Titre: Title:	Low and Variable Frame Rate Face Tracking Using an IP PTZ Camera
Auteur: Author:	Parisa Darvish Zadeh Varcheie
Date:	2010
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Darvish Zadeh Varcheie, P. (2010). Low and Variable Frame Rate Face Tracking Using an IP PTZ Camera [Ph.D. thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/318/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/318/
Directeurs de recherche: Advisors:	Guillaume-Alexandre Bilodeau
Programme: Program:	Génie informatique

UNIVERSITÉ DE MONTRÉAL

LOW AND VARIABLE FRAME RATE FACE TRACKING USING AN IP PTZ CAMERA

PARISA DARVISH ZADEH VARCHEIE

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 2010

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée:

LOW AND VARIABLE FRAME RATE FACE TRACKING USING AN IP PTZ CAMERA

présentée par : DARVISH ZADEH VARCHEIE, Parisa

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

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

M. DESMARAIS Michel, Ph.D., président

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

M. PAL Christopher J., Ph.D., membre

M. PAYEUR Pierre, Ph.D., membre

DEDICATE

To Sina and my family

ACKNOWLEDGEMENT

I would like to thank:

- Prof. Guillaume-Alexandre BILODEAU, my PhD supervisor, for all his supports, his helps and his novel ideas in different research domains.
- Dr. Langis GAGNON, my research director at Centre de Recherche Informatique de Montréal (CRIM), for his financial support during my PhD and his ideas in my research project at CRIM.
- Jury members for their valuable comments on the thesis.
- All of my colleagues at LITIV, Atousa, Pier-Luc, François, Soufiane and my friends in Montréal for their help and suggestions and everybody who contributed to my thesis.
- My husband, Sina, and my parents, and my sisters who supported me in my PhD program.

RÉSUMÉ

En vision par ordinateur, le suivi d'objets avec des caméras PTZ a des applications dans divers domaines, tels que la surveillance vidéo, la surveillance du trafic, la surveillance de personnes et la reconnaissance de visage. Toutefois, un suivi plus précis, efficace, et fiable est requis pour une utilisation courante dans ces domaines. Dans cette thèse, le suivi est appliqué au haut du corps d'un humain, en incluant son visage. Le suivi du visage permet de déterminer son emplacement pour chaque trame d'une vidéo. Il peut être utilisé pour obtenir des images du visage d'un humain dans des poses différentes. Dans ce travail, nous proposons de suivre le visage d'un humain à l’aide d'une caméra IP PTZ (caméra réseau orientable). Une caméra IP PTZ répond à une commande via son serveur Web intégré et permet un accès distribué à partir d'Internet. Le suivi avec ce type de caméra inclut un bon nombre de défis, tels que des temps de réponse irrégulier aux commandes de contrôle, des taux de trame faibles et irréguliers, de grand mouvements de la cible entre deux trames, des occlusions, des modifications au champ de vue, des changements d'échelle, etc.

Dans notre travail, nous souhaitons solutionner les problèmes des grands mouvements de la cible entre deux trames consécutives, du faible taux de trame, des modifications de l'arrière-plan, et du suivi avec divers changements d'échelle. En outre, l'algorithme de suivi doit prévoir les temps de réponse irréguliers de la caméra.

Notre solution se compose d’une phase d’initialisation pour modéliser la cible (haut du corps), d’une adaptation du filtre de particules qui utilise le flux optique pour générer des échantillons à chaque trame (APF-OFS), et du contrôle de la caméra. Chaque composante exige des stratégies différentes.

Lors de l'initialisation, on suppose que la caméra est statique. Ainsi, la détection du mouvement par soustraction d’arrière-plan est utilisée pour détecter l'emplacement initial de la personne. Ensuite, pour supprimer les faux positifs, un classificateur Bayesien est appliqué sur la région détectée afin de localiser les régions avec de la peau. Ensuite, une détection du visage basée sur la méthode de Viola et Jones est effectuée sur les régions de la peau. Si un visage est détecté, le suivi est lancé sur le haut du corps de la personne.

Après la détection de la personne à suivre, l'étape de suivi est démarrée. Nous proposons un filtre de particules qui utilise le flux optique pour générer des échantillons à chaque trame (APF-OFS). Une approche basée sur le flux optique est donc combinée avec la méthode du filtre de particules. Ainsi, le suivi est réalisée en fonction d'un échantillonnage autour de la position prédite de la cible (obtenue par un prédicteur) et les régions avec du mouvement détectées par la méthode du flux optique. La sélection des échantillons est faite à l’aide de distances sur l’apparence au niveau des couleurs de chaque échantillon avec la cible à suivre.

En mode suivi, une fois qu'un visage est détecté dans une image, le panoramique et l'inclinaison de la caméra PTZ sont contrôlés afin d'amener le visage au centre de l'image. Ainsi, normalement, le visage devrait être près du centre de l'image. Donc, la zone de recherche dans la trame suivante n'a pas besoin d'être l'image entière, mais est plutôt une région d’intérêt autour du centre de l’image. À partir du moment où la caméra change d’orientation, la soustraction d'arrière-plan n'est plus efficace pour éliminer les faux positifs. Dans ce cas, des distances sur l’apparence de la cible et un filtre de particules peuvent résoudre le problème de suivi.

Lorsque que le visage est suivi correctement, la caméra commence par faire un zoom sur le visage pour prendre une photo de la personne. Initialement, la caméra couvre un large champ de vue. Afin de reconnaître une personne, il est nécessaire de prendre une photo avec une résolution suffisante et donc de rétrécir le champ de vue.

Les résultats montrent que notre algorithme peut traiter et surmonter de grands mouvements entre deux images consécutives, et que l’emplacement trouvé de la cible correspond aux valeurs témoins. Aussi, la caméra peut se centrer sur la cible avec une bonne précision. Ainsi, le centre de l'objectif de la caméra est presque toujours situé à une distance de 1/6ième du diamètre de l’image à partir de son centre.

L'utilisation d'une caméra IP implique des délais et de grands mouvements entre deux trames consécutives. Nous proposons une nouvelle technique qui est conçue pour être précise, robuste, et appropriée pour cette application en temps réel. En outre, nous avons adapté des algorithmes de suivi afin d'être en mesure d'obtenir de la robustesse à la fois aux occlusions et aux mises à l'échelle. Le suivi des objets et le contrôle d'une caméra IP PTZ est une contribution qui permettra de nombreuses avancées dans le futur.

ABSTRACT

Object tracking with PTZ cameras has various applications in different computer vision topics such as video surveillance, traffic monitoring, people monitoring and face recognition. Accurate, efficient, and reliable tracking is required for this task. Here, object tracking is applied to human upper body tracking and face tracking. Face tracking determines the location of the human face for each input image of a video. It can be used to get images of the face of a human target under different poses. We propose to track the human face by means of an Internet Protocol (IP) Pan-Tilt-Zoom (PTZ) camera (i.e. a network-based camera that pans, tilts and zooms). An IP PTZ camera responds to command via its integrated web server. It allows a distributed access from Internet (access from everywhere, but with non-defined delay). Tracking with such camera includes many challenges such as irregular response times to camera control commands, low and irregular frame rate, large motions of the target between two frames, target occlusion, changing field of view (FOV), various scale changes, etc.

In our work, we want to cope with the problem of large inter-frame motion of targets, low usable frame rate, background changes, and tracking with various scale changes. In addition, the tracking algorithm should handle the camera response time and zooming.

Our solution consists of a system initialization phase which is the processing before camera motion and a tracker based on an Adaptive Particle Filter using Optical Flow based Sampling (APF-OFS) tracker, and camera control that are the processing after the motion of the camera. Each part requires different strategies.

For initialization, when the camera is stationary, motion detection for a static camera is used to detect the initial location of the person face entering an area. For motion detection in the FOV of the camera, a background subtraction method is applied. Then to remove false positives, Bayesian skin classifier is applied on the detected motion region to discriminate skin regions from non skin regions. Face detection based on Viola and Jones face detector can be performed on the detected skin regions independently of their face size and position within the image.

After face detection, the tracking step is started. We propose an adaptive particle filter using optical flow based sampling (APF-OFS) method adapted to the general object tracking problem with an IP PTZ camera. Optical flow is used to extract moving pixels is combined with particle filter that has robustness to non-Gaussian distribution of target movements to extract random motion of the object. Target modeling and tracking are done based on sampling around predicted positions obtained by a position predictor and moving regions are detected by optical flow. The scoring of sample features is done with some reasonable normalization functions. The normalization functions are used to combine different measure values to standardize their magnitudes within similar ranges. Normalization functions are applied to geometric and appearance features.

In the tracking mode, once a face is detected in the first image frame, the pan and tilt of the PTZ camera is controlled to bring the face back in the image center. Depending on the speed of the person motion, the face might be near the image center; so the search area for the next frame does not need to be the whole image and should be an area around it. For processing, because of camera motion, background subtraction is not effective to remove the false positive rate. In this case, the combination of an optical flow method and particle filter tracking can solve the tracking problem. While the camera is tracking the face, it starts to zoom on the face to take a photo of that person. Generally, the camera covers a wide FOV with low resolution so in order to recognize that person, it is necessary to take an image with sufficient resolution.

The general contribution is dynamic face tracking with an IP PTZ camera, a first in the field of computer vision. The specific contributions are (1) modeling and formulating the tracking program as a servo control loop that compensates all the delays resulting from network or processing; and (2) proposing of an adaptive particle filter with optical flow samples method to cover all possible candidate regions, handle large inter-frame motion of the target, low tracking frame rate and recover the tracking target in the case of occlusion or target lost.

Results show that our algorithm can handle and overcome large motion between two consecutive frames, and the detected target location is near to the ground truth. In addition the camera can center on the target with a good precision. The target usually is located in a constant distance within 1/6th of image diameter from the image center.

Using an IP camera entails remote monitoring and delays caused by communicating over a network. We propose a novel technique that is designed to be accurate, robust and appropriate for this real time application. Furthermore, we adapt tracking algorithms to be able to obtain robustness against both occlusion and scaling problems with a mobile camera in real time process. Monitoring the objects, sending the commands and controlling the subject with an IP camera is a robust tool to do many advanced projects in the future.

CONDENSÉ EN FRANÇAIS

Introduction

Un suivi d'objets précis, efficace, et fiable est un problème difficile à résoudre, mais essentiel pour les applications de visions par ordinateur, telles que la surveillance vidéo. Dans le domaine de la sécurité et de la vidéosurveillance, les caméras PTZ fournissent des informations riches et utiles à faible coût grâce à une couverture de scène élevée en comparaison avec les caméras fixes qui n'observent en général qu'une petite partie d'une scène.

Une caméra réseau peut être décrite comme une caméra et un ordinateur combinés en une seule unité. Elle capte et transmet en direct les images par protocole Internet (IP), permettant ainsi aux utilisateurs autorisés d'afficher, de stocker, et de gérer les vidéos localement ou à distance sur les infrastructures de réseaux standards. Une caméra réseau possède sa propre adresse IP. Elle est connectée à un réseau et est dotée d'un serveur web, d'un serveur FTP, d'un client FTP, d'un client courriel, d'un module de gestion d'alarmes, et beaucoup plus. Une caméra réseau n'a pas besoin d'être connectée à un ordinateur, elle fonctionne de façon indépendante et peut être placée là où il y a une connexion réseau IP.

La détection de visages combinée à la versatilité des caméras IP PTZ nous permettra d'augmenter les capacités des systèmes d'identification automatique de personne, notamment dans le domaine de la vidéosurveillance. Les applications des techniques de suivi du visage par des caméras PTZ peuvent être résumées comme le suivi automatiquement et l'identification des personnes à différents endroits tels que les stations de transport, banques, maisons, écoles, laboratoires, bureaux, industries, etc. Donc, ce projet est lié à deux principaux thèmes de recherche, d'abord la détection de visage et le deuxième, le suivi du visage détecté par le contrôle d'une caméra IP PTZ. La détection d'un visage et son suivi permet de déterminer l'emplacement et la taille de celui-ci pour chaque trame d'une séquence vidéo. C'est le problème que nous étudions dans cette thèse. Il se compose de l''initialisation qui est, la représentation de la cible avant tout de mouvement de caméra, ainsi que le suivi et contrôle de la caméra qui sont les traitements après la modélisation de la cible. Chaque partie comprend différentes étapes. L'initialisation consiste en la soustraction du fond, la détection de la peau, et la détection de visage. Les étapes pour le suivi et contrôle de la caméra sont l'estimation du mouvement, de l'extraction des candidats cibles, la sélection des candidats les plus ressemblant et l'orientation de la caméra pour la centrer sur la cible. Après la détection de visage et le suivi, la reconnaissance du visage est souvent effectuée pour distinguer le visage détecté. L'avantage du suivi d'objets en utilisant seulement une caméra PTZ est de supprimer le besoin d'avoir de nombreuses caméras fixes pour réduire le coût tout en gardant une très large couverture de la scène. Ici, le suivi d'un objet est appliqué aux visages humains et au haut du corps. Il peut être utilisé pour obtenir des images du visage d'une cible humaine dans des poses différentes.

Précise, efficace et fiable, le suivi et le classement des objets multiples est une tâche difficile en matière de surveillance vidéo en raison de la résolution d'image, temps de traitement, la précision du suivi, etc. Ces défis sont exacerbés avec des caméras PTZ IP. Ils peuvent être résumés comme suit:

- Temps de réponse irrégulier aux commandes de caméra;
- Faible taux de trame utilisable (alors que l'appareil exécute une commande de mouvement, les images reçues sont souvent inutiles, car aucune nouvelle commande ne peut être envoyée sans délai);
- Le taux de trame irrégulier en raison de retards sur le réseau (le temps entre deux images n'est pas nécessairement constant);
- Modification de FOV résultant de panoramique, inclinaison et zoom;
- Différentes échelles des objets.

Suivi automatique du visage par une caméra IP PTZ est un sujet nouveau qui exige des capacités telles que:

- Détection des visages à des positions différentes (visages pas nécessairement en vue frontale).
- Des scènes avec des échelles différentes.
- Couverture d'un large champ de vision (pan, tilt, zoom).
- Temps de traitement court (utilisation en temps réel').

- Applicabilité des images à basse résolution.
- Gérer de vastes mouvements inter-images de la cible.

Dans notre travail, nous souhaitons solutionner les problèmes des grands mouvements de la cible entre deux trames consécutives, du faible taux de trame, des modifications de l'arrière-plan, et du suivi avec divers changements d'échelle. En outre, l'algorithme de suivi doit prévoir les temps de réponse irrégulier de la caméra.

Méthodologie et l’architecture du système

Nous considérons que notre caméra PTZ est comme un système d'asservissement. En effet, toute machine ou pièce d'équipement contient des systèmes de contrôle un ou plusieurs servo pour contrôler précisément le mouvement de chacune de ses parties mobiles. Le servo de contrôle et de suivi est modélisé par un contrôle en boucle fermée qui a une rétroaction négative. Du point de vue classification de système, le système est discret, stable, variant dans le temps, causal, dynamique et non linéaire. Ces propriétés nous aident dans la modélisation du système. Nous pouvons utiliser les caractéristiques de chaque catégorie pour représenter notre modèle de système.

En raison de la latence du réseau et la réponse de la caméra, il n'est pas possible de contrôler la caméra en permanence. Les commandes sont en attente et réalisées l'une après l'autre. Si une nouvelle commande est envoyée et la file d'attente n'est pas vide, la plupart des commandes seront appliquées trop tard, et la cible sera perdue. Dans notre système d'asservissement, une nouvelle requête / commande de zoom est délivré que si la file d'attente est vide. De cette manière, un bon contrôle de la caméra est maintenue, mais la cible peut se déplacer au cours du dernier mouvement / commande de zoom. Ainsi, le mouvement de la cible doit être estimée pour garder la cible dans le champ de vision de la caméra.

Notre solution se compose d’une phase d’initialisation pour modéliser la cible (haut du corps), d’une adaptation du filtre de particules qui utilise le flux optique pour générer des échantillons à chaque trame (APF-OFS), et du contrôle de la caméra. Chaque composante exige des stratégies différentes. L'initialisation est effectuée avant d'introduire la cible de suivi au tracker. Pour la modélisation de la cible, les caractéristiques doivent être invariant, discriminatoire, et assez stable au cours du suivi. Ces contraintes dépendent de l'application et le des scénarios de suivi. Dans notre application, le système de suivi est à faible taux de trame, sous un éclairage à peu près constante, mais avec une variabilité dans l'arrière-plan à cause des changements de panoramique et d'inclinaison, avec des variations de mise à l'échelle, et avec d'importantes variations du mouvement de la cible. Bien que simple, un modèle de couleur de base satisfait à ces exigences. D'autres modèles pourraient être utilisés dans notre méthodologie, si des détails plus précis doivent être pris en compte. Cependant, le modèle doit être robuste à grande échelle et au changement instantané de point de vue en raison de la faible cadence de trame. Par conséquent, dans notre problème, la cible est représentée par une ellipse qui délimite la tête et une partie du tronc du corps humain. L'initialisation est effectuée automatiquement par l'extraction de la partie supérieure du corps (tête et torse) de la personne à suivre. C'est une partie qui doit toujours être visible, soit lorsque la personne est loin ou quand elle est près de la caméra.

Lors de l'initialisation, on suppose que la caméra est statique. Ainsi, la détection du mouvement par soustraction d’arrière-plan est utilisée pour détecter l'emplacement initial de la personne. Ensuite, pour supprimer les faux positifs, un classificateur Bayesien est appliqué sur la région détectée afin de localiser les régions avec de la peau. Ensuite, une détection du visage basée sur la méthode de Viola et Jones est effectuée sur les régions de la peau. Après la détection du visage, nous construisons une région rectangulaire avec la même largeur que le visage détecté et une hauteur de 4 fois plus long que la longueur face au modèle de la cible. Nous avons ajusté une ellipse à l'intérieur de la boîte englobante de la région choisie, et le modèle résultant est une région elliptique avec des caractéristiques. Si la cible est complète, le suivi est lancé sur le haut du corps de la personne.

Après la détection de la personne à suivre, l'étape de suivi est démarrée. Le suivi se compose d'un objet de suivi et de la prédiction de position. La cible est localisé en permanence par un objet de suivi par catégorie. L'emplacement de la cible détectée est affecté par trois retards. Par conséquent, un bloc de prédiction de position est utilisée pour compenser la latence du système résultant de trois retards et pour modifier la position détectée par le suivi d'objets par catégorie. Ensuite, la position prédite est envoyée au bloc de contrôle de la caméra pour la centrer sur la cible. La commande de caméra et la prédiction de position sont affectées par les résultats de suivi d'objets. Nous proposons un filtre de particules qui utilise le flux optique pour générer des échantillons à chaque trame (APF-OFS). Une approche basée sur le flux optique est donc combinée avec la méthode du filtre de particules. Ainsi, le suivi est réalisée en fonction d'un échantillonnage autour de la position prédite de la cible (obtenue par un prédicteur) et les régions avec du mouvement détectées par la méthode du flux optique. Comme on l'a constaté expérimentalement, les vecteurs de mouvement détectés sont bruités. En outre, les vecteurs de mouvement de caméra ont un effet sur les vecteurs de déplacement de l'objet. Ainsi, pour éliminer cet effet, les vecteurs de mouvement de caméra sont extraits. Pour calculer les vecteurs de mouvement de la caméra, un histogramme de vecteurs de mouvement radial est calculé. La classe qui a le nombre maximum de vecteurs est attribuée à la longueur du vecteur de mouvement et de l'angle. Puisque nous n'avons pas la position 3D des échantillons et que le vecteur de mouvement de la caméra est également une approximation en 2D, nous ne pouvons pas distinguer la position transformée de l’échantillon d'une image à une autre avec précision si le déplacement de la caméra est grand. Nous avons donc adapté le filtre à particules à notre demande. Par conséquent, à partir de l'exemple précédent nous ne prenons l'échantillon cible avec la plus grande probabilité, si la caméra ne bouge pas. Mais si la caméra se déplace, car il est supposé que le centre de caméra est sur la cible, pour le filtre à particules nous ne prenons la position du centre de l'image avec la taille de l'échantillon de la dernière cible. Par conséquent, nous échantillonnons avec des ellipses l'image autour de deux types de ROI et les modélisons:

1. Autour de la position de la cible précédente ou autour du centre de l'image,
2. Dans les zones extraites par flux optique.

Ainsi, le ré-échantillonnage se fait à partir de deux types d'échantillons: position de la cible précédente et des échantillons de base basé sur le mouvement. Si l’objet commence à se déplacer, pendant le mouvement, le processus d'échantillonnage se fait à travers le centre de l'image en mouvement et les zones extraites par flux optique. Pendant le mouvement de la caméra, le centre de l'image peut être une prédiction de la position de la cible, car idéalement, l’objet devrait toujours être au centre de l'image.

La sélection des échantillons est faite à l’aide de distances sur l’apparence au niveau des couleurs de chaque échantillon avec la cible à suivre. Nous appliquons différentes mesures sur la couleur et la position de chaque échantillon, parce que le résultat d'une seule mesure n’est pas assez précis dans toutes les conditions alors que les résultats des autres distances dans les mêmes conditions sont meilleurs. Nous voulons un équilibre entre la précision des mesures dans les conditions générales. En mode suivi, une fois qu'un visage est détecté dans une image, le panoramique et l'inclinaison de la caméra PTZ est contrôlé afin d'amener le visage au centre de l'image. Ainsi, normalement, le visage devrait être près du centre de l'image. Donc, la zone de recherche dans la trame suivante n'a pas besoin d'être l'image entière, mais est plutôt une région d’intérêt autour du centre de l’image. À partir du moment où la caméra change d’orientation, la soustraction d'arrière-plan n'est plus efficace pour éliminer les faux positifs. Dans ce cas, des distances sur l’apparence de la cible et un filtre de particules peuvent résoudre le problème de suivi.

Le bloc de contrôle de caméra ajuste panoramique et d'inclinaison des valeurs fondées sur la position prédite et zoom sur la cible avec une valeur appropriée du zoom. Il faut faire attention sur le déplacement de la caméra ou le zoom pour éviter de perdre l'objet dans FOV de la caméra. Pour atteindre cet objectif, certains critères doivent être vérifiés pour s'assurer qu'elles sont correctement bougées la caméra pour suivre la cible ou pour effectuer un zoom de la caméra sur la cible. Camera effectue des commandes séquentielle et non en parallèle. De plus, chaque zoom ou de déplacement de la caméra correspond à un retard à cause d'une tâche mécanique qui a besoin de temps à faire. Si nous voulons envoyer des commandes consécutivement sans tenir compte de ces retards, la probabilité que nous perdons la cible est élevée. Ainsi, la communication avec l'appareil photo est commandée par un mutex. Le mot vient du mutex exclusion mutuelle, qui assigne à un objet, organise exclusion mutuelle entre les threads. Un mutex est appliquée entre les fils de s'assurer que seulement l'un des fils est à la fois autorisé à exécuter une application spécifique de code à la fois. Par conséquent, une commande de mouvement ou une commande de zoom sera envoyée à l'appareil si le mutex montre que la tâche précédente du fil a été terminée et maintenant il est prêt pour une autre commande.

Lorsque que le visage est suivi correctement, la caméra commence peut faire un zoom sur le visage pour prendre une photo de la personne. Initialement, la caméra couvre un large champ de vue. Afin de reconnaître une personne, il est nécessaire de prendre une photo avec une résolution suffisante et donc de rétrécir le champ de vue.

Nous trouvons filtre à particules classiques comme une solution appropriée pour le suivi avec la caméra PTZ, puisque l'arrière-plan change en raison de la caméra panoramique, d'inclinaison et de zoom. D'un côté, dans notre système, les mouvements de cible peuvent être aléatoires et / ou réguliers. Aussi entre deux images consécutives, la cible peut avoir de grands mouvements. Par conséquent, entre deux images consécutives de la cible peut être situé n'importe où dans l'image. Nous devrions trouver tous les candidats possibles de résulter de tous les types des mouvements de cible pour trouver le candidat qui est le mieux correspondre. Méthode de filtre à particules de nous fournir une représentation de tous les mouvements possibles de la cible découlant de mouvement régulier ou aléatoire de la cible. De l'autre côté, la méthode de suivi doit être rentable d'avoir un petit retard, puisque notre système est basé sur internet, le système de suivi doit traiter les retards de trafic réseau et le temps de traitement.

Ici, seulement la méthode de filtre à particules classiques n'est pas appropriée, parce que nous aurions besoin beaucoup d'échantillons pour couvrir toutes les possibilités partout dans l'image. Évaluation de nombreux échantillons correspond à avoir un temps de traitement élevée et, depuis le suivi d'objet doit être rentable, ce qui les échantillons doivent être générés de façon appropriée et de manière sélectif. Mais dans la méthode d'APF-OFS en utilisant des échantillons de flux optique, il est nécessaire pour générer des échantillons de filtres à particules seulement où il ya un mouvement des résultats de mouvement de la cible et pas de mouvement de la caméra. Le vecteur de mouvement de caméra est retiré de tout mouvement pixels détectés par flux optique en utilisant un schéma histogramme radiale.

Par ailleurs, l'APF-OFS utilise un prédicteur pour modifier la position de la cible détectée par le bloc de suivi sur la base des trois retards du système, les déplacements de la cible et de caméra. En conséquence la méthode d'APF-OFS est une version modifiée de la méthode de filtre à particules qui est appropriée pour notre système de suivi.

Notre filtre de particules peut être utilisé pour les images de faible résolution. L'algorithme proposé permet de distinguer et reconnaître les visages. Il gère les délais réseaux et le temps de réponse de la caméra.

Résultats

Différentes expériences ont été réalisées afin d'évaluer les performances du système de suivi proposé selon les différents paramètres utilisés et pour le comparer avec d'autres méthodes de suivi. Nous avons intégré dans notre système de suivi le zoom et la capture d'images de visages. Les résultats montrent que notre algorithme peut traiter et surmonter de grands mouvements entre deux images consécutives, et que l'emplacement trouvé de la cible correspond aux valeurs témoins. Aussi, la caméra peut se centrer sur la cible avec une bonne précision. C'est parce que les échantillons du filtre de particules sont bien répartis dans chaque trame avec des échelles différentes autour des positions candidates possibles. Notre système perdra une personne si celle-ci change de direction soudainement, ou si elle marche très vite dans la direction opposée de la direction prévue. Ce problème est amplifié dans le cas d'un zoom, alors que le champ de vue est limité et que la personne a une vitesse de mouvement relative plus grande.

Dans la comparaison des performances d'APF-OFS avant et après le zoom, le taux de ralentissement de trame est obtenue en raison de ajoutant une nouvelle tâche qui est la commande de zoom de caméra et prend plus de temps qu'un simple mouvement de caméra (zoom prend entre 1,6 ~ 2,5 fois plus que le seul déplacement). De plus, en cas de zoom, plus faible précision, plus de la fragmentation de suivi et plus d'erreur dans les distances sont obtenus parce que de FOV de la caméra est limitée et le réglage de zoom de la caméra avec la vitesse de cible prend du temps. Les erreurs de distance est plus à cause d'avoir de plus grande taille cible, tout en caméra a zoomé et son déplacement semble plus élevé.

Nous pouvons récupérer le suivi si la personne se déplace de nouveau à l'intérieur du champ de vision de la caméra. La méthode proposée peut traiter les occlusions de courte durée à la condition que l'objet reste dans le champ de vue de la caméra. Nous pouvons également gérer les mouvements aléatoires entre les trames, tant et aussi longtemps que la position de la cible est bien prédite, que son apparence ne change pas de façon significative, et qu'elle reste dans le champ de vue de la caméra. Un prédicteur de mouvement est utilisé pour compenser trois délais (acquisition, traitement, mouvement de la caméra) qui peuvent résulter en une sortie du champ de vue par l'objet.

Nous avons déterminé les limites du système à l'étape d'initialisation et de suivi et de contrôle de la caméra. Il est démontré que la taille minimale de la détection de visage dans notre system à l'étape d'initialisation est limitée en fonction de la taille minimale de la détection dans la méthode de Viola et Jones. En outre, dans le suivi lorsque la cible est très faible et il ya des changements d'éclairage, la cible sera perdue. Nous avons montré que le zoom sur la cible peut améliorer la taille de la cible de petite taille. Toutefois, le zoom augmente la teneur de l'échantillon cible à comparer. Lorsque la localisation échoue, c'est à cause de la similitude ou la proximité de l'histogramme de couleur de la cible avec d'autres échantillons. Nous avons aussi montré les performances de suivi par rapport à la taille de la cible, la vitesse de la cible et la moyenne des taux de trame de la cible.

Conclusion

Dans ce travail, le suivi d'un objet est appliqué au visage humain et au haut du corps. Il peut être utilisé pour obtenir des images du visage d’un humain dans différentes poses. Dans notre travail, nous considérons le problème de grands mouvements inter-trame par les cibles, le faible taux d'images utilisables, des modifications à l'arrière-plan et les changements d'échelle. En outre, l'algorithme de suivi gère le temps de réponse de la caméra et le zoom.

La contribution générale de cette thèse est un suivi dynamique du visage avec une caméra PTZ IP, une première dans le domaine de la vision par ordinateur.

Les contributions spécifiques de cette thèse sont:

- Modélisation et formulation du problème par une boucle de commande avec rétroaction qui assure et compense tous les retards dus aux communications avec la caméra ou aux déplacements de celle-ci. Pour modéliser le système, ses caractéristiques sont déterminées d'un point de vue de classification des systèmes.

- La proposition d'une méthode composée d'un filtre de particules et du flux optique comme solution pour le suivi d'objet et le contrôle de la caméra. La combinaison d'un filtre de particules et du flux optique pour générer des échantillons appropriés qui couvrent toutes les régions candidates possibles est une solution qui permet de tenir compte des grands mouvements de la cible et du faible taux de trames. En outre, elle permet de récupérer le suivi de la cible dans le cas d'occlusion ou de perte de suivi.

La détection automatique d'humains et leur suivi en contrôlant une caméra IP PTZ est un sujet nouveau qui fournit laisse entrevoir des capacités très utiles pour les applications où on souhaite une interaction homme-machine naturelle telle que l'identification de personnes, la surveillance vidéo en temps réel, et la vidéoconférence. L'utilisation d’une caméra PTZ permet de couvrir un large champ de vue en changeant le panoramique, l'inclinaison et le zoom.

Les applications de suivi du visage en vidéosurveillance sont très variées, depuis la surveillance du trafic à l'activité humaine en passant pas la compréhension des comportements.

Les travaux futurs pour faire suite à cette thèse sont l'optimisation du code, comme l'utilisation des GPUs pour accélérer les temps de traitement, et la modélisation plus détaillée des délais pour avoir une estimation plus exacte de ceux-ci. Il est possible de généraliser le système de suivi à d’autres objets selon l’application, telle que le suivi de voiture, et la surveillance du trafic ou des terrains de stationnement. En outre, de multiples caméras PTZ IP pourraient collaborer pour avoir une large couverture avec de meilleures performances et des capacités supplémentaires par rapport à l'utilisation des caméras fixes.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

PTZ	Pan Tilt Zoom
IP	Internet Protocol
APF-OFS	Adaptive Particle Filter using Optical Flow based Sampling
PF	Particle Filter
KLT	Kanade-Lucas-Tomasi
FOV	Field Of View
CCTV	Closed Circuit TV
ROI	Region Of Interest
PDF	Probability Density Function

CHAPTER 1. INTRODUCTION

1.1 Context

Moving object detection is one of the key technologies for intelligent video monitoring systems (Xiaopeng, Zhiqiang, & Yewei, 2006). The problem in video surveillance is how to identify and recognize the on-going events. People detection and tracking are important capabilities for applications that desire to achieve a natural human–machine interaction such as people identification.

Face tracking in video surveillance has a large range of purposes, from traffic monitoring to human activity and behaviour understanding. The applications of face tracking techniques by PTZ cameras can be summarized as automatically monitoring and identifying people in different places such as transportation stations, banks, houses, schools, laboratories, offices, industries, etc. So this project is related to two main research topics, first detecting the face and second tracking the detected face by controlling an IP PTZ camera.

Usually, video surveillance applications imply paying attention to wide areas thus different kinds of cameras are generally used, e.g. fixed cameras, omni directional cameras, mobile cameras, or PTZ cameras (Gagnon, Laliberte, Foucher, Branzan Albu, & Laurendeau, 2006; Lalonde et al., 2007; Micheloni & Foresti, 2006). In the area of security and surveillance, PTZ video cameras provide rich and useful information with low cost and high coverage in comparison with other technologies. Today, most available commercial surveillance systems work with stationary Closed Circuit TV (CCTV) cameras. But CCTV cameras provide maximum quality of 720 x 480 pixels with the resolution of low-end cameras being much lower with a limited FOV. This creates a problem for face detection, face recognition, people identification especially if the subjects are distant. PTZ cameras have the obvious benefit of being able to pan, tilt and zoom either manually or automatically. For manual operation, a PTZ camera can, for example, be used to follow a person in a retail store. PTZ cameras are often used indoor. The optical zoom ranges from 18x to 26x. The PTZ cameras have another benefit which is high speed wide angle pan and tilt functions.

A network camera can be described as a camera and computer combined into a single unit. It captures and transmits live images directly over an IP network, enabling authorized users to locally or remotely view, store, and manage video over standard IP-based network infrastructure. A network camera has its own IP address. It is connected to the network and has a built-in web server, FTP server, FTP client, e-mail client, alarm management, programmability, and much more. A network camera does not need to be connected to a PC; it can operate independently and can be placed wherever there is an IP network connection.

In addition, IP cameras surpass analog camera performance in some important areas: 1) end to interlace problems, 2) Power over Ethernet (PoE) increases saving and reliability, 3) Megapixel resolution, 4) intelligence at the camera level, 5) integrated PTZ and input /output control, 6) secure communication, 7) flexible, cost-effective infrastructure choices, and 8) lower total cost of ownership.

The combination of face detection with the properties of IP PTZ cameras would allow us to increase the abilities of automatic identification systems, especially in the field of video surveillance. Face detection and tracking determine the location and size of each human face for each output image of a video camera. This is the problem we want to deal with in this research. It implies controlling an IP PTZ camera to follow and detect faces of people and track them. After face detection and tracking, face recognition is often performed to distinguish the detected face from other faces. The advantage of object tracking using only one PTZ camera is removing the requirement of many stationary cameras from the surveillance system to reduce the cost while keeping a large coverage of the scene.

Section 1.2 will identify the problem in this project. Section 1.3 presents the objective and in section 1.4 our contributions will be discussed.

1.2 Problem identification

Accurate, efficient, and reliable tracking and classification of multiple objects is a challenging task in video surveillance due to image resolution, processing time, tracking accuracy, etc. These challenges are exacerbated with IP PTZ cameras. They can be summarized as:

- Irregular response time to camera control command;
- Low usable frame rate (while the camera executes a motion command, the frames received are mostly useless, as no new command can be sent without a delay);
- Irregular frame rate because of network delays (the time between two frames is not necessarily constant);
- Changing FOV resulting from panning, tilting and zooming;
- Various scales of objects.

Automatic face tracking by an IP PTZ camera is a new subject that requires capabilities such as:

- Detecting faces at different positions (face not necessarily in frontal view).
- Performing with various face scales.
- Covering a wide FOV (pan, tilt, zoom).
- Short processing time (for real-time usage).
- Applicability for low resolution images.
- Handling large inter-frame motion of target.

