance cues to find proper multispectral correspondences, and to properly segment objects in low contrast regions. We also formulate our model as a frame processing pipeline using higher order terms to improve the temporal coherence of our results. Our method is evaluated under different configurations on multiple multispectral datasets, and our implementation is available online.

7.1 Introduction

The detection and segmentation of objects of interest based on motion analysis in video sequences is a fundamental early vision task. In the context of video surveillance and intelligent environments, objects of interest (or “foreground” objects) are disruptors that temporarily break the natural state of the observed scene (the “background”). Several types of approaches exist to classify image regions as being “of interest” based on this criteria (see Bouwmans, 2014; Perazzi et al., 2016). While these all have different qualities, they suffer from the same fundamental drawback: if the contrast between an observed object and the background becomes too low, our ability to detect and segment it automatically deteriorates. This problem is not specific to the visible light spectrum, as this camouflaging can occur with any imaging

modality. However, interestingly, the phenomena describing the appearance of an object and the conditions under which it becomes harder to identify are rarely shared across several imaging modalities. This is especially true when considering for example the visible and Long-Wavelength Infrared (LWIR) spectra, as the correlation between the temperature of an object and its visible appearance is very weak (see Bilodeau et al., 2011b). We show an example of this in Figure 7.1. In fact, many surveillance systems rely on the complementarity of these two imaging modalities to detect abnormal events: the visible spectrum can easily identify large objects near ambient temperatures (e.g. vehicles), and the LWIR spectrum can easily identify objects that exhibit abnormal temperatures (e.g. animals, engine parts).

Integrating data captured from different spectral bands to attain benefits in recognition tasks is however not trivial. If the optical axes of the sensors are not already aligned using a beam splitter, a registration method has to be used to bring data points back into a common coordinate system. The image registration problem has been thoroughly studied for identical sensor pairs, but multispectral registration is fundamentally more challenging (c.f. Zitová and Flusser, 2003). Since the appearance of objects cannot be directly relied upon to find local correspondences, higher level image features such as edges have to be used instead. These are typically harder to compute, and often result in a loss of registration accuracy at the pixel level when parallax effects are not negligible.

Past research has focused mostly on the problems of binary (or foreground-background) segmentation and multispectral image fusion/registration as separate issues. Yet, holistic approaches such as the ones of Torabi et al. (2012); Zhao and Sen-ching (2014) can outperform combinations of distinct methods on identical tasks. These holistic approaches first optimize registration using foreground object contours or trajectories as high-level features, and then use integrated image data to improve their segmentation. Solving both problems at once would be more beneficial, but this goal implies a “*chicken-and-egg*” dilemma: the result of one task is needed to obtain the other. An ideal holistic method should thus adopt an iterative optimization approach to resolve this issue. In the case of video sequences, proposed solutions should also consider the temporal redundancy of data to improve their performance. Finally, in the context of surveillance applications, the entire process should function without any human supervision, and allow frame pairs to be processed one at a time.

In this paper, we propose a holistic method to address both segmentation and registration problems by inferring their solutions alternately using move-making algorithms on a set of conditional random fields. We use self-similarity descriptors and shape cues to find proper pixel-level matches across imaging modalities in non-planar scenes, and integrate image data to improve foreground-background partitioning. This integration is achieved by iteratively

refining local color models and shape contour positions while continuously realigning data sources. Our two goals are formulated as distinct energy minimization problems, and we use provisional inference results as dynamic priors to converge to a global solution. We also rely on dynamic temporal connections updated via motion cues to improve segmentation coherence over long image sequences.

Our principled bottom-up approach requires no human intervention, and relies on no prior knowledge of the foreground objects’ nature. Our models are formulated so that imaging modalities can be combined without assumptions about their specific characteristics, as image regions containing discriminative data are automatically identified. This power of discrimination is exploited to scale the importance of each imaging modality when registering and integrating pixel-level data. It is also used to speed up shape contour evolution in low contrast regions by reducing penalties for label discontinuities when the other view possesses strong intensity gradients in its corresponding regions. Besides, we tackle foreground-background segmentation in the general case of video surveillance, meaning we assume the scene might contain multiple foreground objects at different depths and scales, and that they might not always be moving. This differs significantly from traditional cosegmentation methods, as we make no assumption regarding the distribution of foreground and background regions in the observed scene.

Through our experiments, we show that our primary goal, mutual foreground segmentation, can be achieved efficiently despite low contrast and other adverse conditions in both visible and LWIR images. Performance evaluations show that our approach outperforms both supervised and unsupervised monocular segmentation methods in terms of F_1 score on the VAP dataset of Palmero et al. (2016). Compared to the recent video segmentation method of St-Charles et al. (2016a), our method improves its average F_1 score by 13%, from 0.766 to 0.866. To help future benchmarking on this task, we offer a new multispectral video dataset for the simultaneous evaluation of registration and segmentation performance¹. Finally, we also offer our source code and testing framework online².

Note that our method was previously introduced (St-Charles et al., 2017). Here, beyond presenting an extended description of our approach, we introduce a spatiotemporal term to our model and study its effect on segmentation accuracy, we discuss new experiments on two pre-existing datasets, and we present new evaluation results on a novel non-planar RGB-LWIR video dataset.

The paper is organized as follows. In Section 7.2, we present previous works related to our

1. <http://www.polymtl.ca/litiv/vid/index.php>

2. <https://github.com/plstcharles/litiv>



Figure 7.1 Examples of mutual foreground segmentation in low contrast conditions for RGB-LWIR image pairs. On the left, the person is only partly perceptible in the LWIR spectrum due to a winter coat, but is clearly perceptible in the visible spectrum. The opposite is true on the right, where legs are hard to perceive in the visible spectrum, but easy to perceive in the LWIR spectrum.

multispectral mutual segmentation problem, and highlight major differences. In Section 7.3, we describe our dual modeling approach, inference strategies, and implementation details. In Section 7.4, we present parameter and configuration studies, and evaluation results on three publicly available datasets. Lastly, we conclude with some remarks in Section 7.5.

7.2 Previous Work

The problem of foreground-background segmentation in images is difficult to tackle without some assumptions or constraints. Monocular segmentation solutions typically rely on visual saliency hypotheses (e.g. single foreground object roughly focused) or human supervision to obtain good results (Arbelaez et al., 2011; Rother et al., 2004). The same problem in the temporal domain (i.e. on image sequences) is easier to address due to the additional assumptions that can be made regarding object or scene motion. Multiple families of methods exist in video segmentation; the main ones are listed here. Background subtraction methods work by building a model representing the background under the assumption that the camera is static. These methods then perform one-class pixel classification to label all outliers as foreground without supervision (Bouwman, 2014). These methods are favored in cases where foreground objects can temporarily become immobile, as they will retain their labeling for some time. Other video object segmentation approaches instead extend the concept of visual saliency into the temporal domain using highly connected graph structures (Perazzi et al.,

2016). These approaches can usually be applied to sequences with changing viewpoints, but are computationally more demanding. Finally, motion clustering methods exist that rely on optical flow or trajectory points partitioning to identify image regions that behave differently from their surroundings (Tron and Vidal, 2007). The strong link between motion partitioning and video object segmentation has also become a focus in recent years (Jain et al., 2017; Cheng et al., 2017). Also, in semi-supervised settings, approaches based on end-to-end neural networks have also become increasingly popular for single object video segmentation (Cheng et al., 2017; Caelles et al., 2017).

Foreground-background segmentation can become easier if multiple images of the object(s) of interest are available. Two families of methods have been developed for this circumstance: cosegmentation methods and mutual segmentation methods. Cosegmentation methods typically rely on visual saliency assumptions (e.g. shared foreground appearance and low background correlation across different views), and assume a single object is targeted and shared throughout all views (Rother et al., 2006; Zhu et al., 2016). Interestingly, cosegmentation methods can also work with different object instances from the same object category (Vicente et al., 2011). On the other hand, mutual segmentation methods typically assume that the same object instance is observed from multiple viewpoints, and optimize the geometric consistency of the extracted foreground region (Djelouah et al., 2015; Jeong et al., 2017; Riklin-Raviv et al., 2008). Our work falls into this second family of methods, as we assume the use of a synchronized stereo pair for data capture.

Previous mutual segmentation methods have typically focused on single-spectrum imaging (Riklin-Raviv et al., 2008; Ju et al., 2015; Bleyer et al., 2011), or have used depth sensors to solve the registration problem and to provide a range-based solution for foreground object detection (Jeong et al., 2017; Djelouah et al., 2015; Zhang et al., 2016). Of these, our proposed method is closest to the work of Riklin-Raviv et al. (2008), who termed the idea of “mutual segmentation” for objects in visible image pairs. Their approach addresses the uncertainty of object boundary localization under occlusions and noise by iteratively optimizing active contours without supervision. Their use of a biased shape term however entails that a free parameter directly controls the elimination of ambiguous shape segments in the image pair. In our work, we avoid this parameterization issue by relying on local saliency and self-refining color models to automatically integrate multiple view data. Our object contours then expand and contract until they naturally converge. Besides, the method of Riklin-Raviv et al. (2008) considers that all images are related only by planar projective homographies, and thus it cannot handle parallax issues in 3D scenes. This latter problem was addressed by Ju et al. (2015), who also proposed a contour-based modeling approach for mutual foreground segmentation in stereo pairs. This more recent approach however relies on the assumption

that near-perfect foreground contours obtained via human supervision are available in at least one of the views. Lastly, the work of Bleyer et al. (2011) is also somewhat related to ours: they tackle disparity (or parallax) estimation for calibrated stereo pairs using a piecewise planar model based on object segmentation. However, their main goal is scene-wide data registration, which is very computationally demanding. According to Tippetts et al. (2016), processing an image pair took the method about 20 minutes. In our case, we only focus on the registration and segmentation of foreground objects classified as such in a video surveillance mindset. This makes our proposed approach much more lightweight and applicable to real data streams.

The use of multispectral data (other than RGBD) has been mostly neglected in the context of mutual segmentation or cosegmentation due to the registration problem. As stated before, this difficulty is due to the (typically) low correlation between the appearances of objects in different spectral bands (see Zitová and Flusser, 2003). Beam splitters can be used to avoid the registration problem altogether (Bienkowski et al., 2012; Hwang et al., 2015). These setups are however very delicate, and they induce color distortions. Moreover, the elimination of parallax also prevents the recovery of depth information from the scene.

In practice, if the chosen spectral bands are not too distant in terms of their imaging characteristics (e.g. visible light and near-infrared), modern image descriptors and similarity measures can be used to find local correspondences with varying degrees of success (see Pinggera et al., 2012). These “close” spectrum pairs are however less interesting to integrate in machine vision systems due to their resemblance. On the other hand, traditional appearance-based matching approaches suffer when distant spectrum pairs such as visible and Long-Wavelength Infrared (LWIR) are selected (Bilodeau et al., 2014). Multispectral registration thus has to rely on higher level features that encapsulate raw object appearance in order to find proper local correspondences. In the recent literature, some have relied on edge matching in local neighborhoods (Coiras et al., 2000; Mouats and Aouf, 2013) or in Hough space (Pistarelli et al., 2013) to resolve this problem. Edge-based approaches are however more suited to man-made environments, and underperform in more general settings (e.g. open terrain) where large intensity gradients are rarer or more weakly correlated between imaging modalities.

Other works have instead addressed the registration problem in the temporal domain by adopting motion-based cues (Torabi et al., 2012; Zhao and Sen-ching, 2014; Nguyen et al., 2016), which is more similar to our approach. In the work of Torabi et al. (2012), the trajectories of foreground objects are used for high-level registration based on the idea that position and motion are fully independent of appearance. In the works of Zhao and Sen-ching

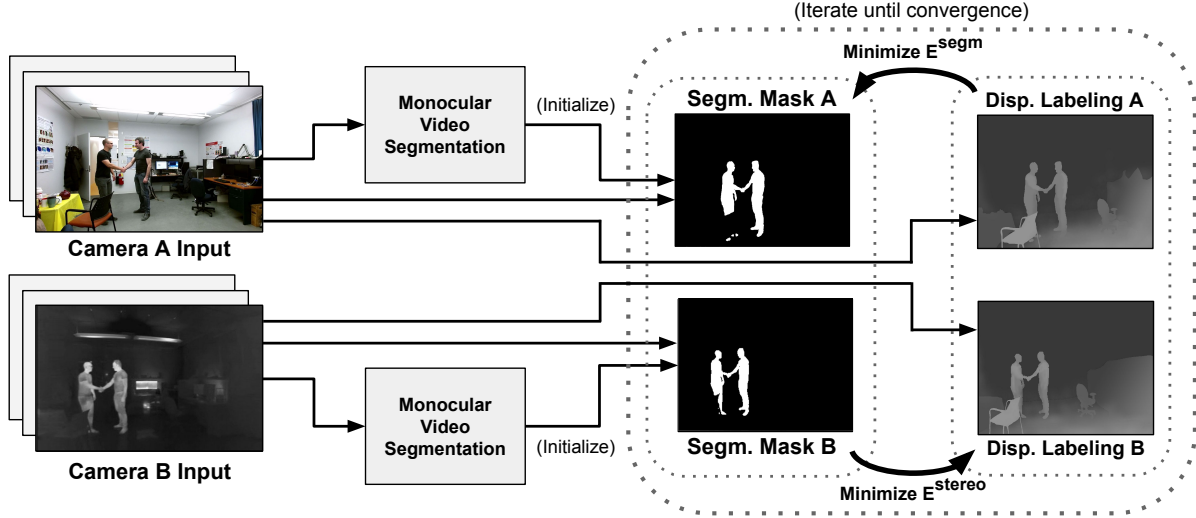


Figure 7.2 Flowchart of the proposed method. A monocular video segmentation method is first used to initialize segmentation masks for both cameras individually. Then, the energies of the stereo and segmentation models (described in Sections 7.3.1 and 7.3.2, respectively) are alternately minimized until a proper global solution is reached. The output of our method then consists of the refined segmentation masks of the input frames, and of the reciprocal disparity labelings computed for both cameras.

(2014) and Nguyen et al. (2016), foreground shapes obtained via background subtraction are used for contour matching. This latter strategy has been shown to be more pixel-accurate for the registration of foreground objects, but it still depends strongly on the performance of the segmentation method used. In our proposed method, we address this problem by combining contour-based registration and segmentation into a global optimization framework.

Finally, as for the combination of multispectral registration and segmentation, we can highlight the existence of a few papers. Torabi et al. (2012) propose a solution based on object-wise planar registration, and improve segmentation masks obtained via background subtraction by combining multispectral data using a sum-rule approach. Zhao and Sen-ching (2014) also rely on object-wise planar registration, and use multiple object trackers to improve the results of parallel segmentors *a posteriori*. In this case, the methods are run in cascade to resolve the “*chicken-and-egg*” optimization dilemma stated earlier. The strategies of Torabi et al. (2012) and Zhao and Sen-ching (2014) do not handle occlusions well due to their high level registration approach, and only provide a single-pass improvement to the segmentation results of a given frame pair. Palmero et al. (2016) introduced a human body segmentation method for trimodal (RGBD-LWIR) image sequences based on feature fusion using a random forest classifier. They also avoid pixel-level registration by predefining a set of homographies to use at runtime based on detected foreground object depth. Davis and Sharma (2007)

proposed a dual background subtraction model and contour extraction technique to improve RGB-LWIR foreground fusion based on local visual saliency evaluation. Similarly, Li et al. (2017) proposed a background subtraction method based on the low-rank decomposition of integrated RGB-LWIR pairs to improve foreground segmentation in a global framework. The main shortcoming of these latter two works is that they only handle planar scenes (i.e. scenes where parallax issues are negligible) using a single predefined homography. To the best of our knowledge, no method has previously been proposed to tackle multispectral non-planar registration and mutual foreground segmentation simultaneously.