In this project, we propose an automatic face tracking method for video surveillance applications with IP PTZ cameras. It consists of initialization that is, the processing before any target representation and camera motion, as well as APF-OFS and camera control that are the processing after target modeling. Each part includes different steps. Initialization consists of background subtraction, skin detection, and face detection. The steps for APF-OFS and camera control are motion estimation, target candidate extraction, selection of best target candidate, and orienting the camera to center it on the target.

In the next section, we will discuss the general and specific objectives of our project.

1.3 Objectives

Face detection and tracking determines the location and size of each human face for each input image of a video sequence. Controlling the PTZ camera is required to track the face and record the events during a surveillance process. As discussed before, this project is related to two main research topics, first detecting faces at the initialization step, and second, in the tracking step controlling the PTZ camera to track and follow faces, and zoom on them for close-up views.

The general objective of this project is to automatically detect and track a face using an IP PTZ camera.

The specific objectives of the project are to:

1. Detect a human entering a room (target).
2. Model the target.
3. Track the target.
4. Control an IP PTZ camera to center on the target.
5. Zoom on the target face, verify the target face and capture an image of the target face.
6. Validate and quantify the performance of the implemented algorithms in a real-time application.

Human detection is done in three steps which are extraction of motion, skin detection, and face detection. Extraction of motion regions is performed by using a backgroud subtraction with high true positive rate and low false positive rate. Then, detection of skin regions is achieved by using a technique with high positive rate and processing to remove the false positives to keep only skin regions. To localize the desired face among the skin regions, Viola and Jones face detector is applied. To model the target, geometric and appearance-based features are calculated inside of a fitted ellipse over the torso and detected face. Then, in the tracking step, an adaptive particle filter with optical flow samples method is applied to find the target and center the camera on it. This implies that the camera should change its pan, tilt, zoom values according to the face motion. The PTZ camera used is an IP camera, so the transmission line to send and receive the commands is internet. The IP camera has the limitation of delays in receiving and executing commands, and for sending images. Controlling the IP PTZ camera to track the face is related to the tracking algorithm; so it is necessary to have algorithms with low computational cost to process the images and move the camera within a short period of time to keep the tracked face inside the FOV of the camera. Zooming on the target face is performed when tracking is good under some criteria. Then, the detected face should be verified as target face for capturing its image and stop the camera movement.

In video surveillance applications, there are a lot of different challenging situations like interactions between the mobile objects, occlusions, and many similar objects. Thus, the utilized tracking algorithm should have the capability of distinguishing the desired object from others. The proposed technique for face tracking is validated and its performance quantified with online and real-time video applications and by comparison with other algorithms.

1.4 Contribution

The impact of this research can be summarized as follows: Face tracking is a type of object tracking that has many applications in different fields of computer vision, surveillance, security, and face recognition. The general contribution of this thesis is dynamic face tracking with an IP PTZ camera, a first in the field of computer vision.

The specific contributions of this thesis are:

- Modeling and formulating the tracking program as a servo control loop with a communication feedback that assumes and compensates all the delays resulting from network or processing. To model it, the characteristics of system from a classification system view are determined.
- Proposing of an adaptive particle filter with optical flow samples method as a solution to track the object and control the camera. Combination of particle filter and optical

flow to generate appropriate samples that cover all possible candidate regions is a suitable solution that can handle large inter-frame motion of the target and low tracking frame rate. In addition, it is able to recover the tracking target in the case of occlusion or target lost. The combination of these two techniques is a novelty that results in extracting all possible candidates only where there is motion resulting from target motion and not camera motion.

CHAPTER 2. OBJECT TRACKING, STATE OF THE ART

The capability of tracking moving objects in a scene is a fundamental and crucial problem in many computer vision systems that include human detection/tracking in a video surveillance environment. In general object tracking, the initial step is to detect the object to initialize tracking and then follow the object. In the context of this work, object detection corresponds to upper body detection since we want to track the face of people with a PTZ camera. The main focus of this project is object tracking rather than object detection. Accordingly, in the following literature review, we will focus first on general tracking of objects, specifically on object tracking with PTZ camera and then on face and upper body tracking using PTZ cameras. During the review, object models needed for detection will be commented.

2.1 General tracking methods

Object tracking is a challenging problem and the complexity of object tracking will be increased due to (Alper, Omar, & Mubarak, 2006):

- Loss of information caused by projection of the 3D world on a 2D image;
- Noise in images;
- Complex object motion;
- Non rigid or articulated nature of objects;
- Partial and full object occlusions, object to object and object to scene occlusions;
- Complex object shapes;
- Scene illumination changes;
- Real-time processing requirements;
- Camera motion.

By utilizing some constraints on the motion and/or appearance of objects, object tracking will be simpler. Even some constraints such as constant velocity or constant acceleration of the object motion based on a priori information can be used. Number and size of objects, or the object appearance and shape, can also be utilized to simplify the problem.

From a recent survey (Alper et al., 2006), the main tracking methods are categorized as point tracking, kernel tracking, and silhouette tracking. In the following, an overview of each category is presented.

2.1.1 Point tracking

Detected objects are defined by points which might be the centroid of the object (Figure 2.1 (a)) or a set of points (Figure 2.1 (b)). Point detectors are utilized to find desired points in images. The current locations of the points are based on the previous object state which can include object position and motion. This approach requires another technique to detect the objects in each frame. Tracking of objects can be considered as correspondence across frames of detected objects represented by points. Point correspondence in the presence of occlusions, misdetections, entries, and exits of objects is much challenging because it is possible that an interesting point is hidden or missing. The methods of point correspondence can be divided into two categories: deterministic and statistical. In the deterministic methods, Veenman et al., (2001), utilize qualitative motion heuristics to constrain the correspondence problem (Veenman, Reinders, & Backer, 2001). By a set of motion constraints such as proximity, maximum velocity, small velocity change, common motion, rigidity and proximal uniformity, they assume a cost of associating each object in frame $t - 1$ to a single object in frame t for point correspondence. A combinatorial optimization problem is applied to minimize the point correspondence cost (Shafique & Shah, 2003; Veenman et al., 2001).

Object motion in the video frames can be affected by random perturbations like maneuvering vehicles and noise from video sensors. This type of problems can be solved by utilizing statistical approaches by taking the measurement and the model uncertainties into account during object state estimation. In statistical approaches, object properties like position, velocity, and acceleration are modeled by a state space approach. Object position in the image is determined by a detection technique, for example, by learning the mean shape (shape from points distribution) and dynamics of the shape change (Shaohua, Chellappa, & Moghaddam, 2003; Vaswani, Roy Chowdhury, & Chellappa, 2003). This problem is solved differently for single and multiple objects.

In the single object case, if the object motion is considered linear with Gaussian noise, the problem is solved by a Kalman Filter to estimate the state of a linear system (Rosales & Sclaroff, 1999). Prediction and correction are the two steps of Kalman filtering. The prediction step uses the state model to predict the new state of the variables:

$$\begin{aligned} \bar{X}^t &= DX^{t-1} + W \\ \bar{\Sigma}^t &= D\Sigma^{t-1}D^T + Q^t, \end{aligned} \tag{2-1}$$

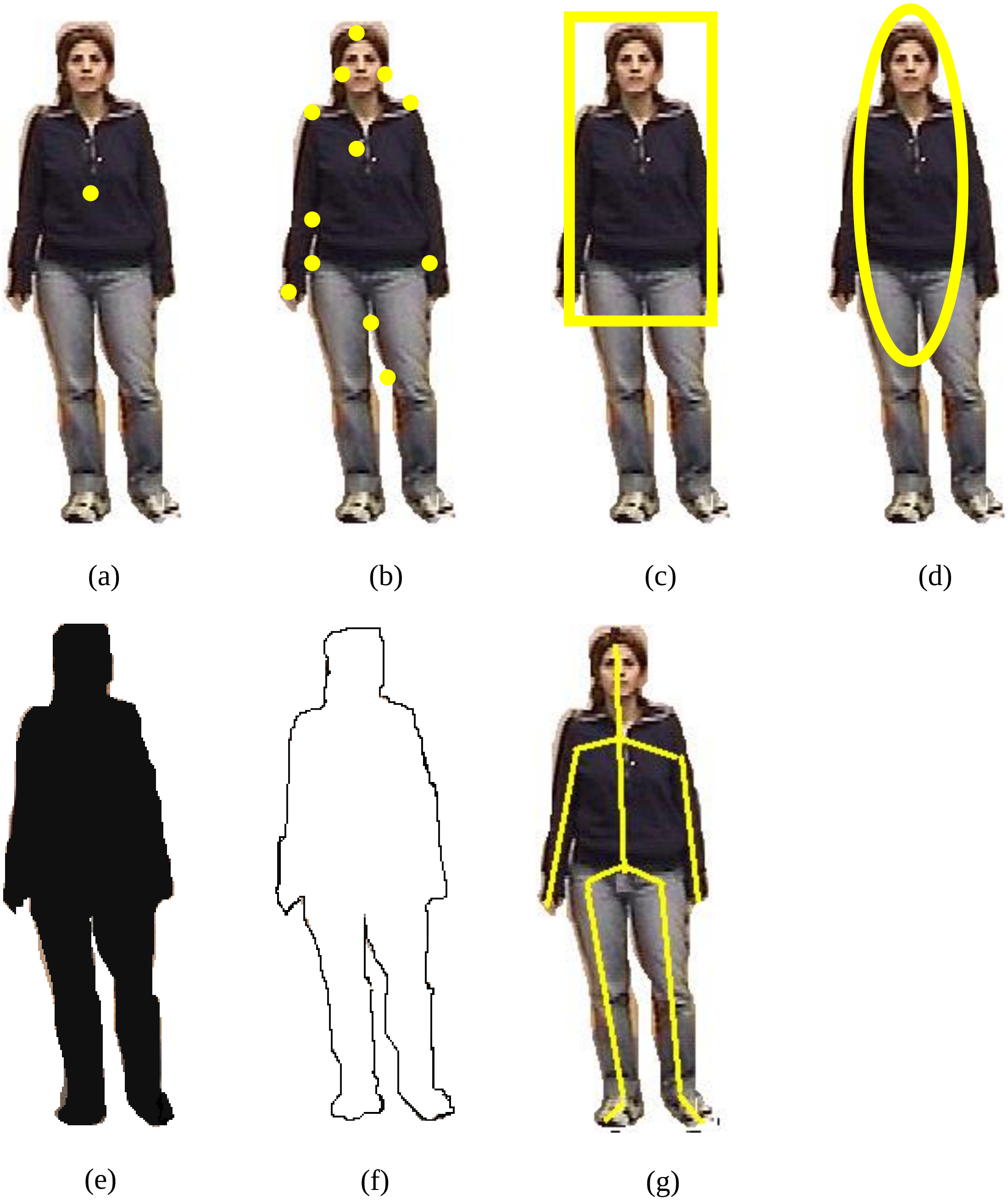
where $\bar{X}^t$ and $\bar{\Sigma}^t$ are the state and covariance predictions at time *t*. The relationship between the state variables at two consecutive time *t* and *t-1* is defined by *D*, the state transition matrix. *Q* is the covariance matrix of the noise *W*. To update the object's state at the correction step, the current observation Z^t is utilized:

$$\begin{aligned} K^t &= \bar{\Sigma}^t M^T \left[M\bar{\Sigma}^t M^T + R^t \right]^{-1} \\ X^t &= \bar{X}^t K^t \left[Z^t - M\bar{X}^t \right] \\ v &= K^t \left[Z^t - M\bar{X}^t \right] \end{aligned} \tag{2-2}$$

$$\Sigma^t = \bar{\Sigma}^t - K^t M\bar{\Sigma}^t \tag{2-3}$$

where *v* is the innovation, *M* is the measurement matrix, *K* is the Kalman gain.

Particle filters for state estimation is proposed in (Genshiro, 1987). The limitation of the Kalman filter is the consideration of a Gaussian distribution for state variables of object motion which is not the general case. Thus, the Kalman filter gives poor estimations of state variables in the case of non-Gaussian distribution. Tracking of human movement is non-linear and non-stationary, so the Kalman filter assumption makes false results in tracking.



(a) (b) (c) (d)

(e) (f) (g)

Figure 2.1 Object representation (a) centroid, (b) multiple points, (c) rectangular patch, (d) elliptical patch, (e) object silhouette, (f) object contour, (g) object skeleton.

To cope with this problem, particle filtering is utilized. $p(X_t \mid Z_t)$ is the conditional state density at time *t* and it is represented by a set of *n* samples $\{\ s_{t-1}^{(n)} : n = 1,..., N\ \}$(particles) with weights $\pi_t^{(n)}$ (sampling probability). The importance of a sample is given by its weight, that is, its observation likelihood.

The initialization of particle filter can be done by using the first measurement, $s_0^{(n)} \sim X_0$ with weight $\pi_0^{(n)} = \frac{1}{N}$ or by training the system utilizing sample sequence. Resampling is necessary to eliminate low weight samples, while keeping track of the best particles. It is not required to have the posterior density defined as a Gaussian function.

In the case of tracking multiple objects, before using the filters (kalman or particle), it is required to deterministically associate the most likely measurement for a particular object to that object's state. It is the correspondence problem that should be solved. Nearest neighbour approach is the simplest solution for correspondence. Although in the case of near objects the incorrect correspondence might occur. A false associated measurement can cause the filter to fail to converge. To cope with this problem, there are some statistical data association methods. Joint Probability Data Association Filtering (JPDAF) (Rasmussen & Hager, 2001) and Multiple Hypothesis Tracking (MHT) (Hue, Le Cadre, & Perez, 2002) are two widely used techniques for data association.

2.1.2 Kernel tracking

Kernel tracking refers to computing the motion of a region. The object motion is usually in the form of parametric motion such as affine, translation, etc. To compute the motion of a region, the object shape and appearance of that region are used as comparison (Figure 2.1 (c) and (d)). So the kernel can be a rectangular template or an elliptical shape with an associated histogram depending on the tracked object. Object tracking is performed by computing the motion of the kernel frame by frame. Parametric transformations like translation, rotation, and affine are usually applied to model the object motion. Object motion can also be in the form of a dense flow field computed in subsequent frames. The choice of the appearance model, number of tracked objects and motion estimation method give rise to different algorithms for kernel-based tracking. Kernel-based tracking methods can be divided in two categories based on the appearance representation used, namely, templates and density-based appearance models, and multi-view appearance models.

Template and density-based appearance models are often chosen because of their relative simplicity and low computational cost. The trackers in this category can be divided into two subcategories based on if objects are tracked individually or jointly. Template matching is a brute force method for searching the image to find a similar region to the template of the desired object. One example of this method is in (Avidan, 2001, 2004) which utilized a Support Vector Machine (SVM) classifier for tracking. SVM is a general classification scheme that, given a set of positive and negative training examples, searches to find the best separating hyperplane between the two classes to classify the data in two partitions. For SVM-based trackers, the positive examples are the desired objects to track, and the negative examples are the other parts of the image. Usually negative examples are the background regions. In Avidan's tracking method, maximizing of the SVM score to estimate the position of the object is utilized instead of minimizing the intensity difference of a template from the image regions. An advantage of this approach is the knowledge of background objects that is explicitly incorporated in the tracker.

Template matching is the most common technique for single object tracking. The best position of the template in the image is determined by a similarity measure, like cross-correlation. High computation cost due to the brute force search is the limitation of this method. To minimize the computational cost, some papers suggest limiting the object search to the vicinity of its previous position (Schweitzer, Bell, & Wu, 2002). In addition, color histograms or mixture models can be used instead of templates to model the appearance of pixels inside the rectangular or ellipsoidal regions to track. For example Comaniciu et al., (2003), proposed a weighted histogram obtained from a circular region for object representation (Comaniciu, Ramesh, & Meer, 2003). In their method, they use the mean-shift procedure instead of a brute force search for object localization. The mean-shift algorithm tries to find maximum similarity by iterative histogram comparison of the object and the window (search space) near the desired object location. The technique of (Comaniciu & Meer, 2002) is the extension of mean-shift tracking technique by utilizing a joint spatial-color histogram instead of using a color histogram. By comparing mean-shift tracking and template matching techniques, mean-shift has the obvious advantage of eliminating the brute force search. In addition, for the mean-shift tracking method, the translation of the object patch can be determined in a smaller number of iterations. However, for mean-shift tracking initialization, a part of the object has to be inside the search window. In addition, mean-shift tracking cannot track objects with large motion between consecutive frames.

Zhaowen et al., (2009), proposed a method for object tracking with an active background in videos recorded by a mobile camera (Zhaowen, Xiaokang, Yi, & Songyu, 2009). They apply camshift tracker (i.e. it is a kind of mean shift tracker) to estimate the next target position and apply particle filter to predict the scale changes of the target. Their method cannot handle target with large inter-frame motion.

Joe-Air et al., (2006) proposed multiple object tracking using segmentation and trajectory estimation for stationary camera (Joe-Air, Ying-Tung, Chuang, & Yen-Ling, 2006). One example of the applications is face tracking. General object tracking in this work is achieved by analyzing the movement of contours frame by frame in the video stream. The proposed algorithm contains three major parts for analyzing the shapes and motions of the objects in the video frames. First of all, a modified mathematical morphology edge detection algorithm is utilized to extract the contour features in the video frames. Then, a contour-based image segmentation algorithm is proposed and applied to the contour features for partitioning the predetermined target objects in the video frames. Partitioning is necessary to distinguish the desired object from others. Finally, to handle the movements of the objects, a trajectory estimation scheme is used. They use stationary camera for the face tracking and also the face in their implementation occupy a large region of the image. In addition, the initial position of the face should be provided manually.

Jepson et al., (2003) proposed a probabilistic appearance model to calculate the affine motion of the object for capturing stable object features (Jepson, Fleet, & El-Maraghi, 2003). The discussed approach compares only the appearance of a region such as an elliptical region to track it and find the orientation of the object. Tracking of complete regions of objects can be performed by analyzing the contour features in the video frames. Yezzi et al., (2001) proposed an approach which compares the contours of two images in two different views (Yezzi, Zollei, & Kapur, 2001). It combines the Mumford-Shah distance with 2D transformations for computing the contour transformations between two images. This method needs a predetermined and learned shape mask to do object tracking. In the discussed approaches, accurate extraction of desired objects in the image or video frames is not possible, and the tracking performances are limited.

Another approach for object tracking is to determine the translation of a region in consecutive frames by utilizing the optical flow method. In the optical flow method, the flow vector of each pixel under the brightness constancy constraint $I(x,y,t)-I(x+dx,y+dy,t+dt)=0$ is computed for generating dense flow fields. This computation is performed in the neighborhood of the pixel either algebraically or geometrically. The approaches based on optical flow (Galic & Loncaric, 2000; Youshan, Weijian, & Yingcai, 2003) depend on the fact that the motion of moving objects causes intensity changes in magnitude such that intensity changes are important cues for locating moving objects in time.

Shi et al., (1994) introduced the KLT tracker (Kanade-Lucas-Tomasi Feature Tracker) to iteratively determine the translation of a region centered on an interest point (Shi & Tomasi, 1994). In KLT, when the new location of a point is obtained (i.e. similar to optical flow), the region around that point is compared with the corresponding region from previous frame by computing the affine transformation between two consecutive frames.

In general, the background subtraction technique is used in tracking to discriminate mobile objects from the background scene. It assumes a static background. There are different background subtraction methods such as temporal averaging (Shoushtarian & Bez, 2005), Single Gaussian (McKenna, Jabri, Duric, Rosenfeld, & Wechsler, 2000), Gaussian mixture (Stauffer & Grimson, 1999), min max (Haritaoglu, Harwood, & Davis, 2000) and texture based (Heikkila & Pietikainen, 2006). Background subtraction techniques are only suitable for video streams which are captured by a stationary camera. Therefore,, when the camera is non-stationary, the background subtraction method is not useful. To overcome this problem, some advanced algorithms have been proposed (Comaniciu et al., 2003; Joe-Air et al., 2006) .

For multiple objects tracking Tao et al., (2002) proposed an object tracking method based on modeling the whole image (Tao, Sawhney, & Kumar, 2002). In this method a single background layer and one layer for each object are considered. This representation consists of object shape model, motion model and Gaussian background subtraction results. At each step, shape model and motion model of the foreground resulting from the background subtraction are compared using 2D affine transformation. Isard et al., (2001) suggested joint modeling of the background and foreground regions for tracking (M. Isard & MacCormick, 2001). Each of the background and foreground appearance is modeled by mixture of Gaussian function separately. The shapes of objects are represented as cylinders. The ground plane is considered to be known to determine the 3D object positions. Tracking is obtained by utilizing particle filters where the state vectors consist of the 3D position, shape, and velocity of all objects in the scene. In their method, a modified prediction and correction scheme for particle filtering is introduced to increase or to decrease the size of the state vector in the case of added or removed objects. The method can also cope with the occlusion between objects. In this method, the maximum number of objects in the scene is predefined. The other limitation of the approach is the use of the same appearance model for all foreground objects. In addition, training to model the foreground regions is required. In the method of (Bernardin, van de Camp, & Stiefelhagen, 2007), the upper body histogram information, KLT feature tracker, and active camera calibration are combined to track the person. It is used for 3D localization.

In the application of some object tracking techniques, sometimes face is used as a desired object. For example, Comaniciu et al., (2003) applied the mean-shift algorithm on face tracking (Comaniciu et al., 2003). They assume an elliptical region whose histogram is represented in the intensity normalized RG space with 128 x 128 bins. To adapt to fast scale changes they also exploit the gradient information in the direction perpendicular to the border of the hypothesized face region. The scale adaptation is thus determined by maximizing the sum of two normalized scores, based on color and gradient features, respectively. They also use background subtraction for their work assuming a stationary camera. This technique can be applied for real-time application. This method is not designed to cope with large inter-frame motion, as the target in the current frame needs to be near the previous position for the mean-shift procedure to work.

In (Kublbeck & Ernst, 2006) the combination of illumination invariant feature to detect the face is applied with a tracking mechanism for improving speed and accuracy. For face detection, the intensity of each point in the image is modeled by linear combination of luminance and reflectance values. The face is detected by the investigation of illumination invariant features of regions. Tracking is performed by means of continuous detection. They use stationary cameras.

The only assumption on the object to track is its maximal speed in the image plane. From this assumption, they derived three conditions for a valid state sequence in time, which are validity, initialization, and extrapolation. To estimate the optimal state of a tracked face from the detection results, a Kalman filter is used. In addition, the face should be in a frontal view and large in the image.

In the previous tracking methods, the appearance models (e.g. histograms, templates, etc.) are usually calculated online. Thus, these models describe the information gathered about the object from the most recent observations. Appearance models might have some problems in the case of multi-view or change of viewpoints during the tracking. To cope with this problem, different views of the object can be learned offline and applied for tracking. Black et al., (1998) introduced a subspace-based approach, that is, Eigenspace, to determine the affine transformation from the current image of the object to the image reconstructed by eigenvectors (Black & Jepson, 1998).

2.1.3 Silhouette tracking

Object tracking is done by estimating the object region in consecutive frames. In silhouette tracking methods, the appearance, density and shape models, which are usually in the form of edge maps, object contour and color histogram, are utilized to model the tracking object (Figure 2.1 (e), (f), and (g)). Either shape matching or contour comparison are applied for silhouette tracking for a given object model. Silhouette tracking is more appropriate in tracking of objects with complex shape, while in kernel tracking, a specific object shape (e.g. ellipse, rectangle, etc) is considered. Indeed, in the silhouette-based object tracking, object region in each frame is tracked by means of an object model determined from the previous frames. Color histogram, object edges or the object contour are models that describe the object. Silhouette trackers are divided into two groups: shape matching and contour tracking. In shape matching approaches, object silhouette should be searched in the current frame. Shape matching is done like tracking based on template matching or silhouette matching, where an object silhouette and its associated model is searched in the current frame (Baoxin, Chellappa, Qinfen, & Der, 2001). Another shape matching approach for finding correspondence is to match detected silhouettes in two consecutive frames. It is similar to point matching technique. Some other tracking techniques in this category such as utilizing Hough transform can be found in (Haritaoglu, Harwood, & Davis, 1998a, 1998b; Haritaoglu et al., 2000). In contour tracking approaches, an initial contour is transformed to its new current position in the current frame, by either utilizing state space models or direct minimization of some energy functional. As mentioned previously, contour tracking methods evolve a contour from frame to frame. Contour evolution requires that some parts of the object in the current frame have an overlap with the object region in the previous frame. Tracking by evolving a contour can be done using two approaches. The first one is to utilize state space models to represent the contour shape and motion (Yunqiang, Yong, & Huang, 2001). In the second one, the contour is directly evolved by minimizing the contour energy by applying direct minimization techniques like gradient descent (Bertalmio, Sapiro, & Randall, 2000; Yilmaz, Xin, & Shah, 2004).

Different object tracking methods that are used with stationary camera were explained. Since in this project, we are interested in tracking with PTZ camera, in the next section object tracking with PTZ cameras is discussed.

2.2 Object tracking with PTZ or moving cameras

Several works have been introduced with combination of PTZ camera with multiple PTZ or stationary cameras in a master-slave configuration to explore the FOV (Chung-Hao et al., 2008; Cindy, Collange, Jurie, & Martinet, 2001; Sangkyu, Abidi, & Abidi, 2004). The important part in the combination of a PTZ camera with other PTZ cameras or stationary cameras is finding the geometrical relationship between them. For this purpose camera calibration is an appropriate solution which is applied in (Everts, Sebe, & Jones, 2007; Lu & Payandeh, 2008; Yan & Payandeh, 2008). In (Bellotto et al., 2009) an architecture for multi-camera, multi-resolution surveillance system for person tracking is proposed. This system contains distributed static and PTZ cameras that are controlled by visual tracking algorithms with a central supervisor unit. There is a calibration step between cameras to know the geometrical relations between them. The moving object detection and tracking is done by static camera and the PTZ camera is commanded to follow the target from the tracking results of the static camera. The tracking algorithm is the method of (Roh, Kim, Park, & Lee, 2007) and will be explained in the following. Zooming is done based on the estimated depth of the person. The face of the person should occupy a fixed fraction of the FOV. After any pan-tilt command, they apply also Viola and Jones face detector (Viola & J. Jones, 2004; Viola & Jones, 2001) to initialize the region-based tracker.

Roh et al., (2007) proposed a contour-based object tracker and use face tracking as an application of their technique (Roh et al., 2007). The proposed method has been tested by selecting only tracked object boundary edges in two steps: First, omission of background edges considering edge motion; second, selection of boundary edges utilizing a normal direction derivative of the tracked contour. Accurate tracking is obtained by reducing irrelevant edges influence by only selecting boundary edge pixels using optical flow. In selecting boundary edges using the normal contour direction, the image gradient values on every edge pixel are calculated, and edge pixels with large gradient values are picked. In this method, the camera is moved manually, not automatically, and it just follows a pre-determined trajectory. In addition, in their work, the face region is large.

Viola and Jones face detector, which will be explained in the face detection part is also used in (Bagdanov, del Bimbo, & Nunziati, 2006; Yuan, Haizhou, Yamashita, Shihong, & Kawade, 2008) which will be explained in the following.

Bagdanov et al., (2006) proposed an active face tracking method using Viola and Jones face detector (Viola & J. Jones, 2004) to detect face regions with Shi-Tomasi feature point tracker to track the face (Bagdanov et al., 2006). They limit camera movement to only eight cardinal directions. They did a continuous zooming at a constant velocity for a pre-specified, constant amount of time. Consecutively Viola and Jones face detector (Viola & J. Jones, 2004) is also used as a face tracking scheme in eye tracking system with PTZ cameras (Chan, Oe, & Lin, 2007). Their method could not handle large target motion.

Furthermore, in the algorithm of (Yuan et al., 2008) each observer or observation model (e.g. a two frame template matching tracker is an observer) should be learned from different ranges of samples, with various subsets of features. An offline trained detector like a Viola and Jones face detector is used for object detection process. Their method needs a learning step that is based on model complexity which increases computation time. The method has limitations in distinguishing between different targets.

Ser-Nam et al., (2003) proposed an image-based pan tilt camera control method in multiple camera surveillance system (Ser-Nam, Elgammal, & Davis, 2003). To extract the foreground region (blob) that contains the movement of the target, they apply the adaptive background subtraction method in (Elgammal, Harwood, & Davis, 2000). The background subtraction did not support large background motions and is appropriate for small motions such as tree branches and bushes. These blobs are tracked by finding the correspondence between the blobs of two consecutive frames with the silhouette edge matching of people explained in (Haritaoglu et al., 1998a, 1998b, 2000). This silhouette tracker is sensitive to the background subtraction results. A Kalman filter is used to improve the tracking results and predict the blob location of the objects. They will then map this position to the other camera to pan and tilt. To know the geometrical relationship between multiple cameras they did not use camera calibration. Instead they apply homographies to project the obtain trajectories from one camera (stationary camera) to the trajectories achieved from others (PTZ cameras).

Kalman filter is similarly used in (Ahmed, Jafri, Shah, & Akbar, 2008; Kwang Ho, Dong Hyun, Sung Uk, & Myung Jin, 2005; Sommerlade & Reid, 2008; Tordoff & Murray, 2007) which are explained in the following. The problem in Kalman filter is poor estimation of state variables in the case of non-Gaussian distribution for state variables.

Kwang Ho et al., (2005) did a multi view face tracking method. Stationary cameras are used for this work (Kwang Ho et al., 2005). Tracking is performed by estimating the position in the next frame of critical rectangular features on a face using a Kalman filter. Rectangular features are selected using Adaboost and they are modeled using a color model. This type of technique needs face images to be in a large enough scale.

Tordoff et al., (2007) proposed a camera zoom control method (Tordoff & Murray, 2007). A Kalman filter based on the object speed is applied in this method to track the object. Zoom control is done by considering that the image measurement error under zooming variation or scale changes are fixed. Indeed zooming is performed based on the measured noise level of Kalman filter. However, there is a delay in their feedback loop between the noise process and zooming which causes a lag.

Sommerlade et al. (2008) presented a scene exploration method that is combined with zoom control (Sommerlade & Reid, 2008). Their goal is to zoom on the object to obtain high resolution images. They minimize the uncertainty of object location using a Kalman filter tracker. They model the background by a Poisson process that learns its parameters from extended scene observation. They assume a simple constant- velocity target motion. For the tracking part they apply the method of (Denzler, Zobel, & Niemann, 2003) that is focused on 3D object tracking. In the experiments, they mentioned that evaluation of their algorithms on live video data is difficult and thus, they tested their algorithm on pre-recorded videos, without any real-time camera zoom control.

Ahmed et al., (2008) proposed an edge based tracking method, which compares the correlation of the target model edge with image edge to find the target (Ahmed et al., 2008). They predict the next target position using a Kalman filter.

Krahnstoever et al., (2008) designed a real-time control system of active cameras for multiple cameras surveillance system (Krahnstoever, Yu, & Lim, 2008). Object tracking is done by stationary cameras using a shape-based method (Krahnstoever, Tu, Sebastian, Perera, & Collins, 2006; Tu et al., 2007) which detects and compares the human body shape in consecutive frames. The cameras are calibrated with a common site-wide metric coordinate system described in (Collins, Lipton, & Kanade, 1999; Krahnstoever & Mendonca, 2005). Thus, the obtain target coordinate are transformed to the appropriate pan and tilt values using geometrical transformation and the camera will move according to the adjusted pan and tilt values. Similarly to previous works they are using tracking methods that are appropriate for stationary cameras. Also the multiple camera control is different from our subject.

Elder et al., (2007) proposed a face tracking method for wide FOV (Elder, Prince, Hou, Sizintsev, & Olevskiy, 2007). It uses two cameras, one a stationary, preattentive, low resolution wide FOV camera and the other is a mobile, attentive, high resolution narrow field camera. Firstly, the face is detected with the low resolution camera and then the attentive camera will track the face. Their algorithm for the head detection has three steps: skin detection, motion detection and foreground extraction. The advantage of their work is wide FOV of the camera but they have to make a feedback communication for the two cameras. This feedback is used for confirmation of face detection which takes time to do. They implement their algorithm for the real-time application but for IP cameras the delay of the network need to be considered.

Funahashi et al., (2004) developed a system for human head and facial parts tracking by a hierarchical tracking method using a stationary camera and a PTZ camera (Funahashi, Fujiwara, & Koshimizu, 2004; Funahashi, Tominaga, Fujiwara, & Koshimizu, 2004). At first, irises are recognized from motion images. Then, detected irises are used as feature for face detection. The face needs to be large enough to detect the irises. The PTZ camera is used to track the human head by detecting the face from stationary camera.

In (Micheloni & Foresti, 2005) object tracking is addressed by first performing moving object detection and then continuous zooming on the detected object if it is located inside a rectangle without change of pan and tilt values. In the first phase, a set of features of the moving target is tracked using Shi-Tomasi-Kanade method (Birchfield, 1997; Shi & Tomasi, 1994). Indeed, it clusters the background features from moving object features. In the second phase, the camera starts to zoom on the moving target. Their method fails in different tracking conditions with more than one moving object in a scene, or if the target gets out of FOV of the camera. Yao et al., (2006) proposed a scale estimation algorithm of 3D object for zooming with linear solution based on affine projection model (Yao, Abidi, & Abidi, 2006; Yi, Besma, & Mongi, 2009). For the tracking part, they apply the dynamic memory method in (Matsuyam, Hiura, Wada, Muease, & Toshioka, 2000). In (Matsuyam et al., 2000) pan and tilt angles are obtained from active background subtraction method. Indeed, they prepare a recorded background images database in a dynamic memory that contains all the background images in different pan and tilt values. Then, they interpolate the image that they want to process with the image database to obtain the approximate pan and tilt values corresponding to current image. The camera is controlled using a PID controller.

Schreiber, (2008) proposed a histogram-based tracking method, which is a generalized template matching Lucas-Kanade algorithm (Schreiber, 2008). Using histograms help to overcome some limitations of kernel-based methods. There is only one example of active background where the camera is installed in a car and tries to track a car that is in front. In this video there are only scale changes, and no panning or tilting is used.

Venkatesh babu et al., (2007) combined two tracking methods which are sum-of-squared differences (SSD) and color-based mean-shift (MS) tracker to overcome their separate shortcomings (Venkatesh babu, Perez, & Bouthemy, 2007). SSD method compares the object appearance frame by frame by minimizing the intensity sum-of-squared differences of the objects in two consecutive frames like a block matching process. MS tracker overcomes rapid model change in SSD tracker, and large displacements handling is done by SSD tracker. They tested their method in 30 fps recorded video sequences with PTZ camera and their problem did not involve the PTZ control elaborations. As we will discuss in the following chapters, in our problem because of the network delay and camera control, we cannot assume high frame rate.

In (Changjiang, Duraiswami, & Davis, 2005), a camera based position tracking system (PCTS) for person tracking is presented. The centroid of human is calculated from the centroid of the detected foreground. It is only based on motion detection. There is no other feature comparison. Their method fails in different tracking conditions with more than one moving object in a scene. Indeed, there is no object segmentation. Similarly, (Chu-Sing, Ren-Hao, Chao-Yang, & Shou-Jen, 2008) utilizes opencv Camshift tracking algorithm for face tracking. In their method, only the tracking of one person is studied. The methods of (Chu-Sing et al., 2008) fails in different tracking conditions with more than one moving object in the scene.

Sangkyu et al., (2003) used a geometric transform-based mosaicing method for person tracking by a PTZ camera (Sangkyu, Joonki, Andreas, Besma, & Mongi, 2003). For each consecutive frame, it finds the good features for the correspondence and then tries to shift the moved image and update the changed background. They are using a high cost background modeling using a calibration scheme, which is not suitable for tracking by internet-based PTZ cameras.

The algorithm in (Leichter, Lindenbaum, & Rivlin, 2008) works by maximizing the Probability Density Function (PDF) of the target's bitmap, which is formulated by the color and location of pixel at each frame. This information allows color and motion to be treated separately. Tracking is based on three assumptions: color constancy, spatial motion continuity, and spatial color coherence or similarity of objects between two consecutive frames. Severe occlusions and large inter-frame motion are not handled, and this algorithm is not very fast.

In several papers, a contour-based person tracking is proposed (Murray & Basu, 1994; Yilmaz et al., 2004). It is assumed that the background motion between two consecutive images can be approximated by an affine transformation. Therefore, an image mapping is done between two consecutive images. The motion is detected by subtracting two consecutive images and some morphological operations are applied to remove a portion of the noise. Their methods are limited to small motion tracking and are time consuming for IP camera. In addition, they can track only continuously moving edges and cannot track temporarily stopping objects (Araki, Matsuoka, Takemura, & Yokoya, 1998).

Available object tracking techniques with PTZ cameras were explained. In the next section, we focus more specifically on face detection methods.

2.3 Face detection

Many works on face and upper body tracking have been reported. However, most of them have difficulties in finding the initial position and size of a face automatically. The challenges associated with face and facial feature detection can be attributed to the following factors (Phimoltares, Lursinsap, & Chamnongthai, 2007):

- Intensity: There are three types of intensity: color, gray, and binary.
- Pose: Face images vary due to the relative camera-face pose (frontal, 45^0, profile), and some facial features such as an eye may become partially or completely occluded.
- Structural components: Facial features such as beards, mustaches, and glasses may or may not be present.
- Image rotation: Face images vary for different rotations.
- Poor quality: Blurry images, distorted images, and images with noise that becomes unusual.
- Facial expression: The appearance of faces depends on a person facial expression.
- Occlusion: Faces may be partially occluded by other objects such as hand, scarf, etc.
- Illumination: Face images vary due to the position of light source.

In the context of our work, some of these challenges will have to be dealt with. They are pose, structural components, image rotation, poor quality, facial expression, occlusion and illumination. Hence, the method needed for this task should perform well for these difficulties.

There are different classifications of face detection techniques. Early division of face detection approaches can be categorized as using correlation templates, deformable templates and image invariants (Sung & Poggio, 1998). The correlation templates method calculates a difference measurement between a fix target pattern and candidate image locations. Thresholding over the output is used to achieve matches (Sim, Sukthankar, Mullin, & Baluja, 2000). However, the class of all face patterns is too varied to be modeled by a single correlation template. View-based eigenspaces is a related approach to correlation templates (Pentland, Moghaddam, & Starner, 1994). In this approach, it is considered that the set of all possible face patterns occupies a low dimensional linear subspace, within a high dimensional vector space of all possible image patterns. The face image is detected if its distance from the subspace of faces is below a threshold. This approach has only been implemented on images with little background clutter. Kim et al., (2003) combined a face region tracking method with a face recognition technique for human identification in intelligent surveillance system (Y. O. Kim et al., 2003). For the face tracking part which is, in fact, consecutive face detection, they apply color skin detection technique to extract the region of interest (ROI). Then, they use template matching to compare the detected skin regions with face model. Their method requires face to be large enough and visible by the camera.

Deformable templates are like classical correlation methods with some built in non-rigid component. This approach uses parameterized curves and surfaces to model the non-rigid elements of faces and facial sub-features such as the eyes, nose, and lips. The parameterized curves and surfaces are fixed to a global template frame to permit variations in position between facial features. The matching process aligns the parameterized curves and surfaces to their corresponding image features (Sangkeun & Hayes, 2004).

In image invariant techniques, specific spatial image relationships common and unique to all face patterns are considered even in the case of different imaging conditions. For face detection, it has to compile a set of image invariants and search the image for the regions where they occur. Image invariants scheme is based on a set of observed brightness invariants between various parts of the face (Walker, Cootes, & Taylor, 1998).

In these three methods, it is necessary to have the face in a simple pose. It means that the face in the image is clear and large enough to be detected. Also the face should be in a frontal view in the image.

One of the most used and robust face detection techniques that is implemented by OpenCV library (Intel open vision library, 2008, Online accessed 15-February-2010), is the Adaboost face detection method proposed by Viola and Jones (Viola & J. Jones, 2004; Viola & Jones, 2001). They extract the Haar-like features that contain the information of image frequency. Then a set of key features from extracted features are selected as a cascade of classifiers. These features are robust to illumination changes and various face colors.

In recent survey, there are five categories of face and facial feature detection (Phimoltares et al., 2007).

1. Geometry-based methods. Geometrical information is used in these methods. Each feature is modeled as a geometrical shape. For example, face is detected as an ellipse shape (Zhang & Liu, 2005). Although an accurate detection of the face and facial features is possible, large variations of the face images cannot be handled. For example, in the case of images with some occluded facial features and images with noise, these variations are not handled.

2. Appearance-based methods. These methods utilize face models learned from a set of face training images. The algorithm has a chain structure to construct and it has a boosting step. Indeed some boosting classifiers are applied to this technique to detect the face. Most of the time, they use Adaboost for this part. Gray value (intensity) has the most important role for the detection (for example (Viola & J. Jones, 2004; Viola & Jones, 2001)). Face detection in images with low quality and occlusion is difficult.

3. Motion-based methods. Face and facial features are detected from an image sequence. First the motion is detected by backgrounds subtraction and then a hierarchical

segmentation technique should be used to segment the face parts (Lievin & Luthon, 2004). Then the parts are labeled. Face features such as lips and eyes are labeled and used for face detection. By using such methods, facial features cannot be detected using only a single still image from one view.

4. Edge-based methods. In this class of methods, faces are detected without information about intensity and motion. The edge information is used as input (Phimoltares, Lursinsap, & Chamnongthai, 2002). These methods can handle large variations of the face images. These methods detect face by using the face edges or using facial edge features.

5. Color-based approaches. These approaches face difficulties in robustly detecting skin colors in the presence of complex background and different illuminations. These algorithms are applicable only for color images (Liu, Yang, & Peng, 2005). This approach not only enables fast localization of potential facial regions but also proves to be highly robust to geometric variations in the face patterns. For this reason, face tracking in color images through the detection of skin colored regions has always gained special attention. However, the success of this approach depends heavily on the accuracy and robustness of the human skin color model. Also, it suffers from the need of a training stage (Kwang Ho et al., 2005).

In all types of face detection techniques (except the color-based approaches) that are explained in the above, the face in the image should be clear, large enough and in a frontal view to be detected. Most of these methods rely on finding facial features that might be too small to detect in the context of our work. These types of face detection are useful for face recognition applications because in most of them, they assume that the face location or at least the initial face position is known. In addition, these categories of techniques require having a lot of models for the various faces. In video surveillance applications, the face can be in any view, with different scales and sizes. In addition it can have poor resolution and be not clear enough. So these methods are not appropriate for video surveillance applications.

The color-based approaches are based on skin colors and skin classifiers. Their main feature is that they are learning the models from training data at all possible locations, scales and orientations with different type of skin colors. However, these methods are not robust to change of illumination, but they can cope with them if used in an indoor environment or in an environment with less changes of illumination. The performance of these methods depends on the size of the training data set.

2.4 Discussion on literature review

In the previous sections, recent object tracking methods, with stationary and PTZ or moving cameras, and face detection methods were presented.

Among the general object tracking methods, point trackers and kernel trackers with modifications are appropriate solution to our problem. Silhouette trackers in our problem have difficulties due to the IP PTZ system set up and image data that require robust algorithms to extract contours precisely.

A qualitative comparison of kernel trackers can be obtained based on tracking single or multiple objects, ability to handle occlusion, and type of motion model in our problem. Object trackers, based on the gradient ascent (descent) approach (e.g. mean-shift tracker), require that some part of the object is at least visible inside the chosen shape whose location is defined by the previous object position. In our problem since large inter-frame motion due to the low frame rate is occurring, to eliminate such requirements, a possible approach is to use Kalman filtering or particle filtering discussed in the context of point trackers to predict the location of the object in the next frame. The limitation of the Kalman filter is the consideration of a Gaussian distribution for state variables of object motion which is not the general case. In the context of human tracking which has non-linear and non-stationary movement, Kalman filter under the assumption of Gaussian distribution for noise and linear relation for system dynamics, is not a good solution. Instead, particle filter can cope with this limitation since it can distribute well samples at each frame in different positions and various scale sizes everywhere. Thus, with a particle filter more possible candidate target positions can be investigated.

According to the type of motion model, which is large motion between two frames, optical flow and KLT method can be used to model the motion but they cannot be used alone since they are not robust enough methods for moving camera.

In the study of surveillance system using multiple cameras, combination is done by a master-slave configuration in which detection and tracking is done by stationary cameras to explore the FOV. The important part in the combination of a PTZ camera with other PTZ cameras or stationary cameras is finding the geometrical relationship between them. For this purpose, camera calibration is an appropriate solution and has its own elaboration. In these typical works, a system contains distributed static and PTZ cameras that are controlled by visual tracking algorithms with a central supervisor unit. There is a calibration between the cameras to know the geometrical relationship between them. The moving object detection and tracking are done by static camera, and the PTZ camera is commanded to follow the target from the tracking results of the static camera. The principal challenge by this system is obvious when the number of tracking subjects is larger than the number of PTZ cameras. In such a case the PTZ camera controlling will be non-trivial. In these works, since many cameras share the same FOV, the PTZ camera used to follow the target does not need to predict its position. It just moves based on the position of the target in other camera's FOV. Thus, it is a different problem from what we are studying, where the PTZ camera must track a target without any other information than the one it can extract from its own FOV. The advantage of object tracking using only one PTZ camera is removing the requirement of stationary cameras to reduce the cost of the surveillance system, while keeping a large coverage of the scene.

Some tracking methods examine human body tracking or face tracking as an application of their tracking methods. People tracking and people detection in general is difficult for several reasons (J. B. Kim & Kim, 2003): 1) there are multiple moving objects, 2) objects of interest are usually small and poorly textured, 3) illumination conditions may be poor and change rapidly, and 4) shadows and multiple occlusions exist.

Robust and applicable face tracking algorithm must at least satisfy the following conditions:

1. Able to automatically detect the initial location and size of a face and track it in complex background.
2. Insensitive to face orientation, scale changes, and partial occlusions.
3. Able to distinguish the desired face from other faces or objects in various conditions and positions.

4. Computationally efficient for real time execution.
5. No false face should be detected.
6. All the faces should be found automatically without the requirement of adjusting the parameters in the process.

The general problem for all of the techniques is the limitation of the scale of face from frontal view. In addition most of them use stationary camera. In one case optical flow is used for non-stationary camera. Kalman filter that is utilized in some techniques for tracking is a special case with Gaussian function model for motion state which is not general. A Kalman filter cannot handle well sudden change of direction by the target. So it is not always appropriate for real-time and surveillance applications. Mean-shift has difficulty in target large inter-frame motion handling.

Available face tracking techniques are limited in their capabilities. In addition, they are applicable in some special conditions with some specific constraints (e.g. related to the application, FOV, face view, face size, etc.).

In summary the characteristics of available face tracking methods are:

- Used for special positions of face and in most cases in a large region, in frontal view at a very short distance or at least with a high resolution and good quality zoom on the face.
- Not necessarily fast.
- High computational cost with large amount of datasets.
- Applicable only for limited application like face recognition application, not real-time application.
- Have been used with stationary camera having limited FOV.
- Cannot handle large inter-frame target motion.

As discussed above, for any object tracking, firstly, it is necessary to detect the desired object. In general face detection techniques, the face in the image should be clear and large enough in frontal view to be detected. These types of face detection methods are useful for face recognition applications because in most of them they assume that the face location or at least the initial face position is known. In addition, these categories of techniques require having a lot of models for the various faces. Viola and Jones face detector is a robust and powerful feature based method that if it is learned with good dataset then it may be an appropriate solution for face detection.

CHAPTER 3. METHODOLOGY

In this chapter, the methodology that is used in this project is explained. In the first part, system architecture and an overview of the method is presented. Then, the different elements of the system, which are the main blocks of the method such as initialization, object tracking, position prediction, and camera control are described.

3.1 System architecture

We consider our PTZ camera as a servo control system. Indeed, any machine or piece of equipment will contain one or more servo control systems to control precisely the motion of each of its moving parts. The servo controlling and tracking system is modeled by a closed-loop control that has a negative feedback as shown in Figure 3.1. From system classification point of view, the system is discrete, stable, time variant, causal, dynamic, and nonlinear. These properties help us in modeling the system. We can use the characteristics of each category to represent our system model. For example, the system is causal, and thus, it means that our system response at time *t*, depends only on values occurring now or that occurred in the past. This causality information helps us to predict the current situation. Also, the system is discrete, which means that we do not capture continuously, and there is a time delay between each image. This is modeled by a delay block τ_1. The system is stable, which means that for finite input (pan-tilt-zoom values), there is finite output (obtained pan-tilt-zoom values). The system is obviously time variant and varies according to the scenario, processing time, traffic network delay, and other events that are occurring. The proposed method makes the system non deterministic which means that for the same input images or angles, the output may be different because of the random sampling of tracking process. It is a dynamic system that is constantly changing. A static system is a system in which its value depends only on the present value of input and neither past nor future has effect on its current value.

Servo system consists of five main blocks which are initialization, image capture, object tracking, position prediction, and camera control. Initialization extracts the target at the initial pan-tilt angles, (θ_0, φ_0), and zoom (ξ_0) and models it, as the input M (θ_0, φ_0, ξ_0), of an Adaptive Particle Filter using Optical Flow based Sampling (APF-OFS).

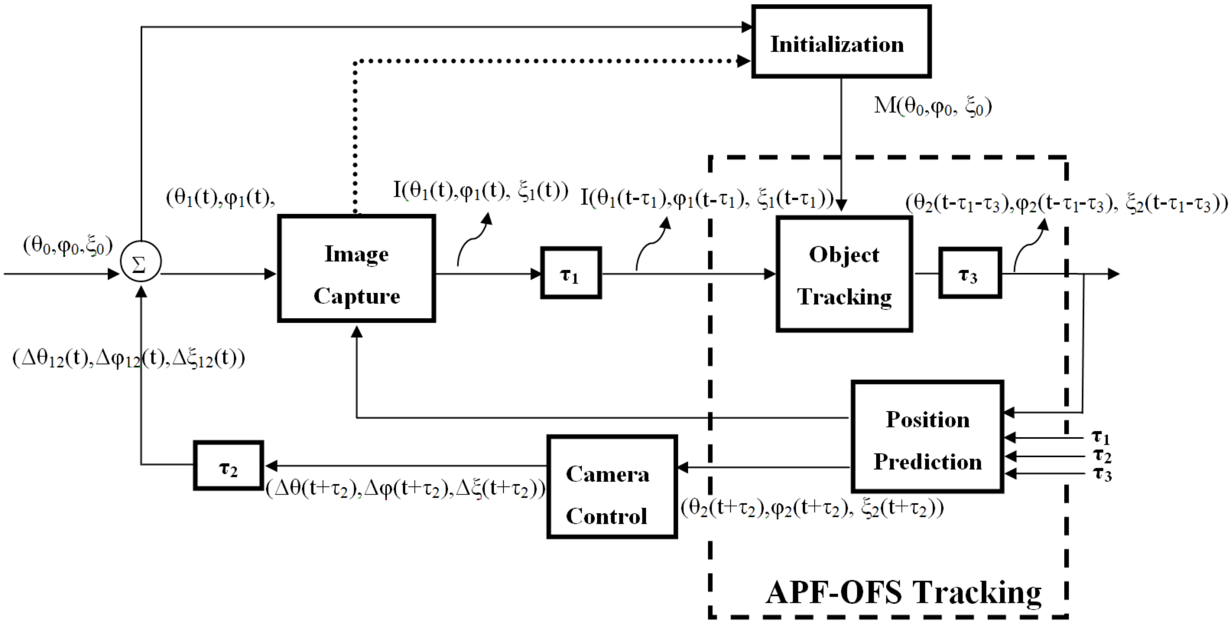

Initialization
M(θ_0,φ_0, ξ_0)
(θ_1(t),φ_1(t),
I(θ_1(t),φ_1(t), ξ_1(t))
I(θ_1(t-τ_1),φ_1(t-τ_1), ξ_1(t-τ_1))
(θ_2(t-τ_1-τ_3),φ_2(t-τ_1-τ_3), ξ_2(t-τ_1-τ_3))
(θ_0,φ_0,ξ_0)
Σ
Image Capture
τ_1
Object Tracking
τ_3
($\Delta\theta_{12}$(t),$\Delta\varphi_{12}$(t),$\Delta\xi_{12}$(t))
Position Prediction
τ_1
τ_2
τ_3
τ_2
($\Delta\theta$(t+τ_2),$\Delta\varphi$(t+τ_2),$\Delta\xi$(t+τ_2))
Camera Control
(θ_2(t+τ_2),φ_2(t+τ_2), ξ_2(t+τ_2))
APF-OFS Tracking

Figure 3.1. The system architecture and servo control model. (θ_0,φ_0) : initial pan-tilt angles, ξ_0: initial zoom value, ($\Delta\theta_{12}$(t),$\Delta\varphi_{12}$(t),$\Delta\xi_{12}$(t)) means (θ_1-θ_2, φ_1-φ_2, ξ_1-ξ_2), M(θ_0,φ_0, ξ_0): target model, I(θ(t),φ(t), ξ(t)): captured image at (θ(t),φ(t), ξ(t)), $\sum$: sum of inputs.

APF-OFS is a particle filter tracker that is combined with optical flow and it is adapted to our tracking problem. Adaptive means that the tracking system can work in both stationary and moving camera conditions. APF-OFS consists of object tracking and position prediction as shown in Figure 3.1. APF-OFS is affected by three delays, which are the delay τ_1 from image capture, the delay τ_2 in the feedback loop from executing camera motion commands, and the delay τ_3 from object tracking. τ_1 depends on the network traffic but it is almost constant for a particular network traffic and is about 0.05*s* for a 640×480 pixels image. τ_1 is affected by a specified image capture action but τ_2 varies in the range of [0.71s, 2s] depending on the pan, tilt, zoom values (That is, the motion amplitude because of PTZ motor speed constraints) and network traffic. τ_3 is the processing time of the particle filter tracker combined with optical flow and varies according to the number of samples in the particle filter, size of samples, and number of moving pixels calculated from optical flow. The input of the system in each time step *t* is the current pan, tilt angles, (θ(t), φ(t)), and zoom (ξ(t)) of the camera. The output will be the determined pan, tilt angles and zoom by the object tracking block. The delays imply that the current position of the target cannot be used for centering the camera. To compensate for motion of the target during the three delays, a position predictor is designed inside the APF-OFS. The position predictor will estimate and predict the next target position according to previous target motion vectors and average of previous system delays that is calculated based on the three system delays and the target displacement.

Because of the network latency and the camera response, it is not possible to control the camera continuously. Camera commands are queued and performed one after the other. If a new command is sent and the queue is not empty, most probably this command will be applied too late, and target will be lost. In our servo control system, a new motion/zoom command is issued only if the queue is empty. By this way, good control of the camera is maintained, but the target may move during the last motion/zoom command. Thus, the target motion needs to be estimated to keep the target inside the FOV of the camera.

The following assumptions for our application of face tracking are considered:

1. People walk at a normal pace or fast with random motions, but do not run.
2. A wide FOV (approximately 48°) and scenes that are not crowded (max 2-3 persons).
3. Indoor scenes with minimum variation in illumination or light sources. We aim to handle the change in illumination in the case of a person near and far from light sources.
4. The people do not wear the same color or do not have the same appearance.

The algorithm of each block of the servo system control is explained in the following parts. First the initialization block is explained since before any object tracking it is necessary to detect the object. Initialization is used to model and represent the target for the tracking system. Then the proposed Adaptive Particle Filter using Optical Flow based Sampling (APF-OFS) method that is used to localize and track the target is explained. APF-OFS consists of two object tracking and position prediction blocks. Finally the camera control process to center the IP PTZ camera on the target with the moving, zooming and stopping criteria are described.

3.2 Initialization

Here, the initialization block shown in Figure 3.1 is explained. Initialization is performed before tracking to introduce the target to the tracker. In this section, the target modeling steps that are background subtraction, skin detection, face detection, ellipse fitting and feature extraction are first described. Then results on initialization are presented.

3.2.1 Target Modeling

To represent the target, it is first required to model it by some features, like edges, feature points, or color. One possibility is the method of (Dalal & Triggs, 2005). In their algorithm, a dense grid of Histograms of Oriented Gradients (HOG) is calculated over blocks of 16 × 16 pixels used as detection window. A linear SVM classifier is used to classify the human. This method is robust enough but it can only process images with 320 × 240 size at a low frame rate (1 *fps*). (Han, Zhu, Comaniciu, & Davis, 2005) have speeded up the algorithm of (Dalal & Triggs, 2005) to process 320 × 240 images at 5 *fps*. The problem with both methods is high computational cost. Also, HOG is sensitive to the viewpoint of the person, arms and legs position. The model should be learned for all possible viewpoints and poses of the human. In addition, it is required to process all edges or contours inside the whole image to extract the body edges.

For modeling the target, the features should be invariant, discriminative, and stable enough during the tracking. These constraints are dependent on the application and tracking scenarios. In our application, the tracking system is low frame rate, under about constant illumination, but with variability in the background because of pan and tilt changes, with scaling variations, and with large target motion variations. Although simple, a color-based model satisfies these requirements. Other models could be used with our methodology if more precise details need to be accounted for. However, the model must be robust to instantaneous large scale and viewpoint change because of the low frame rate. This is coherent with the rational of the work of (Leichter et al., 2008), which also deals with low frame rate.

Therefore, in our problem the target is represented by an ellipse that circumscribes the head and torso part of the human body. The following geometric and appearance-based features are used as the observation model to represent the target for our particle filter tracking:

1. Quantized normalized *HSV* color histogram with 162 bins (i.e.$18 \times 3 \times 3$).
2. Mean of *H, S* and *V* color components of *HSV* color space of all the pixels inside of the region.
3. Center coordinates of each sample or candidate ellipse.

The selected simple features are appropriate for low-resolution object modeling, robust enough to scaling variations, illumination changes, and fast to compute. Histogram quantization is used to reduce computation time and for an adequate level of precision; we use 18 × 3 × 3 = 162 bins color histogram recommended in (Sangoh, Online accessed 15-February-2010) which is 18 levels of Hue color component, 3 levels of Saturation color component and 3 levels of intensity color component of HSV color space. The histogram is normalized. The second feature is the mean of *H, S* and *V* components of *HSV* color space of all the pixels inside of the elliptical region.

The initial model *M* is obtained from initialization block discussed in the system architecture and consists of the steps shown in the diagram of Figure 3.2. Initialization is done automatically by extraction of the top part of the body (head and torso) of the person to track. It is a part that should always be visible, either when the person is far away or when it is close to the camera. Initially, in our scenario, the camera is stationary and looks at the entrance of a room. Background subtraction is used to extract the foreground region of the scene (Darvish Zadeh Varcheie, Sills-Lavoie, & Bilodeau, 2008, 2010). At the same time a Bayesian skin classifier is applied to distinguish the skin regions of the extracted foreground from non-skin regions (Bourbakis, Kakumanu, & Makrogiannis, 2007). Then, the Viola and Jones face detector is applied to detect the face among skin regions (Viola & J. Jones, 2004). If there is no face or skin, this process is continued until the system finds a face and captures the desired face and torso image. The torso is 1/3 of the total height of the body as considered in (Bellotto & Huosheng, 2007). We are using a rectangular region with the same width as detected face and height of 4 times longer than face length, to model the target. We fit an ellipse inside the bounding box of the selected region, and model the resulting elliptical region with the features. This is the initial model *M*. Figure 3.6 (e) and (j) shows elliptical region (torso and the head). We use an elliptical region because it better fits the shape of the head and torso. In the following, the steps used in initialization are explained in detail.

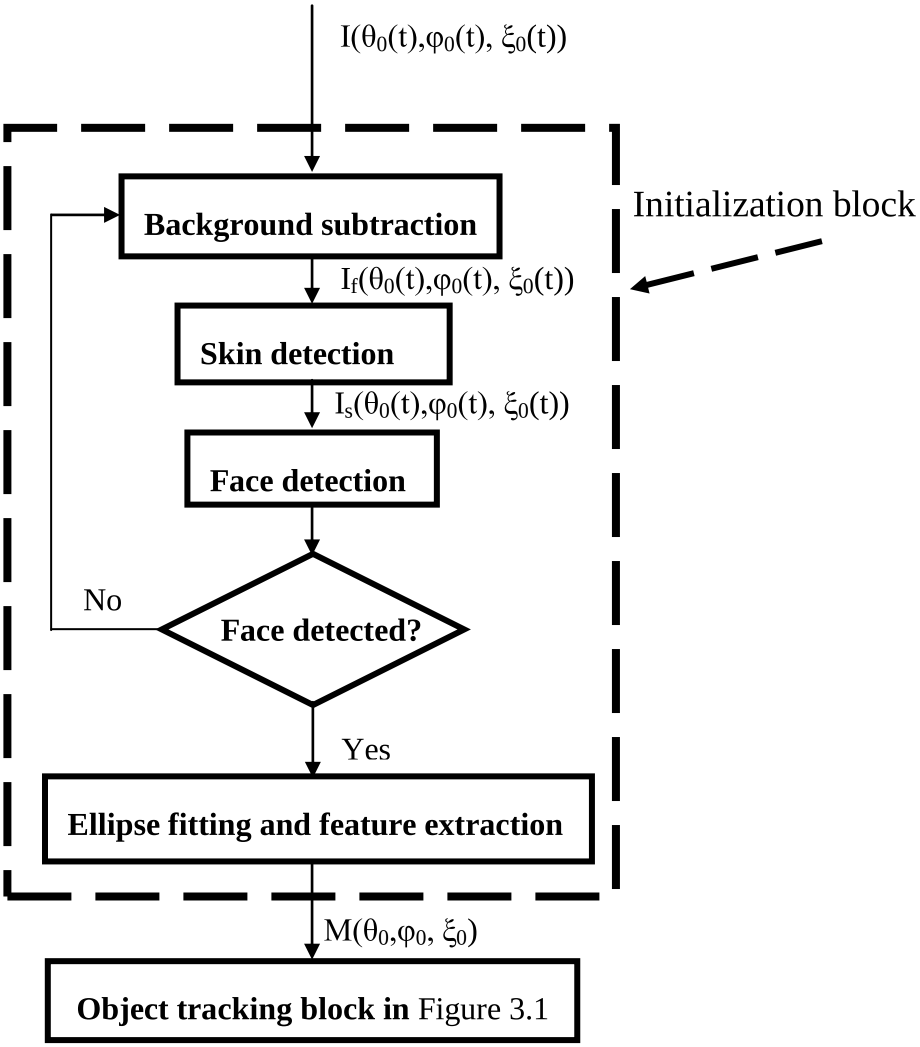


Figure 3.2 Principal steps inside initialization block shown in Figure 3.1.

3.2.1.1 Background subtraction

The PTZ camera is connected to the network and starts to capture the real-time video in a stationary initial position. The camera views, for example, the entrance of the monitored place.

For face tracking, the initial location of the face has to be known. We assume no camera motion when a subject first enters the FOV of the camera, so a background subtraction technique step can be used to distinguish the foreground region from the background in each image. There are different background subtraction methods that are available at our laboratory. For an environment with few skin color similarities and few lighting changes, TempAVG is appropriate. For scenes with lots of skin color similarities, RectGauss-Tex-MDPA is recommended to obtain less false positive rate (Darvish Zadeh Varcheie et al., 2008, 2010).

As all background subtraction approaches, the RectGauss-Tex-MDPA method, developed with collaborators at Polytechnique in another project, is composed of a background model that is regularly updated, and a similarity measure to compare a given frame with the background model. The background is modeled at different scales with color histograms of rectangular regions. Histograms are used to model the region, because they are robust to local noise and larger region histograms can be recursively built from the combination of smaller ones. This is not true for descriptors that account for pixel locations. Thus, histograms can be computed rapidly and filter noise. The current frame, in which motion is detected, is modeled in the same way. Motion is detected by comparing the corresponding rectangular regions from the coarsest scale to the finest scale. Comparisons are done at a finer scale only if motion is detected at a coarser scale.

After background subtraction, we will apply the Bayesian skin classifier over the extracted foreground.

3.2.1.2 Skin detection

Background subtraction has also some false positive regions that should be removed. A Bayesian skin classifier helps to reduce the false positive rate of the background subtraction results. A skin classifier can remove some part of the background colors that might contain non skin colors. This reduces the combined false positive rate.

By applying a Bayesian classifier and background subtraction, the skin regions of the image are extracted. Bayesian skin classification is trained in an offline process. Because skin detection is the preprocessing step of face detection, it is necessary to select an optimum solution to have high true positive rate and low false positive rate values. As found by our analysis, a Bayesian skin classifier is the best choice for the purpose of face detection because of the high true positive rates and simple implementation (Bourbakis et al., 2007).

A Bayesian skin classifier determines skin and non-skin probability functions from a labelled training data set of various skin and non-skin colors. Figure 3.3 shows the steps to obtain skin classification with the Bayesian skin classifier. This corresponds to the training of the Bayesian classifier. Skin detection will be done over the yellow, white, light brown, and pink color skin types from number 1 to 26 on Von Luschan's skin chromatic scale (almost 73% of all skin types in the world (Wikipedia, 2008, Online accessed 15-February-2010). For other skin types, since they are less discriminative other approaches might be required.

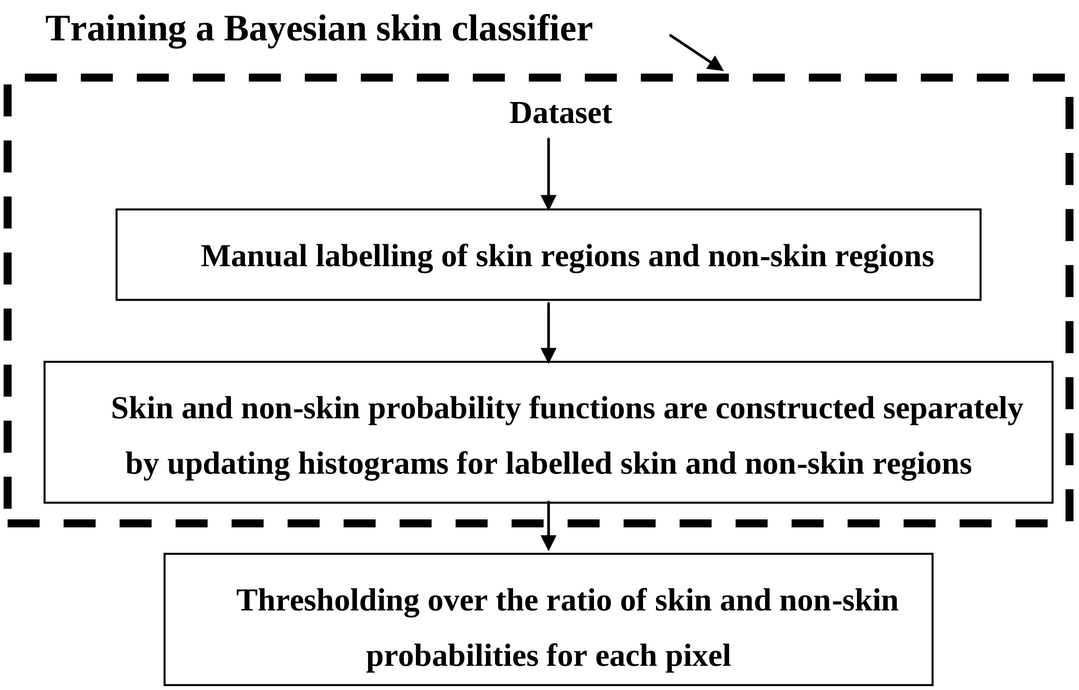

Training a Bayesian skin classifier
Dataset
Manual labelling of skin regions and non-skin regions
Skin and non-skin probability functions are constructed separately by updating histograms for labelled skin and non-skin regions
Thresholding over the ratio of skin and non-skin probabilities for each pixel

Figure 3.3 Steps for constructing a Bayesian skin classifier and Bayesian skin classification

Training is performed as an offline process. The skin and non-skin probabilities are determined from manual labelling of skin and non-skin regions from the images of the Caltech frontal face image dataset (Weber, 1999, Online accessed 8-March-2010). Then, from the labelled areas, the 3D color histograms of skin and non-skin pixels are calculated. The skin and non-skin probability functions are constructed separately by updating the 3D histograms for each image. Finally, by applying the Bayesian constraint and by using a threshold over the ratio of the skin and non-skin probability density functions, the skin classification can be performed.

$$\begin{cases} if \ \dfrac{p(z \mid \varpi_1)}{p(z \mid \varpi_2)} > \kappa & skin \\ else & non\text{-}skin \end{cases} \qquad (3\text{-}1)$$

where $p(z \mid \varpi_1)$ and $p(z \mid \varpi_2)$ are the class-conditional probability density functions of skin and non-skin color respectively. They are estimated in the off-line process as discussed before. From a large number of labelled skin and non-skin pixels in a training data set, normalized skin and non-skin histograms are calculated. κ represents the adjustable threshold (here κ=0). After training, the classifier is used in the online process. We remove all the skin regions that contain less than 40 skin pixels (less than the half of the minimum face size). However, the face must be separated from other skin regions. To do that, we use Viola and Jones face detector to find the face (Viola & J. Jones, 2004). This is discussed in the next section.

3.2.1.3 Face detection

As discussed before Viola and Jones face detector is used to detect the face (Viola & J. Jones, 2004). Viola and Jones face detector is implemented on opencv (Intel open vision library, 2008, Online accessed 15-February-2010). The face detector is applied over the detected skin regions to detect the target face. In brief, the face detector consists of three main parts:

1. Integral image which is a new image representation that allows fast rectangle features evaluation as shown in Figure 3.4. It looks like to the summed area table utilized in computer graphics. It is computed for an image by applying some operations per pixel. The integral image at coordinate (x,y) is the sum of the pixels above and to the left of (x,y). Then Haar-like features of the face images at any scales or locations in the image are computed and evaluated. Rectangle features are sensitive to edges, boundaries, image texture, and other simple image structures. This provides an appropriate image representation for the classification step.

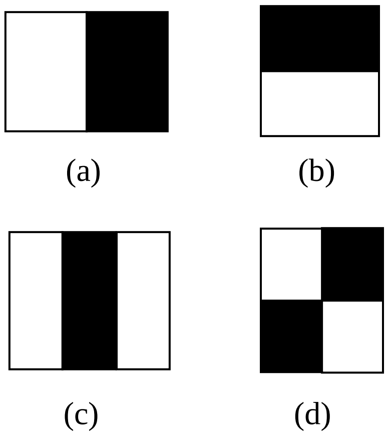


Figure 3.4. Rectangle features, the sum of the pixels on the white rectangles are subtracted from sum of the pixels in black rectangles shown in (a) two-rectangle features, (b) two-rectangle features, (c) three-rectangle features, and (d) four-rectangle features (Viola & J. Jones, 2004).

2. Adaboost classification of Haar-like features obtained from Haar basis functions or Haar filters (Papageorgiou, Oren, & Poggio, 1998). Here, Adaboost is used both in selection and training of features. It combines a set of weak classifiers to form a stronger classifier.
3. Combination of classifiers in a cascade structure to increase the detection speed. In each step of a cascade structure, the sub-windows are reduced. Indeed, a large number of false negatives are removed by adjusting the classifier thresholds with small processing time. This structure results in keeping only strong classifiers for the final face detection.