7.3 Proposed Approach

Our approach can be described based on its two main components: the stereo matching model for disparity (or parallax) estimation on epipolar lines, described in Section 7.3.1, and the shape matching model for binary image segmentation, described in Section 7.3.2. These two models are conditional random fields formulated as discrete energy functions that tackle the multispectral registration and segmentation problems in an integrated fashion. Our energy functions are minimized alternately using move-making algorithms, as described in Section 7.3.3. The flowchart in Figure 7.2 illustrates our approach.

We begin with an introduction of the general terms and notation used in this section. Given a set of rectified images $\mathcal{I} = \{I_k\}$ (with $k = \{0, 1\}$ in the case of a stereo pair), the disparity label space $\mathcal{L}_D = \{0, \dots, d_{\max}\}$, and the background-foreground label space $\mathcal{L}_S = \{0, 1\}$, our goal is to find the optimal pixel-wise disparity and segmentation labelings $\mathcal{D} = \{\mathcal{D}_k\}$ and $\mathcal{S} = \{\mathcal{S}_k\}$ such that:

$$\mathcal{D}_k = \underset{D_k}{\operatorname{argmin}} E_k^{\text{stereo}}(D_k), \quad (7.1)$$

$$\mathcal{S}_k = \underset{S_k}{\operatorname{argmin}} E_k^{\text{segm}}(S_k), \quad (7.2)$$

where $D_k = \{d_p : p \in I_k, d_p \in \mathcal{L}_D\}$ is a disparity labeling, $S_k = \{s_p : p \in I_k, s_p \in \mathcal{L}_S\}$ is a segmentation labeling (or mask), and where the energy cost functions E_k^{stereo} and E_k^{segm} are described in Sections 7.3.1 and 7.3.2, respectively. For now, note that these functions are linked through their estimation results, $\mathcal{D}_k$ and $\mathcal{S}_k$, which are used as dynamic priors throughout the minimization. In other words, disparity labels d_p for each pixel p in I_k are used in E_k^{segm} for appearance data integration, and segmentation labels s_p are used in E_k^{stereo} to improve stereo matching. Lastly, note that we sometimes omit the k subscript in the following subsections to simplify the notation, as most equations only deal with one image of the stereo pair at a time.

7.3.1 Stereo Registration Model

We tackle the multispectral stereo registration problem for non-planar scenes using a sliding window strategy for pixel matching. This search for correspondences is limited to an horizontal axis on the image plane due to epipolar geometry constraints. These constraints restrict the disparity (or parallax) between the 2D projections of an observed 3D object point to one dimension (see Hartley and Zisserman, 2003). In short, given the intrinsic and extrinsic parameters of the stereo pair obtained via calibration, we can rectify the input images. This forces the corresponding projection of a 2D point in one view to be located somewhere on the same horizontal line in the other view. While calibration does require human intervention, it is a one-time effort generally accepted in an unsupervised system. It could also be replaced by an automatic approach (e.g. Nguyen et al., 2016).

For a pixel-wise disparity label map D , we define its energy (or cost) to be minimized as

$$E^{\text{stereo}}(D) = E^{\text{appearance}}(D) + E^{\text{shape}}(D) + E^{\text{uniqueness}}(D) + E^{\text{smooth1}}(D). \quad (7.3)$$

Each term in this cost function is crafted to promote a desired property of the output disparity labeling, and is described in detail in the following paragraphs. The first three terms are unary costs summed over all pixels of the image. The appearance and shape terms evaluate the local affinity between a pixel p and its corresponding pixel shifted by d_p in the other view. The uniqueness term penalizes multiple matches with p in the other view. The last term is a sum of pairwise smoothness costs used to penalize irregular disparities in uniform image regions. Note that in order to maximize processing speed for image pair sequences, we keep our stereo model simple. Our results would undoubtedly improve with second-order terms such as those of Woodford et al. (2009) or Kohli et al. (2009), but at an important increase in computational complexity. Moreover, since we only focus on the registration of foreground objects, higher-order surface smoothness priors are not as important here.

Appearance and shape terms. These two terms convey the cost of matching an image patch centered on a pixel $p \in I$ to another one in the second view which is offset according to its disparity label d_p . The terms are both defined as

$$E^{\{\text{appearance}, \text{shape}\}}(D) = \sum_{p \in I} \mathcal{A}(p, r(p, d_p)) \cdot \mathcal{W}(p), \quad (7.4)$$

where $r(p, d_p)$ returns the pixel location in the other view obtained by shifting p by d_p on its epipolar line, $\mathcal{A}(p, q)$ encodes the affinity cost for matching descriptor patches centered at p

and q in each image, and $\mathcal{W}(p)$ encodes the saliency coefficient for pixel p (detailed further down). For the appearance term, the affinity cost map $\mathcal{A}$ is obtained by densely computing local image descriptors over I_0 and I_1 , and by matching them using L2 distance in 15x15 patches to dampen noise. As stated in Section 7.2, classic appearance-based descriptors are not ideal for wide spectrum pairs such as RGB-LWIR. To address this issue, we used DASC descriptors (Kim et al., 2015), which are based on local self-similarity measures. We also tested the LSS descriptor (Shechtman and Irani, 2007) during our preliminary experiments, and found a slight decrease in terms of overall registration performance. For the shape term, we densely compute Shape Context descriptors (Belongie et al., 2002) over S_0 and S_1 , which are the provisional segmentation masks. We then match these descriptors using the same approach as for the appearance term to obtain the shape affinity cost map $\mathcal{A}$. Our hypothesis here is that the combination of these two types of descriptors can provide better matching results than either one alone. However, remember that multispectral matches are often unreliable due to non-discriminative descriptors in uniform image regions or in regions with very low multispectral correlation. To avoid increasing pixel matching penalties in such cases, we multiply the affinity cost by a local saliency coefficient. In both the appearance and shape terms, this local saliency coefficient for a given pixel p is defined as

$$\mathcal{W}(p) = \max \left\{ \mathcal{H} \left(\left[\mathcal{A}(p, r(p, d)) \forall d \in \mathcal{L}_D \right] \right), \mathcal{H} \left(K(p) \right) \right\}, \quad (7.5)$$

where $K(p)$ returns the matrix of local descriptors in the patch centered on pixel p , and $\mathcal{H}(\cdot)$ computes the sparseness metric of Hoyer (2004) over a vector or matrix. This metric returns a value $\in [0, 1]$, meaning $\mathcal{W}(p)$ is also in that interval. In simple terms, if all affinity values are uniform (i.e. all disparity offsets have the same cost), and if the local patch's descriptor bins are all uniform, then $\mathcal{W}(p)$ will take a low value. In turn, this will lower the cost for d_p evaluated through the affinity map $\mathcal{A}$, and make local labeling depend more on neighboring decisions through the smoothness term. A simplified case of this is illustrated in Figure 7.3. Besides, note that in E^{shape} , we nullify the saliency outside foreground regions to avoid influencing background disparity estimation around object contours. We can assume that disparity estimation for background regions will be less accurate due to this missing term contribution, but since we focus on the registration of foreground shapes, this is inconsequential. We study the individual contributions of the appearance and shape terms to the overall performance of our approach in Section 7.4.

Uniqueness term. This unary term is used to penalize having multiple epipolar correspondences tied to the same pixel. This helps spread and equalize disparity labels in occluded and weakly discriminative image regions. Our formulation for this term is different from

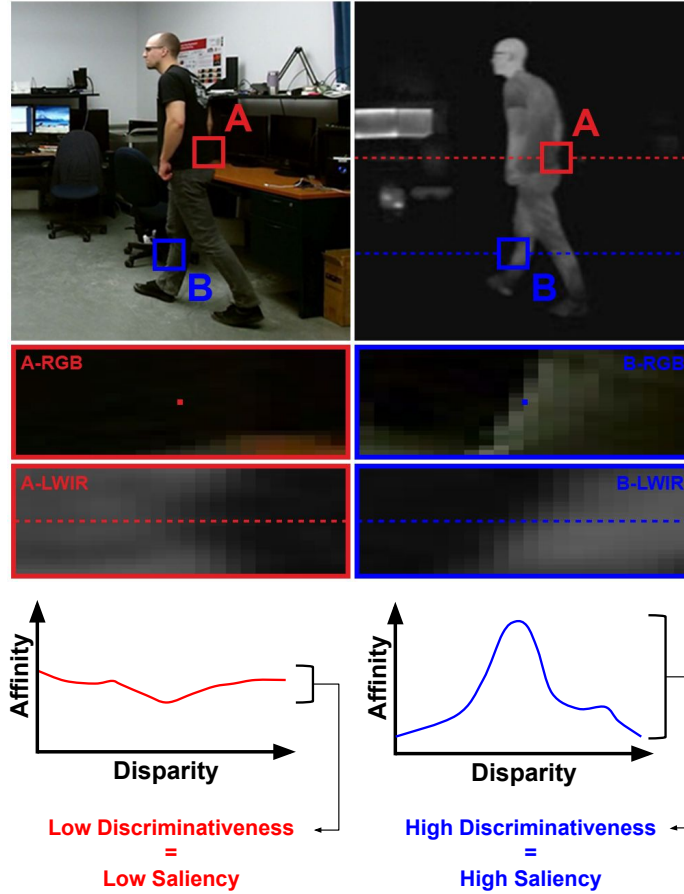


Figure 7.3 Simplified case of saliency evaluation during a correspondence search on an epipolar line. On the left, for the “A” pair, low contrast in one image leads to roughly uniform affinity scores and matching costs, which translate into a low local saliency value. On the right, for the “B” pair, good contrast leads to varied affinity scores and matching costs, and a high local saliency value.

the classic mutual exclusion constraint proposed by Kolmogorov and Zabih (2001), which assigns an infinite cost to all extra correspondences found for a pixel p . Instead, we rely on a soft constraint that permits many-to-one correspondences with gradually increasing costs. This strategy allows our stereo model to temporarily stack extra correspondences during label swaps if the extra cost is worth absorbing. This translates into faster and larger label moves in the early steps of our inference approach, and redistribution of extra correspondence costs over future iterations. Since our method only requires a rough registration of foreground shapes to start properly segmenting them, this allow us to bootstrap the segmentation model without spending too much time on disparity estimation. We define the uniqueness cost for

a pixel p as

$$\mathcal{U}(p) = \begin{cases} \sum_{n=1}^{N(p)-1} \frac{w \cdot n}{w+n-1} & \text{if } N(p) > 1 \\ 0 & \text{otherwise} \end{cases}, \quad (7.6)$$

where $N(p)$ returns p 's current correspondence count with pixels in the other view, and w is a small weight (we used $w=3$ in our tests). For this to work, we need to keep track of pixel correspondence counts ($N(p)$) as latent variables in our model. However, since we use a move-making strategy for model inference, many correspondences might be removed in a single iteration. This makes the total cost of a move over several pixels hard to predict with (7.6) due to its nonlinearity. To solve this problem, we define our uniqueness term as

$$E^{\text{unique}}(D) = \lambda_u \cdot \sum_{p \in I} \left(\frac{-\mathcal{U}(r(p, d'_p))}{N(r(p, d'_p))} + \frac{w \cdot N(r(p, d_p))}{w + N(r(p, d_p)) - 1} \right), \quad (7.7)$$

where d'_p is the previous disparity label of p , and λ_u is a fixed scaling factor. Note that we specify the values used for important factors such as λ_u in Section 7.3.3, and test their contribution to overall performance in Section 7.4.4. The formulation behind (7.7) provides a worst-case energy variation between two labeling states, and guarantees that estimated pixel move costs provided to the move-making algorithm will always be similar but greater than their real evaluated costs once the full move is complete. This is required so that computed label updates always minimize (7.3), which lets us avoid having to fall back to an older solution if the energy increases.

Smoothness term. Lastly, we rely on a classic truncated pairwise (first-order) smoothness term to enforce the spatial coherence of our model. This term penalizes cases where neighboring pixels have irregular disparity labels despite being located in a roughly uniform image region, as described by a weak local gradient magnitude. If the gradient detected between the two pixels is instead strong, the penalty is lowered, as object edges more likely correspond with breaks in labeling. We define this term as

$$E^{\text{smooth1}}(D) = \lambda_{s1} \cdot \sum_{\langle p, q \rangle \in \mathcal{N}} \min(|d_p - d_q|, 10)^2 \cdot G_I^s(p, q), \quad (7.8)$$

with

$$G_I^s(p, q) = \max\left(\exp\left(1 - \frac{|\nabla I(p, q)|}{g}\right) - 0.5, 0\right), \quad (7.9)$$

where λ_{s1} is a fixed scaling factor, $\mathcal{N}$ is the set of first order cliques in the graph model, $\nabla I(p, q)$ returns the normalized local image gradient magnitude between pixels p and q of image I , and g is a constant value defining the expected object contour gradient magnitude

(also specified in Section 7.3.3). The truncation value (10 is used) allows large discontinuities to occur by capping the maximum smoothness penalty.

7.3.2 Segmentation Model

Our segmentation model’s role is to integrate multispectral image data so that foreground objects can be properly segmented in both views, even in low contrast imaging conditions. Our model also needs to be lightweight enough so that cost updates and inference is fast, as shape priors are continuously modified. Since our goal is to build an unsupervised approach, we initialize the priors described below using the approximate masks provided by a monocular segmentation method (i.e. the one of St-Charles et al., 2016a). This method was chosen because it can detect multiple foreground objects at once, and it can keep segmenting them at least partially if they become immobile. In Section 7.4, we show that our method works even when an initialization mask is provided for only one of the two views.

We describe the energy cost of a pixel-wise segmentation proposal S as

$$E^{\text{segm}}(S) = E^{\text{color}}(S) + E^{\text{contour}}(S) + E^{\text{smooth2}}(S) + E^{\text{temp}}(S). \quad (7.10)$$

Once again, the terms of this cost function are defined so that various characteristics expected of the segmentation masks can be promoted. The first two terms are unary costs summed over all pixels, and their role is to influence local segmentation decisions based on image data. The color data term maximizes the separation between the color distributions of foreground and background pixels, while the contour data term penalizes shape mismatches between the views based on distance transforms. The third term is a pairwise smoothness sum similar to (7.8), and is used to penalize labeling irregularities in uniform image regions. Lastly, the temporal term is a sum of higher order clique costs used to enforce temporal labeling coherence. These terms are all described in the following paragraphs. Note that due to the presence of the higher order temporal term in (7.10), our model is built as a multi-layer lattice, as illustrated in Figure 7.4. The top layer’s nodes correspond to the pixels of the latest frame of the video sequence, and lower layers’ nodes correspond to the pixels of older frames. This effectively creates a pipeline where segmentation masks can be improved over time based on new image data. We discuss the improvement achieved using this approach with various pipeline depths in Section 7.4.

Color term. We define the cost for this unary term using a color mixture model for each modality of the stereo pair. We employ the classic approach of Rother et al. (2004) which

relies on Gaussian mixture models to represent foreground and background regions. These models can provide us with the probability that a pixel belongs to the background or foreground based on its color value. In our implementation, we use six mixture components, and use our initial and updated segmentation masks to refine our models after each iteration, in each frame. We define the color cost of all pixels as

$$E^{\text{color}}(S) = \sum_{p \in I} \begin{cases} -\log \left(h(I_p; \beta_1, \mu_1, \Sigma_1) \right) & \text{if } s_p = 1 \\ -\log \left(h(I_p; \beta_0, \mu_0, \Sigma_0) \right) & \text{otherwise} \end{cases}, \quad (7.11)$$

where $h(x; \beta, \mu, \Sigma)$ returns the relative likelihood that the pixel color x fits a Gaussian mixture model with component weights β , means μ and covariance matrices Σ . Note that the parameter subscripts in (7.11) indicate that either the foreground or background model is used based on s_p . These parameters are initialized using *k-means*, and refitted after every minimization step using the new estimated segmentation masks.

Contour term. Next, we define another data term that penalizes label swaps far from shape boundaries, and that combines these boundaries across the stereo pair. Its value is computed using shape distance transforms: first, we build maps in which each pixel is assigned its Euclidean distance to the closest pre-existing foreground or background pixel in the current view. We then use these maps to deduce the label update costs for each pixel in our graph, considering a mix of distances in both views at once (note the use of subscript k below). More specifically, we define our contour term as

$$E_k^{\text{cont.}}(S_k) = \lambda_c \cdot \sum_{p \in I_k} \begin{cases} F_k(p) + \lambda_m \cdot F_{k'}(r(p, d_p)) & \text{if } s_p = 1 \\ B_k(p) + \lambda_m \cdot B_{k'}(r(p, d_p)) & \text{otherwise} \end{cases}, \quad (7.12)$$

where λ_c and λ_m are fixed scaling factors, k' is the opposite index of k in the stereo pair, $F_k(p)$ returns a nonlinear distance cost (described below) for pixel p based on its distance to the closest foreground pixel in the previous segmentation of view k , and similarly for $B_k(p)$ with background pixels. Note in (7.12) that λ_m scales the term's multispectral cost contribution. During our tests, we give it a value $\in]0, 1[$, meaning that shape contours will prefer sticking to their own previous results. This improves the stability of the segmentation while optimizing, reducing the risk of eliminating relevant shape fragments too rapidly. For the nonlinear distance cost function behind $F_k(p)$ and $B_k(p)$, we use an exponential to increase the contrast between close range and long range contour overlaps. More specifically, we use a relation of the form

$$\text{distance-cost}(p) \propto \frac{1}{\exp(-t(p))}, \quad (7.13)$$

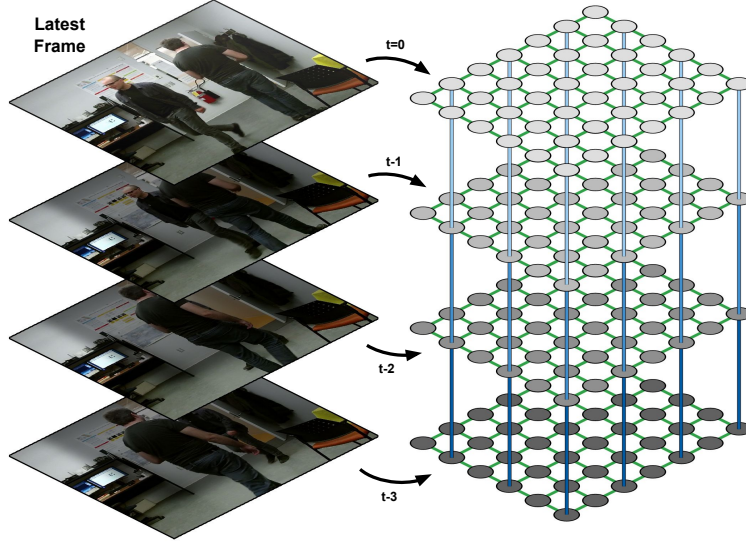


Figure 7.4 Illustration of the simplified frame layering used in our segmentation model for temporal labeling refinement. In green, the first-order cliques that form E^{smooth2} are used to enforce spatial coherence in every layer. In blue, the higher order cliques that form E^{temp} are used to enforce temporal coherence across layers. Note that due to foreground motion, these cliques would not all be linked to the same underlying nodes; in reality, the links are dictated by image realignment based on optical flow.

where $t(p)$ returns the actual Euclidean distance between p and its nearest pixel with a foreground or background label in the previous inference result, depending on the current value of s_p . The contour term's main responsibility is to control the evolution of object contours over several optimization passes. The multispectral contribution allows contours to be modified in regions where only one modality contributes meaningful information. The simple formulation of our contour term also avoids the needless filling of cavities, and it makes no assumption on the foreground-to-background ratio in the images.

Smoothness term. This pairwise term is similar to the one used in (7.8); its role is to penalize label discontinuities everywhere except for image regions where local gradients are strong. In this case, however, we reuse the multispectral contribution idea of (7.12), and apply it to the gradient scaling factor. We define this term as

$$E_k^{\text{smooth2}}(S_k) = \lambda_{s2} \sum_{\langle p, q \rangle \in \mathcal{N}_k} (s_p \oplus s_q) \cdot \left(G_{I_k}^s(p, q) + \lambda_m \cdot G_{I_{k'}}^s(p', q') \right), \quad (7.14)$$

where λ_{s2} is a fixed scaling factor, $\oplus$ is the XOR operator, p' is a shorthand for $r(p, d_p)$, and q' is a shorthand for $r(q, d_q)$. In (7.14), the right-hand parentheses group returns the gradient coefficient with its multispectral contribution, and the left-hand group returns 1 or 0 based

on whether a label discontinuity is found. As before, the use of local image gradients helps “snap” these discontinuities to real object contours. However, the multispectral contribution allows shape contours to settle in uniform regions if the other view possesses a strong local gradient there. Paired with the contour term, this allows our model to properly expand and contract shape boundaries across image modalities. We study the effect of λ_m on the performance of our method in Section 7.4.

Temporal term. Lastly, we present the formulation and role of our temporal term. Unlike the other terms presented so far, this term is based on higher order cliques that are composed at runtime, and updated for each frame. The role of these cliques is to enforce spatiotemporal labeling coherence despite foreground object motion. Our graph structure can be visualized as a stack of analyzed frames; this structure is shown in Figure 7.4. While the depth (or layer count) of this stack is predetermined, its temporal cliques are composed based on node realignments provided by optical flow maps. This allows cliques to remain attached to the same object part despite movement, and thus enforce labeling smoothness across frames. We compute optical flow maps using the method proposed by Kroeger et al. (2016). As for the cost term itself: given $\mathcal{C}$, the set of all temporal cliques in our model, and using the subscript l to identify different temporal layers in these cliques, we define it as

$$E^{\text{temp}}(D) = \lambda_{s2} \cdot \sum_{c \in \mathcal{C}} \sum_{l=1}^{L-1} (s_{c,l} \oplus s_{c,(l+1)}) \cdot G^t(c, l), \quad (7.15)$$

where λ_{s2} is the same scaling factor as in (7.14), L is the pipeline’s depth in frames, $s_{c,l}$ returns the label of the l -th node in clique c , and $G^t(c, l)$ returns a scaling factor for clique c at layer l (described next). Overall, this term is similar to the pairwise smoothness terms described earlier, but it can link more nodes together. However, instead of scaling costs via local image gradients, we rely on temporal image gradients in $G^t(c, l)$. These new gradient values are obtained by computing the absolute color differences between pixels of consecutive frames realigned using optical flow maps. Strong color differences are indicative of uncertain regions where consistency costs should be reduced due to occlusions or bad optical flow estimation. Here, similarly to (7.9), we define the new gradient scale term as

$$G^t(c, l) = \max \left(\exp \left(1 - \frac{|i_{c,l} - i_{c,(l+1)}|}{g} \right) - 0.5, 0 \right), \quad (7.16)$$

where $i_{c,l}$ returns the color value of the l -th node in clique c . We study the contribution of this new term in Section 7.4 given various layer count configurations.

7.3.3 Inference and Implementation Details

Simultaneously minimizing the cost functions defined in (7.3) and (7.10) is not trivial. Both functions rely on each other’s provisional results as dynamic priors, and (7.10) contains a higher order term. Fortunately, recently proposed move-making algorithms can deal with these issues iteratively without having to resort to a move-and-check or rollback strategy (c.f. Lempitsky et al., 2010; Kappes et al., 2013). Such algorithms do not guarantee that a globally optimal solution will be found for either the stereo model, due to its non-binary label space, or for the segmentation model, due to its higher order term. However, in general, both labelings settle in acceptable local minima (i.e. non-degenerate cases) after some time, provided that the initialization masks used are not both completely empty. Our dual-model formulation also naturally converges to such local minima without having to use a “cooling” metaparameter to force a solution after a fixed number of iterations.

We rely on the move-making algorithms of Fix et al. (2011) and Fix et al. (2014) for the inference of our stereo and segmentation models, respectively. Both are modified for use in a dynamic graph structure. While faster inference solutions do exist, these were deemed fast enough for our experiments, even without having to parallelize label moves. In both cases, our move proposals only consist of uniform labeling maps, meaning our inference approach is similar to the α -expansion strategy of Boykov et al. (2001). We build our graphical models using OpenGM (Andres et al., 2012), and reuse the same structure for all frames in a video, updating only the composition of temporal factors in (7.15) as required.

We tackle the alternating minimization of energies (7.3) and (7.10) for each frame of a video by first minimizing the stereo model’s energy using unary terms only, or by realigning its previous disparity labeling result via optical flow. Simultaneously, the segmentation model is initialized using the masks provided by an unsupervised monocular method, as stated earlier. Then, segmentation and disparity label moves are iteratively computed in small batches until no more moves in $\mathcal{L}_S$ can reduce the energy of (7.10). This typically happens after less than three passes over the disparity label space ($\mathcal{L}_D$), and less than 50 moves in the segmentation label space ($\mathcal{L}_S$), the exact number depending on the quality of the initialization. For reference, with our baseline implementation, this is equivalent to approximately 30 seconds worth of processing time on a single core of a 3.7 GHz Intel i7 processor for a VGA-sized image pair.

As for the free parameters listed earlier, we use the following configuration for our tests in the next section:

- Stereo model uniqueness term weight: $\lambda_u = 0.4$

- Stereo model smoothness term weight: $\lambda_{s1} = 0.001$
- Expected object contour gradient intensity: $g = 30$
- Segmentation model contour term weight: $\lambda_c = 7$
- Segmentation model smoothness weight: $\lambda_{s2} = 7$
- Multispectral contribution term weight: $\lambda_m = 0.5$

The values listed above have been empirically found to provide good overall segmentation performance on a small subset of our test data via grid search. As previously noted, we study the effect of several of these parameters on the overall performance of our method in the next section. For optical flow and DASC descriptors computations, we kept the default parameters provided by their original authors. For Shape Context computations, we used 50 pixel-wide descriptors with 10 angular bins and 3 radial bins. For the depth of our frame processing pipeline, we used two temporal layers (i.e. the current frame and the previous one), as adding more did not improve overall performance significantly over the extra processing cost; this is discussed in Section 7.4.4. Finally, to reduce the computational cost when using higher order terms in our segmentation model, we use a stride of two pixels when creating the temporal cliques used in (7.15). For more implementation details, we refer the reader to our source code³.

7.4 Experiments

In this section, we first discuss our evaluation methodology, and then present evaluation results for mutual segmentation and stereo registration. Since close-range (non-planar) multispectral video datasets are quite uncommon in the literature, we had to adapt existing datasets to our problem. For multispectral mutual segmentation, we rely on a modified version of the VAP trimodal dataset of Palmero et al. (2016); the modifications we made are detailed in Section 7.4.2. For stereo registration, we rely on the benchmark of Bilodeau et al. (2014). We follow up with an ablation study of our method in which we remove key terms from our energy functions, and then study the effect of tuning key parameters of these terms. Finally, we provide evaluation results for both segmentation and stereo registration on a newly captured and annotated RGB-LWIR dataset for future comparisons.

7.4.1 Evaluation Methodology

Since our primary goal is mutual foreground segmentation, we employ binary classification metrics for the first part of our evaluation. Commonly used metrics in the context of video

3. <https://github.com/plstcharles/litiv>

Table 7.1 Evaluation results on the multispectral video segmentation dataset of Palmero et al. (2016). Bold results are the best in that category across all methods.

Method	Metric	Scene 1		Scene 3		Overall		
		visible	LWIR	visible	LWIR	visible	LWIR	Average
St-Charles et al. (2016a) (unsupervised)	Pr	0.820	0.755	0.716	0.514	0.768	0.635	0.701
	Re	0.810	0.975	0.688	0.969	0.749	0.972	0.861
	F_1	0.815	0.851	0.702	0.672	0.758	0.762	0.760
Rother et al. (2004) (GrabCut; supervised)	Pr	0.685	0.808	0.653	0.847	0.669	0.828	0.748
	Re	0.759	0.896	0.929	0.916	0.844	0.906	0.875
	F_1	0.721	0.850	0.767	0.880	0.744	0.865	0.804
Proposed method (unsupervised)	Pr	0.894	0.860	0.788	0.749	0.841	0.804	0.821
	Re	0.902	0.901	0.918	0.937	0.910	0.919	0.914
	F_1	0.898	0.880	0.848	0.833	0.873	0.857	0.866

segmentation are Precision (Pr), Recall (Re), and F_1 score. These are based on three types of pixel-wise classification result counts, namely True Positives (TP), False Positives (FP), and False Negatives (FN). These metrics are defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (7.17)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (7.18)$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (7.19)$$

In all three cases, higher values indicate better performance. The F_1 score corresponds to the harmonic mean of the precision and recall scores. We use it as an overall indicator of binary segmentation performance, as it was shown in the work of Goyette et al. (2012) to be strongly correlated with the final ranking of methods on a large binary segmentation dataset based on numerous other metrics.

Our second goal is to evaluate stereo registration performance. For this, we employ the strategy of the Middlebury dataset (Scharstein et al., 2014), and report the percentage of pixels labeled with disparity errors larger than some fixed distance thresholds (in pixels). We also report average frame-wide pixel disparity errors, noted $\bar{d}_{\text{err}}$ below. In this case, lower values indicate better performance.

7.4.2 VAP 2016 Dataset

For this first part of our evaluation, we adapted the dataset of Palmero et al. (2016) to our needs. This dataset was originally intended for the trimodal (RGBD-LWIR) detection and segmentation of people in images, and it is provided as a set of videos. It consists of 5724 image triplets split into three scenes, with their associated groundtruth foreground-background segmentation masks. We obtained the calibration data used by the original authors to roughly register scene contents via homographies, and rectified all RGB and LWIR image pairs using the OpenCV calibration toolbox. The depth images were left unused during all our experiments, and the second scene was removed due to missing calibration data. Finally, to avoid skewing the performance evaluation by continuously segmenting empty frames or frames with purely static and/or unoccluded foreground regions, we manually selected a subset of groundtruth masks for our experiments. These masks were picked at a rate of roughly 2 Hz from all originally available masks while focusing on time spans with people interacting.

We present the segmentation performance of our proposed method, as well as the performance of baseline video and image segmentation methods in Table 7.1. We could not evaluate the performance of the works listed in the last paragraph of Section 7.2 that simultaneously tackle segmentation and registration due to a lack of open-source code and datasets. Besides, comparing our results to those of other methods that assume single-spectrum data or planar scenes would also be unfair. For the video segmentation baseline, we rely on the method of St-Charles et al. (2016a), which is fully unsupervised. For the image segmentation baseline, we rely on the GrabCut method of Rother et al. (2004), and provide this method with bounding boxes for all foreground objects. We used OpenCV’s GrabCut implementation, and ran five iterations per image.