It is possible to retrain the face detector with more face images to make a more robust face detector. Here, we use the default face detector trained by OpenCV. The output of the face detector is a bounded rectangle on the face region.

The face detector is filtering the detected skin regions by extracting the face region even if there are false positives in the detected skin region results from similar skin color objects in the scene.

3.2.1.4 Ellipse fitting and feature extraction

After detection of the face bounding box, we build a rectangular region with the same width as the detected face and a height 4 times longer than the face length to model the target. We fit an ellipse inside the bounding box of the selected region, and model the resulting elliptical region with the features. To extract the pixels inside of the elliptical region from the selected rectangular region, a mathematical rule is applied (Math Forum, 2003, Online accessed 15February-2010). Let C_1 and C_2 be the two focus points of an ellipse (foci of an ellipse) and a and b be the lengths of the ellipse major and minor axis respectively, obtained from the rectangular region. According to Figure 3.5, the location of foci is obtained using (Math Open Reference, 2008, Online accessed 15-February-2010):

$$F = \sqrt{a^2 - b^2}, \qquad (3\text{-}2)$$

F is the distance of the foci from ellipse center.

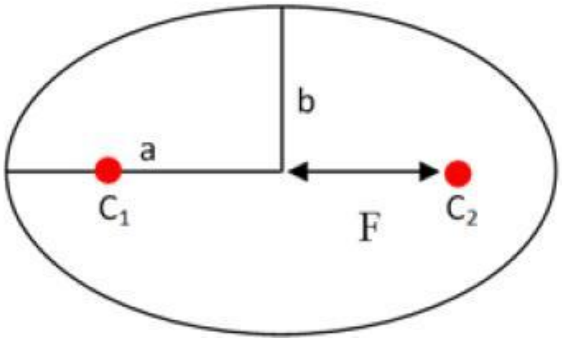


Figure 3.5 The relationship of ellipse and the foci.

A point P is inside of an ellipse if the sum of its distances from the two focus points is smaller than the length of major axis:

$$|P - C_1| + |P - C_2| < 2a \qquad (3\text{-}3)$$

Now the final initialization step is to calculate the discussed features (see Section 3.2.1) for the pixels inside the ellipse to represent the initial model M for the tracker.

3.2.2 Results

As we discussed before, to track the face, the initial position of the face should be determined. According to the hypotheses, we will use a combination of background subtraction, skin detection and Viola and Jones face detector to obtain the initial position of the face. Then an ellipse will be fitted over the face area, and the torso region which is 4 times longer than the total height of the detected face height.

Our method has been implemented in C++ using OpenCV and a custom library to control the IP PTZ cameras. We used two Sony IP PTZ cameras (SNC-RZ50N and SNC-RZ25N) for our tests. Table 3-1 shows the capabilities and characteristics of both cameras used in our experiments. The cameras have a lot of movement capabilities such as change of the pan and tilt values, zoom to change the FOV or see the details, and speed of the motion. The SNC-RZ50N supports three compression formats, JPEG, MPEG-4, and H.264.

Table 3-1 Cameras specifications.

Camera Model	*Max (R)*	*Max (fr)*	*P*	*Max(PS)*	*T*	*Max(TS)*	*OZ*
SNC-RZ50N	640×480	30*fps*	-170° to 170°	300°/*s*	-90° to 25°	90°/*s*	26x
SNC-RZ25N	640×480	30*fps*	-170° to 170°	100°/*s*	-90° to 30°	100°/*s*	18x

Max(R): Maximum Resolution, *Max(fr):* Maximum frame rate, *P*:Pan range, *Max(PS):* Maximum Pan Speed, *T:* Tilt range, *Max(TS):* Maximum Tilt Speed, *OZ*: Optical Zoom.

As discussed before the camera views the entrance of a place and different people are asked several times to enter into the place from that entrance to see if the camera can detect their initial face position or not. We have done eight experiment sets with two IP cameras on 5 people and different background scenes. The experiment sets are classified into four different classes based on initial target distance from camera, and image resolution. The experiment sets are classified into four different classes based on initial target distance from camera, and image resolution. The experiments are described in Table 3-2. We recorded all the experiments to extract their ground-truth manually for performance evaluation. Figure 3.6 shows the procedure of automatic initial target detection and modeling. We recorded all the experiments to extract their ground-truth manually for performance evaluation. Figure 3.6 shows the procedure of automatic initial target detection and modeling.

Table 3-2 Detection precision results of automatic initialization.

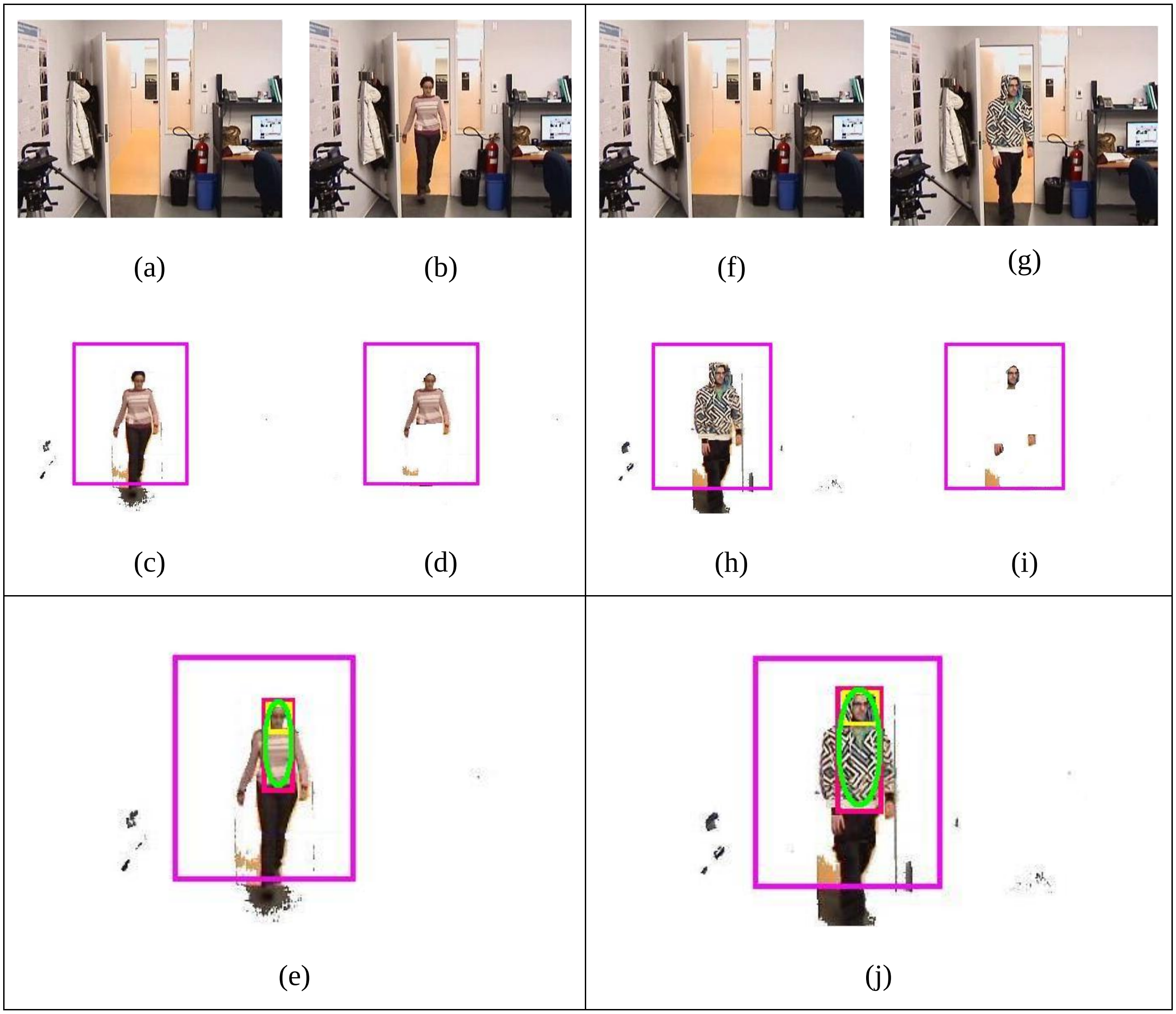
	Experiments	Camera Model	Image Size	Initial Model Position	DP(%)	NE
Class 1	E_1	SNC-RZ25	640×480	Near	100	10
	E_2	SNC-RZ50	640×480	Near	100	8
Class 2	E_3	SNC-RZ25	640×480	Far	95	6
	E_4	SNC-RZ50	640×480	Far	100	9
Class 3	E_5	SNC-RZ25	320×240	Near	90	9
	E_6	SNC-RZ50	320×240	Near	95	6
Class 4	E_7	SNC-RZ25	320×240	Far	85	8
	E_8	SNC-RZ50	320×240	Far	90	10

Near ~ 1 meter, Far ~ 6 meters.

To evaluate our initialization method, Detection Precision (*DP*) is used as a metric to calculate the target face detection accuracy. It is defined as

$$DP = \frac{CFD}{NE} \tag{3-4}$$

where CFD is the numbers of experiments in which the target face is detected correctly and NE is total number of experiments. We just count the number of experiments in which the system could detect the target face correctly and automatically find the initial target model. DP is used for this purpose.



(a) (b) (f) (g)

(c) (d) (h) (i)

(e) (j)

Figure 3.6 Automatic initial target detection procedure for E_4 ((a)-(e)) and E_8 ((f)-(j)), (a)and (f) background image, (b) and (g) target enters in a room, (c) and (h) background subtraction results, (d) and (i) skin detection over detected foreground, (e) and (j) face detection and ellipse fitting over the face and estimated torso.

Table 3-2 shows the *DP* results of automatic target initialization for two different image size resolutions of 640×480 and 320×240 and also two target initial distances near (1 meter) and far (6 meters) from camera. We have done our tests in a room with dimension of 6×4 meters.

For classes with smaller image size and target position far from the camera, the method has lower *DP* results. It is because of the minimum face size detection limitation. Viola and Jones can detect face size with minimum size of 20×20. During the smaller image size experiments and far target position, size could be even smaller than 20×20. That might be solved by an initial camera zooming on the initial target position. However, camera zooming will limit the camera FOV.

Another solution to increase the *DP* and make the Viola and Jones face detector more robust is to re-train the face detector with more face images as positive samples and more cluttered background images as negative samples. This causes a face detector with a higher true positive rate and lower false positive rate.

Figure 3.6 shows two examples of automatic target detection procedure. As discussed before first the background subtraction method is applied then skin classification is performed over the foreground region. Face detection is used over the specified region which is a room entrance and shown as pink rectangle in Figure 3.6. Finally, an ellipse is fitted (green ellipse in Figure 3.6 (e) and (j)) over the detected face area (yellow rectangle in Figure 3.6 (e) and (j)) and estimated torso region to model the target. As shown in Figure 3.6 (d) some false positive are extracted as skin region and face detector is helpful to remove those false regions.

The initialization block models and represents the target for the APF-OFS tracker. In the following the APF-OFS tracking method is explained.

3.3 Adaptive Particle Filter using Optical Flow based Sampling (APF-OFS) tracking

APF-OFS tracking shown in Figure 3.1 consists of an object tracking and position prediction block which are explained in this section. Target is localized continuously by an object tracking block. The detected target location has been effected by three delays shown in Figure 3.1. Therefore a position prediction block is used to compensate the system latency resulting from three delays and modify the detected position by object tracking block. Then the predicted position is sent to camera control block to center the PTZ camera on the target. Camera control and position prediction are affected by the results of object tracking.

3.3.1 Object tracking

Object tracking block shown in Figure 3.1 is described in this part. We developed a particle filter tracker with optical flow samples adapted to our face tracking problem. Some normalization functions are applied on the samples scoring. Normalization functions are applied to geometric and appearance features. The normalization functions are used to combine different measure values to uniform their magnitudes. Indeed, normalization of the sample scores might be done by any function. But more importantly is the adjustability property of the score values that is allowed by using the reasonable normalization functions. For example, the camera is always moved to put its image center on the target. This event allows the probability of target location around the image center to be higher than at other places. Thus, for the second measure of Euclidean distance between the sample center coordinates, $(x(s), y(s))$ and the image center coordinates (x_{im}, y_{m}), Gaussian normalization function is an appropriate choice. Optical flow that extracts the moving pixels is combined with particle filter that has robustness to non-Gaussian distribution of target movements to extract random motion of the object. Target modeling and tracking are done based on sampling around predicted position obtained by the position prediction block and detected moving regions determined by the optical flow algorithm. Our tracker is a combination of point- based and kernel-based methods since particle filter and optical flow are used in it.

The APF-OFS tracking algorithm must not lose the face for the tracking duration and must not be distracted by any other objects. The particle filter is a Bayesian method that recursively estimates the state of the tracking target as a posterior distribution with a finite set of weighted samples (Torabi et al., 2008). It operates in prediction and update phases. A sample is a prediction due to the state of tracking target.

We find classical particle filter as an appropriate solution for tracking with PTZ camera since the background is changing due to the camera panning, tilting and zooming. On one side, in our system, the target might have random and/or regular motions and also large inter frame motion between two consecutive frames. Therefore, between two consecutive frames the target may be located anywhere in the image. We should be able to find all possible candidates resulting from all types of target motions to find the best match candidate. Particle filter scheme provide us a representation of all target position possibilities resulting from both regular and random target motions. On the other side, the object tracking should be cost effective to have a small delay $\tau_{3,}$ since our system is internet based, the tracking system should handle the delays from traffic network and processing time. Now the question is how many samples are required to cover all target possible positions? Where the samples should be located? How the samples should be distributed, locally, globally or randomly? How much the processing time of particle filter will be? The answer to these four questions provides the main difference of our APF-OFS method with classical particle filter method. Here, classical particle filter alone is not appropriate, because we should have many samples to cover all possibilities everywhere in the image. Evaluation of many samples corresponds in having a high processing time and since the object tracking should be cost effective, thus the samples should be generated appropriately and selectively.

But in APF-OFS by using optical flow samples it is required to generate particle filter samples where only there is a motion results from target motion and not camera motion. The camera motion vector is removed from all motion pixels detected by optical flow by using a radial histogram scheme that is explained in the following.

Furthermore, APF-OFS uses a position predictor to modify the detected target position based on the three system delays, target and camera displacement. As a result APF-OFS tracker is a modified version of particle filter method that is appropriate for our tracking system.

3.3.1.1 Observation model

In our problem, as discussed in the initialization, the target is represented by an ellipse that circumscribes the head and torso part of the human body. The state of the particle filter at each time *t* is defined as a vector P_t of the coordinates the ellipse center, *c(t),* and its height, *h(t),* and width, *wid(t),* as:

$$P_t = (c(t), h(t), wid(t)) \qquad (3\text{-}5)$$

The state vector P_t was determined in the initialization step. The following geometric and appearance-based features as discussed in the initialization are used as the observation model to represent the target for our particle filter tracking:

1. Quantized normalized *HSV* color histogram with 162 bins (i.e.$18 \times 3 \times 3$).
2. Mean of *H, S* and *V* color components of *HSV* color space of all the pixels inside of the region.
3. Center coordinates of each sample or candidate ellipse.

3.3.1.2 Sample scoring using reasonable normalization functions

To localize the target, features of each sample s_i are compared with the initial model M, and a weight or a score ω_i is given to each s_i using a group of normalization functions. In addition of comparing the whole sample region, we divide vertically each sample to *b* (e.g. *b=5*) small regions to compare the content of each sample locally and in more detail for each portion (Figure 3.7). Indeed, we are interested to have a more accurate semi-local comparison than just a general comparison of whole sample size. Optimum value of *b (b=5),* is obtained experimentally according to the approximated ratio of face length to torso length and will be discussed in the results part.

The samples are divided vertically since we track a human and if the target turns from left to right or right to left around its main axis, the vertical content of the small regions does not change.

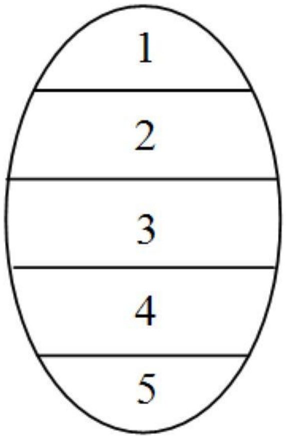


Figure 3.7 Vertical divisions of samples.

Various measures are applied on both whole sample region and *b* divided into small pieces of the same region. The goal is to define the most discriminating measures to compare robustly the target features with sample features. The following appearance and geometric based measures such as mean *HSV* color distance, quantized normalized *HSV* color histogram distance, 2-D correlation coefficient (similarity) of two histograms, and histograms intersection, are described in the following:

1. $\varphi_{EC}(s)$, Euclidean distance between mean *HSV* of the sample s, $(\overline{H}(s), \overline{S}(s), \overline{V}(s))$, and mean *HSV* of M, $(\overline{H}_m, \overline{S}_m, \overline{V}_m)$ is the first measure on an appearance feature and is defined as:

$$\varphi_{EC}(s) = \sqrt{(\overline{H}(s) - \overline{H}_m)^2 + (\overline{S}(s) - \overline{S}_m)^2 + (\overline{V}(s) - \overline{V}_m)^2} \qquad (3\text{-}6)$$

This distance is normalized between 0 and 1 by a linear normalization function, $\overline{\varphi}_{EC}(s)$ given by:

$$\overline{\varphi}_{EC}(s) = 1 - \frac{\varphi_{EC}(s)}{255\sqrt{3}} \qquad (3\text{-}7)$$

This distance is also applied on five divided small regions of each sample. Their correspondent normalization functions are $\overline{\varphi}_{EC1}(s)$, $\overline{\varphi}_{EC2}(s)$, $\overline{\varphi}_{EC3}(s)$,

$\overline{\varphi}_{EC4}(s)$, and $\overline{\varphi}_{EC5}(s)$. *HSV* color space is robust to illumination changes. This distance is a color-based comparison of the average color content of each sample with target model to compare the similarity of the sample appearance with target.

2. $\varphi_{EP}(s)$, Euclidean distance between the sample center coordinates, $(x(s), y(s))$ and the image center coordinates (x_{im}, y_m) is the measure on geometric feature as:

$$\varphi_{EP}(s) = \sqrt{(x(s) - x_{im})^2 + (y(s) - y_{im})^2} \tag{3-8}$$

Indeed, normally the person should be near the image center or ideally it should always be at the image center. Thus, a normalized Gaussian function $\overline{\varphi}_{EP}(s)$, is used to normalize it as:

$$\overline{\varphi}_{EP}(s) = \exp(-\frac{\varphi_{EP}^2(s)}{2\sigma^2}) \tag{3-9}$$

σ^2 is equal to a quarter of the image area around the image center. This distance is an appropriate measure to evaluate the sample location according to our goal (i.e camera center should be always on the target).

3. $\varphi_{EH}(s)$, Euclidean distance between quantized normalized *HSV* color histogram of sample *s*, $H(s)$, and the histogram of *M*, H_m, is the measure for this appearance feature as (Cha & Srihari, 2002):

$$\varphi_{EH}(s) = \sqrt{\sum_n (H(s)[n] - H_m[n])^2} \tag{3-10}$$

Where *n* is the histogram bin number. The normalization function $\overline{\varphi}_{EH}(s)$ is considered as:

$$\overline{\varphi}_{EH}(s) = 1 - \frac{\varphi_{EH}(s)}{\sqrt{2}} \tag{3-11}$$

The color histogram of a region represents the color distribution of that region. Color histogram comparison is another measure used to compare the appearance closeness of sample with target model by comparing their color distributions. This normalization function is also determined for the 5 divided small regions, as $\overline{\varphi}_{EH1}(s), \overline{\varphi}_{EH2}(s), \overline{\varphi}_{EH3}(s), \overline{\varphi}_{EH4}(s)$, and $\overline{\varphi}_{EH5}(s)$ to compare their color distributions in more detail.

4. $\varphi_H(s)$, 2-D correlation coefficient between $H(s)$, and H_m is given by:

$$\varphi_H(s) = \frac{\sum_n (H(s)[n] - \overline{H})(H_m[n] - \overline{H}_m)}{\sqrt{\sum_n (H(s)[n] - \overline{H})^2} \sqrt{\sum_n (H_m[n] - \overline{H}_m)^2}} \qquad (3\text{-}12)$$

where $\overline{H}$ and $\overline{H}_m$ denote the average of $H(s)$ and H_m and its normalization function $\overline{\varphi}_H(s)$, is defined as :

$$\overline{\varphi}_H(s) = \frac{1 + \varphi_H(s)}{2} \qquad (3\text{-}13)$$

$\varphi_H(s)$ is a metric to evaluate the color histogram similarity of target model with each sample. It is used to find the correlation of two color distributions which is a mathematical representation of appearance similarity. This normalization function is also determined for 5 divided small regions as $\overline{\varphi}_{H1}(s)$, $\overline{\varphi}_{H2}(s)$, $\overline{\varphi}_{H3}(s)$, $\overline{\varphi}_{H4}(s)$, and $\overline{\varphi}_{H5}(s)$.

5. $\varphi_{HI}(s)$, the intersection of histograms $H(s)$ and H_m is the last measure on appearance features defined as:

$$\varphi_{HI}(s) = \sum_n \min(H(s)[n], H_m[n]) \qquad (3\text{-}14)$$

Since $\varphi_{HI}(s)$ is between 0 and 1, $\overline{\varphi}_{HI}(s)$ is equal to $\varphi_{HI}(s)$. Histogram intersection shows the same content amount from target model histogram and each sample. It illustrates the likeness amount of two color distributions. This normalization function is also determined for 5 divided small regions as $\overline{\varphi}_{HI1}(s), \overline{\varphi}_{HI2}(s), \overline{\varphi}_{HI3}(s), \overline{\varphi}_{HI4}(s)$, and $\overline{\varphi}_{HI5}(s)$.

The weight (score) of the sample i at time t, ω_i^t , is obtained by sum of all normalization function values of the whole sample and five divided regions as:

$$\begin{aligned}\omega_i^t &= \overline{\varphi}_{EC}(s_i^t) + \overline{\varphi}_{EP}(s_i^t) + \overline{\varphi}_{EH}(s_i^t) + \overline{\varphi}_{H}(s_i^t) + \overline{\varphi}_{HI}(s_i^t) + \\ &\sum_{j=1}^{5}(\overline{\varphi}_{ECj}(s_i^t) + \overline{\varphi}_{EHj}(s_i^t) + \overline{\varphi}_{Hj}(s_i^t) + \overline{\varphi}_{HIj}(s_i^t))\end{aligned} \qquad (3\text{-}15)$$

Target, s_f is the best sample in each time t which has the maximum weights and is selected by:

$$s_f = \arg\max_{s_i \in S} \left\{ \begin{aligned} &\overline{\varphi}_{EC}(s_i) + \overline{\varphi}_{EP}(s_i) + \overline{\varphi}_{EH}(s_i) + \overline{\varphi}_{H}(s_i) + \overline{\varphi}_{HI}(s_i) + \\ &\sum_{j=1}^{5}(\overline{\varphi}_{ECj}(s_i) + \overline{\varphi}_{EHj}(s_i) + \overline{\varphi}_{Hj}(s_i) + \overline{\varphi}_{HIj}(s_i)) \end{aligned} \right\} \qquad (3\text{-}16)$$

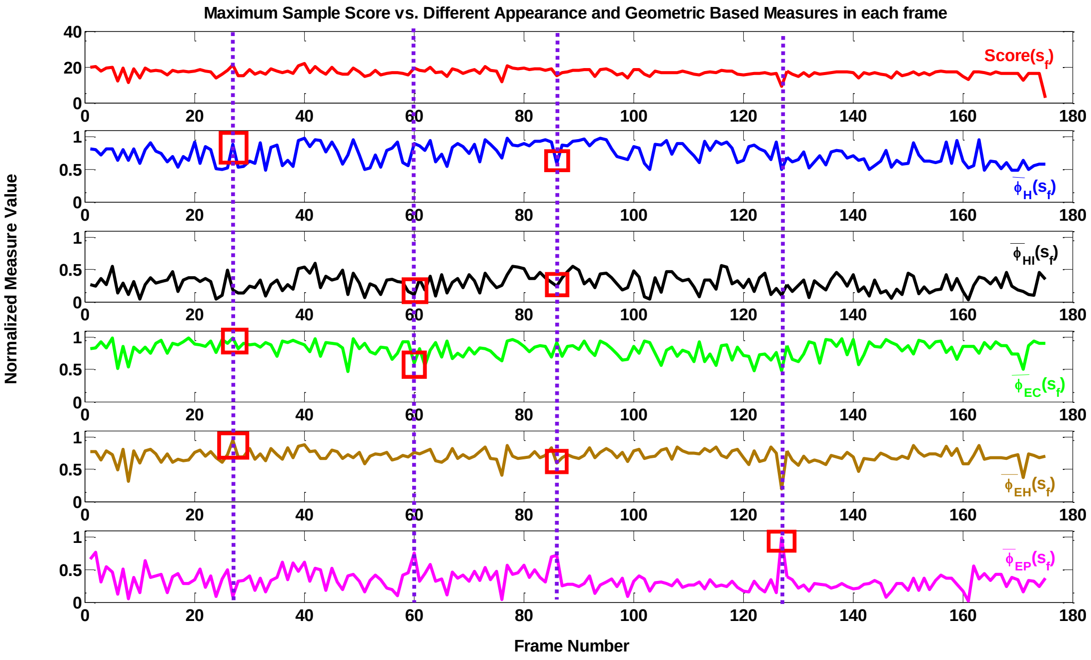
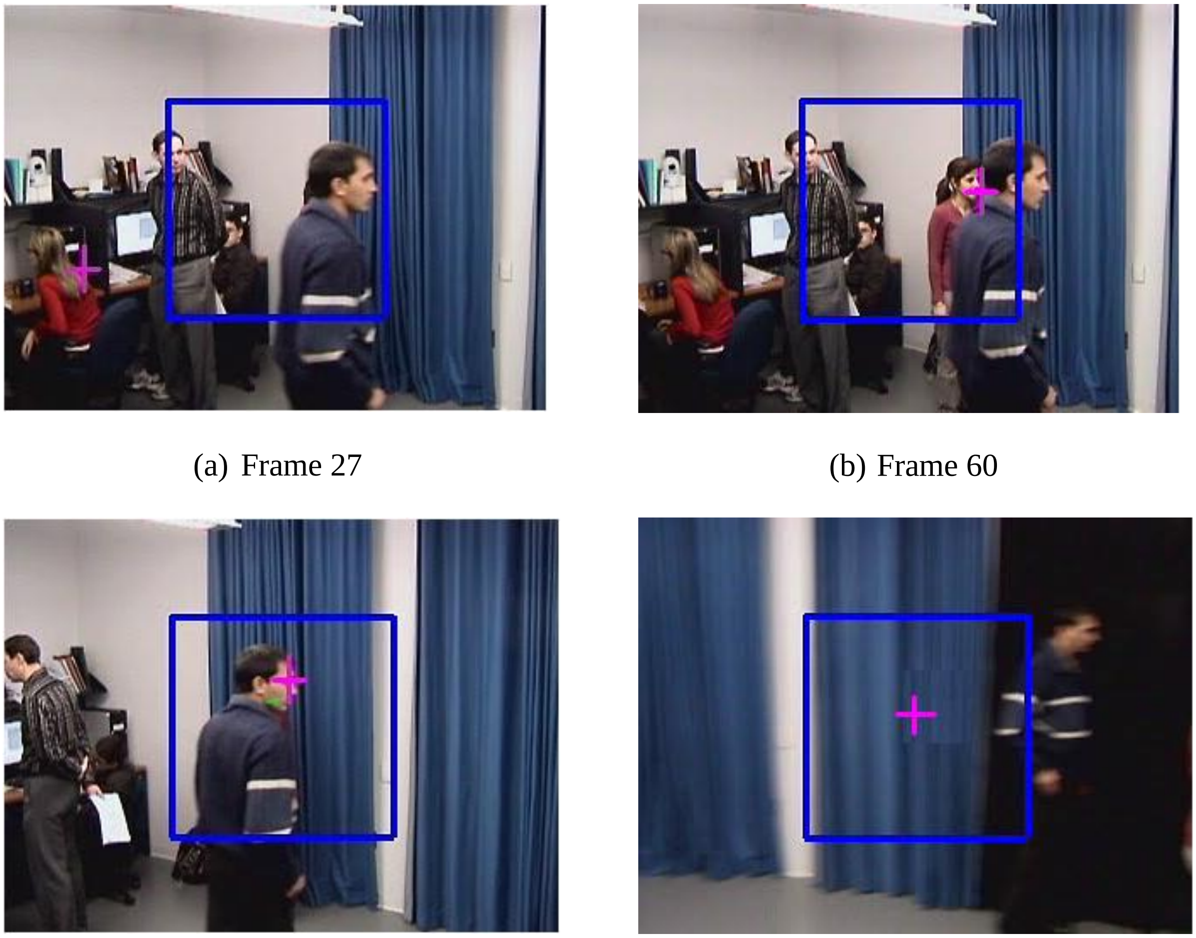


Figure 3.8 Maximum sample score versus different appearance and geometric based measures, red line: Score (s_f), blue line: $\overline{\varphi}_H(s_f)$, black line : $\overline{\varphi}_{HI}(s_f)$, green line: $\overline{\varphi}_{EC}(s_f)$, brown line: $\overline{\varphi}_{EH}(s_f)$, pink line : $\overline{\varphi}_{EP}(s_f)$.

We apply different measures on color because the results from only one measure are not accurate enough in all conditions while the results from other distances in the same conditions are better. We want to balance the precision of measures in general conditions. Figure 3.8 shows maximum sample score value in each frame versus different geometric and appearance based measures.

In Figure 3.8, the red rectangles and dashed lines illustrate interesting observation of various measures together and with total final score in different frames of an experiment. At frame 27 the maximum sample score does not belong to the real target because the target has been occluded. It is a false positive detection because of similarity of other object color to the target as shown in Figure 3.9 (a). However from the result of $\overline{\varphi}_H(s)$, $\overline{\varphi}_{EC}(s)$ and $\overline{\varphi}_{EH}(s)$ measures, a high score might be assigned because of the color closeness of the false target but other measures ($\overline{\varphi}_{EP}(s)$ and $\overline{\varphi}_{HI}(s)$) have a small value at this frame. This is also true at frame 60 in which the results of $\overline{\varphi}_{HI}(s)$ and $\overline{\varphi}_{EC}(s)$ are small because of partial occlusion as illustrated in Figure 3.9 (b) while the results of other measures are large and cause the true target detection. At frame 85, there is a false target detection because in a complete occlusion condition as shown in Figure 3.9 (c). It results in $\overline{\varphi}_H(s)$, $\overline{\varphi}_{HI}(s)$ and $\overline{\varphi}_{EH}(s)$ to have low score values, while the two others are high. Indeed according to the appearance features, the tracker cannot find any similar color object in the scene and the sample with highest score (target) is founded near to the image center. At frame 126, there is a tracking fragmentation resulting of target being out of FOV, Figure 3.9 (d). All the four score measures are small while $\overline{\varphi}_{EP}(s)$ has a high value because the centre of the image is selected.



(a) Frame 27

(b) Frame 60

(c) Frame 85

(d) Frame 126

Figure 3.9 Various events during target tracking (a) false target detection at frame27, (b) partial occlusion at frame 60, (c) complete occlusion at frame 85, (d) target gets out of FOV at frame 126.

Table 3-3 shows the mean and variances of all 5 measures for the best sample (target) and the average for other samples. In this experiment, $\overline{\varphi}_{EH}(s_f), \overline{\varphi}_{H}(s_f)$ and $\overline{\varphi}_{EC}(s_f)$ have a high mean value (close to 1) which means in general target sample has similar color histogram to initial target model. $\overline{\varphi}_{HI}(s_f)$ has a small mean because of partial occlusion and occlusion that happens a lot in this experiment. $\overline{\varphi}_{EP}(s_f)$ is small because the target sample in this experiment has lots of large motion that causes to be far from camera center. In average target sample has a higher discriminative measure values than other samples. The score value is almost smooth. It means that the features do not change or fail generally in the scenario or after a while. In other words, the features are robust, discriminative and stable to different situations during the scenario. Therefore, the usage of sum of all metrics is considered. Since the false alarms from one measure will be compensated from the rest.

Table 3-3 Means and variances of different measures shown in Figure 3.9.

	$\overline{\varphi}_{H}(s_f)$	$\overline{\varphi}_{HI}(s_f)$	$\overline{\varphi}_{EC}(s_f)$	$\overline{\varphi}_{EH}(s_f)$	$\overline{\varphi}_{EP}(s_f)$	*Score (s_f)*
Mean	0.7378	0.2898	0.8159	0.7120	0.3255	16.9129
Variance	0.0221	0.0177	0.0139	0.0080	0.0174	3.9469
Average of Mean for other samples	0.4511	0.0479	0.5201	0.3618	0.1272	-

3.3.1.3 Re-sampling and updating

Among all N_T samples existing in each frame, N_{s_i} samples with high probabilities (weights) are selected.

Thus, current sample set S_t is determined from N_T samples centered on c_i^t with probability ω_i^t at time *t*,

$$S_t = \left\{c_i^t(h_i^t, wid_i^t), \quad \omega_i^t\right\}_{i=1}^{N_T} \qquad (3\text{-}17)$$

where c_i^t is the *i*th sample coordinates with height and width h_i^t and wid_i^t at time *t* and ω_i^t is the related weight (score) to *i*th sample. Sample set S_t is an approximation of posterior distribution of the target state at time *t*.

The particle filter state in two consecutive frames does not change significantly if the camera does not move. Here, it is a translation of sample coordinates around its previous position and scaling of previous sample size. No rotation is applied in our application. It is because people are walking upright.

In each time *t*, the motion of the target is assumed to correspond to a dynamical first order auto-regressive model given by:

$$P_t = P_{t-1} + \omega_t \tag{3-18}$$

P_t and P_{t-1} are the particle filter states at time *t* and *t-1* respectively. ω_t is a multivariate Gaussian random variable and it related to random translation of the sample center coordinates and scaling of previous sample size.

Thus, in the resampling step, *N* samples are generated by a Gaussian random function in a circular region with radius of r_g (e.g. $1/5^{th}$of image height length) around the centroid of ROI using the motion assumption of equation 3-18. The size of these samples (width and height of the ellipses) is varied randomly s_s (e.g$\pm 5\%$) of the previous target size in the case that the target approaching or moving back from the camera. This choice is based on experiments and on the normal speed of people walking toward or away from the camera. This value is changed proportionally to the camera zooming parameter (e.g. two times zooming results $\pm(2\times 5\%)$ scaling changes). Indeed for our particle filter we are using scaling from the previous target size and translation around a pixel of interest. The scaling and translation values are obtained experimentally and are explained in the results section.