We can observe that our proposed method outperforms the unsupervised video segmentation approach of St-Charles et al. (2016a) in terms of overall F_1 score by a margin of 0.1, equal to a relative improvement of over 13%. This confirms that our approach can properly integrate multispectral information through stereo registration in order to improve segmentation performance beyond that of a state-of-the-art monocular method. Interestingly, our proposed method even outperforms the supervised image segmentation approach of Rother et al. (2004), which relies on manual annotations to pinpoint all foreground objects in every frame. This can be explained by the fact that foreground objects in this dataset have better contrast in the LWIR spectrum than in the visible spectrum, and because our approach propagates this contrast information across the stereo pair. Finally, we show in Figure 7.5 some qualitative results for this dataset. The last row of this figure presents an interesting

case: in this frame pair, the initial foreground masks provided to our method both contain important errors in different regions, but the output is excellent. This shows that despite not having a proper foreground shape template, the real underlying shape can be found and extracted correctly via our iterative process.

7.4.3 Bilodeau *et al.* 2014 Dataset

We now evaluate our proposed method’s stereo registration accuracy using the benchmark dataset of Bilodeau et al. (2014). This dataset was originally intended for the evaluation of image descriptors and similarity measures in the context of multispectral stereo matching, once again provided as a set of videos. It consists of 5390 RGB-LWIR frame pairs split into three scenes, with over 25,000 sparse correspondences annotated on visible foreground objects.

As stated before, we evaluate performance on this dataset by analyzing the accuracy of disparity labelings. Unfortunately, previous works tackling multispectral registration have often relied on their own foreground overlap ratios to assess their performance (e.g. Nguyen et al., 2016), meaning comparisons here are impossible. Here, to provide a reusable evaluation baseline, we compare our results to those obtained using a simple sliding window approach based on the matching of image patches. In short, local disparities are assigned based on the best match (or smallest distance) found between these patches in a winner-takes-all fashion. To describe the similarity between image patches, we rely on descriptors, namely LSS (Shechtman and Irani, 2007) and DASC (Kim et al., 2015), and on Mutual Information (MI) scores (Maes et al., 1997). Finally, to highlight the issue of applying traditional stereo registration methods on multispectral datasets, we evaluate the block matching algorithm of K. Konolige implemented in OpenCV. These results are presented in Table 7.2.

We can note that our proposed method performs very well compared to the baseline methods. Unsurprisingly, OpenCV’s block matching method fails on this dataset as it tries to compare image textures directly across the pair, despite their low correlation. The approaches based on self-similarity descriptors (LSS, DASC) and mutual information perform slightly better, but still produce highly inaccurate results. On average, above 50% of all the evaluated points are labeled with disparities at least four pixels off from the groundtruth. On the other hand, our approach manages to label 51.8% of all evaluated points within a single pixel of the groundtruth, and provides an average disparity error of only 3.21 pixels. Note however that while this performance is good enough for our primary task (mutual foreground segmentation), it is still far from the current state-of-the-art in single-spectrum stereo registration. For example, on the Middlebury dataset (Scharstein et al., 2014), top-performing methods

Table 7.2 Evaluation results on the multispectral video registration dataset of Bilodeau et al. (2014). Bold results are the best in that category across all methods.

Method	Video 1			Video 2			Video 3			Overall		
	% err. >1px	% err. >4px	$\bar{d}_{\text{err}}$	% err. >1px	% err. >4px	$\bar{d}_{\text{err}}$	% err. >1px	% err. >4px	$\bar{d}_{\text{err}}$	% err. >1px	% err. >4px	$\bar{d}_{\text{err}}$
OpenCV’s Block Matcher	95.5	95.3	27.51	99.8	99.7	38.99	99.8	99.6	34.76	98.3	98.2	33.75
LSS Sliding Window	78.3	61.3	9.81	88.8	76.5	8.73	80.4	55.5	9.11	82.5	64.4	9.22
MI Sliding Window	79.2	59.5	9.81	86.8	59.8	8.62	82.6	60.9	10.81	82.9	60.1	9.74
DASC Sliding Window	81.1	58.3	8.96	79.3	57.1	6.55	73.6	43.5	6.94	78.0	52.0	7.48
Proposed method	47.7	17.3	3.28	55.6	25.4	3.11	52.0	17.5	3.26	51.8	20.1	3.21

typically label less than 20% of all points with a disparity error larger than a single pixel. This highlights the difficulty of multispectral stereo registration.

7.4.4 Parameters and Ablation Study

In this section, we study the behavior of our method when key terms and parameters are modified from the default configuration listed in Section 7.3.3 on the two previously introduced datasets. First, we perform an ablation study to determine which energy terms are the most important in our models; this study is presented in Table 7.3.

According to the F_1 scores, modifying the stereo energy formulation only has a small effect on segmentation performance. On the other hand, removing the color or contour terms from the segmentation energy has larger impacts, and the latter of the two is the most important contributor to overall performance. As for the registration performance, the shape term seems to be the most important, but all terms contribute to the overall performance of the method. The positive contribution of both appearance and shape terms thus confirms the hypothesis set in Section 7.3.1. Besides, interestingly, when our model is initialized in only one of the two modalities using approximative masks, its segmentation performance is still at least as good as GrabCut’s (as reported in Table 7.1). This highlights the robustness of our approach, and shows that it can perform well even in adverse initialization conditions.

Next, we show the effect of parameter tuning. The segmentation and registration performance for our proposed method in terms of overall F_1 score and average disparity error ($\bar{d}_{\text{err}}$, in pixels) is presented for various configurations in Figure 7.6. Note that we roughly tuned our method with segmentation performance as a priority to obtain our default configuration. Nonetheless, registration performance is usually near-optimal or stable around the same parameter values. In general, we can note that the choice of parameters does not seem to drastically alter our method’s performance, as both metrics fairly remain stable over large

Table 7.3 Overall performance for various configurations of the proposed method on the datasets of Palmero et al. (2016); Bilodeau et al. (2014).

Method Configuration	$\bar{d}_{\text{err}}$	F_1
No Shape Term (E^{shape})	8.71	0.860
No Appearance Term ($E^{\text{appearance}}$)	3.69	0.851
No Saliency Maps ($\mathcal{W}$)	3.47	0.856
No Uniqueness Term ($E^{\text{uniqueness}}$)	3.33	0.865
No Color Term (E^{color})	3.46	0.822
No Contour Term (E^{contour})	4.16	0.624
No Temporal Term (E^{temporal})	3.29	0.855
No Initial LWIR Segm. Mask	10.82	0.820
No Initial Visible Segm. Mask	8.32	0.800
Default Configuration	3.21	0.866

Table 7.4 Overall segmentation performance for various temporal pipeline depths on the dataset of Palmero et al. (2016).

Method Configuration	Pr	Re	F_1
2 Layers, Real-time	0.817	0.910	0.863
3 Layers, Real-time	0.821	0.915	0.866
4 Layers, Real-time	0.825	0.918	0.867
5 Layers, Real-time	0.826	0.918	0.868
2 Layers, Deferred	0.821	0.914	0.866
3 Layers, Deferred	0.824	0.920	0.870
4 Layers, Deferred	0.827	0.921	0.870
5 Layers, Deferred	0.826	0.919	0.868
No Temporal Term	0.801	0.919	0.855

value intervals.

Finally, in Table 7.4, we evaluate our approach configured with different temporal pipeline depths, and while allowing deferred output or not. The notion of “pipeline depth” here corresponds to the number of edges in the higher order temporal terms introduced in Section 7.3.2.

Deferred segmentation outputs are masks generated by our method with the added latency of the full pipeline, meaning the results are evaluated with a delay equal to the pipeline depth. These masks are thus allowed more iterations in our graphical model, and benefit from more temporal information (i.e. past and future frame data). On the other hand, the real-time segmentation outputs are the masks generated by our method for all new image pairs, provided without delay. From these results, we can note that the difference between deferred and real-time output is surprisingly small. This means that our model’s temporal inertia allows it to smooth out shape variations without having to peek at future frame data, which is useful for real-time surveillance systems. Besides, the overall improvements obtained by using more than two temporal layers is marginal, as more temporally consistent results also entail that some relevant shape fragments around non-rigid objects are discarded. Finally, note that using more layers results in an important increase in computational complexity: using four layers roughly triples the time required for model inference compared to the default configuration.

7.4.5 LITIV 2018 Dataset

To help others compare their work on multispectral segmentation and registration, we developed and annotated a new dataset. We recorded video sequences using a stereo pair composed of a Kinect v2 for Windows (at Full HD resolution) and a FLIR A40 LWIR camera (at QVGA resolution). The sensors were roughly aligned on a fixed baseline support (approximately 50 centimeters apart) and synchronized via software to capture frame pairs at 30 Hz. Calibration data for image rectification was obtained by capturing snapshots of a foam core checkerboard pattern heated using halogen lamps to make it visible in LWIR images. For the annotations, we simultaneously recorded depth and user segmentation masks provided by the Kinect SDK, and transformed this data into foreground-background segmentation masks, adding manual touch-ups where needed. Stereo correspondences were also manually annotated like in the work of Bilodeau et al. (2014) to allow an approximate evaluation of registration performance in foreground image regions. In total, this dataset contains over 6000 frame pairs split into three videos, and its groundtruth is composed of 866 binary segmentation masks and 14716 point correspondences roughly distributed among frames with visible foreground. As for the capture conditions, we deliberately recorded sequences with both strong and weak contrast between foreground and background regions in the two image modalities. More specifically, we used two different temperature calibrations to make individuals more or less perceptible in LWIR images, we introduced some cluttered background in part of the visible images, and we had people carry and exchange objects that modify their appearance in both spectral bands. Overall, this dataset should be more challenging than already available RGB-LWIR video

Table 7.5 Evaluation results for the proposed method on our newly captured multispectral video dataset.

Evaluation Type (Method)	Metric	Video 1		Video 2		Video 3		Overall		
		visible	LWIR	visible	LWIR	visible	LWIR	visible	LWIR	Average
Segmentation (St-Charles et al., 2016a)	Pr	0.933	0.716	0.938	0.763	0.935	0.821	0.935	0.767	0.851
	Re	0.721	0.997	0.834	0.938	0.750	0.996	0.768	0.977	0.872
	F_1	0.813	0.834	0.883	0.841	0.832	0.900	0.843	0.858	0.851
Segmentation (Proposed)	Pr	0.867	0.931	0.874	0.923	0.918	0.941	0.887	0.931	0.909
	Re	0.741	0.822	0.860	0.860	0.917	0.911	0.839	0.865	0.852
	F_1	0.799	0.873	0.867	0.891	0.918	0.926	0.861	0.897	0.879
Registration (DASC Sliding Window)	% err. >1px	90.6	88.6	92.1	90.8	88.5	87.2	90.4	88.8	89.6
	% err. >2px	85.2	81.9	86.6	83.7	81.9	80.2	84.6	81.9	83.3
	% err. >4px	75.5	71.6	78.6	74.2	72.3	70.0	75.5	71.9	73.7
	$\bar{d}_{err}$	30.26	21.90	31.22	29.11	26.48	23.34	29.32	24.79	27.05
Registration (Proposed)	% err. >1px	76.3	80.5	71.3	71.6	66.5	66.9	71.4	73.0	72.2
	% err. >2px	63.9	70.5	55.4	55.6	48.7	48.9	56.0	56.7	57.3
	% err. >4px	46.8	56.2	36.4	37.1	28.9	29.5	37.4	40.9	39.2
	$\bar{d}_{err}$	16.76	19.99	6.68	7.63	5.17	5.47	9.53	11.03	10.28

datasets. The fact that it also allows the simultaneous evaluation of foreground segmentation and stereo registration also makes it quite unique in the current literature.

We have made this new dataset available online along with our modified version of the VAP dataset for other authors⁴. Our Kinect’s raw data which includes depth images and mapping information is also provided for those interested in trimodal segmentation tasks.

We offer our proposed method’s results on this new dataset as a baseline for future comparisons in Table 7.5. We can note that compared to the other two datasets, segmentation results here are still good, but registration errors are much higher. This is primarily due to the fact that our camera baseline is very large (≈ 50 cm), which leads to high disparities for close-range objects (over 150 pixels in some cases), and because our images are higher resolution than those of Bilodeau et al. (2014). Also, we can note that registration errors are higher in the first video sequence: this is caused by the loss of some small foreground segments near image borders which were annotated with correspondences. As for the segmentation results, there are cases where foreground objects are only partly detected, which results in slightly lower Recall scores in some videos. Nonetheless, these results show that our method is capable of segmentating foreground objects in difficult imaging conditions. Finally, we present qualitative segmentation results for this dataset in Figure 7.7.

4. <http://www.polymtl.ca/litiv/vid/index.php>

7.5 Conclusion

We have presented a new method for simultaneous multispectral foreground segmentation and stereo registration, and validated its capabilities on several datasets. Our approach is based on the alternating minimization of two linked energy functions that integrate multispectral shape and appearance cues. We have shown that both segmentation masks and disparity maps can simultaneously converge to good local minima without any human supervision. Furthermore, with the help of higher order factors, we achieve strong temporal coherence in our segmentation results by linking consecutive video frames inside our graphical models. To make the comparison of methods tackling this problem easier in the future, we provide our full implementation online, as well as a newly created multispectral dataset for evaluation.

If supporting large stereo baselines is unnecessary, the method could use a stronger constraint on multispectral contour similarity to improve coherence between views. Besides, explicit occlusion handling in our stereo model would further improve overall performance on the current datasets. Our model could also be generalized to provide instance-level segmentation by using a separate foreground appearance model for each object. Finally, a three-way energy minimization solution tackling foreground segmentation, stereo registration, and optical flow could be designed based on our current inference approach.



Figure 7.5 Examples of typical segmentation results from the VAP dataset of Palmero et al. (2016); the left two columns show the segmentation masks obtained via the method of St-Charles et al. (2016a) and used to initialize our method, and the right two columns show our final segmentation masks. Image regions properly classified as foreground are highlighted in green over the original images, while regions highlighted in orange and magenta show false positives and false negatives, respectively. Images have been cropped to show more details.

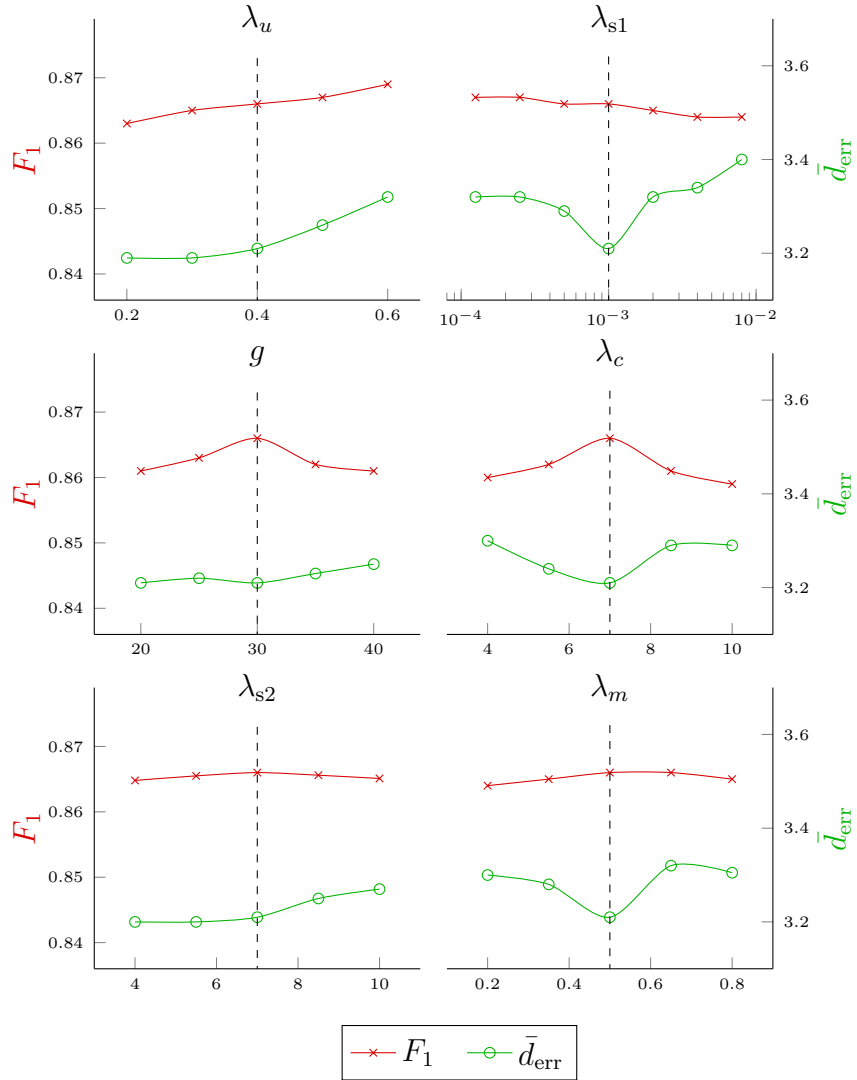


Figure 7.6 Overall performance for various parameter values of the proposed method on the datasets of Palmero et al. (2016); Bilodeau et al. (2014). The default configuration of each parameter is shown with the dashed line. Remember that for F_1 , higher is better, and for $\bar{d}_{err}$, lower is better.

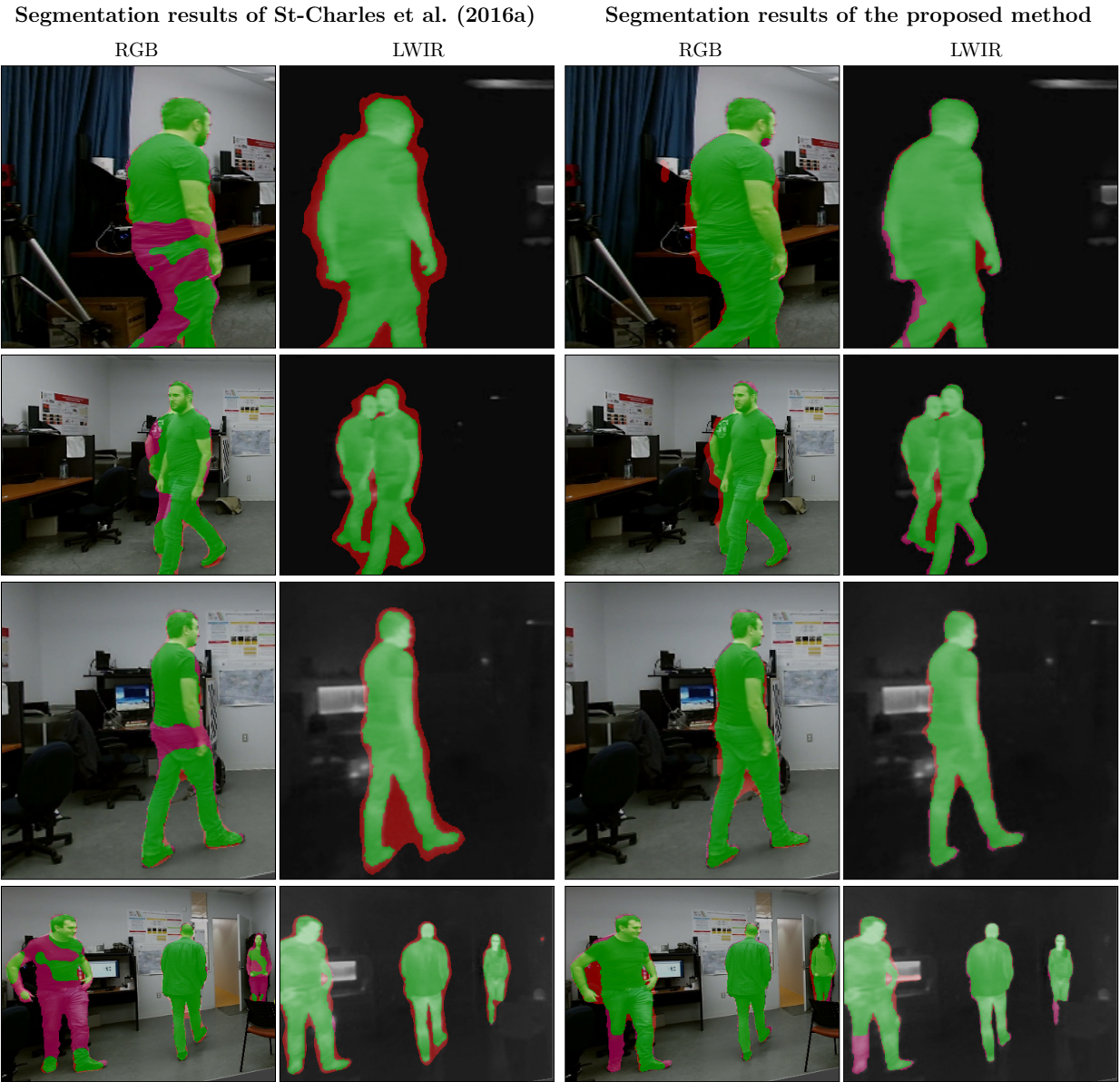


Figure 7.7 Examples of typical segmentation results from our newly captured dataset; the left two columns show the segmentation masks obtained via St-Charles et al. (2016a) and used to initialize our method, and the right two columns show our final segmentation masks. Image regions properly classified as foreground are highlighted in green over the original images, while regions highlighted in orange and magenta show false positives and false negatives, respectively. Images have been cropped to show more details.

CHAPITRE 8 DISCUSSION GÉNÉRALE

Ce chapitre présente une synthèse de chaque volet du travail accompli. Plus spécifiquement, nous abordons les limitations identifiées *a posteriori* pour chacune des méthodes développées ainsi que les améliorations qui pourraient être apportées à ces méthodes. Nous discutons aussi de l’efficacité globale du pipeline de traitement bas niveau développé.

8.1 Segmentation vidéo par soustraction d’arrière-plan

Les deux méthodes de soustraction d’arrière-plan proposées dans les chapitres 4 et 5 offrent des solutions différentes aux défis typiquement rencontrés en segmentation vidéo monoculaire. Bien que leurs performances globales évaluées sur l’ensemble de données de Wang et al. (2014a) soient similaires, la méthode du chapitre 4 (SuBSENSE) est mieux adaptée aux conditions des scénarios plus extrêmes (e.g. surveillance de nuit, surveillance sous la pluie/neige) que la méthode du chapitre 5 (PAWCS). Par contre, PAWCS offre des résultats largement supérieurs à ceux de SuBSENSE en présence d’objets immobilisés à long terme. Puisque de tels objets sont relativement communs en vidéosurveillance, PAWCS est la méthode la plus adéquate pour effectuer un pré-traitement dans un système de surveillance réel. Au final, cette méthode répond bien aux objectifs de segmentation monoculaire énoncés dans la section 1.3 ; elle est capable de détecter et de segmenter n’importe quel type d’objet faisant irruption dans une scène tout en s’adaptant aux variations naturelles de son arrière-plan. La conception de cette méthode fait d’ailleurs complètement abstraction de la modalité d’imagerie traitée, ce qui est idéal pour effectuer l’initialisation des méthodes des chapitres 6 et 7.

Notons ici que les masques de segmentation produits par les méthodes de soustraction d’arrière-plan des chapitres 4 et 5, dans des scénarios de type “*baseline*”, sont de qualité comparable au *groundtruth* utilisé pour leur évaluation (i.e. dans l’ordre de l’incertitude des annotations). Cela démontre que la segmentation vidéo n’est plus problématique lorsque les images ne possèdent pas d’arrière-plan dynamique, lorsque le contraste entre l’arrière-plan et l’avant-plan n’est pas faible, et lorsque les objets d’intérêt ne projettent pas d’ombres trop prononcées au sol (*hard shadows*). Dans les cas où une partie de l’arrière-plan varie légèrement et où les ombres sont adoucies par l’éclairage ambiant, la segmentation vidéo est relativement près d’être considérée comme non-problématique.

En ce qui concerne les limitations des méthodes proposées, nous pouvons identifier d’abord que celles-ci sont assez sensibles à la résolution des images traitées. Cela s’explique par le

descripteur binaire qu’elles utilisent pour la caractérisation de représentations d’images, qui est de taille fixe (5x5). La conséquence liée à cette utilisation est que des objets d’intérêt de taille inférieure à ce descripteur peuvent être considérés comme du bruit dans l’arrière-plan, tandis que les textures d’échelle plus grande ne peuvent pas être capturées dans les modèles de pixels. Une solution à ce problème consiste à utiliser une approche de description multi-échelle, telle que celle proposée par St-Charles et al. (2016b). Par ailleurs, ces méthodes de segmentation par pixel n’utilisent aucunement le concept d’*objectness* souvent exploité en segmentation vidéo afin de mieux identifier l’ensemble des pixels appartenant à un objet d’intérêt. L’utilisation de ce concept en post-traitement à travers un modèle graphique simple pourrait aider à mieux refermer les formes d’objets fragmentés.

8.2 Recalage de paires d’images multispectrales

Les stratégies de recalage de paires d’images proposées dans les chapitres 6 et 7 sont conçues pour être appliquées dans des scénarios très différents, c’est à dire sur des scènes planaires ou non-planaires. Ces stratégies partagent toutefois l’idée générale que les points provenant de contours d’objets détectés par segmentation peuvent fournir une quantité suffisante d’information discriminative permettant de recalibrer ces objets. Bien que la qualité des résultats obtenue ici est loin de celle typiquement atteignable en recalage monospectral, il est démontré au chapitre 7 que cette stratégie permet une intégration appropriée des données multispectrales, ce qui résulte en une amélioration de la qualité des masques de segmentation. Cela répond donc aussi aux exigences fixées dans la section 1.3 pour nos objectifs de recalage multispectral.

L’approche de recalage du chapitre 6 pour des scènes planaires pourrait être améliorée en considérant des segments de contours (i.e. groupes de points) plutôt que des points individuels. Cela nous permettrait de distinguer les sections de contours fiables (i.e. qui possèdent une courbure semblable dans les deux images) des sections peu fiables pour le recalage de formes. Les sections de formes peu fiables, qui sont causées par des imperfections dans les masques de segmentation, pourraient alors être recalées à partir de la mise en correspondance d’un autre type de représentation de haut niveau (e.g. les arêtes de l’image).

Pour l’approche de recalage de scènes non-planaires du chapitre 7, il serait possible d’améliorer la rectification (et ainsi le recalage) des images en perfectionnant les paramètres de calibration utilisés au temps d’exécution. En effet, les paramètres de calibration initiaux obtenus préalablement pour nos ensembles de données ne sont pas idéaux, car l’erreur de reprojection des points sur leur plan de calibration se situe au-delà d’un pixel. Ce problème induit des erreurs au niveau de la mise en correspondance des pixels sur les épipoles, ce qui

diminue la qualité des disparités estimées localement. Le peaufinage des paramètres de calibration au temps d'exécution permettrait donc d'obtenir de meilleurs résultats. Cela pourrait être réalisé en combinant les points de calibration initiaux aux points de contours tirés des résultats de segmentation afin d'estimer de nouveaux paramètres. Une approche semblable a d'ailleurs été proposée dans le chapitre 6 pour l'estimation par RANSAC des paramètres d'une transformation projective en continu.

Enfin, l'approche du chapitre 7 pourrait aussi être améliorée par la modélisation explicite des pixels en occultation dans chaque image traitée. En bref, dans les cas où la parallaxe n'est pas négligeable, la disparité (ou profondeur) réelle des pixels cachés par un objet d'avant-plan dans une des images ne peut pas être déterminée avec précision. Toutefois, cette disparité devrait être semblable à celle de ses voisins. Dans la méthode actuelle, ces pixels sont souvent assignés une valeur adéquate par lissage, mais il est possible qu'ils créent un îlot de mauvaise valeurs dans la carte de disparités générée. Une modélisation explicite des occultations permettrait de forcer leur lissage à partir d'autres pixels dans leur voisinage.

8.3 Segmentation mutuelle d'images multispectrales

L'approche de segmentation mutuelle d'objets d'intérêt proposée dans le chapitre 7 peut être réalisée grâce au recalage dense des images traitées et grâce à la détection *a priori* des objets d'intérêt s'y retrouvant. Cette approche n'est pas basée sur l'apprentissage des caractéristiques de chaque modalité d'imagerie ou bien de la forme des types d'objets ciblés, mais plutôt sur la combinaison et sur l'évolution de modèles d'avant-plan multispectraux. L'évolution de ces modèles par notre processus d'optimisation itératif permet d'identifier et de segmenter adéquatement les objets d'intérêt de la scène malgré une initialisation potentiellement imparfaite, ce qui répond bien aux deux derniers objectifs spécifiques de la section 1.3.

En rétrospective, nous pouvons noter qu'une des faiblesses de l'approche de modélisation proposée au chapitre 7 est la complexité des graphes construits. Autant au niveau du modèle de recalage que du modèle de segmentation, la quantité de nœuds utilisée pour représenter les régions de chaque image semble parfois nuire au lissage des cartes de résultats produites. Dans la méthode proposée, chaque pixel des images traitées constitue un nœud, et ces nœuds sont connectés à leurs voisins immédiats à l'aide de facteurs d'ordre un (voir l'illustration de la figure 8.1.a). Ces facteurs sont responsables du lissage final des cartes de résultats produites, mais en pratique, ils ne semblent pas être en mesure de toujours bien jumeler les changements d'étiquettes de disparités ou de segmentation aux contours des objets d'intérêt, peu importe les hyperparamètres utilisés. Une des raisons pouvant expliquer ce phénomène est qu'il existe trop d'incertitude dans les coûts associés aux facteurs unaires du graphe. Cela s'explique par

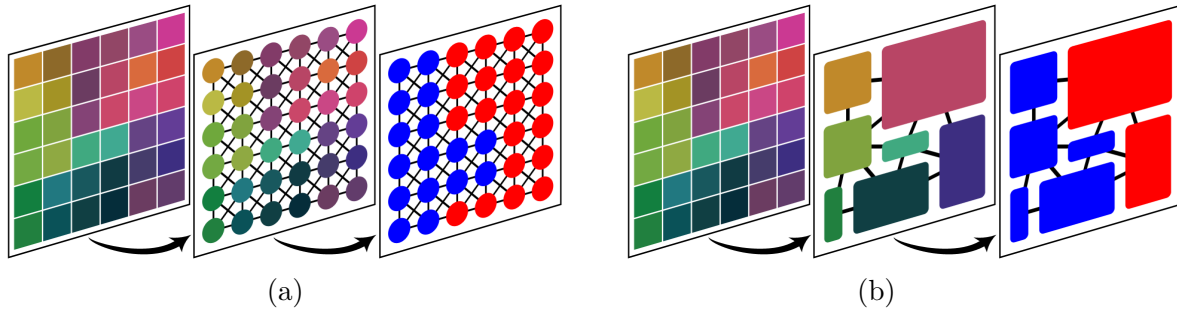


Figure 8.1 Exemple de modélisation graphique où chaque pixel d’une image constitue un nœud (a), et où les superpixels de l’image sont calculés et utilisés comme nœuds (b). La deuxième approche permet de réduire le nombre total de nœuds et de liens dans le graphe, ce qui accélère le temps de traitement, en plus d’offrir un meilleur lissage des données.

le fait que la modélisation par pixel n’est pas appropriée pour le recalage et la segmentation d’images sur la base de représentations de haut niveau (i.e. les contours d’objets). Par conséquent, il faudrait utiliser une approche de modélisation où les nœuds du graphe sont établis à un niveau légèrement plus élevé, par exemple en se basant sur des superpixels (voir l’exemple de la figure 8.1.b). Des superpixels de différentes tailles respectant les contours des objets de la scène peuvent d’ailleurs être calculés facilement en suivant par exemple l’approche de Achanta et al. (2012). La modélisation par superpixels permettrait alors de mieux lisser les étiquettes de régions voisines dans les cartes de disparités et de segmentation produites tout en gardant une partie de la précision obtenue grâce à une approche de modélisation de bas niveau.