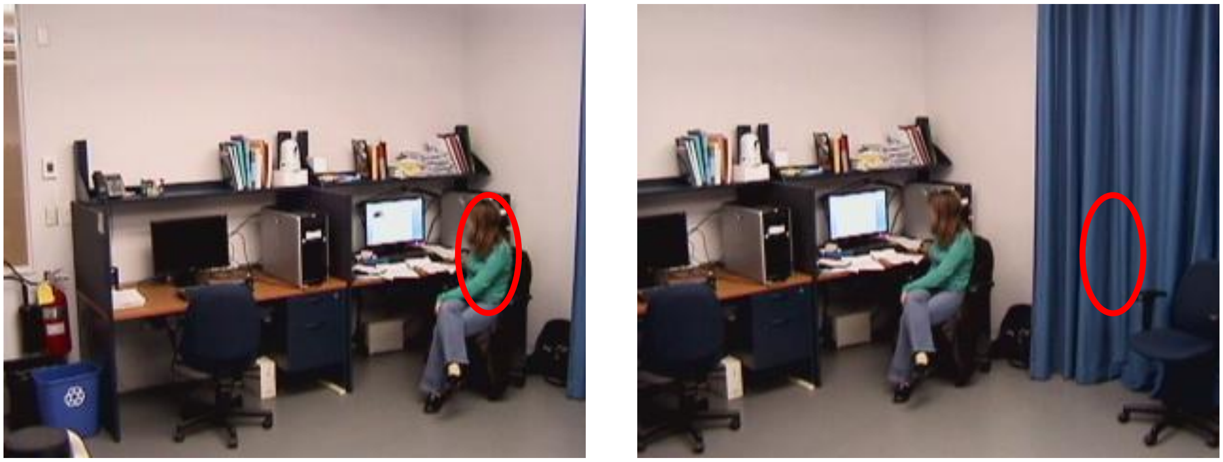
In our APF-OFS, current sample set S_t is composed of two sample sets which are S_t^{target} and S_t^{motion}:

$$S_t = S_t^{\text{target}} + S_t^{\text{motion}} \tag{3-19}$$

S_t^{target} is the current sample set that is composed of previous sample set at time *t-1* and $S_t^{\text{motion}}$is the current sample set composed of moving areas extracted by optical flow. It is because of using only previous sample set is not appropriate in our application since the camera is moving and the position of the previous state is changed. Indeed, in this work, we do not make any 3D position estimation of the scene, we do not know how the samples should be translated or scaled if the camera is moving or zooming. Thus, we do not have the 3D position of the samples and the camera motion vector is also an approximation in 2D, we cannot distinguish the sample position transformation from one frame to another accurately if the camera displacement is large. So we adjusted the particle filter to our application. Therefore from the previous sample set we only take the target sample with highest probability, s_f (e.g. $N_{s_i} = 1$), if the camera does not move. But if the camera moves, since it is supposed that the camera centers on the target, for the particle filter we only take the image center position with the last target sample size. The image center is used during the camera movement to generate N samples around it for the current sample set.

Therefore, we sample with ellipses the image around two types of ROI and model them:

1. Area around the previous target position coordinates or around image center,
2. Moving areas extracted by optical flow.



(a)

(b)

Figure 3.10 Target sample in two consecutive frames (a) before camera movement (b) after camera movement.

Thus, re-sampling is done based on two types of samples: previous target position based samples and motion-based samples. If the camera starts to move, during the movement, the sampling process is done around the image center and moving areas extracted by optical flow. During the camera movement, image center may be a prediction of target position since ideally target should always be at the image center.

Figure 3.10 shows two consecutive frames during camera movement. If we still do re-sampling around previous target position instead of image center we will wrongly do the sampling. In the following the re-sampling process for the motion-based samples is explained.

The second type of samples, S_t^{motion}, is detected by estimating the motion of the target from two consecutive images I_t and I_{t+1}, using pyramidal Lucas Kanade optical flow (Bouguet, 2000). In optical flow (Shi & Tomasi, 1994), strong corners in the image which have large eigenvalues are detected for comparison. To solve pixel correspondence problem for a given pixel in I_t, we look for nearby pixels of the same color in I_{t+1}. The basic concept in optical flow is to define the motion vector d for two consecutive images I_t and I_{t+1} by minimizing the following residual function ε:

$$\varepsilon(d) = \varepsilon(d_x, d_y) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} (I_t(x,y) - I_{t+1}(x+d_x, y+d_y)) \qquad \text{(3-20)}$$

where $(2w_x+1)\times(2w_y+1)$ is the integration window size to evaluate the ε value. Usually w_x and w_y are equal to 2,3,..7 pixels (Here we select 3, which is recommended in (Bouguet, 2000). For the pyramidal representation of the images, in each level this residual function will be minimized. We use 4 pyramid levels with 10 iterations. The threshold on stopping the iteration for minimizing of the residual function is 0.3. Optical flow extracts motion-based pixels with their related motion vectors resulting from camera movement or object movement.

As found experimentally, the detected motion vectors are noisy. In addition, the camera motion vectors have effect on object motion vectors. Thus, to remove this effect, camera motion vectors are extracted. To calculate the camera motion vectors, a radial histogram of motion vectors is calculated. In a radial histogram, each bin is based on the quantized length *(r)* and angle *(θ)* of the motion vector. Our radial histogram, *h(r,θ)* has 36180 bins. *r* has 201 values, and is varied based on the image size between 0 and image diameter (e.g for an image size 640 × 480: the radial bin number, N_r =201 , radius is quantized by factor 4 pixels). θ has 180 values and is varied between 0° and 360° (e.g. the angular bin number, N_θ =180, angle is quantized by factor 2). These values are obtained experimentally and are explained in the results section. The r and θ of the bin that has the maximum number of vectors is assigned to be the camera's motion vector length and angle. The detected motion vectors that have this length and angle are removed and then the camera motion vector is subtracted from the rest of the motion vectors using estimated bin values. The bin value is computed based on the low limit range value of the quantized values. For example for r between 0-3 the bin value is 0. Motion vectors are then grouped according to their distances from each other and their lengths and orientations (Chung et al., 2005). Motion-based samples are extracted around object motion vectors groups.

3.3.1.4 Number of samples per frame

In our APF-OFS, the total number of samples in each frame, N_T is composed of samples generated around previous target position (or image center) and the centroid of clustered moving pixels. The number of samples generated by Gaussian function, N, is fixed but N_T , varies due to the moving pixel clusters. However, N_T , has a finite value since there is a finite number of moving pixels clusters. The sample generation process is stopped whenever it covers all the motion clusters.

3.3.2 Position prediction

Position prediction block shown in Figure 3.1 is explained here. As discussed before, because of the three delays in the control loop, the system does not put the camera center on the target at the right time using only the determined object position found by the object tracking block. To compensate for motion of the target during the delays, a position prediction based on the D last motion vectors of the target (i.e. when the camera is not moving) has been designed. This motion predictor considers the angles between D consecutive motion vectors. If the angle difference is smaller than ϑ° (i.e. ϑ° is obtained experimentally), it is assumed that the target is moving in the same direction and its average speed can be estimated. The average speed of the target, $\bar{v}$, during the *D* last target motion vectors is thus, equal to:

$$\bar{v} = \frac{\sum_{i=1}^{D} \Delta x_i}{\sum_{i=1}^{D} (\tau_1^i + \tau_3^i)} \tag{3-21}$$

Where *Δx_i*, is the *ith* target displacement vectors (i.e. target motion vectors). Motion vectors are calculated when the camera is not moving.

τ_1^i, is the *i*th delay τ_1 and τ_3^i, is the *i*th delay τ_3 for the *D* last captured images. The system will put the camera center on the predicted position which is:

$$c_p = c_E + \bar{\tau}_2 \times \bar{v} \tag{3-22}$$

Where c_p is the predicted target coordinate and c_E is the extracted target coordinate from the object tracking block. $\bar{\tau}_2$ is the average delay time τ_2 obtained proportionally to displacement required to reach c_E. For example each camera movement creates a delay of τ_2^i where *i* is the *i*th movement of the camera with displacements of c_c - c_E^i, where c_c is the image center. $\bar{\tau}_2$ is calculated based on average of all previous camera displacements (e.g. all c_c - c_E^i).

$$\bar{\tau}_2 = (c_c - c_E) \times \frac{\sum_i \tau_2^i}{\sum_i \Delta x_i} \tag{3-23}$$

Thus, the camera is controlled and moved adaptively with the speed of the network and camera response time, and the motion vector of the target.

The face tracking algorithm using APF-OFS tracking can be summarized as the following steps:

1. *Initialization*

- Automatically fit an ellipse over the target head and torso.

2. *Resampling*

- To construct S_t^{target} :
  - If the camera does not move, choose best sample with highest score from samples at time *t-1*, s_f, based on the probabilities determined by equation (3-15).
  - If the camera is moving, choose the image center with previous target size.
- To construct S_t^{motion}:
  - For two consecutive images, apply optical flow and extract the group of object motion vectors from camera motion vectors.
- Compose the current sample set of S_t from S_t^{target} and S_t^{motion} using equation (3-19) .
- Reproducing samples using translation and scaling changes equation (3-19) .

3. *Update*

- Update new samples weights from obtained observation measurement using equation (3-15).

4. *Feature scoring*

- Select the best sample using equation (3-16).

5. *Position predication*

- Predict the position of the target based on the *D* last motion vectors of the target.
- Send the predicted position to the camera. The predicted target coordinate is now ready to send to the camera if the camera is free for moving.
- Process next frame and iterate from step 2 to step 5.

The camera control is presented in the next section.

3.4 Camera control

Then the predicted position is sent to camera control block to center the PTZ camera on the target. Camera control block shown in Figure 3.1 adjusts pan and tilt values based on the predicted position and zoom on the target with an appropriate zooming value. Figure 3.11 shows the perspective transform from 3D world coordinates to the 2D image plane coordinates.

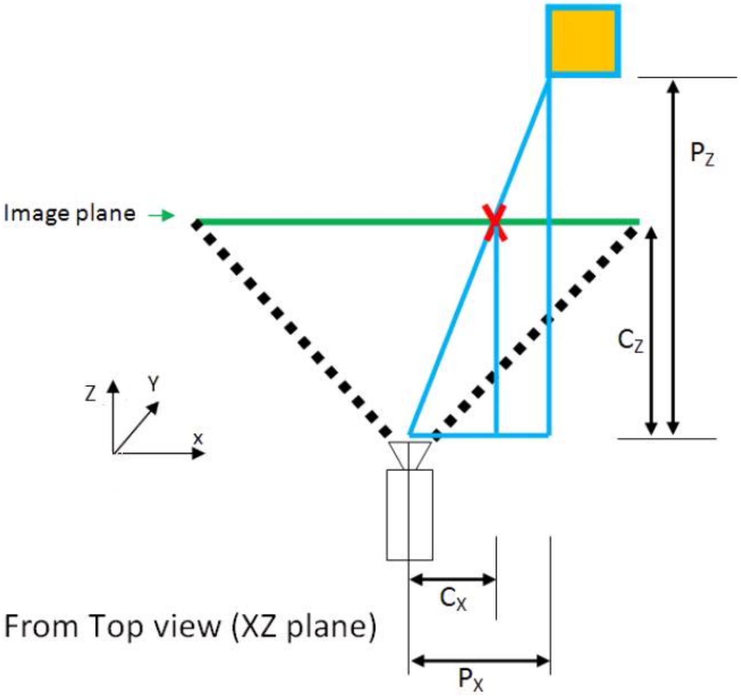


Figure 3.11 Perspective transform geometry

To obtain which image coordinate $c_P=(C_x,C_y)$ corresponds to a world point, P, the following relations are used:

$$C_x = P_x \times \frac{C_z}{P_z} \tag{3-24}$$

$$C_y = P_y \times \frac{C_z}{P_z} \tag{3-25}$$

In our case, the camera is controlled based on the target motion on the 2D image plane. That is, we assume that the object is moving on a plane parallel to the image plane ($\frac{C_z}{P_z}$ is assumed constant). Because of the low frame rate, this is not exact, but it is an acceptable approximation for a walking human.

To follow the target, the PTZ motors are commanded based on c_p. As discussed earlier, in our servo control loop, c_p is computed and camera is moved to keep well tracking of target in the case of variable delays. Sometimes, we have good tracking and sometimes not. Indeed sometimes the camera continuously tracks the true target with a high score but sometimes the target score is small. Thus, the zooming on the target is done if the target is well tracked. We should be careful about moving the camera or zooming to avoid losing the object from observed camera FOV. To achieve this goal, some criteria should be verified to ensure we are correctly moving the camera to follow the target or to zoom the camera on the target. We will describe the moving and zooming criteria with their corresponding camera functions in the following.

3.4.1 Moving criteria

As a first criterion, a moving command is sent to the camera if it is not moving or zooming. This is to avoid mixing two different commands that is sent consecutively to network. Camera performs commands sequentially and not in parallel. In addition, each camera zooming or moving corresponds to a delay because a mechanical task needs time to be done. If we want to send commands consecutively without considering these time periods, the probability that we lose the target is high. Thus, communication with camera is controlled by a mutex. The word mutex comes from MUTual Exclusion, which assigned to an object, arranges MUTual EXclusion between threads. A mutex is applied between threads to make sure that only one of the threads is simultaneously permitted to execute a specific application code at a time. Therefore, a moving or a zooming command will be sent to the camera if the mutex shows that the previous task of the thread has been finished and now it is ready for another command.

As discussed before, sometimes we have good tracking and sometimes not. Thus, the moving on the target is done if the target score is at least η percent similar to the previous target scores. The tracking is fine if the sample score remains the same. To evaluate the similarity of samples, the mean of the sample scores are calculated and compared with the current sample score. Therefore, in each frame we are calculating the average μ_{s_f} of all target sample scores that are available from the first frame up to the current frame. After approbation of the criterion, a moving command is sent if the score of target sample, s_f that is obtained by APF-OFS, is higher than η% of the average score, μ_{s_f} . The value of η is obtained experimentally and is explained in the results part (e.g. no zooming, $\eta = 85$).

$$if\ s_f > \eta \times \mu_{s_f} \qquad (3\text{-}26)$$

3.4.2 Zooming criteria

Similarly, zooming should be done under certain zooming criteria to prevent losing the target. Zooming is done when the program is sure about tracking results. Like moving criterion, a zooming command is sent to the camera if it is not currently moving or zooming.

As explained earlier, to avoid losing the object from observed camera FOV, zooming should be done if the target is well tracked. Therefore, the zooming is done under a more precise constraint on target score similarity. The same as for moving, in each frame, the average μ_{s_f} and variance σ_{s_f} of all target sample scores from first frame up to current frame are calculated. Then, a zoom in command is sent if the following constraint is respected:

$$s_f > \mu_{s_f} + \sigma_{s_f} \qquad (3\text{-}27)$$

Also the zoom out command is sent if this constraint is true:

$$s_f < \mu_{s_f} - \sigma_{s_f} \qquad (3\text{-}28)$$

The target size is changed adaptively to the zooming parameter. For example if we do zooming two times, the target size is enlarged two times (e.g. h_i^t and w_i^t will be $2 \times h_i^t$ and $2 \times w_i^t$ respectively). Also zooming has effect on the score value. For example for four times zooming η is 70% instead of 85% which is explained in the results part.

3.4.3 Stopping criteria

After doing zooming, moving and tracking it is required that the camera tracking is stopped somewhere. As discussed earlier the goal is to find the target face, track it, zoom on it, and take the target face image. Therefore, some criteria should be applied to verify that if the camera correctly zooms on the target face or not. To do so, after camera zoom in, the Viola and Jones face detector (Viola & J. Jones, 2004) is applied on a search region on the top part of the target sample to verify if a face is found or not. If a face is found the target face image is captured and tracking process is stopped but if not the camera continues the tracking process until the stopping criteria is satisfied.

3.4.4 Camera functions

To control the camera, to move/zoom it, we have implemented three methods. The angles and zoom can be computed by the camera using the on-board *AreaZoom* function, or the camera can be controlled by computing the angles and zoom on a workstation and sending computed values to the camera using the *RelativePanTiltZoom* function. For zooming case, we have specifically a function which is called *MultipleZoom* that only does the zooming *m* times (e.g. *1X*, *2X*, *3X*, *etc*) larger than the original target size. In the following only the best choices are explained in details.

3.4.4.1 RelativePanTiltZoom function

In this case, the angles required to move the camera are computed by the workstation. This way, computations are faster than using the on-board function. The target pixel coordinates (x,y) in the current frame is converted in pan-tilt values for centering the camera. The pan and tilt values are obtained according to the FOV of the lens and the range of pan and tilt angles for each camera (i.e. it is dependent on the camera model.). The angles and zoom values are sent to the camera by sending an HTTP POST request using the CGI scripts of the camera (Sony corporation, 2005, Online accessed 15-February-2010) and the *RelativePanTiltZoom* function. For this function, it is possible to specify the speed of motion. It is set to the maximum value.

3.4.4.2 MultipleZoom

MultipleZoom is used only for the zooming case. It will be used if we are interested to enlarge the target *m* times (e.g. *1X, 2X, 3X, etc.*) larger than the original target size. *m* is the input of this function which is send by an HTTP POST request using the CGI scripts to the camera.

CHAPTER 4. Experimental Results and Discussion

The goal of this chapter is to evaluate the performance of the proposed tracking and controlling system. To evaluate the performance of the system we have done experiments in different categories which are:

- Study of different parameters on tracking system performance such as image size and camera model.
- Calibration of tracking parameters to determine the optimum parameters values.
- Study of system limits.
- Comparison of proposed tracking system performance with well-known tracking methods such as PF and KLT trackers.
- Study of tracking algorithm on camera control performance by using two similar IP PTZ cameras and two different tracking methods executed on each.
- Study of zooming control on tracking system performance to test and evaluate the whole system performance and determine the maximum level of zooming with an acceptable tracking error result.
- Study of tracking performance versus available network bandwidth and versus different camera to network connection mode to evaluate the tracking performance versus the environmental communication parameters.

The details of each experiment are presented in detail in the following.

4.1 Data acquisition and cameras characteristics

Our method has been implemented in C++ using OpenCV and a custom library to control the IP PTZ cameras. We used two Sony IP PTZ cameras (SNC-RZ50N and SNC-RZ25N) as previously shown in Table 3-1 for our tests. For validation, we tested the complete system in online experiments. No dataset is available for testing the complete tracking system because of its dynamic nature. The tracking algorithm has been tested over events such as entering or leaving the FOV of the camera and occlusion with other people in the scene. We recorded all the experiments to extract their ground-truth manually for performance evaluation. Only usable frames that are captured and processed by camera are recorded. Thus, the recorded video frame rate is not 30 fps, but corresponds to the frame processing rate by the servo control loop. The general scenario of the experiments is the following. An actor from the frontal view is selected for initial modeling. She starts to walk around in a room. Between two to four actors can walk at the same time in different directions, crossing and occluding with the target. The target actor makes some pauses while walking to verify the performance for stationary target. The target actor also moves parallel, toward or away from the camera. In all experiments, there are scale changes to verify tracking against scaling.

The camera parameters such as video compression format, level of image compression, and bandwidth control are set to the recommended values in (Sony corporation, 2005, Online accessed 15-February-2010).

4.2 Evaluation metrics

To evaluate our method, four metrics are used:

- Precision (*P*) to calculate the target localization accuracy. It is defined as

$$P = \frac{TP}{TP + FP} \qquad (4\text{-}1)$$

where *TP* and *FP* are true positive and false positive respectively. *TP* is the number of frames in which the target is correctly localized by the camera when the target is in the camera FOV. *FP* is the number of frames where the target is not localized correctly when the target is in the camera FOV.

- Normalized Euclidean distance (d_{gc}) to evaluate the dynamic performance of the tracking system. It is defined as:

$$d_{gc} = \frac{\sqrt{(x_c - x_g)^2 + (y_c - y_g)^2}}{a} \qquad (4\text{-}2)$$

where (x_g,y_g) is the ground-truth target coordinate and (x_c,y_c) is the center of the image. It is the spatial latency of the tracking system, as ideally, the target should be at the image center. *a* is the radius of the circle which circumscribes the image (the maximum distance). By this definition if the target gets out of FOV the maximum penalty of 1 is considered.

- Normalized Euclidean distance (d_{gp}) which shows the error of tracking algorithm. It is the target position error. It is defined as

$$d_{gp} = \frac{\sqrt{(x_p - x_g)^2 + (y_p - y_g)^2}}{2a} \tag{4-3}$$

where (x_p,y_p) is the tracked object coordinates. Ideally, d_{gp} should be zero.

- Track fragmentation (*TF*) indicates the lack of continuity of the tracking system for a single target track (Yin, Makris, & Velastin, 2007).

$$TF = \frac{T_{out} + FP}{NF} \tag{4-4}$$

T_{OUT} is the number of frames where the target is out of the *FOV* and *NF* is the total number of frames. Therefore, *TF* shows target lost either by false detection or by target being out of camera FOV.

4.3 Study of different parameters on tracking system performance

The goal of this section is to study and compare the effect of different parameters such as image size and camera model on tracking system performance. We have done twenty experiments with the two IP cameras. The experiments are described in Table 4-1. Experiments are classified into four classes based on the camera model, and image size. The experiments are done in a room with dimension of 6 × 4 meters. The distances of the initial model position from camera are written in Table 4-1.

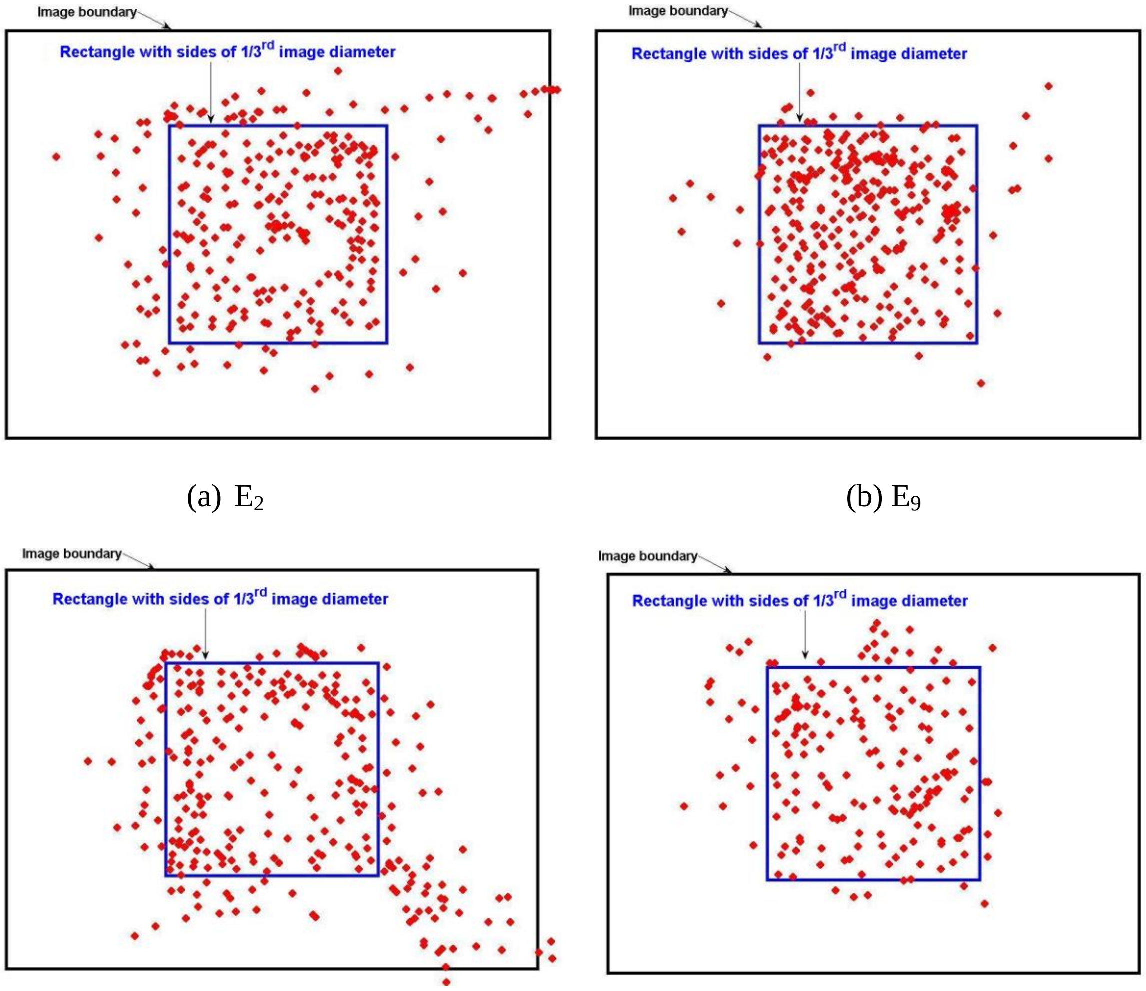
Figure 4.1 shows the target position in image plane at different frames for one experiment of each class of Table 4-1. d_{gc} is the normalized distance of each point from image center. This distance allows us to verify if the tracker has lost the target. If d_{gc} is more than 1, target is lost (outside of FOV).

Table 4-1 Experiments to compare the effect of camera model and image size.

Class	**Experiment**	**Camera Model**	**Image Size**	**Initial Model Position**
Class 1	E_1	SNC-RZ50	640×480	Near
	E_2	SNC-RZ50	640×480	Near
	E_3	SNC-RZ50	640×480	Far
	E_4	SNC-RZ50	640×480	Far
	E_5	SNC-RZ50	640×480	Middle
Class 2	E_6	SNC-RZ50	320×240	Near
	E_7	SNC-RZ50	320×240	Near
	E_8	SNC-RZ50	320×240	Far
	E_9	SNC-RZ50	320×240	Far
	E_{10}	SNC-RZ50	320×240	Middle
Class 3	E_{11}	SNC-RZ25	640×480	Near
	E_{12}	SNC-RZ25	640×480	Near
	E_{13}	SNC-RZ25	640×480	Far
	E_{14}	SNC-RZ25	640×480	Far
	E_{15}	SNC-RZ25	640×480	Middle
Class4	E_{16}	SNC-RZ25	320×240	Near
	E_{17}	SNC-RZ25	320×240	Near
	E_{18}	SNC-RZ25	320×240	Far
	E_{19}	SNC-RZ25	320×240	Far
	E_{20}	SNC-RZ25	320×240	Middle

Near ~ 1 meter, Far ~ 6 meters, Middle ~ 3 meters.

For distances smaller than 0.6, the object is in the FOV. For the range of (0.6 < d_{gc} < 1), it depends if the centroid coordinates of the target are in the range [0, height-1] and [0, width-1]. For E_9, and E_{17} the target is always in the FOV. For E_2, the target was out of FOV one time at frame 124. In E_{14}, the target was out of FOV several times at frame 136, 166 and 184. When the target moves very fast in the opposite predicted position, the target will be out of FOV.



(a) E_2

(b) E_9

(c) E_{14}

(d) E_{17}

Figure 4.1 Target position (center of best sample (s_f)) in image plane at different frames for various experiments (a) E_2, (b) E_9, (c) E_{14}, and (d) E_{17}.

Table 4-2 Experimental results to compare the effect of camera model and image size, (Class 1 and Class 2).

E	P(%)	TF(%)	$\mu_{d_{gc}}$	$\sigma^2_{d_{gc}}$	$\mu_{d_{gp}}$	$\sigma^2_{d_{gp}}$	*FR(fps)*	*NF*	*min(Δx)*	*max(Δx)*
E_1	93	5	0.2554	0.0436	0.0580	0.0015	6.67	511	0.018	0.33
E_2	96	2	0.2121	0.0352	0.0517	0.0021	6.86	489	0.013	0.31
E_3	98	1	0.1561	0.0271	0.0483	0.0032	7.31	494	0.016	0.36
E_4	92	7	0.1763	0.0394	0.0619	0.0027	6.04	506	0.006	0.25
E_5	91	6	0.1779	0.0613	0.0617	0.0023	6.19	517	0.008	0.29
Class 1	**93.95**	**4.2**	**0.1956**	**0.0413**	**0.0563**	**0.0024**	**6.57**	**2517**	-	-
E_6	96	1	0.1124	0.0273	0.0310	0.0017	10.29	897	0.015	0.45
E_7	100	0	0.0930	0.0227	0.0242	0.0018	12.72	908	0.007	0.26
E_8	98	0.6	0.1015	0.0231	0.0246	0.0019	11.63	916	0	0.42
E_9	95	2	0.1209	0.0261	0.0391	0.0022	10.84	903	0.017	0.47
E_{10}	97	0.7	0.1086	0.0242	0.0263	0.0016	11.35	910	0.012	0.43
Class 2	**97.20**	**0.858**	**0.1073**	**0.0246**	**0.0290**	**0.0018**	**11.31**	**4534**	-	-

E: Experiments, P:Precision, $\mu_{d_{gc}}$: mean of d_{gc}, $\mu_{d_{gp}}$: mean of d_{gp}, $\sigma^2_{d_{gc}}$: variance of d_{gc}, $\sigma^2_{d_{gp}}$: variance of d_{gp}, FR: System frame rate and NF:Number of frames, *min(Δx):* minimum normalized motion vector length, *max(Δx)*: maximum normalized motion vector length.

Table 4-2 and Table 4-3 show the results of four metrics and the system frame rate for all experiments. For d_{gc} and d_{gp}, we show the mean and variance of all experiments. For classes with larger image size, the method has lost the target several times, but eventually recovers. For experiments using a smaller image resolution, there are fewer target lost (e.g. TF is smaller), less distance error ($\mu_{d_{gc}}$ and $\mu_{d_{gp}}$) and the results of precision are better (P is larger). With smaller image resolution, the results of tracking are significantly better. Indeed, it results to a faster system rate because less data is processed and transferred on the network (for the same JPEG compression level).

Table 4-3 Experimental results to compare the effect of camera model and image size (Class 3 and Class 4).

E	P(%)	TF(%)	$\mu_{d_{gc}}$	$\sigma^2_{d_{gc}}$	$\mu_{d_{gp}}$	$\sigma^2_{d_{gp}}$	*FR(fps)*	*NF*	*min(Δx)*	*max(Δx)*
E_{11}	89	10.5	0.2431	0.0324	0.0702	0.0029	4.96	502	0.025	0.33
E_{12}	92	8	0.2242	0.04872	0.0626	0.0041	5.07	467	0.010	0.34
E_{13}	88	11	0.2437	0.0569	0.0689	0.0037	4.21	481	0.017	0.23
E_{14}	91	9	0.2379	0.0431	0.0658	0.0036	4.24	523	0.012	0.29
E_{15}	93	7	0.2216	0.0388	0.0592	0.0038	5.14	514	0.015	0.26
Class 3	**90.61**	**9**	**0.2341**	**0.0440**	**0.0653**	**0.0036**	**4.68**	**2487**	**-**	**-**
E_{16}	95	2.1	0.1220	0.0213	0.0321	0.0019	9.68	897	0.027	0.50
E_{17}	93	2.5	0.1381	0.0117	0.0387	0.0018	9.47	869	0.012	0.38
E_{18}	96	1.9	0.1117	0.0171	0.0317	0.0021	10.47	904	0.007	0.43
E_{19}	94	2.3	0.1315	0.0112	0.0339	0.0022	9.51	888	0.010	0.32
E_{20}	92	2.6	0.1428	0.0526	0.0382	0.0020	9.36	893	0.005	0.29
Class 4	**94.01**	**2.2**	**0.1292**	**0.0228**	**0.0349**	**0.0020**	**9.68**	**4451**	**-**	**-**

E: Experiments, P:Precision, $\mu_{d_{gc}}$: mean of d_{gc}, $\mu_{d_{gp}}$: mean of d_{gp}, $\sigma^2_{d_{gc}}$: variance of d_{gc}, $\sigma^2_{d_{gp}}$: variance of d_{gp}, FR: System frame rate and NF:Number of frames, *min(Δx):* minimum normalized motion vector length, *max(Δx)*: maximum normalized motion vector length.

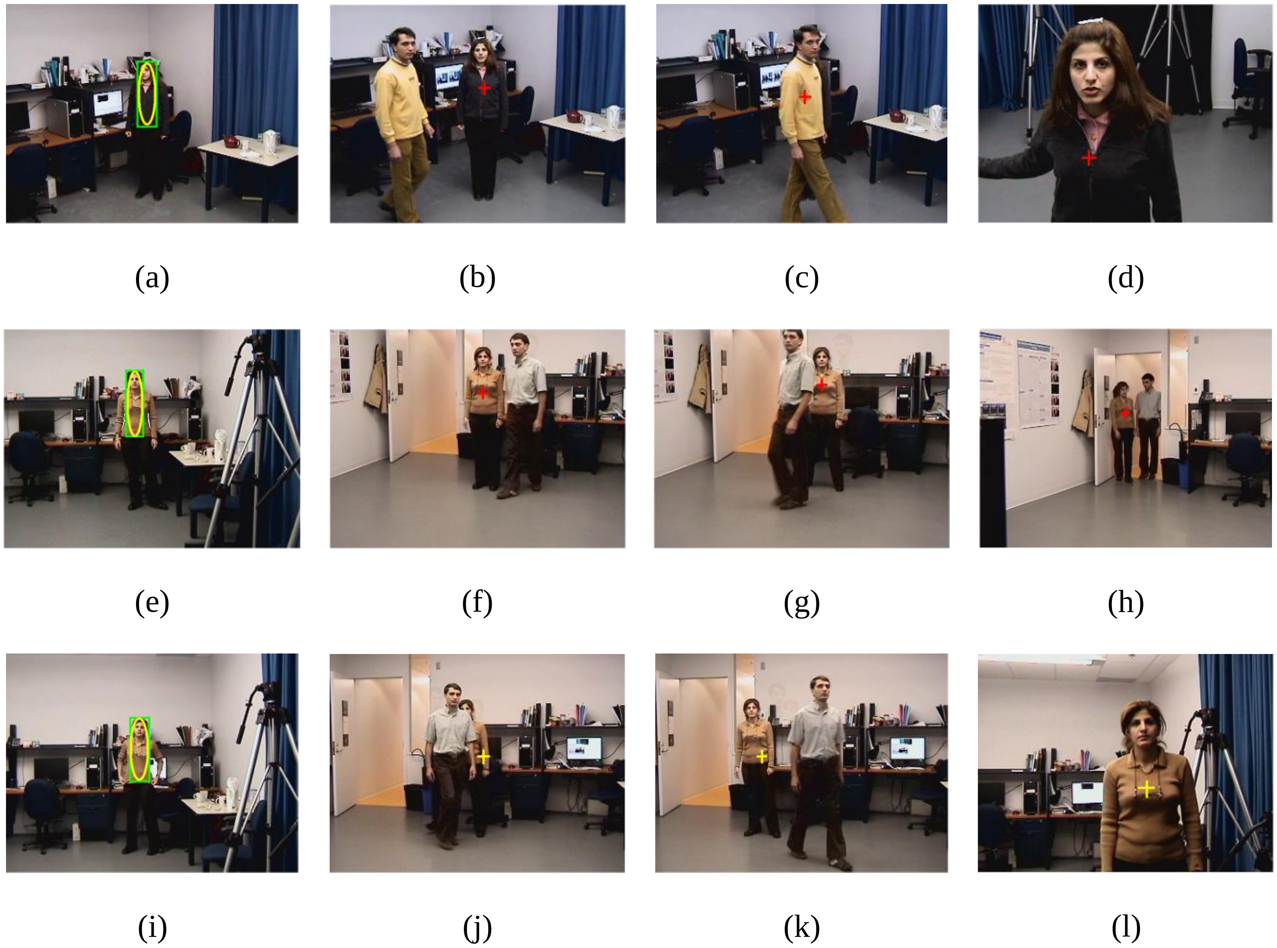
In fact, because of larger image size, the camera sends only 16fps in 640 × 480, while it can send 30fps in 320 × 240. If the camera sends only 16fps, delays τ_1 and τ_3 increase. This means that a moving target will have larger motion between two frames than for 320 × 240 (e.g.

$\max(\Delta x) = \dfrac{\text{target displacement in pixel}}{\text{image diameter in pixel}}$ = 0.5 means, 200 target displacement in pixels). In turn, because the target has larger motion, the camera needs to move a larger angular distance (activates its motors longer), and thus, delay τ_2 also increases. Because delays τ_1 and τ_2 both increases (τ_2 being the more significant), larger image results into a larger localization and prediction errors. This is confirmed by comparing the 4 classes, where the mean d_{gc}, mean d_{gp}, and P are improved with smaller image size.

Because of φ_{EP} and camera control, the error on $\mu_{d_{gc}}$ has effect on $\mu_{d_{gp}}$ and vice versa. That is, if $\mu_{d_{gc}}$ is large, the target will have a smaller value for φ_{EP}, and in turn, the object might then not be localized correctly and increase $\mu_{d_{gp}}$. Indeed the algorithm tries to find similar object to the target which is nearest to the image center. Furthermore, if the target moves very fast, to minimize d_{gp}, d_{gc} will be increased. The best results are obtained for class 2 and 4, in which the system frame rate is faster. *TF* for class 2 and 4 are the smallest. A faster system frame rate improves the results of TF, $\mu_{d_{gc}}$, $\mu_{d_{gp}}$ and P. By comparing the results of class 2 and 4 with results of class 1 and 3, the effect of image resolution size for both cameras is evaluated. As shown in Table 4-2 and Table 4-3, with smaller resolution, faster frame rate and better tracking performance are obtained.

With smaller image size, we obtain a higher target tracking precision P with less tracking fragmentation and faster frame rate. The localization of the target is better ($\mu_{d_{gc}}$ and $\mu_{d_{gp}}$ values are smaller). It means the location of the target is near to the ground truth and the camera could center on the target with a good precision. This result is explained by the use of particle filter that well distributes samples at each frame in different positions and various scale sizes around detected motion regions or previous target position. That gives more candidate regions to the tracker to find the best match. Localization accuracy is thus improved. This also results in the target being usually located in a constant distance within 1/6th of image diameter from the image center (e.g. blue rectangle in Figure 4.1). When localization fails, it is because of similarity or closeness of the color histogram of the target with other samples. The image resolution and camera model have effects on the system frame rate and thus on tracking error. The last column in Figure 4.2 and Figure 4.3 shows the scale variation of the target during experiments. Also there are some short term occlusion examples in the second and third column of Figure 4.2 and Figure 4.3.

By comparing the results for both cameras (class 1 and 2 with class 3 and 4), results of SNC-RZ50 are better than SNC-RZ25 because of the camera characteristics such as maximum pan and tilt speeds, maximum FOV, and etc.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 4.2 Examples of tracking frames for E_5 (a) to (d), E_8 (e) to (h), and E_9 (i) to (l). E_5 (a) initial model selection, (b)before short occlusion, (c) short term occlusion, (d) scale variation; E_8 (e) initial model selection, (f) before short term occlusion, (g) after short term occlusion, (h) scale variation; E_9 (i) initial model selection, (j) short term occlusion, (k) after short term occlusion, (l) scale variation.

The last two columns in Table 4-2 and Table 4-3 are the minimum and maximum length of target motion vector in number of pixels which is normalized due to the image diameter. These values vary according to the image resolution size, frame rate and target movement. max (Δx) for smaller image size experiments are higher. In smaller image size experiments subjects are asked to walk faster. Because of having higher frame rate in smaller image size we want to evaluate the system performance in the case of fast walking and higher large inter-frame motion. Results show that our algorithm can handle and overcome large inter-frame motion (i.e. high values of max (Δx)) because of using a sampling scheme and motion prediction technique that does not rely on spatial proximity. It will lose a target only if the target changes its motion direction suddenly and walks very fast in the opposite of the predicted direction (e.g. experiments with TF≠ 0).

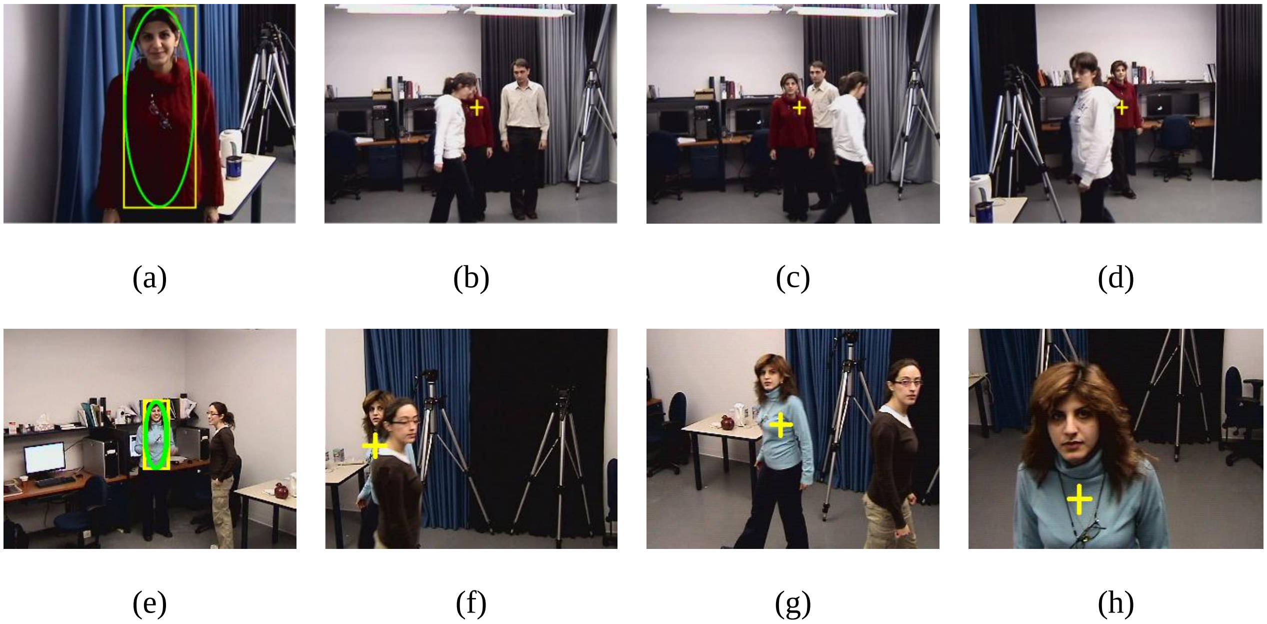


Figure 4.3 Examples of tracking frames for E_{11} (a) to (d), and E_{20} (e) to (h). E_{11} (a) initial model selection, (b)short term occlusion, (c) after short term occlusion, (d) scale variation; E_{20} (e) initial model selection, (f) short term occlusion, (g) after short term occlusion, (h) scale variation.

Our method can also recover the tracking if it loses the object (e.g. experiments with TF $\neq 0$), because of the samples around regions with detected motion. Of course, it is conditional to the object being in the FOV of the camera.

The duration of the experiments is short because the goal of the algorithm will be zooming on target face and capturing it for identification purpose.

Figure 4.2 and Figure 4.3 show the initial target detection and some frame obtained during tracking for different experiments explained in Table 4-2 and Table 4-3.

4.4 Calibration of tracking parameters

As discussed before in our APF-OFS method some parameters that should be tuned for better performance. These parameters are: *b* the number of vertical divisions of samples, bin numbers of *r* and θ the radius and angle in radial histogram, *N* number of samples generated by a Gaussian random function, r_g radius of circular region that samples are generated in, s_s variation of sample size and η the similarity percentage used for moving criteria. Therefore, to determine these seven parameters some experiments are done which are explained in the following. In these experiments the image size is set to be 320 × 240 and the camera is SNC-RZ50. In the experiments, actors have been asked to come close and go far away from the camera during the tracking to have some scale changes. The same scenario as explained before is used for experiments but in each case, tracking performance is compared versus many variations of related parameters.

Calibration of each of the seven parameters is done as explained in the following:

- *b,* the number of vertical division of samples is selected based on ten values varied from 1 to 10. For each value of *b,* 5 experiments are done in which the values of P and TF have been calculated. *b* has direct effect on P and TF, because *b* corresponds to how much accuracy the sample content is compared with the target model. *b* does not have significant effect on other metrics or even on the frame rate.

  Table 4-4 shows the P and TF results for various numbers of vertical sample divisions. The optimum value of b is obtained when P is maximized and TF is minimized which is obtained between 5 to 9 division. More division of *b* does not result in significant changes in P or TF.

Table 4-4 Precision and tracking fragmentation results for various numbers of vertical sample divisions

b	1	2	3	4	5	6	7	8	9	10
P(%)	79	84	87	89	**94**	93	**94**	92	93	91
TF(%)	3.7	2.1	1.8	1.6	**1.3**	**1.2**	1.4	1.5	1.3	1.6

That can be explained by nature of scaling that from one side might increase/decrease the detail that causes to have higher detection precision and on the other side increase/decrease the errors because of insufficient content to compare. Here we select *b* equal to 5. For example, in the initial modeling if the target was small, large scaling of the target will increase the detail and increase the content to compare therefore the error will be increased.

- N_r and N_θ, numbers of bins of radial histogram parameters of r and θ, are selected based on the prediction results of position predictor block since it has direct effect on estimation of camera motion vectors. Radial histogram parameters are required to be quantized since their combination to build a histogram may have many possible values and it takes time to be computed. Therefore, fifteen different experiments are done in which the bin numbers are varied and two metrics of $\mu_{d_{gc}}$ and $\mu_{d_{gp}}$ with τ_3 (the processing time) of experiment are calculated as shown in Table 4-5. There is a tradeoff between quantization to minimize the time processing of τ_3 while minimizing the tracking error. Therefore, an acceptable level of quantization for r and θ will be obtained in E_9 in which τ_3 $\mu_{d_{gc}}$ and $\mu_{d_{gp}}$ are all small and acceptable. A bin with length of 0 is also mentioned for N_r.

Table 4-5 Experimental results versus various bin numbers of radial histogram parameters values.

E	$\mu_{d_{gc}}$	$\mu_{d_{gp}}$	$\bar{\tau}_3\,(ms)$	N_r	N_θ
E_1	0.1611	0.0206	731	801	360
E_2	0.1503	0.0219	516	401	360
E_3	0.1422	0.0243	406	266	360
E_4	0.1363	0.0268	331	201	360
E_5	0.1216	0.0319	225	161	360
E_6	0.1451	0.0209	544	801	180
E_7	0.1367	0.0227	378	401	180
E_8	0.1184	0.0249	318	266	180
E_9	**0.0992**	**0.0275**	**202**	**201**	**180**
E_{10}	0.1208	0.0339	182	161	180
E_{11}	0.1541	0.0295	493	801	120
E_{12}	0.1426	0.0326	352	401	120
E_{13}	0.1495	0.0382	291	266	120
E_{14}	0.1510	0.0409	196	201	120
E_{15}	0.1606	0.0452	158	161	120

- Obviously, generating samples locally around a candidate point gives better performance than just generating samples everywhere. But the question is how far and how many samples should be generated around a candidate point? How much these samples should be scaled? The answer to these three questions will determine *N*, number of samples generated by a Gaussian random function, r_g radius of circular region for generating samples inside of it and s_s variation of sample size. They are determined based on the tracking error results (the four discussed metrics). r_g define the local distribution of samples. It is obtained from maximum displacement of a walking person at normal speed during two consecutive frames with the minimum frame rate. This value is equal to 1/5th of image height. s_s variation of sample size is determined in the same way. It is obtained from maximum scale change of a walking person at normal speed toward the camera during two consecutive frames while minimum frame rate is achieved. This value is obtained to be equal ±5%. *N* has direct effect on the number of possible target candidates and if sufficient samples are not generated good tracking results were never obtained. Although higher values of *N* will increase the processing time of τ_3. Thus, there is a trade off between having enough number of samples and obtaining a good frame rate. It is required that average processing time of generating *N* samples around each candidate pixel calculated. Since the number of candidate pixels are varied in each frame. Thus, in this part we have done 4 experiments in which *N* varies and the four metrics and the average required processing time with the average total number of samples per frame are calculated as shown in Table 4-6.

Table 4-6 Experimental results versus various number of samples.

E	P(%)	TF (%)	$\mu_{d_{gc}}$	$\mu_{d_{gp}}$	$\bar{\tau}_3\,(ms)$	N	$\overline{N}_t$
E_1	85	2.6	0.1316	0.0327	230	25	132
$\mathbf{E_2}$	**94**	**1.1**	**0.1135**	**0.0248**	**415**	**50**	**195**
E_3	94	1.05	0.1063	0.0215	581	75	246
E_4	95	1.03	0.1021	0.0201	702	100	574

To find the optimum number of samples, time processing should be small with high values of precision and smaller tracking fragmentation. Therefore, in experiments between E_2 up to E_4 no significant changes of P and TF are observed while the processing time is increased. While in experiments between E_1 up to E_2 an improvement of P and TF with an acceptable processing time is gained. As a result N is selected to be equal 50.

- η, the similarity percentage used for moving criteria is determined based on the tracking error results (the precision and tracking fragmentation). Different values of η, are tested in different scenarios and the values of P and TF have been calculated. η has direct effect on P and TF, because *η* is corresponds to how much similar the sample content to target model. *η* does not have significant effect on other metrics or even on frame rate. In these experiments, subjects were all asked to be far away from the camera at the initialization and then move close and away from the camera.

Table 4-7 Precision and tracking fragmentation results versus various similarity percentage values

η(%)	100	95	90	**85**	**80**	75	70	65	60
P(%)	76	78	85	**91**	**90**	86	82	73	69
TF(%)	16.1	13.5	7.5	**3.2**	**3.5**	5.7	8.7	14.9	19.2

Table 4-7 shows the P and TF values of ten experiments versus different similarity percentage values. The optimum value of η is obtained equal to either 80 or 85 when P is maximized and TF is minimized. Higher values of η result smaller P and higher TF. That can be explained by nature of scaling that from one side might increase/decrease the detail that causes to have higher detection precision and on the other side increase/decrease the errors because of lack of enough content detail/less content to compare. For the zooming case Table 4-8, the detail content variations of samples are more significant and that is why η is obtained to be 70%.

Table 4-8 Precision and tracking fragmentation results versus various similarity percentage values in the zooming case.

η(%)	100	95	90	85	80	75	**70**	65	60
P(%)	54	57	61	70	79	87	**92**	88	83
TF(%)	14.3	13.1	12.7	9.7	7.6	6.3	**5.8**	6.1	7.2

4.5 Study of system limits

In this part, some experiments are done to determine the limits of the algorithm in both initialization step and whole tracking and camera control system. The SNC-RZ50N with 320 × 240 image size resolution is used. The system has been tested on wide areas at different floors of Pavillon Mackay-Lassonde at École Polytechnique de Montréal. We have done 9 experiments with three different actors. On the initialization step, an actor has been asked to enter to a corridor with length of 30 meters (Figure 4.4 (a) and (b)). The camera is located at the end of the corridor without any zooming. Figure 4.4 shows one of the results of target detection in initialization step. In average of all experiments, target face can be detected at the distance of 7 meters far from the camera by our system (Figure 4.4 (c)).

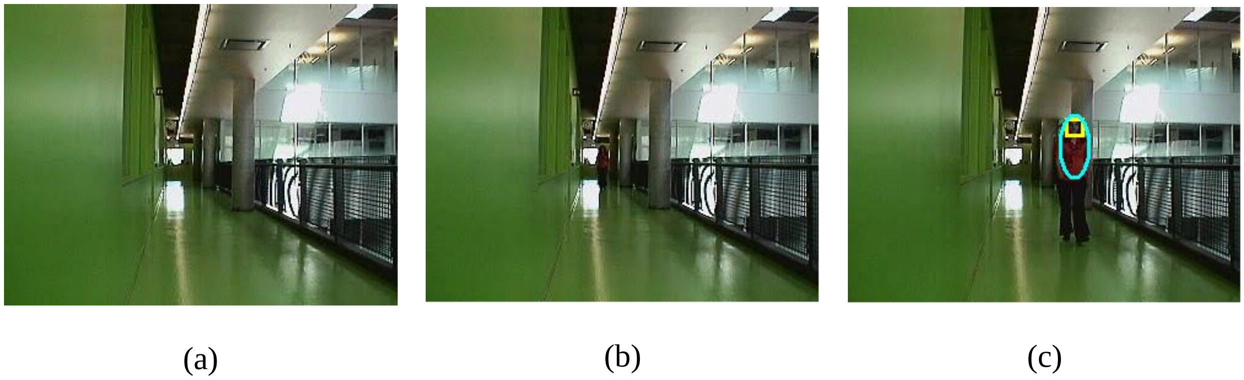
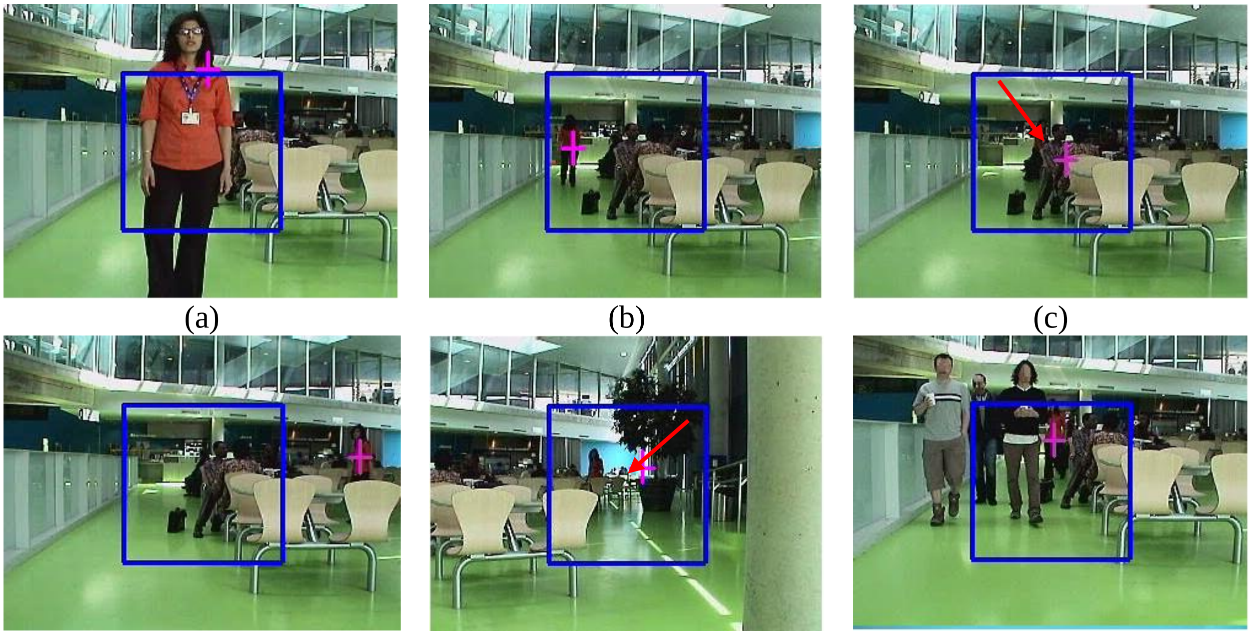

(a) (b) (c)

Figure 4.4 Target modeling in the initialization step (a) reference frame, corridor with 30 meters length, (b) target enters into the corridor, (c) face is detected and target is modeled at the size of 20 × 20 pixels and distance of 7 meters far from the camera.

This average distance can be explained by the minimum face size that can be detected by Viola and Jones which is 20 × 20 pixels. The solution can be initial zooming on the area which results in the target face to be detected even further than this distance. The purpose of the following experiments is to determine if whether the target is lost because of the target size resulting from being far from the camera without zooming, tracking of similar target color objects, and target speed. The general scenario of the experiments is the following. An actor from the frontal view is selected for initial modeling. Then she starts to walk in a wide and large room where the people might sit, stand, and/or walk. She crosses with other people or objects in the scene. She might be occluded during the walking. The actor walks in the scene and go farther from the camera until the camera loses track. The target sample size where the camera loses the target is assumed as the smallest target size that can be tracked by our system. In average, this distance in our experiments is about equal to 12 meters when there is no camera zooming. Figure 4.5 shows the frames where the camera has lost the target because of small target size (14×25pixels in Figure 4.5 (c), 16×24 pixels in Figure 4.5 (e)) and poor quality of the target color at these sizes. Indeed the target color is not distinctive and is almost close to black. Also, in this figure the frames where the system recovers the tracking are shown (Figure 4.5 (d) and (f)). However, in this figure, the target is occluded with other people but the target is larger and its color is distinguishable by the system. Blue rectangle in Figure 4.5 shows a constant distance of 1/6th of image diameter from the image center for target localization comparison.



(a) (b) (c)

(d) (e) (f)

Figure 4.5 Some frames of tracking in a wide scene, (a) initial target detection, (b) target goes far from camera, (c) target lost, (d) tracking recovery, (e) target lost, (f) target recovery with partial occlusion.

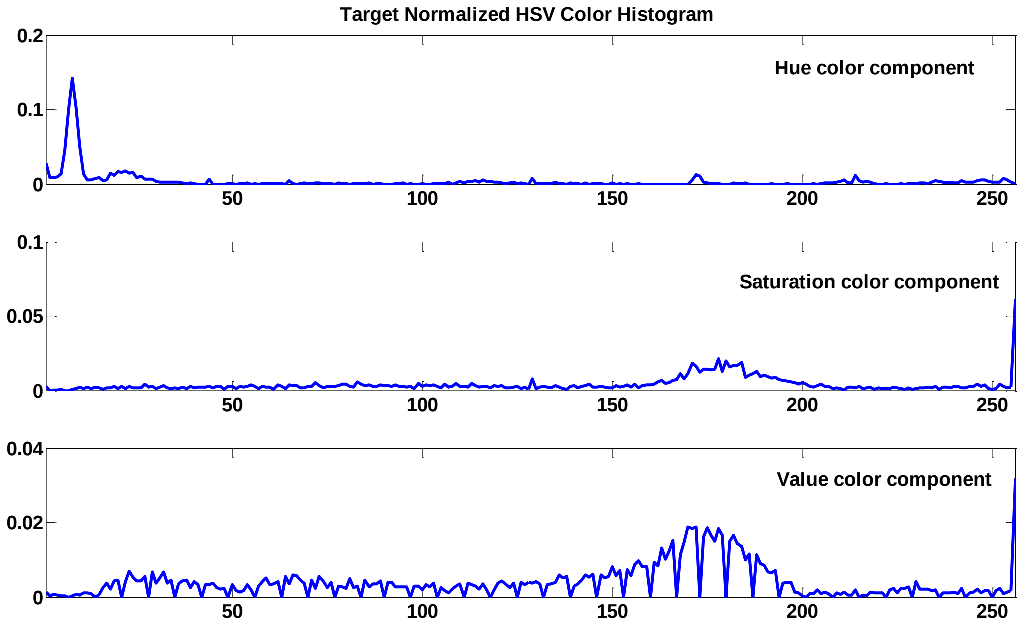


Figure 4.6 Target Normalized HSV color histogram shown in Figure 4.5 (a).

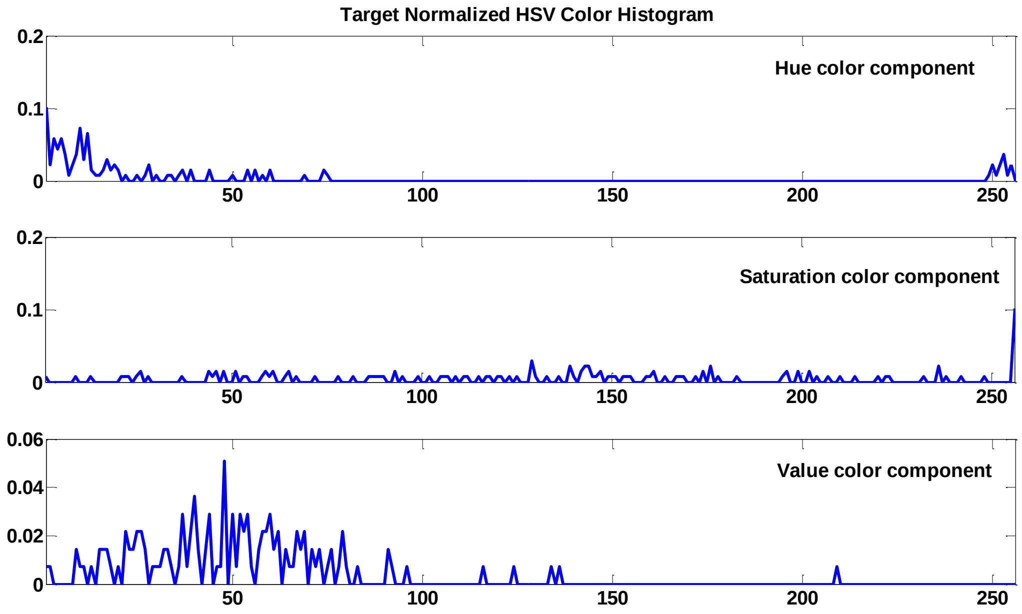


Figure 4.7 Target Normalized HSV color histogram shown in Figure 4.5 (c).

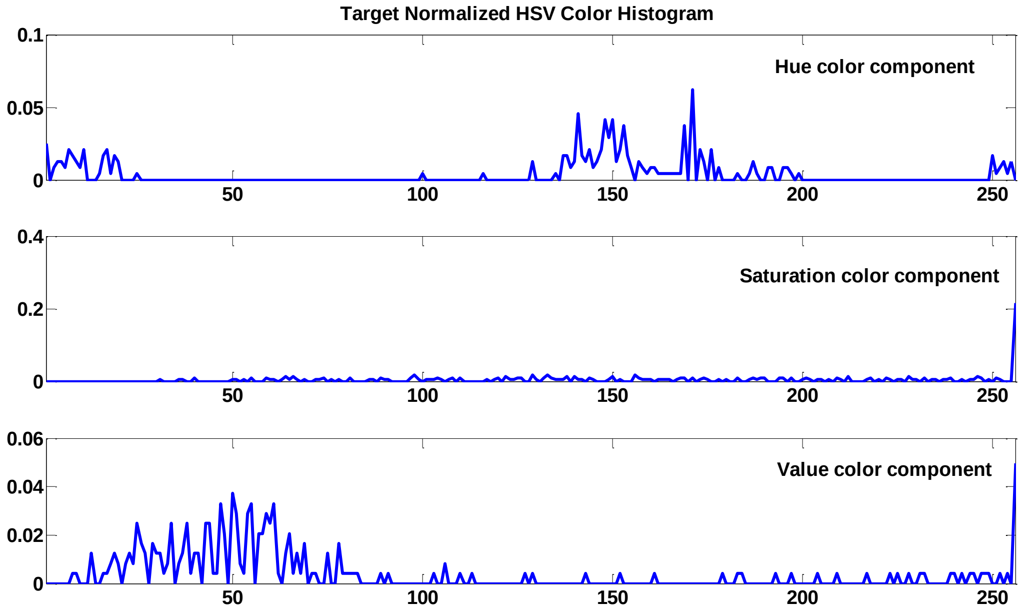


Figure 4.8 Target Normalized HSV color histogram shown in Figure 4.5 (e).

Figure 4.7, and Figure 4.8 show the target normalized HSV color histogram in different frames shown in Figure 4.5. According to the lighting changes of the target color, the value color components of the histogram in these three figures are totally different. In addition because of the small target size in Figure 4.7, and Figure 4.8 the hue and saturation color components are changed.

Figure 4.9 shows the average target detection precision (defined in equation 4-1) versus target sample size in different experiments of 320 × 240 image size. In average when the target sample size is smaller than 600 pixels, the detection precision is less than 35%. Similarly, the detection precision is less than 60% for target size smaller than 1000 pixels. The precision increases when the target size is increased. However, when the target is too large the detection precision is decreased. Because the FOV of the camera is filled with the target and a small movement of the target results in the whole or a portion of the target to be out of camera FOV.

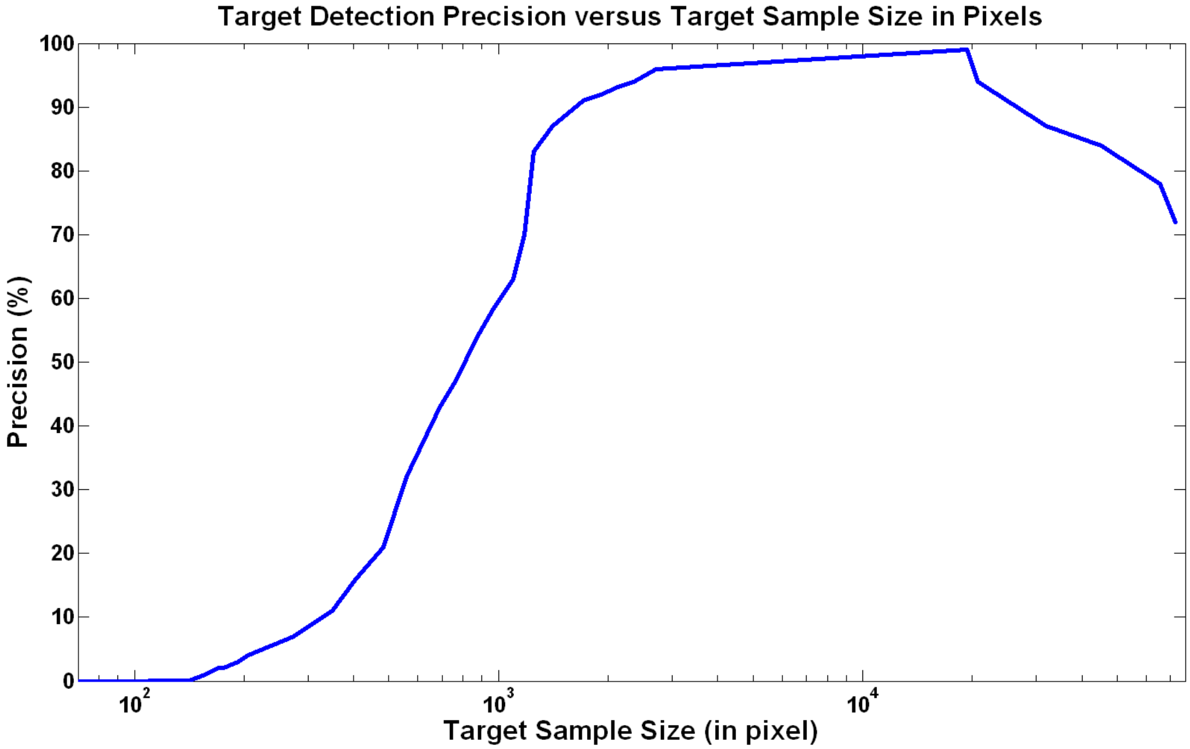
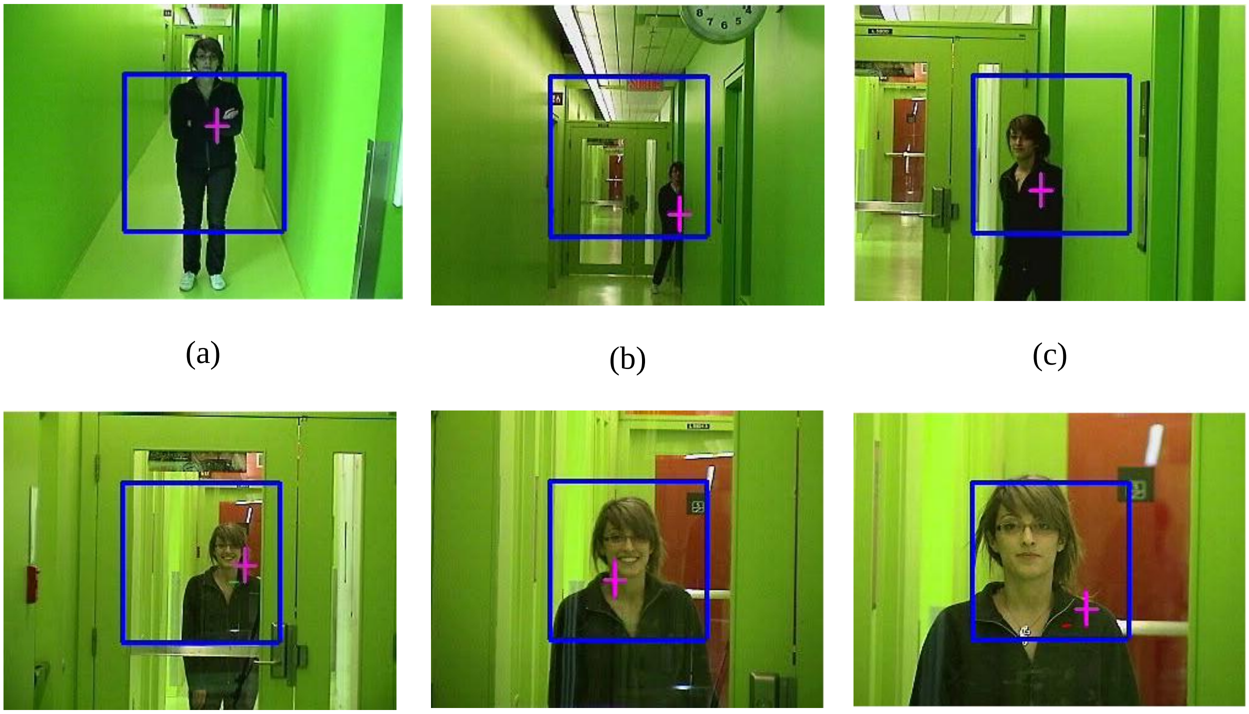


Figure 4.9 Average target detection precision versus target sample size in pixels.

In Figure 4.10, we show that zooming on the target while the target goes very far from the camera results in having larger image size and high quality of target color. The length of corridor is about 25 meters and the target size is 1700 and 360 pixels when it is located almost in the middle and at the end of corridor respectively. Camera zooming enlarges the target size at farther distances. Indeed camera zooming provides more details about the target and can increase the ability of object tracking in farther distances or smaller objects. However, zooming from one side enhances the details which results in higher detection precision and on the other side increases the errors because of insufficient content to compare. Therefore, when a higher zoom level is applied the target score is in average decreased. This experiment also shows the robustness of our method due to the scaling changes.



(a) (b) (c)

(d) (e) (f)

Figure 4.10 Target tracking and zooming frames, (a) initial target detection, (b)-(d) multi level zooming on the target which is in the middle of corridor, (e) and (f), multi level zooming on the target which is at the end of corridor.

Previously in Figure 3.9 (a) we have shown that another similar color object is tracked when the target is occluded. Here in Figure 4.11 (b) - (d) it is shown that the target is wrongly detected because of the small size of the real target and its poor color quality. In addition similar to Figure 4.6, 4.7 and 4.8 the target histogram has been changed because of light changes and target size. Therefore it results to false target detection.

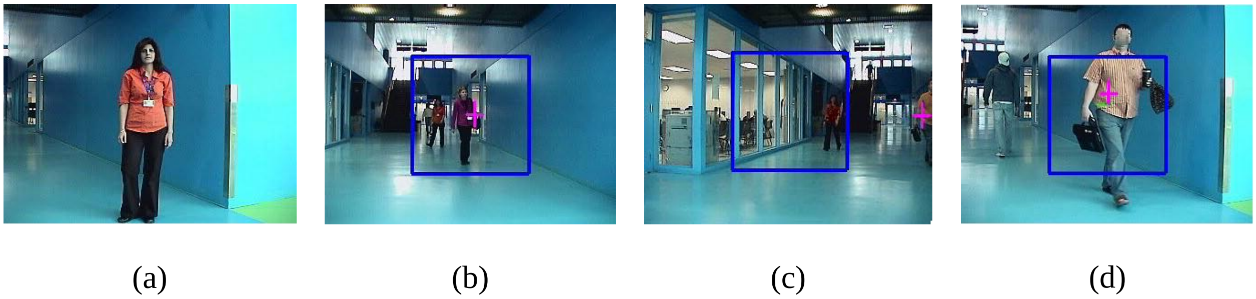


Figure 4.11 Target tracking frames, (a) initial target detection, (b) false target detection, (c) false target detection, (d) target lost.