D’autre part, l’utilisation d’un mécanisme d’ajustement pour déterminer la valeur adéquate du facteur contrôlant la contribution multimodale (λ_m) dans la section 7.3.2 aurait aussi été bénéfique. Ce mécanisme aurait pu être basé sur l’estimation de saillance locale effectuée pour chaque image dans la section 7.3.1. L’aboutissement aurait alors permis une évolution plus rapide des contours d’objets détectés dans la paire d’images, et une diminution des chances de trouver un minimum local imparfait lors de l’évolution de ces contours.

8.4 Efficacité du pipeline de traitement pour usage temps-réel

Finalement, nous discutons ici de l’applicabilité de notre pipeline de traitement dans un contexte de vidéosurveillance en temps réel. Rappelons-nous d’abord que le pipeline de traitement complet est essentiellement réalisé et testé dans l’article du chapitre 7. La solution proposée dans ce chapitre utilise la méthode de segmentation vidéo du chapitre 5 pour le pré-traitement des séquences d’images brutes, et propose sa propre stratégie de recalage et

de segmentation mutuelle de paires d’images. L’implémentation des différentes composantes de cette solution a été effectuée dans un même cadriciel, ce qui nous permet de considérer son efficacité de façon globale. Notons toutefois que l’optimisation de cette efficacité n’a pas été notre première priorité lors de cette phase de développement. Notre propre code ne profite donc pas de toute la parallélisation et de toute la vectorisation d’instructions auquel il aurait eu droit.

Tel que mentionné dans la section 7.3.3, la génération des cartes de disparités et de segmentation finales, pour une paire d’images de taille 640x480, demande environ 30 secondes de calcul sur CPU, ce qui est loin de satisfaire les exigences du “temps réel” (≈ 30 Hz). Par contre, plus de 90% de ce temps est consacré à l’inférence des cartes de résultats à partir des modèles graphiques construits. Les méthodes d’inférence utilisées proviennent des travaux de Fix et al. (2011) et Fix et al. (2014), et celles-ci sont implémentées sur CPU sans aucune parallélisation. Il est donc possible de considérer qu’une implémentation parallèle sur GPU similaire à celle de Vineet and Narayanan (2008) pourrait grandement améliorer l’efficacité de notre pipeline. La simplification des modèles graphiques suggérée dans la section 8.3 (figure 8.1) pourrait aussi diminuer le temps d’inférence par un facteur important (i.e. 10 ou plus).

En ce qui concerne la méthode de pré-traitement du chapitre 5, celle-ci est déjà en mesure de traiter des séquences d’images de manière assez rapide, c’est à dire à plus de 10 trames par seconde sur CPU pour une résolution de 640x480. Par ailleurs, l’implémentation GPU d’une version simplifiée de la méthode du chapitre 4 (qui est similaire à la méthode utilisée ici) a été réalisée dans le cadre d’autres travaux. Cette implémentation parallèle peut segmenter des images à un rythme plus rapide que le taux de versement vers le GPU, c’est à dire à plusieurs milliers de trames par seconde. Cela est dû au fait que la modélisation et la détection d’objets d’intérêt selon une approche par pixel sans régularisation globale est un problème de parallélisation d’une facilité dite “embarrassante” (*embarrassingly parallel*). Nous pouvons donc conclure que la parallélisation de notre méthode finale de soustraction d’arrière-plan pourrait aussi profiter d’un gain d’efficacité semblable.

En résumé, l’implémentation présentement disponible en ligne de notre pipeline complet ne permet pas de produire des résultats en temps réel. Par contre, cet objectif n’est pas hors de portée si nous tenons compte de la capacité de calcul des GPUs présentement sur le marché et du potentiel de parallélisation des différentes méthodes de notre solution.

CHAPITRE 9 CONCLUSION

Dans cette thèse, nous avons présenté de nouvelles méthodes permettant de recalcr et de segmenter des objets d'intérêt dans des paires d'images multispectrales provenant de séquences vidéo synchronisées. Ces méthodes sont conçues de manière à pouvoir atteindre leur objectif sans aucune intervention humaine et sans connaissance *a priori* des objets ciblés. Cela simplifie leur intégration dans une application de vidéosurveillance automatisée.

Deux méthodes de segmentation vidéo basées sur la soustraction d'arrière-plan ont d'abord été proposées. Chaque méthode utilise une approche de modélisation non-paramétrique par pixel combinant l'utilisation de descripteurs locaux binaires et d'intensités locales dans la description des représentations de l'arrière-plan. De plus, ces méthodes utilisent différents mécanismes de rétroaction permettant d'ajuster les hyperparamètres de leurs modèles afin de mieux contrôler leur adaptabilité et leur sensibilité en fonction du dynamisme observé dans la scène. Dans sa stratégie de modélisation, la deuxième méthode propose aussi un indice de persistance, qui aide à identifier les représentations récurrentes de l'arrière-plan ; cela lui permet de beaucoup mieux détecter les objets d'intérêt immobilisés dans une scène. Les deux méthodes de segmentation proposées ont été évaluées sur un vaste ensemble de données, et il a été démontré que chacune d'entre elles surpasse des dizaines de solutions modernes, selon de nombreux critères de performance, et dans différentes catégories.

Nous avons ensuite proposé une méthode de recalage de paires d'images tirées de séquences vidéo pour des scènes planaires. Cette méthode est basée sur la mise en correspondance de points de contours obtenus à partir de cartes de segmentation. Ces points sont décrits à l'aide d'un descripteur de forme et appariés à l'aide d'une approche exhaustive. Les correspondances alors obtenues sont ajoutées et maintenues dans un réservoir temporel de manière stochastique. Ce réservoir est enfin utilisé pour estimer une transformation projective permettant de recalcr tout le contenu des images analysées. Cette méthode a permis de confirmer qu'il est possible de recalcr des images multispectrales à partir de formes d'objets bruitées obtenues par segmentation vidéo.

Finalement, une solution double a été proposée afin de recalcr et de segmenter simultanément des paires d'images multispectrales provenant de scènes non-planaires. Cette solution est basée sur la minimisation itérative de deux fonctions d'énergie axées sur l'estimation de cartes de disparités et sur la segmentation des d'objets d'intérêt. Cette approche d'optimisation permet de résoudre le paradoxe énoncé dans la section 1.2.4, c'est à dire qu'une segmentation de qualité est nécessaire pour obtenir un recalage de qualité, et vice-versa. Les fonctions

d'énergie sont exprimées à l'aide de modèles graphiques initialisés grâce aux résultats fournis par une des méthodes de segmentation vidéo monoculaire déjà proposées. Cette solution combinée représente d'ailleurs un pipeline de traitement capable de traiter des séquences multispectrales de façon complètement automatisée.

Le développement de ces différentes méthodes a mené à de nombreuses contributions au domaine scientifique, sous forme d'articles de journaux et d'actes de conférences. L'accent mis sur la reproductibilité dans le cadre de cette thèse a aussi motivé la publication de tout le code source et de toutes les bases de données utilisées lors de nos expériences.

9.1 Recommandations pour travaux futurs

Dans une perspective plus générale, même s'il semble aujourd'hui que la soustraction d'arrière-plan est surpassée par les approches basées sur l'apprentissage machine dans le cadre de la segmentation vidéo, il n'existe toujours pas d'alternative fiable pour résoudre les problèmes reliés à l'évolution de l'importance des objets dans un contexte de surveillance (e.g. bagages déposés versus abandonnés, voitures stationnées versus à l'arrêt). En effet, la plupart des approches déjà publiées qui utilisent l'apprentissage profond pour détecter des objets en mouvement ne considèrent aucunement leur historique de mouvement. En pratique, l'historique de mouvement est un indice aussi important que l'apparence pour effectuer la détection d'objets d'intérêt dans un contexte d'application comme le nôtre. De futurs travaux devraient donc tenter de combiner certains principes d'analyse de processus dynamiques provenant de la théorie du contrôle (*control theory*) aux techniques d'apprentissage machine afin de développer une approche capable de déterminer la durée de vie utile d'un objet d'intérêt dans une scène complexe.

Pour ce qui est du recalage stéréo multispectral, il serait intéressant de développer une méthode capable d'apprendre automatiquement quel indice de similarité donner à deux pixels provenant d'images différentes en fonction de leur voisinage local. Une telle méthode servirait alors à définir les valeurs utilisées dans les termes unaires (*data terms*) de modèles de recalage stéréo par estimation de disparités. La phase d'apprentissage pourrait se baser sur la comparaison de vecteurs de caractéristiques, incluant des descripteurs de similarités locales (e.g. DASC, Kim et al., 2015) et des cartes de distances de contours (*distance transform maps*). Les résultats de prédiction pourraient être obtenus à l'aide d'une forêt d'arbres décisionnels structurés (*structured random forest*) tel que proposé par Dollar and Zitnick (2015). Il serait aussi envisageable de considérer l'apprentissage bout à bout (*end-to-end learning*) de ces indices de similarité à l'aide d'un réseau de neurones. Un ensemble de données permettant ce type d'apprentissage a d'ailleurs récemment été proposé par Treible et al. (2017).

Enfin, notons que les résultats de segmentation que nous obtenons grâce à nos méthodes ne contiennent pas d'identifiant unique pour chaque objet (i.e. il s'agit simplement de cartes binaires indiquant la présence d'un objet d'intérêt ou non). En d'autres termes, si deux objets d'intérêt se rapprochent jusqu'au point d'entrer en contact, ceux-ci seront détectés comme étant un seul objet par nos méthodes, et ce jusqu'à ce qu'ils se séparent à nouveau. De plus, il nous est impossible d'établir avec certitude les correspondances d'identité entre les objets détectés dans deux images consécutives. Le développement d'une méthode d'identification et d'association de fragments de formes pourrait donc être considérée dans de futurs travaux afin d'améliorer l'applicabilité de notre pipeline de traitement dans un système réel. Cette méthode pourrait être basée sur une approche de segmentation sémantique par instance, e.g. Mask-RCNN (He et al., 2017).

RÉFÉRENCES

- R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, et S. Süssstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012. DOI : 10.1109/TPAMI.2012.120
- C. Aguilera, F. Barrera, F. Lumbreras, A. D. Sappa, et R. Toledo, “Multispectral image feature points”, *Sensors*, vol. 12, no. 9, pp. 12 661–12 672, 2012. DOI : 10.3390/s120912661
- G. Allebosch, F. Deboeverie, P. Veelaert, et W. Philips, “EFIC : Edge based foreground background segmentation and interior classification for dynamic camera viewpoints”, dans *Proc. Advanced Concepts for Intell. Vis. Syst.*, 2015, pp. 130–141.
- B. Andres, T. Beier, et J. Kappes, “OpenGM : A C++ library for discrete graphical models”, *CoRR*, vol. abs/1206.0111, 2012. En ligne : <http://arxiv.org/abs/1206.0111>
- P. Arbelaez, M. Maire, C. Fowlkes, et J. Malik, “Contour detection and hierarchical image segmentation”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011. DOI : 10.1109/TPAMI.2010.161
- O. Barnich et M. Van Droogenbroeck, “ViBe : A universal background subtraction algorithm for video sequences”, *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, 2011. DOI : 10.1109/TIP.2010.2101613
- F. Barrera, F. Lumbreras, et A. D. Sappa, “Multispectral piecewise planar stereo using manhattan-world assumption”, *Pattern Recognit. Lett.*, vol. 34, no. 1, pp. 52–61, 2013. DOI : 10.1016/j.patrec.2012.08.009
- D. Batra, A. Kowdle, D. Parikh, J. Luo, et T. Chen, “iCoseg : Interactive co-segmentation with intelligent scribble guidance”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2010, pp. 3169–3176. DOI : 10.1109/CVPR.2010.5540080
- S. Belongie, J. Malik, et J. Puzicha, “Shape matching and object recognition using shape contexts”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr 2002. DOI : 10.1109/34.993558
- Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, et C. Rosenberger, “Comparative study of background subtraction algorithms”, *J. Electron. Imaging*, vol. 19, 07 2010. DOI : 10.1117/1.3456695

- S. Bianco, G. Ciocca, et R. Schettini, “How far can you get by combining change detection algorithms?” *CoRR*, vol. abs/1505.02921, 2015. En ligne : <http://arxiv.org/abs/1505.02921>
- L. Bienkowski, C. Homma, K. Eisler, et C. Boller, “Hybrid camera and real-view thermography for nondestructive evaluation”, *Quantitative Infrared Thermography*, vol. 254, 2012.
- G.-A. Bilodeau, P. St-Onge, et R. Garnier, “Silhouette-based features for visible-infrared registration”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, June 2011, pp. 68–73. DOI : 10.1109/CVPRW.2011.5981676
- G.-A. Bilodeau, A. Torabi, et F. Morin, “Visible and infrared image registration using trajectories and composite foreground images”, *Image and Vis. Comp.*, vol. 29, no. 1, pp. 41–50, 2011. DOI : 10.1016/j.imavis.2010.08.002
- G.-A. Bilodeau, J.-P. Jodoin, et N. Saunier, “Change detection in feature space using local binary similarity patterns”, dans *Proc. Int. Conf. Comput. Robot Vis.*, 2013, pp. 106–112. DOI : 10.1109/CRV.2013.29
- G.-A. Bilodeau, A. Torabi, P.-L. St-Charles, et D. Riahi, “Thermal-visible registration of human silhouettes : A similarity measure performance evaluation”, *Infrared Physics & Technology*, vol. 64, no. 0, pp. 79–86, 2014. DOI : 10.1016/j.infrared.2014.02.005
- M. Bleyer, C. Rother, P. Kohli, D. Scharstein, et S. Sinha, “Object stereo - joint stereo matching and object segmentation”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2011, pp. 3081–3088. DOI : 10.1109/CVPR.2011.5995581
- T. Bouwmans, “Traditional and recent approaches in background modeling for foreground detection : An overview”, *Computer Science Review*, vol. 11, pp. 31–66, 2014. DOI : 10.1016/j.cosrev.2014.04.001
- T. Bouwmans et E. H. Zahzah, “Robust PCA via principal component pursuit : A review for a comparative evaluation in video surveillance”, *Comput. Vis. and Image Understanding*, vol. 122, pp. 22–34, 2014. DOI : 10.1016/j.cviu.2013.11.009
- T. Bouwmans, N. S. Aybat, et E.-h. Zahzah, *Handbook of robust low-rank and sparse matrix decomposition : Applications in image and video processing*. CRC Press, 2016.
- Y. Boykov, O. Veksler, et R. Zabih, “Fast approximate energy minimization via graph cuts”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, 2001. DOI : 10.1109/34.969114

M. Braham et M. Van Droogenbroeck, “A generic feature selection method for background subtraction using global foreground models”, dans *Proc. Advanced Concepts for Intell. Vis. Syst.*, 2015. En ligne : <http://hdl.handle.net/2268/185047>

——, “Deep background subtraction with scene-specific convolutional neural networks”, dans *Proc. Int. Conf. Systems, Signals and Image Proc.*, 2016. En ligne : <http://hdl.handle.net/2268/195180>

L. G. Brown, “A survey of image registration techniques”, *ACM Comput. Surv.*, vol. 24, no. 4, pp. 325–376, 1992. DOI : 10.1145/146370.146374

S. Brutzer, B. Hoferlin, et G. Heidemann, “Evaluation of background subtraction techniques for video surveillance”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2011, pp. 1937–1944. DOI : 10.1109/CVPR.2011.5995508

S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, et L. Van Gool, “One-shot video object segmentation”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.