In the last set of experiments, we have calculated the average maximum target speed in which the system can track the target without losing it versus target size in pixels and the system frame rate. We have done 24 experiments classified in 6 classes. Each class belongs to a specified target size. In these experiments, an actor has been asked to walk with various speed on a straight line at different distances far from camera but parallel to the image plane from right to left or left to right directions. We asked from each actor to use a chronometer to record the time that takes to walk a specified linear path (5 meters length). The image size is 320 × 240.

Table 4-9 Average maximum target speed tracked by the system versus target size, target distance from camera and system frame rate

Experiment	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Average Target Size (pixels)	15×45	20×60	30×90	40×120	70×210	80×240
Average Target distance (m)	10	8	5.5	4.2	2.3	1.5
Average Target Speed (m/s)	2.5	1.9	1.5	1.1	0.8	0.5
Average Frame Rate (fps)	12.3	12.2	12.6	12.5	12.1	12.7

Table 4-9 shows the average maximum target speed that can be tracked by our system versus different target size. The average target size, target distance from the camera, average target speed and average frame rate in each class are calculated. The average target speed is computed by division of the distance it walks over the total time. The average human normal speed is about 4.51 km/h which is 1.25 m/s (Wikipedia, 2010, Online accessed 17-May-2010).

The system can track a target with higher speed when it is smaller and far from the camera. Similarly when the target is close to camera since the camera FOV is limited, the camera needs more time to center on the target. Therefore, a smaller maximum target speed is obtained. It is also true when the camera zooms on the target which limits the FOV and also the target speed.

4.6 Comparison of proposed tracking system performance with well-known tracking methods

The goal of this section is to study and compare the performance of the proposed Adaptive Particle Filter with Optical Flow Samples method with the performance of well-known tracking methods such as classical particle filter (PF) tracker (Michael Isard & Blake, 1998) and Kanade-Lucas-Tomasi (KLT) Feature Tracker (Birchfield, 1997). We have used the optimized parameters from the previous section for APF-OFS method. We have done ten experiments with one IP camera (SNC-RZ50) in 320×240 image resolution. The experiments are described in Table 4-10. The experiments are done in a room with dimension of 6 × 4 meters. The distances of the initial model position from camera are written in Table 4-10. To avoid the camera control effect from tracking performance, KLT and PF are applied separately on the recorded experiments obtained from APF-OFS tracker method. To take the results of all three trackers, all the experiments are performed first by applying the APF-OFS, and the obtained sequences are recorded to be tested by KLT and PF tracker methods.

Table 4-10 Experiments to compare results of APF-OFS tracker with KLT and PF trackers.

Experiments	Camera Model	Image Size	Initial Model Position
E_1	SNC-RZ50	320×240	Near
E_2	SNC-RZ50	320×240	Near
E_3	SNC-RZ50	320×240	Near
E_4	SNC-RZ50	320×240	Near
E_5	SNC-RZ50	320×240	Far
E_6	SNC-RZ50	320×240	Far
E_7	SNC-RZ50	320×240	Far
E_8	SNC-RZ50	320×240	Far
E_9	SNC-RZ50	320×240	Middle
E_{10}	SNC-RZ50	320×240	Middle

Near ~ 1 meter, Far ~ 6 meters, Middle ~ 3 meters.

Table 4-11 and Table 4-12 show the result of three trackers on three metrics and the system frame rate for all experiments. The mean and variance of d_{gc} and d_{gp}, for all experiments are shown. TF is not evaluated here since it is the same for all three trackers.

For the KLT tracker, we keep the default parameters available in version 1.3.4 of the source code. Thus we keep extraction of maximum 40 number of feature in each frame. It is experimentally tested that in our experiments more numbers of features are useless and provide us the same results. To obtain the KLT results, in the first frame, the target is introduced to the KLT tracker by cropping the target silhouette image Figure 4.12. Then the KLT tracker is applied to this target area to extract 40 features. We have modified the KLT tracker to be applicable with our scenarios since the background is changed resulting from camera panning and tilting. We find the center of a rectangular window that encloses all the features. This is assigned as the tracked target position by KLT. Then for the next frame based on the maximum value of all max(Δx) values that is obtained from Table 4-11 (e.g. 228 pixels), we find the correspondent features inside a region with the center of previous rectangular window region and width of 228 pixels. If no feature is founded by KLT, the previous target position and its size is used. In each frame we do not generate new features.

Table 4-11 Experimental results to compare three tracker performances (E_1 to E_5).

E	TM	P(%)	$\mu_{d_{gc}}$	$\sigma^2_{d_{gc}}$	$\mu_{d_{gp}}$	$\sigma^2_{d_{gp}}$	*FR(fps)*	*NF*	*min(Δx)*	*max(Δx)*
E_1	APF-OFS	**96**	**0.1124**	**0.0273**	**0.0310**	**0.0017**	10.29	897	0.015	0.45
	PF	88	0.1336	0.0657	0.0498	0.0089				
	KLT	63	0.4121	0.1132	0.1426	0.1001				
E_2	APF-OFS	**100**	**0.0930**	**0.0227**	**0.0242**	**0.0018**	12.72	908	0.007	0.26
	PF	85	0.1422	0.0814	0.0562	0.0120				
	KLT	66	0.4036	0.1015	0.1306	0.1121				
E_3	APF-OFS	**95**	**0.1156**	**0.0246**	**0.0362**	**0.0028**	9.57	851	0.022	0.29
	PF	83	0.1669	0.0561	0.0597	0.0132				
	KLT	59	0.4251	0.1251	0.1573	0.1069				
E_4	APF-OFS	**94**	**0.1184**	**0.0259**	**0.0392**	**0.0025**	9.19	902	0.010	0.57
	PF	82	0.1748	0.0632	0.0613	0.0970				
	KLT	58	0.4284	0.1134	0.1679	0.1278				
E_5	APF-OFS	**98**	**0.1015**	**0.0231**	**0.0246**	**0.0019**	11.63	916	0	0.42
	PF	84	0.1555	0.0891	0.0622	0.0886				
	KLT	65	0.3941	0.1204	0.1411	0.0992				

E: Experiments, P:Precision, $\mu_{d_{gc}}$: mean of d_{gc}, $\mu_{d_{gp}}$: mean of d_{gp}, $\sigma^2_{d_{gc}}$: variance of d_{gc}, $\sigma^2_{d_{gp}}$: variance of d_{gp}, FR: System frame rate and NF: Number of frames, *min(Δx):* minimum normalized motion vector length, *max(Δx)*: maximum normalized motion vector length, TM: Tracking method.

Indeed in each frame a part of background is involved in the rectangular window size that contains the target and regeneration of features result in detection of new features that do not belong to the target.

Table 4-12 Experimental results to compare three tracker performances (E_6 to E_{10}).

E	TM	P(%)	$\mu_{d_{gc}}$	$\sigma^2_{d_{gc}}$	$\mu_{d_{gp}}$	$\sigma^2_{d_{gp}}$	*FR(fps)*	*NF*	*min(Δx)*	*max(Δx)*
E_6	APF-OFS	**95**	**0.1209**	**0.0261**	**0.0391**	**0.0022**	10.84	903	0.017	0.475
	PF	79	0.1842	0.0728	0.0695	0.0104				
	KLT	69	0.3849	0.0984	0.1408	0.0973				
E_7	APF-OFS	**96**	**0.1238**	**0.0217**	**0.0399**	**0.0025**	10.08	844	0.005	0.547
	PF	86	0.1398	0.0547	0.0553	0.0111				
	KLT	60	0.4173	0.0871	0.1654	0.0951				
E_8	APF-OFS	**93**	**0.1279**	**0.0200**	**0.0406**	**0.0027**	10.03	876	0.020	0.522
	PF	77	0.1872	0.0856	0.0780	0.0974				
	KLT	62	0.4044	0.0895	0.1560	0.0976				
E_9	APF-OFS	**97**	**0.1086**	**0.0242**	**0.0263**	**0.0016**	11.35	910	0.012	0.432
	PF	81	0.1770	0.0572	0.0691	0.0897				
	KLT	61	0.4096	0.0906	0.1527	0.1010				
E_{10}	APF-OFS	**96**	**0.1116**	**0.0209**	**0.0280**	**0.0020**	9.66	859	0.007	0.285
	PF	84	0.1680	0.0702	0.0679	0.0115				
	KLT	57	0.4304	0.1021	0.1625	0.1155				

E: Experiments, P:Precision, $\mu_{d_{gc}}$: mean of d_{gc}, $\mu_{d_{gp}}$: mean of d_{gp}, $\sigma^2_{d_{gc}}$: variance of d_{gc}, $\sigma^2_{d_{gp}}$: variance of d_{gp}, FR: System frame rate and NF:Number of frames, *min(Δx):* minimum normalized motionvector length, *max(Δx)*: maximum normalized motionvector length, TM: Tracking method.

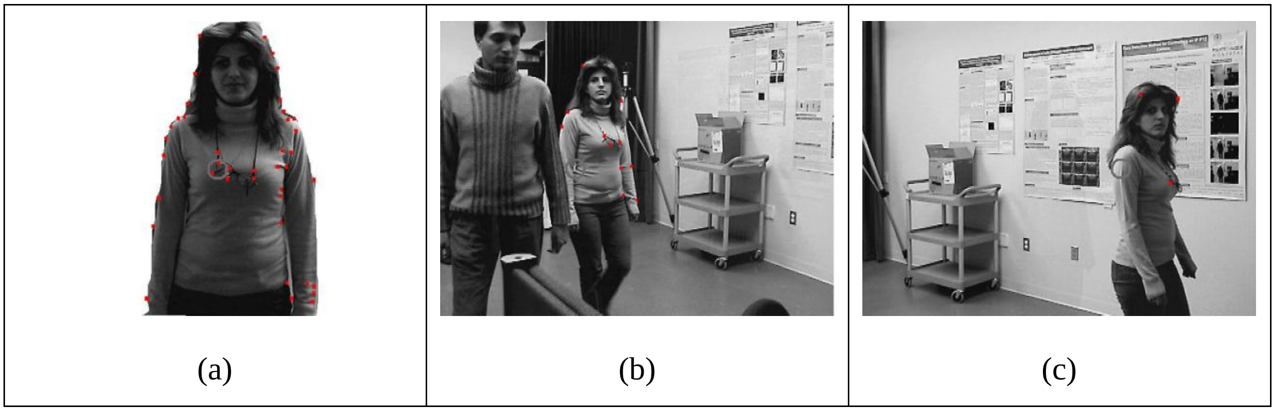
For PF the target is modeled in the same way as APF-OFS. The sample generator of PF generates 250 samples by a Gaussian random function in a circular region with radius of image height around the image center. The number of PF samples are obtained from average frame rate processing of our method which is 10.45 (*fps*) for smaller image size. We have obtained that 250 is the maximum number of samples that should be used to have such a frame rate by applying our PF in the video sequences. This number of samples is not large enough to cover everywhere in the whole image. Since the whole image should be covered the radius of the circular region in sample generator is assumed to be the height of the image.

The size of these samples (width and height of the ellipses) is varied randomly $\pm 5\%$ of the previous target size similar to our APF-OFS tracker. This value is obtained experimentally based on the maximum or minimum displacement of the target in the case that the target is approaching or moving back from the camera between two consecutive frames.

In the comparison of our method with PF and KLT feature tracker, PF outperforms the KLT tracker and APF-OFS outperforms both of the PF and KLT. For the KLT feature tracker it is because of the appearance that changes too much between two images and all features are often lost since they cannot be tracked.

It is also obvious from Table 4-13 which shows the average tracking results of APF-OFS tracker against PF and KLT results. With APF-OFS larger P, smallest $\mu_{d_{gc}}$, $\sigma^2_{d_{gc}}$, $\mu_{d_{gp}}$ and $\sigma^2_{d_{gp}}$ are obtained.

Figure 4.13 and Figure 4.14 show the initial model selection and some frames obtained during experiments explained in Table 4-10. There are some scale change examples in Figure 4.13 (l) and (f) and Figure 4.14 (d). Also short term occlusions are occurring several times in different experiments as shown in second and third columns of Figure 4.13 and Figure 4.14.



(a) (b) (c)

Figure 4.12 (a) Target introduced to the KLT feature tracker at frame 1, detected and tracked KLT features of target at (b) Frame 22, (c) Frame 23.

Table 4-13 Average tracking results of APF-OFS tracker with PF and KLT trackers

TM	P(%)	$\mu_{d_{gc}}$	$\sigma^2_{d_{gc}}$	$\mu_{d_{gp}}$	$\sigma^2_{d_{gp}}$
APF-OFS	**96.02**	**0.1134**	**0.0236**	**0.0329**	**0.0022**
PF	82.88	0.1629	0.0696	0.0629	0.0440
KLT	62.06	0.4110	0.1041	0.1517	0.1053

By comparison of APF-OFS and PF methods with KLT feature tracker, both have a higher target tracking precision P with less tracking fragmentation. The localization of the target is better ($\mu_{d_{gc}}$ and $\mu_{d_{gp}}$ values are smaller). It means the location of the target is closer to the ground truth than the KLT feature tracker results. This result is explained by the particle filter characteristic that well distributes samples at each frame in different positions and various scale sizes everywhere.

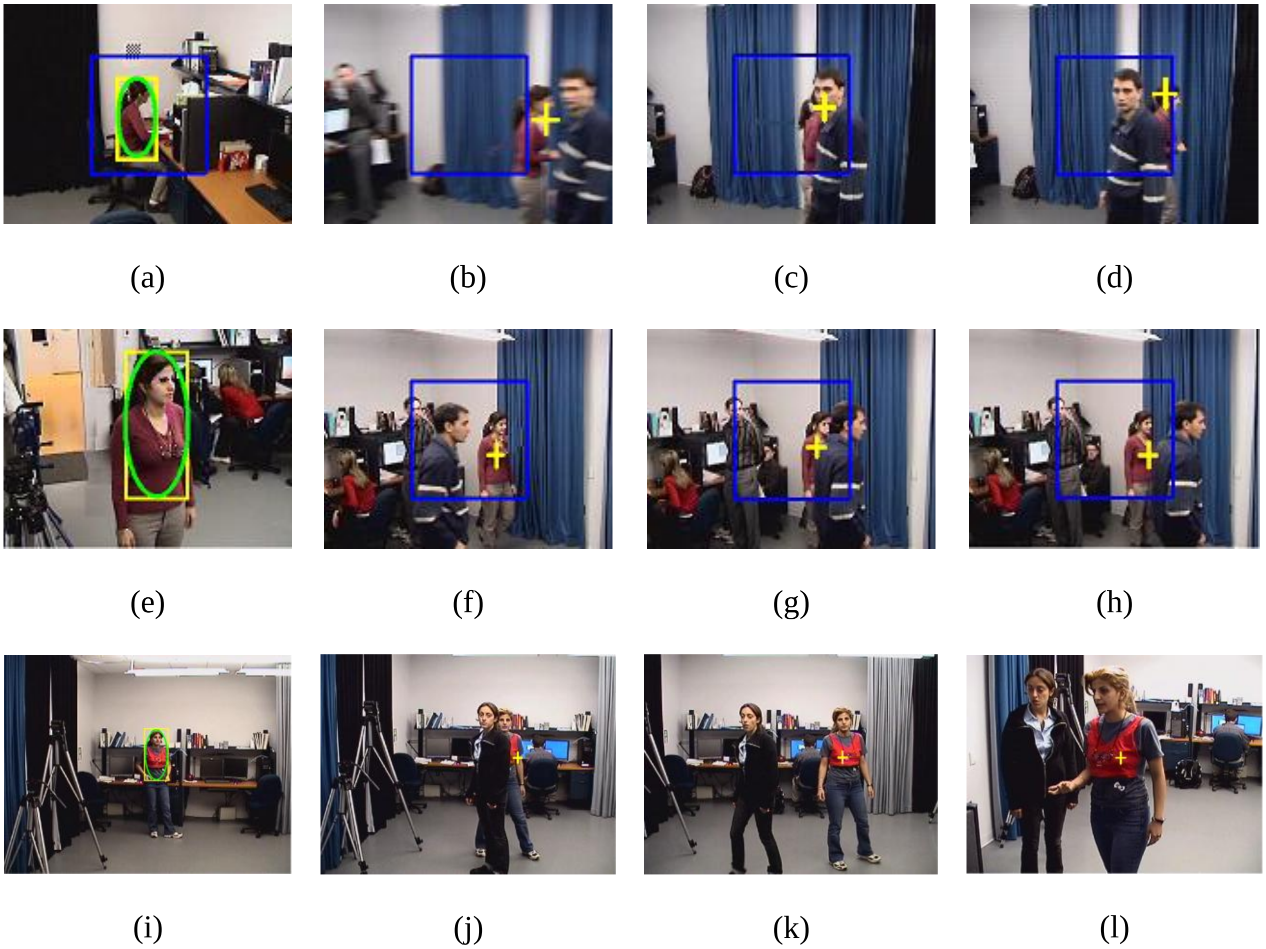
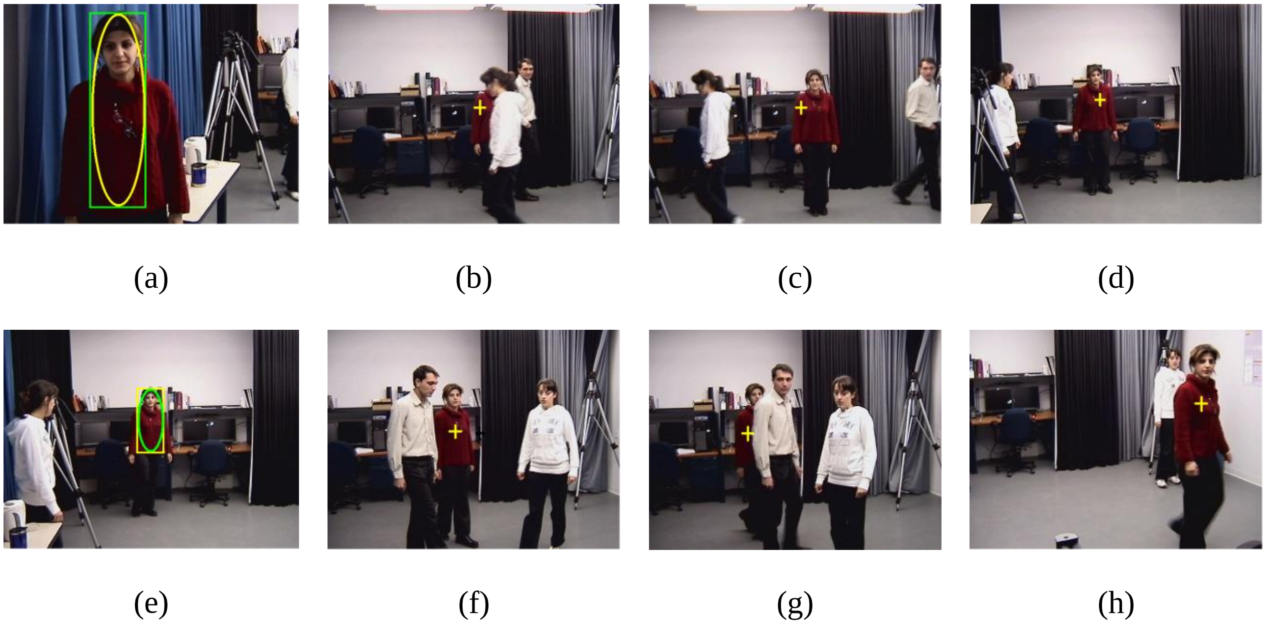


Figure 4.13 Examples of tracking frames for E_9 (a) to (d), E_1 (e) to (h), and E_5 (i) to (l). E_9 (a) initial model selection, (b)before short occlusion, (c) short term occlusion, (d) after short term occlusion; E_1 (e) initial model selection, (f) before short term occlusion, (g) short term occlusion, (h) after short term occlusion; E_5 (i) initial model selection, (j) short term occlusion, (k) after short term occlusion, (l) scale variation.

That gives more candidate regions to the tracker. Localization accuracy is thus, improved. When localization in both APF-OFS and PF fails, it is because of similarity or closeness of the color histogram of the target with other samples that are generated. It is even worst for PF that generates samples everywhere and not necessary around detected motion regions or previous target position. This also causes to have more false positive samples or less precision (smaller P values) than the APF-OFS tracker. Indeed optical flow has an important role that causes to remove this type of false positive samples. In addition it results in the target being usually located within 1/6th of image diameter from the image center.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.14 Examples of tracking frames for E_2 (a) to (d), and E_6 (e) to (h). E_2 (a) initial model selection, (b)short term occlusion, (c) after short term occlusion, (d) scale variation; E_6 (e) initial model selection, (f) short term occlusion, (g) after short term occlusion, (h) scale variation.

Tracking with our camera setup and context requires using features that are not too scale and pose dependent, because large instantaneous changes can occur between two frames. Figure 4.12 shows an example of KLT features detection and tracking in one of the video sequences. Because of the large scale and pose changes, all features may be lost between two frames. This causes target losses. Although, using color information seems simple, it is more suited to our application because it is a more generic description.

For PF a larger number of samples is needed because the PF samples must cover almost all the image. As discussed previously on one side, with higher number of samples the tracking performance could be improved because of having more target candidates to compare. But it also increases the processing time. On the other side, this will result to have a lower frame rate and more tracking fragmentation or even losing target. Thus, to minimize this effect the number of samples is set first and then compared to our APF-OFS performance.

4.7 Study of tracking algorithm on camera control performance

The basis of our APF-OFS comes from a PF method. As it was presented in the previous section, PF results are close to APF-OFS but APF-OFS outperforms PF, because of better repartitioning of samples. Therefore, here we compare the performance of our proposed APF-OFS method with PF on the camera control to observe how much the proposed APF-OFS improves tracking and camera control.

We have done ten experiments with two cameras of the same model (SNC-RZ25) in smaller image resolution. The experiments are described in Table 4-14. The experiments are done in a room with dimension of 6 × 4 meters. The distances of the initial model position from camera are written in Table 4-14. Two IP cameras are located besides together and connected directly to the network adaptor of the workstation (called direct mode) of two same model computers. PF and APF-OFS run on each computer separately and the tracking results obtained from each camera are recorded to extract their ground-truth manually for performance evaluation.

Table 4-15 shows the results of APF-OFS and PF trackers on camera control performance. Both cameras and both tracking methods are initialized with target at the same time and they track the target simultaneously. The four metrics with frame rate are calculated for each tracker separately. Number of samples in PF is set to be 250 as discussed previously which is the maximum number of samples allows the target tracking with almost similar frame rate as APF-OFS.

In the comparison of our method with PF tracker, APF-OFS outperforms the PF. It is also obvious from Table 4-13 which shows the average tracking results of APF-OFS tracker against PF trackers results. With APF-OFS largest P, smallest TF, $\mu_{d_{gc}}$, $\sigma^2_{d_{gc}}$, $\mu_{d_{gp}}$ **and** $\sigma^2_{d_{gp}}$ are obtained while higher frame rate is gained.

Table 4-14 Experiments to compare camera control performance of APF-OFS and PF trackers.

Experiments	**Camera Model**	**Image Size**	**Initial Model Position**
E_1	SNC-RZ25	320×240	Near
E_2	SNC-RZ25	320×240	Near
E_3	SNC-RZ25	320×240	Near
E_4	SNC-RZ25	320×240	Near
E_5	SNC-RZ25	320×240	Far
E_6	SNC-RZ25	320×240	Far
E_7	SNC-RZ25	320×240	Far
E_8	SNC-RZ25	320×240	Far
E_9	SNC-RZ25	320×240	Middle
E_{10}	SNC-RZ25	320×240	Middle

Near ~ 1 meter, Far ~ 6 meters, Middle ~ 3 meters.

Figure 4.15 shows examples of tracking frames in PF and APF-OFS, for one of the experiments. As it is shown because of lack of position prediction in PF tracker it has false target detection which is obvious in Figure 4.15 (a) and (c). In addition since APF-OFS distributes samples where the motion is extracted by optical flow, it has a well distribution of samples. It causes better localization of target. But PF only distributes samples everywhere Figure 4.15 (c) and (e).

Table 4-15 Experimental results to evaluate camera control performance versus APF-OFS and PF trackers.

E	TM	P(%)	TF(%)	$\mu_{d_{gc}}$	$\sigma^2_{d_{gc}}$	$\mu_{d_{gp}}$	$\sigma^2_{d_{gp}}$	FR(fps)	NF	min(Δx)	max(Δx)
E1	APF-OFS	95	2.1	0.1217	0.0213	0.0321	0.0019	10.47	897	0.027	0.50
	PF	82	36	0.1662	0.0526	0.0881	0.0602	10.06	890	0.025	0.51
E2	APF-OFS	93	2.6	0.1361	0.0117	0.0362	0.0018	9.68	869	0.012	0.38
	PF	81	39	0.1731	0.0614	0.0965	0.0753	9.42	872	0.0175	0.36
E3	APF-OFS	96	2	0.1187	0.0171	0.0317	0.0021	10.62	904	0.007	0.43
	PF	82	37	0.1682	0.0581	0.0805	0.0662	10.33	899	0.010	0.49
E4	APF-OFS	94	2.3	0.1205	0.0112	0.0396	0.0022	10.11	888	0.010	0.32
	PF	79	41	0.1743	0.0475	0.0903	0.0617	9.74	885	0.012	0.35
E5	APF-OFS	92	2.8	0.1274	0.0526	0.0412	0.0020	9.66	893	0.005	0.29
	PF	83	38	0.1673	0.0626	0.0822	0.0723	9.27	895	0.007	0.31
E6	APF-OFS	91	2.8	0.1306	0.0375	0.0403	0.0017	8.79	862	0.012	0.40
	PF	82	39	0.1696	0.0735	0.0872	0.0824	8.60	859	0.005	0.39
E7	APF-OFS	92	2.9	0.1262	0.0261	0.0398	0.0019	9.16	738	0.007	0.36
	PF	78	42	0.1686	0.0429	0.0915	0.0936	8.83	740	0.012	0.34
E8	APF-OFS	94	2.7	0.1134	0.0159	0.0370	0.0021	9.85	801	0.015	0.47
	PF	83	38	0.1563	0.0712	0.0872	0.0730	8.74	800	0.017	0.44
E9	APF-OFS	95	2.1	0.1083	0.0266	0.0357	0.0016	9.69	865	0.020	0.51
	PF	77	44	0.1749	0.0518	0.0962	0.0593	8.89	863	0.015	0.51
E10	APF-OFS	93	2.5	0.1178	0.0280	0.0334	0.0023	9.58	796	0.017	0.41
	PF	80	41	0.1620	0.0667	0.0919	0.0841	9.37	789	0.015	0.52

TM: Tracking method.

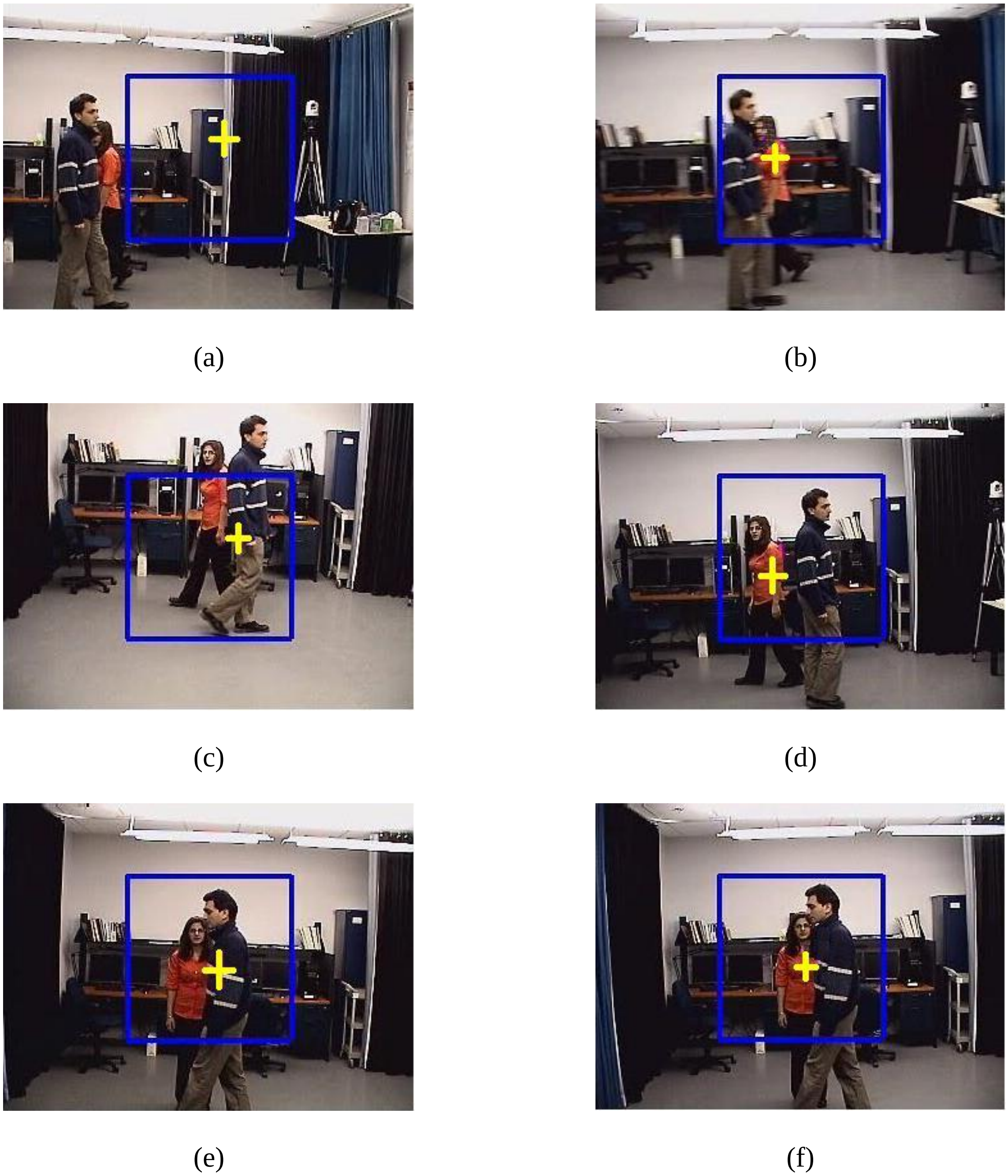
Table 4-16 Average tracking results of APF-OFS and PF tracker.

TM	P(%)	TF(%)	$\mu_{d_{gc}}$	$\sigma^2_{d_{gc}}$	$\mu_{d_{gp}}$	$\sigma^2_{d_{gp}}$	FR(fps)
APF-OFS	**93.53**	**2.46**	**0.1221**	**0.0248**	**0.0367**	**0.0020**	**9.74**
PF	80.74	39.4	0.1680	0.0588	0.0892	0.0728	9.31

By comparison of APF-OFS and PF methods, APF-OFS has a higher target tracking precision P with less tracking fragmentation. The localization of the target is better ($\mu_{d_{gc}}$ and $\mu_{d_{gp}}$ values are smaller). It means the location of the target is closer to the ground truth than the PF tracker results. Since the nature of both methods are similar the results difference is explained by first, better distribution of samples in APF-OFS that uses motion pixels detected by optical flow as candidate sample positions. It improves the localization accuracy in APF-OFS. For example the target is not moving in the same direction, it might be detected from the motion-based samples if it is still in the FOV of camera. When localization in both APF-OFS and PF fails, it is because of similarity or closeness of the color histogram of the target with other samples that are generated. It is even worst for PF that generates samples everywhere and not necessary around detected motion regions or previous target position.

By using a sampling scheme and combining it with a motion predictor, we can handle random motion between frames, as long as the target position is well predicted, and its appearance does not change significantly. The motion predictor is used to compensate the three delays τ_1, $\tau_{2,}$ and τ_3, which may cause the target to exit the FOV.

Occlusions are handled in the same way. However, when the object is occluded, another similar object will be tracked (the most likely candidate blob) until the occlusion ends. This could cause the real target to become out of the FOV of the camera. Figure 4.15 shows some examples of short-term occlusion handling. The proposed method can handle it in this case. In the reported experiments, occlusions did not cause difficulties for scene that are not crowded and people with distinctive clothing.



(a) (b)

(c) (d)

(e) (f)

Figure 4.15 Examples of tracking frames for E_6, left column PF, right column APF-OFS at approximately the same time; (a) PF short term occlusion, (b) APF-OFS short term occlusion, (c)PF after short term occlusion, (d) APF-OFS after short term occlusion, (e) PF partial occlusion, (f) APF-OFS partial occlusion.

The duration of the experiments is short because the goal of the algorithm will be zooming on target face and capturing it for identification purpose.

The second reason comes from the position predictor block, in which the next target position is estimated and the camera can compensate the three delays discussed before to center on the target while PF cannot perform such a thing. In addition it results in the target being usually located in a constant distance within 1/6th of image diameter from the image center in APF-OFS.

The number of PF samples is obtained from average frame rate processing of our method for smaller image size. With larger number of samples higher time processing and lower frame rate is obtained. It is obvious that with the selected number of samples APF-OFS outperforms PF while the frame rate in APF-OFS is also higher.

4.8 Study of zoom control on tracking system performance

The goal of this section is to study the effect of zoom control on tracking system performance.

Table 4-17 Experiments to study zooming control effect on APF-OFS tracker performance.

Experiments	Camera Model	Image Size	Initial Model Position
E_1	SNC-RZ50	320×240	Near
E_2	SNC-RZ50	320×240	Near
E_3	SNC-RZ50	320×240	Near
E_4	SNC-RZ50	320×240	Near
E_5	SNC-RZ50	320×240	Far
E_6	SNC-RZ50	320×240	Far
E_7	SNC-RZ50	320×240	Far
E_8	SNC-RZ50	320×240	Far
E_9	SNC-RZ50	320×240	Middle
E_{10}	SNC-RZ50	320×240	Middle

Near ~ 1 meter, Far ~ 6 meters, Middle ~ 3 meters.

We have done ten experiments with SNC-RZ50 in 320×240 image resolution. We recorded all the experiments to extract their ground-truth manually for performance evaluation. The experiments are described in Table 4-17. The experiments are done in a room with dimension of 6 × 4 meters. The distances of the initial model position from camera are written in Table 4-17.

Table 4-18 shows the result of four metrics, the system frame rate and the maximum zooming level that obtained for all experiments. The algorithm is implemented on an Intel Xeon(R) 5150 in C++ using OpenCV. We have limited the maximum zooming level of the camera (SNC-RZ50) by a factor of *m* to avoid target lost. Indeed we zoom in the camera on the target up to four times (e.g. *m=4*) larger than the initial size. However this value is obtained from maximum far distance in our experiments which is 6 meters. It is obvious that during the zooming, if the camera is close to the person less zooming level is required. Here during the tracking if the system can zoom on the target face, the target face image is recorded.