E. J. Candès, X. Li, Y. Ma, et J. Wright, “Robust principal component analysis?” *J. ACM*, vol. 58, no. 3, pp. 11–37, 2011. DOI : 10.1145/1970392.1970395

J. Carreira et C. Sminchisescu, “Constrained parametric min-cuts for automatic object segmentation”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2010, pp. 3241–3248. DOI : 10.1109/CVPR.2010.5540063

Y. Caspi, D. Simakov, et M. Irani, “Feature-based sequence-to-sequence matching”, dans *Proc. 8th European Conf. Comput. Vis. Workshops*, 2002.

Y. Chen, J. Wang, et H. Lu, “Learning sharable models for robust background subtraction”, dans *Proc. IEEE Int. Conf. MultiMultimedia and Expo*, 2015, pp. 1–6. DOI : 10.1109/ICME.2015.7177419

Y.-T. Chen, C.-S. Chen, C.-R. Huang, et Y.-P. Hung, “Efficient hierarchical method for background subtraction”, *Pattern Recognit.*, vol. 40, no. 10, pp. 2706–2715, 2007. DOI : 10.1016/j.patcog.2006.11.023

Z. Chen et T. Ellis, “A self-adaptive gaussian mixture model”, *Comput. Vis. and Image Understanding*, vol. 122, pp. 35–46, 2014. DOI : 10.1016/j.cviu.2014.01.004

J. Cheng, Y.-H. Tsai, S. Wang, et M.-H. Yang, “SegFlow : Joint learning for video object segmentation and optical flow”, dans *Proc. IEEE Int. Conf. Comput. Vis.*, 2017.

W.-C. Chiu et M. Fritz, “Multi-class video co-segmentation with a generative multi-video model”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2013, pp. 321–328. DOI : 10.1109/CVPR.2013.48

E. Coiras, J. Santamaria, et C. Miravet, “Segment-based registration technique for visual-infrared images”, *Optical Engineering*, vol. 39, pp. 282–289, 2000.

R. Cucchiara, C. Grana, M. Piccardi, et A. Prati, “Detecting moving objects, ghosts, and shadows in video streams”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1337–1342, Oct 2003. DOI : 10.1109/TPAMI.2003.1233909

J. W. Davis et V. Sharma, “Background-subtraction using contour-based fusion of thermal and visible imagery”, *Comput. Vis. and Image Understanding*, vol. 106, no. 2-3, pp. 162–182, 2007. DOI : 10.1016/j.cviu.2006.06.010

B. Dey et M. Kundu, “Efficient foreground extraction from HEVC compressed video for application to real-time analysis of surveillance ‘big’ data”, *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3574–3585, 2015. DOI : 10.1109/TIP.2015.2445631

A. Djelouah, J.-S. Franco, E. Boyer, F. Le Clerc, et P. Perez, “Sparse multi-view consistency for object segmentation”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, no. 99, pp. 1–1, 2015. DOI : 10.1109/TPAMI.2014.2385704

P. Dollar et C. Zitnick, “Fast edge detection using structured forests”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1558–1570, 2015. DOI : 10.1109/TPAMI.2014.2377715

J. Duchon, “Splines minimizing rotation-invariant semi-norms in sobolev spaces”, dans *Constructive Theory of Functions of Several Variables*. Springer, 1977, vol. 571, pp. 85–100. DOI : 10.1007/BFb0086566

G. Egnal et K. Daniilidis, “Image registration using mutual information”, University of Pennsylvania, Rapp. tech., 2000.

A. M. Elgammal, D. Harwood, et L. S. Davis, “Non-parametric model for background subtraction”, dans *Proc. 6th European Conf. Comput. Vis.*, 2000, pp. 751–767.

- R. Evangelio et T. Sikora, “Complementary background models for the detection of static and moving objects in crowded environments”, dans *Proc. IEEE Int. Conf. Advanced Video and Signal Based Surveillance*, Aug 2011, pp. 71–76. DOI : 10.1109/AVSS.2011.6027297
- R. Evangelio, M. Patzold, I. Keller, et T. Sikora, “Adaptively splitted GMM with feedback improvement for the task of background subtraction”, *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 5, pp. 863–874, 2014. DOI : 10.1109/TIFS.2014.2313919
- A. Faktor et M. Irani, “Co-segmentation by composition”, dans *Proc. IEEE Int. Conf. Comput. Vis.*, Dec 2013, pp. 1297–1304. DOI : 10.1109/ICCV.2013.164
- J. Feng, H. Xu, et S. Yan, “Online robust PCA via stochastic optimization”, dans *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 404–412.
- M. A. Fischler et R. C. Bolles, “Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography”, *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981. DOI : 10.1145/358669.358692
- A. Fix, A. Gruber, E. Boros, et R. Zabih, “A graph cut algorithm for higher-order markov random fields”, dans *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1020–1027.
- A. Fix, C. Wang, et R. Zabih, “A primal-dual algorithm for higher-order multilabel markov random fields”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1138–1145.
- N. Friedman et S. Russell, “Image segmentation in video sequences : A probabilistic approach”, dans *Proc. 13th Conf. Uncertainty in Artificial Intell.* Morgan Kaufmann Publishers Inc., 1997, pp. 175–181.
- D. Gallup, J.-M. Frahm, et M. Pollefeys, “Piecewise planar and non-planar stereo for urban scene reconstruction”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2010, pp. 1418–1425. DOI : 10.1109/CVPR.2010.5539804
- Z. Gao, L.-F. Cheong, et Y.-X. Wang, “Block-sparse RPCA for salient motion detection”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 1975–1987, 2014. DOI : 10.1109/TPAMI.2014.2314663
- D. O. Gorodnichy, D. Bissessar, E. Granger, et R. Laganière, “Recognizing people and their activities in surveillance video : technology state of readiness and roadmap”, dans *Proc. Int. Conf. Comput. Robot Vis.*, 2016, pp. 250–259.

A. A. Goshtasby, *2-D and 3-D Image Registration : For Medical, Remote Sensing, and Industrial Applications*. Wiley-Interscience, 2005.

N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, et P. Ishwar, “Changetection.net : A new change detection benchmark dataset”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, June 2012, pp. 1–8. DOI : 10.1109/CVPRW.2012.6238919

—, “A novel video dataset for change detection benchmarking”, *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4663–4679, 2014. DOI : 10.1109/TIP.2014.2346013

M. D. Gregorio et M. Giordano, “A WiSARD-based approach to CDnet”, dans *Proc. 1st BRICS Countries Congress*, 2013.

—, “Change detection with weightless neural networks”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, June 2014.

J. Guo, Z. Li, L.-F. Cheong, et S. Zhou, “Video co-segmentation for meaningful action extraction”, dans *Proc. IEEE Int. Conf. Comput. Vis.*, Dec 2013, pp. 2232–2239. DOI : 10.1109/ICCV.2013.278

T. Haines et T. Xiang, “Background subtraction with dirichlet process mixture models”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 670–683, April 2014. DOI : 10.1109/TPAMI.2013.239

J. Han et B. Bhanu, “Fusion of color and infrared video for moving human detection”, *Pattern Recognit.*, vol. 40, no. 6, pp. 1771–1784, 2007. DOI : 10.1016/j.patcog.2006.11.010

R. Hartley et A. Zisserman, *Multiple View Geometry in Computer Vision*, 2e éd. New York, NY, USA : Cambridge University Press, 2003.

J. He, L. Balzano, et A. Szlam, “Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1568–1575. DOI : 10.1109/CVPR.2012.6247848

K. He, G. Gkioxari, P. Dollár, et R. Girshick, “Mask R-CNN”, dans *Proc. IEEE Int. Conf. Comput. Vis.*, 2017.

M. Heikkilä et M. Pietikäinen, “A texture-based method for modeling the background and detecting moving objects”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 657–662, 2006. DOI : 10.1109/TPAMI.2006.68

- F. Hernandez-Lopez et M. Rivera, “Change detection by probabilistic segmentation from monocular view”, *Mach. Vis. Appl.*, vol. 25, no. 5, pp. 1175–1195, 2014. DOI : 10.1007/s00138-013-0564-3
- S. Herrero et J. Bescós, “Background subtraction techniques : Systematic evaluation and comparative analysis”, dans *Proc. Advanced Concepts for Intell. Vis. Syst.*, 2009, pp. 33–42. DOI : 10.1007/978-3-642-04697-1-4
- M. Hofmann, P. Tiefenbacher, et G. Rigoll, “Background segmentation with feedback : The pixel-based adaptive segmenter”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 38–43. DOI : 10.1109/CVPRW.2012.6238925
- P. O. Hoyer, “Non-negative matrix factorization with sparseness constraints”, *J. Mach. Learn. Research*, vol. 5, no. Nov, pp. 1457–1469, 2004.
- S. Hwang, J. Park, N. Kim, Y. Choi, et I. So Kweon, “Multispectral pedestrian detection : Benchmark dataset and baseline”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1037–1045.
- S. D. Jain, B. Xiong, et K. Grauman, “Fusionseg : Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.
- S. Jeong, J. Lee, B. Kim, Y. Kim, et J. Noh, “Object segmentation ensuring consistency across multi-viewpoint images”, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017.
- J.-P. Jodoin, G.-A. Bilodeau, et N. Saunier, “Background subtraction based on local shape”, *CoRR*, vol. abs/1204.6326, 2012.
- P.-M. Jodoin, S. Piérard, Y. Wang, et M. Van Droogenbroeck, “Overview and benchmarking of motion detection methods”, dans *Background Modeling and Foreground Detection for Video Surveillance*, T. Bouwmans, F. Porikli, B. Hoferlin, et A. Vacavant, éd. Chapman and Hall/CRC, June 2014, ch. 1.
- R. Ju, T. Ren, et G. Wu, “Stereosnakes : contour based consistent object extraction for stereo images”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1724–1732.
- P. KaewTraKulPong et R. Bowden, “An improved adaptive background mixture model for real-time tracking with shadow detection”, dans *Video-Based Surveillance Systems*. Springer, 2002, pp. 135–144. DOI : 10.1007/978-1-4615-0913-4-11

- J. Kappes, B. Andres, F. Hamprecht, C. Schnorr, S. Nowozin, D. Batra, S. Kim, B. Kausler, J. Lellmann, N. Komodakis *et al.*, “A comparative study of modern inference techniques for discrete energy minimization problems”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1328–1335.
- K. Kim, T. H. Chalidabhongse, D. Harwood, et L. S. Davis, “Background modeling and subtraction by codebook construction”, dans *Proc. IEEE Int. Conf. Image Process.*, 2004, pp. 3061–3064.
- K. Kim, T. H. Chalidabhongse, D. Harwood, et L. Davis, “Real-time foreground-background segmentation using codebook model”, *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, Juin 2005. DOI : 10.1016/j.rti.2004.12.004
- S. Kim, D. Min, B. Ham, S. Ryu, M. N. Do, et K. Sohn, “DASC : Dense adaptive self-correlation descriptor for multi-modal and multi-spectral correspondence”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2103–2112.
- P. Kohli, L. Ladický, et P. H. Torr, “Robust higher order potentials for enforcing label consistency”, *Int. J. Comput. Vis.*, vol. 82, no. 3, pp. 302–324, 2009. DOI : 10.1007/s11263-008-0202-0
- V. Kolmogorov et R. Zabih, “Computing visual correspondence with occlusions using graph cuts”, dans *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, 2001, pp. 508–515.
- A. Kowdle, Y.-J. Chang, A. Gallagher, et T. Chen, “Active learning for piecewise planar 3d reconstruction”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 929–936.
- T. Kroeger, R. Timofte, D. Dai, et L. Van Gool, “Fast optical flow using dense inverse search”, dans *Proc. European Conf. Comput. Vis.*, 2016, pp. 471–488.
- S. J. Krotosky et M. M. Trivedi, “Mutual information based registration of multimodal stereo videos for person tracking”, *Comput. Vis. and Image Understanding*, vol. 106, no. 2, pp. 270–287, 2007. DOI : 10.1016/j.cviu.2006.10.008
- H. W. Kuhn, “The hungarian method for the assignment problem”, *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955. DOI : 10.1002/nav.3800020109
- L. J. Latecki et R. Lakämper, “Shape similarity measure based on correspondence of visual parts”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1185–1190, 2000. DOI : 10.1109/34.879802

- D.-S. Lee, “Effective gaussian mixture learning for video background subtraction”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, May 2005. DOI : 10.1109/TPAMI.2005.102
- V. Lempitsky, C. Rother, S. Roth, et A. Blake, “Fusion moves for markov random field optimization”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1392–1405, 2010.
- C. Li, X. Wang, L. Zhang, J. Tang, H. Wu, et L. Lin, “Weighted low-rank decomposition for robust grayscale-thermal foreground detection”, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 4, pp. 725–738, 2017.
- F. Li, T. Kim, A. Humayun, D. Tsai, et J. M. Rehg, “Video segmentation by tracking many figure-ground segments”, dans *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2192–2199.
- L. Li, W. Huang, I.-H. Gu, et Q. Tian, “Statistical modeling of complex backgrounds for foreground object detection”, *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov 2004. DOI : 10.1109/TIP.2004.836169
- S. Liao, G. Zhao, V. Kellokumpu, M. Pietikäinen, et S. Li, “Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1301–1306. DOI : 10.1109/CVPR.2010.5539817
- H.-H. Lin, J.-H. Chuang, et T.-L. Liu, “Regularized background adaptation : A novel learning rate control scheme for gaussian mixture modeling”, *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 822–836, 2011. DOI : 10.1109/TIP.2010.2075938
- L. Lin, Y. Xu, X. Liang, et J. Lai, “Complex background subtraction by pursuing dynamic spatio-temporal models”, *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 3191–3202, 2014. DOI : 10.1109/TIP.2014.2326776
- D. G. Lowe, “Distinctive image features from scale-invariant keypoints”, *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004. DOI : 10.1023/B:VISI.0000029664.99615.94
- L. Maddalena et A. Petrosino, “The SOBS algorithm : What are the limits?” dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 21–26. DOI : 10.1109/CVPRW.2012.6238922
- , “A self-organizing approach to background subtraction for visual surveillance applications”, *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1168–1177, Juil. 2008. DOI :

10.1109/TIP.2008.924285

F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, et P. Suetens, “Multimodality image registration by maximization of mutual information”, *IEEE Trans. Med. Imag.*, vol. 16, no. 2, pp. 187–198, 1997.

B. Mayer et J. Mundy, “Duration dependent codebooks for change detection”, dans *Proc. of the British Machine Vis. Conf.*, 2014.

A. Mittal et N. Paragios, “Motion-based background subtraction using adaptive kernel density estimation”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 2004, pp. II–302–II–309 Vol.2. DOI : 10.1109/CVPR.2004.1315179

A. Morde, X. Ma, et S. Guler, “Learning a background model for change detection”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 15–20. DOI : 10.1109/CVPRW.2012.6238921

T. Mouats et N. Aouf, “Multimodal stereo correspondence based on phase congruency and edge histogram descriptor”, dans *Proc. 16th Int. Conf. on Inf. Fusion*, July 2013, pp. 1981–1987.

D.-L. Nguyen, P.-L. St-Charles, et G.-A. Bilodeau, “Non-planar infrared-visible registration for uncalibrated stereo pairs”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2016, pp. 63–71. DOI : 10.1109/CVPRW.2016.48

Y. Nonaka, A. Shimada, H. Nagahara, et R. Taniguchi, “Evaluation report of integrated background modeling based on spatio-temporal features”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 9–14. DOI : 10.1109/CVPRW.2012.6238920

F. P. Oliveira et J. M. R. Tavares, “Medical image registration : a review”, *Comput. Meth. in Biomech. and Biomed. Eng.*, vol. 17, no. 2, pp. 73–93, 2014. DOI : 10.1080/10255842.2012.670855

N. Oliver, B. Rosario, et A. Pentland, “A bayesian computer vision system for modeling human interactions”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 831–843, Aug 2000. DOI : 10.1109/34.868684

C. Palmero, A. Clapés, C. Bahnsen, A. Møgelmoose, T. B. Moeslund, et S. Escalera, “Multi-modal rgb–depth–thermal human body segmentation”, *Int. J. Comput. Vis.*, vol. 118, no. 2, pp. 217–239, 2016. DOI : 10.1007/s11263-016-0901-x

- D. Parks et S. Fels, “Evaluation of background subtraction algorithms with post-processing”, dans *Proc. IEEE Int. Conf. Advanced Video and Signal Based Surveillance*, Sept 2008, pp. 192–199. DOI : 10.1109/AVSS.2008.19
- E. Parzen, “On estimation of a probability density function and mode”, *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962. DOI : 10.1214/aoms/1177704472
- F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, et A. Sorkine-Hornung, “A benchmark dataset and evaluation methodology for video object segmentation”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2016, pp. 724–732.
- P. Pinggera, T. Breckon, et H. Bischof, “On cross-spectral stereo matching using dense gradient features”, dans *Proc. British Mach. Vis. Conf.*, 2012. DOI : <http://dx.doi.org/10.5244/C.26.103>
- M. D. Pistarelli, A. D. Sappa, et R. Toledo, “Multispectral stereo image correspondence”, dans *Computer Analysis of Images and Patterns*. Springer, 2013, pp. 217–224. DOI : 10.1007/978-3-642-40246-3_27
- J. Pluim, J. Maintz, et M. Viergever, “Image registration by maximization of combined mutual information and gradient information”, *IEEE Trans. Med. Imag.*, vol. 19, no. 8, pp. 809–814, 2000. DOI : 10.1109/42.876307
- , “Mutual-information-based registration of medical images : a survey”, *IEEE Trans. Med. Imag.*, vol. 22, no. 8, pp. 986–1004, 2003. DOI : 10.1109/TMI.2003.815867
- G. Ramírez-Alonso et M. I. Chacón-Murguía, “Auto-adaptive parallel SOM architecture with a modular analysis for dynamic object segmentation in videos”, *Neurocomputing*, vol. 175B, pp. 990–1000, 2016. DOI : 10.1016/j.neucom.2015.04.118
- P. Ricaurte, C. Chilán, C. A. Aguilera-Carrasco, B. X. Vintimilla, et A. D. Sappa, “Feature point descriptors : Infrared and visible spectra”, *Sensors*, vol. 14, no. 2, pp. 3690–3701, 2014. DOI : 10.3390/s140203690
- T. Riklin-Raviv, N. Sochen, et N. Kiryati, “Shape-based mutual segmentation”, *Int. J. Comput. Vis.*, vol. 79, no. 3, pp. 231–245, 2008. DOI : 10.1007/s11263-007-0115-3
- O. Ronneberger, P. Fischer, et T. Brox, “U-Net : Convolutional networks for biomedical image segmentation”, dans *Int. Conf. Medical Image Comput. Computer-Assist. Intervention*, 2015, pp. 234–241.

- C. Rother, T. Minka, A. Blake, et V. Kolmogorov, "Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs", dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2006, pp. 993–1000. DOI : 10.1109/CVPR.2006.91
- C. Rother, V. Kolmogorov, et A. Blake, "'GrabCut' : Interactive foreground extraction using iterated graph cuts", *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, Août 2004. DOI : 10.1145/1015706.1015720
- J. Rubio, J. Serrat, et A. López, "Video co-segmentation", dans *Proc. 11th Asian Conf. Comput. Vis.*, Nov 2012, pp. 13–24. DOI : 10.1007/978-3-642-37444-9_2
- H. Sajid et S.-C. S. Cheung, "Background subtraction for static and moving camera", dans *Proc. IEEE Int. Conf. Image Process.*, 2015.
- D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, et P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth", dans *Proc. German Conf. Pattern Recognit.*, 2014, pp. 31–42.
- A. Schick, M. Bauml, et R. Stiefelhagen, "Improving foreground segmentations with probabilistic superpixel markov random fields", dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 27–31. DOI : 10.1109/CVPRW.2012.6238923
- M. Sedky, M. Moniri, et C. C. Chibelushi, "Spectral-360 : A physics-based technique for change detection", dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, June 2014.
- F. Seidel, C. Hage, et M. Kleinsteuber, "pROST : A smoothed l_p -norm robust online subspace tracking method for background subtraction in video", *Mach. Vis. Appl.*, vol. 25, no. 5, pp. 1227–1240, 2014. DOI : 10.1007/s00138-013-0555-4
- E. Shechtman et M. Irani, "Matching local self-similarities across images and videos", dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8. DOI : 10.1109/CVPR.2007.383198
- Y. Sheikh et M. Shah, "Bayesian modeling of dynamic scenes for object detection", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1778–1792, Nov 2005. DOI : 10.1109/TPAMI.2005.213
- C. Silva, T. Bouwmans, et C. Frelicot, "An eXtended center-symmetric local binary pattern for background modeling and subtraction in videos", dans *Proc. Int. Conf. Comput. Vis. Theory and Appl.*, 2015, pp. 395–402. DOI : 10.5220/0005266303950402

A. Sobral et A. Vacavant, “A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos”, *Comput. Vis. and Image Understanding*, vol. 122, pp. 4–21, 2014. DOI : 10.1016/j.cviu.2013.12.005

S. Sonn, G.-A. Bilodeau, et P. Galinier, “Fast and accurate registration of visible and infrared videos”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, June 2013, pp. 308–313. DOI : 10.1109/CVPRW.2013.53

P.-L. St-Charles, G.-A. Bilodeau, et R. Bergevin, “Flexible background subtraction with self-balanced local sensitivity”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, June 2014, pp. 408–413.

—, “SuBSENSE : A universal change detection method with local adaptive sensitivity”, *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 359–373, 2015. DOI : 10.1109/TIP.2014.2378053

P. L. St-Charles, G. A. Bilodeau, et R. Bergevin, “Fast image gradients using binary feature convolutions”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2016, pp. 1074–1082. DOI : 10.1109/CVPRW.2016.138

P.-L. St-Charles et G.-A. Bilodeau, “Improving background subtraction using local binary similarity patterns”, dans *Proc. IEEE Winter Conf. Applicat. Comput. Vis.*, March 2014, pp. 509–515. DOI : 10.1109/WACV.2014.6836059

P.-L. St-Charles, G.-A. Bilodeau, et R. Bergevin, “A self-adjusting approach to change detection based on background word consensus”, dans *Proc. IEEE Winter Conf. Applicat. Comput. Vis.*, Jan 2015, pp. 990–997. DOI : 10.1109/WACV.2015.137

—, “Online multimodal video registration based on shape matching”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2015, pp. 26–34.

—, “Universal background subtraction using word consensus models”, *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4768–4781, 2016.

—, “Mutual foreground segmentation with multispectral stereo pairs”, dans *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Oct 2017.

L. St-Laurent, D. Prévost, et X. Maldague, “Thermal imaging for enhanced foreground-background segmentation”, dans *Proc. of Quantitative Infrared Thermography*, Jul 2010, pp. 28–38.

- A. Stagliano, N. Noceti, A. Verri, et F. Odone, “Online space-variant background modeling with sparse coding”, *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2415–2428, 2015. DOI : 10.1109/TIP.2015.2421435
- C. Stauffer et W. E. L. Grimson, “Adaptive background mixture models for real-time tracking”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 1999, pp. 246–252. DOI : 10.1109/CVPR.1999.784637
- Y. Sun, X. Tao, Y. Li, et J. Lu, “Robust 2D principal component analysis : A structured sparsity regularized approach”, *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2515–2526, 2015. DOI : 10.1109/TIP.2015.2419075
- T. Tian, X. Mei, Y. Yu, C. Zhang, et X. Zhang, “Automatic visible and infrared face registration based on silhouette matching and robust transformation estimation”, *Infrared Physics & Technology*, vol. 69, no. 0, pp. 145–154, 2015. DOI : 10.1016/j.infrared.2014.12.011
- B. Tippetts, D. J. Lee, K. Lillywhite, et J. Archibald, “Review of stereo vision algorithms and their suitability for resource-limited systems”, *J. Real-Time Image Proc.*, vol. 11, no. 1, pp. 5–25, 2016.
- A. Torabi et G.-A. Bilodeau, “Local self-similarity-based registration of human ROIs in pairs of stereo thermal-visible videos”, *Pattern Recognit.*, vol. 46, no. 2, pp. 578–589, 2013. DOI : 10.1016/j.patcog.2012.07.026
- A. Torabi, G. Massé, et G.-A. Bilodeau, “An iterative integrated framework for thermal-visible image registration, sensor fusion, and people tracking for video surveillance applications”, *Comput. Vis. and Image Understanding*, vol. 116, no. 2, pp. 210–221, 2012. DOI : 10.1016/j.cviu.2011.10.006
- K. Toyama, J. Krumm, B. Brumitt, et B. Meyers, “Wallflower : principles and practice of background maintenance”, dans *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1, 1999, pp. 255–261 vol.1. DOI : 10.1109/ICCV.1999.791228
- W. Treible, P. Saponaro, S. Sorensen, A. Kolagunda, M. O’Neal, B. Phelan, K. Sherbondy, et C. Kambhamettu, “CATS : A color and thermal stereo benchmark”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2961–2969.
- R. Tron et R. Vidal, “A benchmark for the comparison of 3-d motion segmentation algorithms”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007.

D.-M. Tsai et S.-C. Lai, “Independent component analysis-based background subtraction for indoor surveillance”, *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 158–167, Jan. 2009. DOI : 10.1109/TIP.2008.2007558

M. Van Droogenbroeck et O. Barnich, “ViBe : A disruptive method for background subtraction”, dans *Background Modeling and Foreground Detection for Video Surveillance*, T. Bouwmans, F. Porikli, B. Hoferlin, et A. Vacavant, éd. Chapman and Hall/CRC, June 2014, ch. 7.

M. Van Droogenbroeck et O. Paquot, “Background subtraction : Experiments and improvements for ViBe”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 32–37.

S. Varadarajan, P. Miller, et H. Zhou, “Spatial mixture of gaussians for dynamic background modelling”, dans *Proc. IEEE Int. Conf. Adv. Video and Signal Based Surveillance*, 2013, pp. 63–68. DOI : 10.1109/AVSS.2013.6636617

S. Vicente, C. Rother, et V. Kolmogorov, “Object cosegmentation”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2011, pp. 2217–2224. DOI : 10.1109/CVPR.2011.5995530

V. Vineet et P. Narayanan, “CUDA cuts : Fast graph cuts on the GPU”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008.

B. Wang et P. Dudek, “A fast self-tuning background subtraction algorithm”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, June 2014.

H. Wang et D. Suter, “Background subtraction based on a robust consensus method”, dans *Proc. IEEE Int. Conf. Pattern Recognit.*, vol. 1, 2006, pp. 223–226. DOI : 10.1109/ICPR.2006.312

—, “A consensus-based method for tracking : Modelling background scenario and foreground appearance”, *Pattern Recognit.*, vol. 40, no. 3, pp. 1091–1105, 2007. DOI : 10.1016/j.patcog.2006.05.024

R. Wang, F. Bunyak, G. Seetharaman, et K. Palaniappan, “Static and moving object detection using flux tensor with split gaussian models”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, June 2014.

Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, et P. Ishwar, “CDnet 2014 :

An expanded change detection benchmark dataset”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, June 2014, pp. 387–394.

Y. Wang, Z. Luo, et P.-M. Jodoin, “Interactive deep learning method for segmenting moving objects”, *Pattern Recognit. Lett.*, vol. 96, pp. 66–75, 2017.

J. Winn, A. Criminisi, et T. Minka, “Object categorization by learned universal visual dictionary”, dans *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, 2005, pp. 1800–1807.

O. Woodford, P. Torr, I. Reid, et A. Fitzgibbon, “Global stereo reconstruction under second-order smoothness priors”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2115–2128, 2009.

M. Wu et X. Peng, “Spatio-temporal context for codebook-based dynamic background subtraction”, *Int. J. Electron. Commun.*, vol. 64, no. 8, pp. 739–747, 2010. DOI : 10.1016/j.aeue.2009.05.004

M.-H. Yang, C.-R. Huang, W.-C. Liu, S.-Z. Lin, et K.-T. Chuang, “Binary descriptor based nonparametric background modeling for foreground extraction by using detection theory”, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 4, pp. 595–608, 2015. DOI : 10.1109/TCSVT.2014.2361418

Y. Ye et J. Shan, “A local descriptor based registration method for multispectral remote sensing images with non-linear intensity differences”, *J. Photogrammetry and Remote Sens.*, vol. 90, no. 0, pp. 83–95, 2014. DOI : 10.1016/j.isprsjprs.2014.01.009

A. Yilmaz, O. Javed, et M. Shah, “Object tracking : A survey”, *ACM Computing Surveys*, vol. 38, no. 4, p. 13, 2006.

C. Zhang, Z. Li, Y. Cheng, R. Cai, H. Chao, et Y. Rui, “Meshstereo : A global stereo model with mesh alignment regularization for view interpolation”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2057–2065.

C. Zhang, Z. Li, R. Cai, H. Chao, et Y. Rui, “Joint multiview segmentation and localization of RGB-D images using depth-induced silhouette consistency”, dans *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4031–4039.

S. Zhang, H. Yao, et S. Liu, “Dynamic background modeling and subtraction using spatio-temporal local binary patterns”, dans *Proc. IEEE Int. Conf. Image Process.*, 2008, pp. 1556–1559. DOI : 10.1109/ICIP.2008.4712065

J. Zhao et S. C. Sen-ching, “Human segmentation by geometrically fusing visible-light and thermal imageries”, *Multimedia Tools and Applicat.*, vol. 73, no. 1, pp. 61–89, 2014.

Z. Zhao, X. Zhang, et Y. Fang, “Stacked multilayer self-organizing map for background modeling”, *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2841–2850, Sept 2015. DOI : 10.1109/TIP.2015.2427519

X. Zhou, C. Yang, et W. Yu, “Moving object detection by detecting contiguous outliers in the low-rank representation”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 597–610, 2013. DOI : 10.1109/TPAMI.2012.132

H. Zhu, F. Meng, J. Cai, et S. Lu, “Beyond pixels : A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation”, *J. Visual Commun. and Image Represent.*, vol. 34, pp. 12–27, 2016.

B. Zitová et J. Flusser, “Image registration methods : a survey”, *Image and Vis. Comp.*, vol. 21, no. 11, pp. 977–1000, 2003. DOI : 10.1016/S0262-8856(03)00137-9

Z. Zivkovic, “Improved adaptive gaussian mixture model for background subtraction”, dans *Proc. IEEE Int. Conf. Pattern Recognit.*, vol. 2, 2004, pp. 28–31 Vol.2. DOI : 10.1109/ICPR.2004.1333992

Z. Zivkovic et F. van der Heijden, “Efficient adaptive density estimation per image pixel for the task of background subtraction”, *Pattern Recognit. Lett.*, vol. 27, no. 7, pp. 773–780, 2006. DOI : 10.1016/j.patrec.2005.11.005