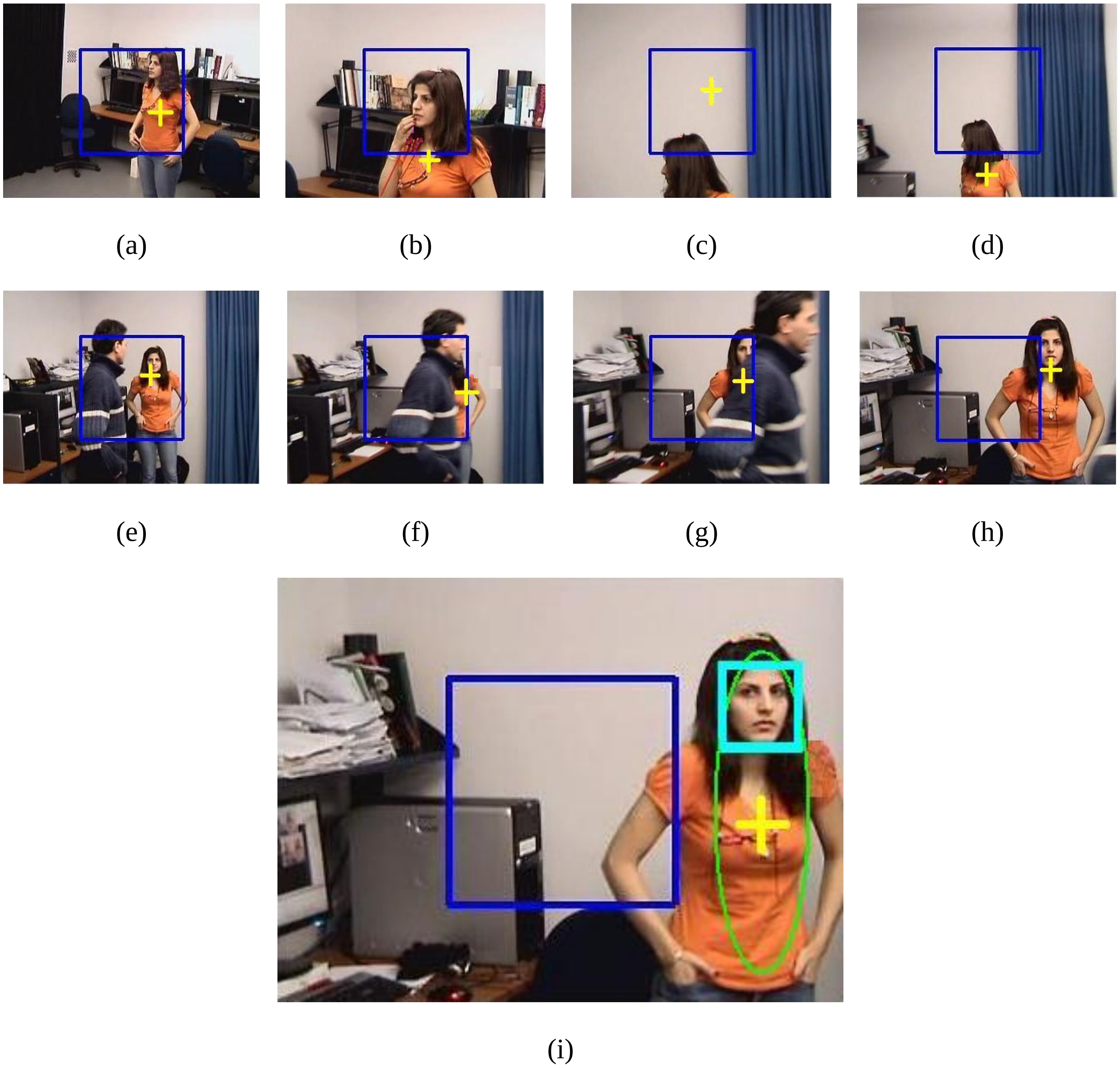
Table 4-18 Experimental results to study zooming camera control on tracking performance, m: zooming level, m_t: zooming level in which target face is detected.

E	P(%)	TF(%)	$\mu_{d_{gc}}$	$\sigma^2_{d_{gc}}$	$\mu_{d_{gp}}$	$\sigma^2_{d_{gp}}$	FR(fps)	NF	min(Δx)	max(Δx)	m	m_t
E_1	94	13	0.2211	0.0912	0.0462	0.0140	8.62	674	0.005	0.32	3	3
E_2	92	21	0.2452	0.1056	0.0417	0.0162	7.59	594	0.015	0.24	3	2
E_3	97	7	0.1961	0.0875	0.0352	0.0137	8.41	631	0.007	0.13	4	3
E_4	95	12	0.2086	0.0762	0.0439	0.0155	8.83	645	0.012	0.42	4	3
E_5	90	26	0.2445	0.0945	0.0527	0.0149	7.06	522	0.002	0.49	3	3
E_6	91	19	0.2487	0.0863	0.0536	0.0176	7.69	510	0.017	0.44	4	1
E_7	94	9.5	0.2166	0.0922	0.0416	0.0144	8.11	497	0.010	0.30	3	2
E_8	88	26.2	0.2542	0.0974	0.0508	0.0150	7.93	505	0.015	0.28	3	3
E_9	93	11	0.2114	0.0944	0.0427	0.0146	7.49	420	0.025	0.51	3	-
E_{10}	92	12	0.2217	0.0893	0.0419	0.0155	7.35	637	0.005	0.19	4	4

For d_{gc} and d_{gp}, we show the mean and variance of all experiments. In the comparison of results of Table 4-18 and Table 4-15, in Table 4-18, there is more target lost (e.g. TF is larger), more distance error ($\mu_{d_{gc}}$ and $\mu_{d_{gp}}$) and the results of precision are worst (P is smaller). Indeed it is a comparison of the APF-OFS performance before and after zooming. Camera control with both moving and zooming has lower frame rate than just moving. It is because of lens adjustment that requires additional time than only moving. The last two columns in Table 4-18 shows the maximum zooming level that occurred in an experiment, *m*, and the zooming level in which the target face has been detected correctly, m_t. *m* and m_t can vary from 1 to 4 and be different from each other depending on the target movement.

For example *m* in E_6 is equal to 4 while m_t is equal to 1 which means that the camera did different zooming level to find the target face but target face criteria discussed before were approved when the target was close to camera. However *m* in E_9 is equal to 3 while the target face detection criteria were not approved and the target face is not detected.

In the comparison of the APF-OFS performance before and after zooming, slower frame rate is obtained because of adding another camera control task (zooming) takes more time than just camera moving (zooming takes between 1.6 ~ 2.5 times longer than moving). It causes in zooming to have less frame rate than moving. In addition camera zooming reduces the camera FOV, it means that maximum normal walking speed that can be tracked by our system will be limited in the camera zooming. Therefore, we cannot keep the camera zoomed in and we have to zoom out. This process will take more time than just a simple moving to follow the target. In addition in zooming case lower precision, more tracking fragmentation and more distances error are obtained. The reason of lower precision and more tracking fragmentation is because of limited FOV and adjusting of camera zooming speed with target speed that takes time. Higher distance errors are because of having larger target size while camera has zoomed in and its displacement seems higher.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i)

Figure 4.16 Examples of tracking and zooming frames for E$_3$, (a) target tracking before zooming in, (b) target tracking after zooming in, (c) false target tracking, (d) target tracking after zooming out, (e) tracking before short term occlusion, (f) tracking while partial occlusion, (g) zooming while partial occlusion, (h) tracking and zooming after occlusion, (i) target face detection.

Similarly, localization fails, when the closeness of the color histogram of the target with other samples occurs. It is even worse in the case of zooming while the FOV is limited and target has some fast and sudden movement that causes the target lost. The random motions between frames are handled by combination of sampling scheme and position prediction method, as long as the target appearance does not change significantly and it remains in the camera FOV. The three delays τ_1, $\tau_{2,}$ and τ_3 are compensated by position predictor and avoid the target to get out of the FOV.

Occlusions are handled in the same way. However, when the object is occluded, another similar object will be tracked (the most likely candidate blob) until the occlusion ends. This could cause the real target to become out of the FOV of the camera. If occlusion occurs when the camera zooms and no other similar object is found the camera will zoom out to get larger FOV. Figure 4.16 (f) to (h), shows an example of short-term occlusion handling during zooming. The proposed method can handle it in this case. The zooming on the target is done when a good tracking is obtained Figure 4.16 (a) to (b) and (e) to (h). The camera zooms out if the quality of tracking is not appropriate as in Figure 4.16 (c) to (d). The target face image is captured when the face and torso is verified as in Figure 4.16 (i).

4.9 Study of tracking performance versus available network bandwidth and versus different camera to network connection mode

Up to now, different parameters that have effect on our tracking method have been evaluated. However the camera is IP based and is connected through a network with the traffic that is not predictable. Therefore, it is needed to know the effect of network on our system. First we study the tracking performance versus camera to network connection mode effect and network bandwidth effect. We have done ten experiments with one IP camera (SNC_RZ50) in 640×480 image resolution since we want more intensive data transferring to occur. We recorded all the experiments to extract their ground-truth manually for performance evaluation. Experiments are classified into two classes based on the camera connection mode. The experiments are done in a room with dimension of 6 × 4 meters. The experiments are described in Table 4-19. The distances of the initial model position from the camera are written in Table 4-19.

Table 4-20 shows the result of tracking performance with different camera to network connection mode. Camera connection mode may be direct or indirect. The direct mode is when the camera is connected directly to the network adaptor of the workstation. The indirect mode is when the camera is connected to the workstation through the local network.

Table 4-19 Experiments to compare tracking performance versus different camera to network connection mode.

Class	Experiments	Camera Model	Image Size	Initial Model Position	CM
Class 1	E_1	SNC-RZ50	640×480	Near	Direct
	E_2	SNC-RZ50	640×480	Near	Direct
	E_3	SNC-RZ50	640×480	Far	Direct
	E_4	SNC-RZ50	640×480	Far	Direct
	E_5	SNC-RZ50	640×480	Middle	Direct
Class 2	E_6	SNC-RZ50	640×480	Near	Indirect
	E_7	SNC-RZ50	640×480	Near	Indirect
	E_8	SNC-RZ50	640×480	Far	Indirect
	E_9	SNC-RZ50	640×480	Far	Indirect
	E_{10}	SNC-RZ50	640×480	Middle	Indirect

Near ~ 1 meter, Far ~ 6 meters, Middle ~ 3 meters, CM: Connection mode.

By comparing class 1 with class 2 in Table 4-20, with direct camera connection to the computer network, a faster system frame rate is obtained because of smaller network delays. This results to have even better tracking performance in direct mode connection (e.g. higher P, lower TF, $\mu_{d_{gc}}$, and $\mu_{d_{gp}}$).

We also tested our method over different network traffic loads to see how well it can track with different network delays. This is not a thorough evaluation as the traffic on the network is not a simulation of real network conditions. However, it gives an idea of how the algorithm performs when there is other activity on the network. Actually this test is similar to setting manually the camera to capture images at a lower frame rate. But since the traffic on the network has effect on the frame rate we interpret this test by this way. The generated network traffic is constant and it is set to occupy a certain bandwidth. On a 100Mps network, the camera uses about 15% of the bandwidth. In our experiment, the total bandwidth used (traffic+camera) has been set to 15% (about only camera), 67% (traffic+camera) and 93% (heavy traffic+camera).

Table 4-20 Experimental results to evaluate tracking performance versus different camera to network connection mode.

E	P(%)	TF(%)	$\mu_{d_{gc}}$	$\sigma^2_{d_{gc}}$	$\mu_{d_{gp}}$	$\sigma^2_{d_{gp}}$	FR(fps)	NF	min(Δx)	max(Δx)
E_1	95	3.1	0.1471	0.0214	0.0316	0.0014	7.45	664	0.010	0.65
E_2	94	3.5	0.1522	0.0352	0.0403	0.0011	7.93	582	0.025	0.38
E_3	94	3.7	0.1563	0.0236	0.0398	0.0018	7.54	636	0.017	0.53
E_4	96	3.0	0.1381	0.0297	0.0305	0.0015	8.87	571	0.015	0.59
E_5	93	3.9	0.1665	0.0242	0.0489	0.0012	6.6	568	0.022	0.51
Class 1	**94.40**	**3.4**	**0.1520**	**0.0268**	**0.0382**	**0.0014**	**7.60**	**3021**	**-**	**-**
E_6	93	5	0.2554	0.0436	0.0580	0.0015	6.67	511	0.018	0.33
E_7	96	2	0.2121	0.0352	0.0517	0.0021	6.86	489	0.013	0.31
E_8	98	1	0.1561	0.0271	0.0483	0.0032	7.31	494	0.016	0.36
E_9	92	7	0.1763	0.0394	0.0619	0.0027	6.04	506	0.006	0.25
E_{10}	91	6	0.1779	0.0613	0.0617	0.0023	6.19	517	0.008	0.29
Class 2	**93.95**	**4.2**	**0.1956**	**0.0413**	**0.0563**	**0.0024**	**6.57**	**2517**	**-**	**-**

E: Experiments, P:Precision, $\mu_{d_{gc}}$: mean of d_{gc}, $\mu_{d_{gp}}$: mean of d_{gp}, $\sigma^2_{d_{gc}}$: variance of d_{gc}, $\sigma^2_{d_{gp}}$: variance of d_{gp}, FR: System frame rate and NF:Number of frames, *min(Δx):* minimum normalized motionvector length, *max(Δx)*: maximum normalized motionvector length.

Table 4-21 Experimental results to evaluate tracking performance versus various network bandwidth usage.

	TF(%)	$\mu_{d_{gc}}$	$\mu_{d_{gp}}$	FR(*fps*)	NF	Used Network Bandwidth (%)
E_{L1}	3.3	0.2019	0.0452	5.87	643	16
E_{L2}	2.7	0.1736	0.0467	6.21	578	15
E_{L3}	3.1	0.1942	0.0501	6.37	605	17
E_{L4}	2.9	0.2215	0.0493	6.72	584	13
E_{L5}	2.6	0.2147	0.0464	5.95	596	14
Class L_1	**2.9**	**0.2012**	**0.0475**	**6.21**	**3006**	**15**
E_{L6}	6.6	0.2614	0.0531	4.46	521	71
E_{L7}	5.4	0.2356	0.0498	5.08	437	65
E_{L8}	7.2	0.2897	0.0552	4.89	409	69
E_{L9}	6.3	0.2466	0.0586	5.22	462	64
E_{L10}	6.9	0.2541	0.0519	4.97	440	66
Class L_2	**6.4**	**0.2575**	**0.0537**	**4.91**	**2269**	**67**
E_{L11}	21	0.3219	0.0857	3.23	342	93
E_{L12}	18.6	0.2854	0.0965	2.97	376	96
E_{L13}	19.4	0.3347	0.1024	3.16	351	91
E_{L14}	17.5	0.3242	0.1167	2.89	361	94
E_{L15}	22.3	0.3565	0.1193	2.73	309	95
Class L_3	**19.6**	**0.3245**	**0.1041**	**3.00**	**1739**	**93.8**

Table 4-21 shows the effect of network utilization on the tracking performance. *TF*, $\mu_{d_{gc}}$ and $\mu_{d_{gp}}$ are increased as the network usage traffic is increased. It means that the traffic on the network has direct effect on the system performance as expected. Indeed when there is a high traffic on the network, the allocated bandwidth for camera is smaller. Therefore, all commands are queued and performed according to the smaller specified bandwidth. It results to take time and have delay on the communication through network. This delay causes smaller frame rate and higher *TF*, $\mu_{d_{gc}}$ and $\mu_{d_{gp}}$.

4.10 General Discussion

The performance of the proposed tracking and controlling system versus different parameters and tracking methods was evaluated. Also the tracking performance in the case of moving only and moving accompanied with zooming has been compared.

Results show that with smaller image resolution, the results of tracking are significantly better. Indeed, it results in a faster system rate because less data is processed and transferred on the network. In addition with smaller image size, we obtain a higher target tracking precision P with less tracking fragmentation and faster frame rate. The localization of the target is better ($\mu_{d_{gc}}$ and $\mu_{d_{gp}}$ values are smaller). It means the location of the target is near to the ground truth and the camera could center on the target with a good precision. This result is explained by the use of particle filter that well distributes samples at each frame in different positions and various scale sizes around detected motion regions or previous target position. That gives more candidate regions to the tracker to find the best match. Localization accuracy is thus, improved. This also results in the target being usually located in a constant distance within 1/6th of the image diameter from the image center. The image resolution and camera model have effects on the system frame rate and thus, on tracking error.

We have determined the limits of the system in both initialization step and whole tracking and camera control. It is shown that our minimum face size detection in initialization step is limited according to the minimum face size detection in Viola and Jones method. In addition in tracking when the target is very small and there are lighting changes, the target will be lost. We have shown that zooming on the farther target improves the small target size. However, zooming increases the target sample content to compare. When localization fails, it is because of similarity or closeness of the color histogram of the target with other samples. We have also shown the tracking performance versus the target size, target speed and average target frame rate.

Results show that our algorithm can handle and overcome large inter-frame motion because of using a sampling scheme and motion prediction technique that does not rely on spatial proximity. It will lose a target only if the target changes its motion direction suddenly and walks very fast in the opposite of the predicted direction. Our method can also recover the tracking if it loses the object, because of the samples around regions with detected motion. Of course, it is conditional to the object being in the FOV of the camera. SNC-RZ50N has better performance because of higher maximum pan-tilt speed than SNC-RZ25N.

Then, we explained how the tracking parameters are calibrated. Indeed, APF-OFS requires to be calibrated for better performance. These parameter are: b the number of vertical division of samples, bin numbers of r and θ the radius and angle in radial histogram, N number of samples generated by a Gaussian random function, r_g radius of circular region that samples are generated in, s_s variation of sample size and η the similarity percentage used for moving criteria. These parameters are determined based on the tracking error results. b corresponds with the amount of sample content accuracy which is compared with target model. N_r and N_θ have direct effect on the estimation of camera motion vector and therefore, on the target position prediction results. N, r_g and s_s have direct effect on the tracking performance since they determine how much well the target candidates are selected due to the time cost problem.

APF-OFS and PF methods outperform the KLT feature tracker, since both have a higher target tracking precision P with less tracking fragmentation and the localization of the target is better ($\mu_{d_{gc}}$ and $\mu_{d_{gp}}$ values are smaller). It is explained by the characteristic of particle filter which is distributing of samples at each frame in different positions and various scale sizes to create more target candidates for comparison. Localization accuracy is improved in the same way. Localization in both APF-OFS and PF fails when the color histogram of the target with other samples is similar. Since PF generates samples everywhere, more false positive is observed. In APF-OFS, by using optical flow this type of false positive samples are reduced. Also it helps to keep the target within an area with 1/6th of image diameter from the image center.

To compare and test the effect of tracking algorithm on camera control part, two similar IP PTZ cameras were used and APF-OFS was compared with PF tracking method. APF-OFS outperforms PF with a higher target tracking precision P and less tracking fragmentation. The target is better localized due to the better distribution of samples in APF-OFS which uses motion pixels detected by optical flow as candidate sample position. Indeed optical flow has an important role that causes to remove redundant false positive samples. By using a sampling scheme and combining it with a motion predictor, we can handle random motion between frames, as long as the target position is well predicted, and its appearance does not change significantly. Occlusions are handled in the same way. However, when the object is occluded, another similar object will be tracked (the most likely candidate blob) until the occlusion ends. This could cause the real target to become out of the FOV of the camera. The proposed method can handle it in this case. The position predictor block, estimates the next target position and therefore, camera can compensate the three delays discussed before to center on the target while PF cannot perform such a thing. The number of PF samples is obtained from average frame rate processing of our method for smaller image size. With larger number of samples higher time processing and lower frame rate is obtained. It is obvious that with less samples APF-OFS outperforms PF while the frame rate in APF-OFS is also higher. In all experiments, there are scale changes to verify tracking against scaling. It is because of using normalized color histogram and average color features. These two features are independent of the size of the target.

In the comparison of the APF-OFS performance before and after zooming, slower frame rate is obtained because of adding another camera control task (zooming) which takes more time than just camera moving (zooming takes between 1.6 ~ 2.5 times longer than moving). Camera zooming limits the camera FOV, and therefore, the maximum normal walk speed that can be tracked by our system will be limited. As a result we have to do the zoom out to keep target tracking. This process takes more time than just a simple moving to follow the target. In addition in zooming case lower precision, more tracking fragmentation and more distances error are obtained because of limited FOV and adjusting of camera zooming speed with target speed that takes time. Higher distance errors are because of having larger target size while camera has zoomed in and its displacement seems higher.

When localization fails more in the zooming since camera FOV is limited and target has some fast and sudden movement that causes the target lost. In the same way, by using a sampling scheme and combining it with a motion predictor, random motion between frames are handled.

Occlusions are handled similarly however if occlusion occurred when the camera zooms and no other similar object is found, the camera will zoom out to get larger FOV. Using normalized color histogram and average color features which are independent of the size of the target helps to overcome to the scaling changes during the zooming.

CHAPTER 5. CONCLUSION AND FUTURE WORK

Accurate, efficient and reliable moving object tracking is a challenging task in different computer vision applications such as video surveillance and intelligent video monitoring systems. Here object tracking is applied to human face and upper body tracking. It can be used to get images of the face of a human target in different poses. Automatic human detection and tracking by controlling an IP PTZ camera are new subjects that provide valuable capabilities for applications that desire to achieve a natural human–machine interaction such as people identification. Using PTZ cameras allows covering a wide FOV by changing the pan, tilt and zoom values. The IP-based property allows monitoring and controlling from everywhere in the network. This subject is applicable for on-line and real-time video surveillance, video conference and face identification.

In our work, we cope with the problem of large inter-frame motion of targets, low usable frame rate, background changes, and tracking with various scale changes. In addition, the tracking algorithm handles the camera response time and zooming. Our system architecture consists of automatic initialization where the target face is detected and modeled for the system tracker, image capture, APF-OFS and camera control. APF-OFS consists of two main blocks which are object tracking and position prediction. We propose an APF-OFS method adapted to our tracking problem to track human face by means of an IP PTZ camera. Optical flow that extracts the moving pixels is combined with particle filter that has robustness to non-Gaussian distribution of target movements to extract random motion of the object.

Target modeling and tracking are done based on sampling around predicted position obtained by a position predictor and moving regions detected by optical flow. The scoring of sample features is done with some reasonable normalization functions to allow us more flexibility in the sample score values. Normalization functions are applied to geometric and appearance features. In our particle filter, target modeling and tracking are done based on sampling around predicted position obtained by a position predictor and moving regions detected by optical flow. The scoring of sample features is done with some normalization functions. In addition it can be used for low resolution images. The proposed algorithm is able to distinguish and recognize the detected faces from other objects or faces. It handle the Internet delay and camera response time.

Various experiments were done to evaluate and compare the system tracking performance versus different parameters, and other tracking methods. Results show that our algorithm can handle large motion between two consecutive frames, and the detected target location is near to the ground truth. Also the camera can center on the target with a good precision. It is because particle filter samples are well distributed in each frame with various scale sizes around candidate locations. We will lose a target if the person changes its motion direction suddenly and walks very fast in the opposite of the predicted direction. It is even worse in the case of zooming while the FOV is limited and target has some fast and sudden movement that causes the target lost. We can recover the track if the target moves inside the FOV of the camera again. The proposed method can handle the short-term occlusion at the condition that the object stays in the FOV. We can handle random motion between frames, as long as the target position is well predicted, its appearance does not change significantly and it remains in the camera FOV. The motion predictor is used to compensate the three delays τ_1, $\tau_{2,}$ and τ_3, which may cause the target to exit the FOV.

The general contribution of this thesis is dynamic face tracking with an IP PTZ camera which is first in the field of computer vision. The specific contributions of this thesis are (1) modeling of the system as a servo control loop that formulates and compensates all the delays resulting from network or processing (Darvish Zadeh Varcheie & Bilodeau, 2009b, 2009c, 2009d); and (2) proposing of an adaptive particle filter with optical flow samples method as a solution that tracks well the object and controls the camera. Combination of particle filter and optical flow to generate appropriate samples helps to cover all possible candidate regions and to handle large inter-frame motion of the target (Darvish Zadeh Varcheie & Bilodeau, 2009a) and low tracking frame rate (Darvish Zadeh Varcheie & Bilodeau, 2009b). Furthermore, recovering the tracking target in the case of occlusion or lost is done in the same way.

Future work of this system can be optimization of the code like a GPU-based system to accelerate processing time to achieve better performance. Complex modeling of delays can be helpful to have an accurate estimation of the available delays through the system communication.

It is possible to generalize the tracking system to any object tracking for different applications such as car tracking in traffic monitoring or parking lots. In addition multiple IP PTZ cameras can be integrated to have a wide coverage of FOV with better performance and more capabilities instead of using stationary and PTZ cameras.

REFERENCES

Ahmed, J., Jafri, M., Shah, M., & Akbar, M. (2008). Real-time edge-enhanced dynamic correlation and predictive open-loop car-following control for robust tracking. *Machine Vision and Applications, 19*(1), 1-25.

Alper, Y., Omar, J., & Mubarak, S. (2006). Object tracking: A survey. *ACM Comput. Surv., 38*(4), 13.

Araki, S., Matsuoka, T., Takemura, H., & Yokoya, N. (1998). *Real-time tracking of multiple moving objects in moving camera image sequences using robust statistics*. Paper presented at the Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on.

Avidan, S. (2001). *Support Vector Tracking.* Paper presented at the Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on.

Avidan, S. (2004). Support vector tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 26*(8), 1064-1072.

Bagdanov, A. D., del Bimbo, A., & Nunziati, W. (2006). Improving evidential quality of surveillance imagery through active face tracking. *Proc. of International Conference on Pattern Recognition (ICPR)*, 1200-1203.

Baoxin, L., Chellappa, R., Qinfen, Z., & Der, S. Z. (2001). Model-based temporal object verification using video. *Image Processing, IEEE Transactions on, 10*(6), 897-908.

Bellotto, N., & Huosheng, H. (2007). *People Tracking and Identification with a Mobile Robot.* Paper presented at the Mechatronics and Automation, 2007. ICMA 2007. International Conference on.

Bellotto, N., Sommerlade, E., Benfold, B., Bibby, C., Reid, I., Roth, D., et al. (2009). a distributed camera system for multi-resolution surveillance. *Proc. of the 3rd ACM/IEEE Int. Conf. on Distributed Smart Cameras (ICDSC).*

Bernardin, K., van de Camp, F., & Stiefelhagen, R. (2007). *Automatic Person Detection and Tracking using Fuzzy Controlled Active Cameras.* Paper presented at the Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on.

Bertalmio, M., Sapiro, G., & Randall, G. (2000). Morphing active contours. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22*(7), 733-737.

Birchfield, S. (1997). Kanade-Lucas-Tomasi Feature Tracker, Online accessed 15-February-2010. from http://www.ces.clemson.edu1stb/klt/

Black, M. J., & Jepson, A. D. (1998). EigenTracking: robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision, 26*(1), 63-84.

Bouguet, J.-Y. (2000). Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm: Intel Corporation Microprocessor Research Labs.

Bourbakis, N., Kakumanu, P., & Makrogiannis, S. (2007). A survey of skin-color modeling and detection methods. *Pattern Recognition, 40*(3), 1106-1122.

Cha, S.-H., & Srihari, S. N. (2002). On measuring the distance between histograms. *Pattern Recognition, 35*(6), 1355-1370.

Chan, C., Oe, S., & Lin, C. (2007). Active Eye-tracking System by Using Quad PTZ Cameras. *IEEE conference on industrial electronics society (IECON), 5.*

Changjiang, Y., Duraiswami, R., & Davis, L. (2005). *Fast multiple object tracking via a hierarchical particle filter.* Paper presented at the Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on.

Chu-Sing, Y., Ren-Hao, C., Chao-Yang, L., & Shou-Jen, L. (2008). *PTZ camera based position tracking in IP-surveillance system.* Paper presented at the Sensing Technology, 2008. ICST 2008. 3rd International Conference on.

Chung-Hao, C., Yi, Y., Page, D., Abidi, B., Koschan, A., & Abidi, M. (2008). Heterogeneous Fusion of Omnidirectional and PTZ Cameras for Multiple Object Tracking. *Circuits and Systems for Video Technology, IEEE Transactions on, 18*(8), 1052-1063.

Chung, R. H. Y., Chin, F. Y. L., K. Wong, K.-Y., Chow, K. P., Luo, T. B., B., & Fung, H. S. K. (2005). Efficient Block-based Motion Segmentation Method using Motion Vector Consistency. *Conference on Machine VIsion Applications (MVA)*, 13-28.

Cindy, X., Collange, F., Jurie, F., & Martinet, P. (2001). *Object tracking with a pan-tilt-zoom camera: application to car driving assistance.* Paper presented at the Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on.

Collins, R., Lipton, A., & Kanade, T. (1999). A system for video surveillance and monitoring. *Proceedings of the American Nuclear Society (ANS) Eighth International Topical Meeting on Robotics and Remote Systems*.

Comaniciu, D., & Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24*(5), 603-619.

Comaniciu, D., Ramesh, V., & Meer, P. (2003). Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*(5), 564-577.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2,* 886-893.

Darvish Zadeh Varcheie, P., & Bilodeau, G. A. (2009a). *Active People Tracking by a PTZ Camera in IP Surveillance System.* Paper presented at the IEEE International workshop on Robotic and Sensors Environments (ROSE).

Darvish Zadeh Varcheie, P., & Bilodeau, G. A. (2009b). *Fuzzy Feature-Based Upper Body Tracking with IP PTZ Camera Control.* Paper presented at the 14th Iberoamerican Congress on Pattern Recognition (CIARP).

Darvish Zadeh Varcheie, P., & Bilodeau, G. A. (2009c). Human Tracking by IP PTZ Camera Control in the Context of Video Surveillance. *Lecture Notes in Computer Science: Image Analysis and Recognition (ICIAR), 5627*, 657-667.

Darvish Zadeh Varcheie, P., & Bilodeau, G. A. (2009d). Online Body Tracking by a PTZ Camera in IP Surveillance System. *The first IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 27-32.

Darvish Zadeh Varcheie, P., Sills-Lavoie, M., & Bilodeau, G. A. (2008). *An Efficient Region-Based Background Subtraction Technique.* Paper presented at the Computer and Robot Vision, 2008. CRV '08. Canadian Conference on.

Darvish Zadeh Varcheie, P., Sills-Lavoie, M., & Bilodeau, G. A. (2010). A Multiscale Region-Based Motion Detection and Background Subtraction Algorithm. *Sensors, 10*(2), 1041-1061.

Denzler, J., Zobel, M., & Niemann, H. (2003). *Information Theoretic Focal Length Selection for Real-Time Active 3-D Object Tracking*. Paper presented at the Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2.

Elder, J. H., Prince, S. J. D., Hou, Y., Sizintsev, M., & Olevskiy, E. (2007). Pre-attentive and attentive detection of humans in wide-field scenes. *International Journal of Computer Vision, 72*(1), 47-66.

Elgammal, A., Harwood, D., & Davis, L. (2000). *Non-parametric Model for Background Subtraction*. Paper presented at the Proceedings of the 6th European Conference on Computer Vision-Part II.

Everts, I., Sebe, N., & Jones, G. (2007). *Cooperative Object Tracking with Multiple PTZ Cameras.* Paper presented at the Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on.

Funahashi, T., Fujiwara, T., & Koshimizu, H. (2004). Hierarchical tracking of face, facial parts and their contours with PTZ camera. *IEEE Int. Conf. on Industrial Technology (ICIT), 1*, 198-203.

Funahashi, T., Tominaga, M., Fujiwara, T., & Koshimizu, H. (2004). Hierarchical face tracking by using PTZ camera. *IEEE Int. Conf. on Automatic Face and Gesture Recognition (FGR)*, 427-432.

Gagnon, L., Laliberte, F., Foucher, S., Branzan Albu, A., & Laurendeau, D. (2006). *A system for tracking and recognizing pedestrian faces using a network of loosely coupled cameras.* Paper presented at the SPIE Defense & Security: Visual Information Processing XV Orlando.

Galic, S., & Loncaric, S. (2000, 14-15 June 2000). *Spatio-temporal image segmentation using optical flow and clustering algorithm.* Paper presented at the IWISPA 2000. Proceedings of the First International Workshop on Image and Signal Processing and Analysis in conjunction with 22nd International Conference on Information Technology Interfaces, Pula, Croatia.

Genshiro, K. (1987). Non-Gaussian State-Space Modeling of Nonstationary Time Series. *Journal of the American Statistical Association, 82*(400), 1032-1041.

Han, B., Zhu, Y., Comaniciu, D., & Davis, L. (2005, Jun 20-25 2005). *Kernel-based Bayesian filtering for object tracking.* Paper presented at the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, San Diego, CA, United States.

Haritaoglu, I., Harwood, D., & Davis, L. S. (1998a). *W4: A Real Time System for Detecting and Tracking People.* Paper presented at the Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on.

Haritaoglu, I., Harwood, D., & Davis, L. S. (1998b). *W4: Who? When? Where? What? A real time system for detecting and tracking people.* Paper presented at the Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on.

Haritaoglu, I., Harwood, D., & Davis, L. S. (2000). W4: real-time surveillance of people and their activities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22*(8), 809-830.

Heikkila, M., & Pietikainen, M. (2006). A texture-based method for modeling the background and detecting moving objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 28*(4), 657-662.

Hue, C., Le Cadre, J. P., & Perez, P. (2002). Sequential Monte Carlo methods for multiple target tracking and data fusion. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on], 50*(2), 309-325.

Intel open vision library. (2008, Online accessed 15-February-2010). Face detection using opencv. from http://opencv.willowgarage.com/wiki/FaceDetection

Isard, M., & Blake, A. (1998). CONDENSATION—Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision, 29*(1), 5-28.

Isard, M., & MacCormick, J. (2001). *BraMBLe: a Bayesian multiple-blob tracker.* Paper presented at the Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on.

Jepson, A. D., Fleet, D. J., & El-Maraghi, T. F. (2003). Robust online appearance models for visual tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 25*(10), 1296-1311.

Joe-Air, J., Ying-Tung, H., Chuang, C. L., & Yen-Ling, L. (2006). Robust multiple objects tracking using image segmentation and trajectory estimation scheme in video frames. *Image and Vision Computing, 24*(10), 1123-1136.

Kim, J. B., & Kim, H. J. (2003). Efficient region-based motion segmentation for a video monitoring system. *Pattern Recognition Letters, 24*(1-3), 113-128.

Kim, Y. O., Paik, J., Jingu, H., Koschan, A., Abidi, B., & Abidi, M. (2003). *Automatic face region tracking for highly accurate face recognition in unconstrained environments.* Paper presented at the Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.

Krahnstoever, N., & Mendonca, P. R. S. (2005). Bayesian autocalibration for surveillance. *IEEE International conference on computer vision (ICCV), 2,* 1858-1865.

Krahnstoever, N., Tu, P., Sebastian, T., Perera, A., & Collins, R. (2006). multi-view detection and tracking of travelers and luggage in mass transit environments. *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance and CVPR*.

Krahnstoever, N., Yu, T., & Lim, S. (2008). Collaborative real-time control of active cameras in large scale surveillance systems. *European Conference on Computer Vision (ECCV)*.

Kublbeck, C., & Ernst, A. (2006). Face detection and tracking in video sequences using the modified census transformation. *Image and Vision Computing, 24*(6), 564-572.

Kwang Ho, A., Dong Hyun, Y., Sung Uk, J., & Myung Jin, C. (2005, 2-6 Aug. 2005). *Robust multi-view face tracking.* Paper presented at the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alta., Canada.

Lalonde, M., Foucher, S., Gagnon, L., Pronovost, E., Derenne, M., & Janelle, A. (2007). *A system to automatically track humans and vehicles with a PTZ camera.* Paper presented at the SPIE: Visual Information Processing XVI

Leichter, I., Lindenbaum, M., & Rivlin, E. (2008). Bittracker&#x02014;A Bitmap Tracker for Visual Tracking under Very General Conditions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 30*(9), 1572-1588.

Lievin, M., & Luthon, F. (2004). Nonlinear color space and spatiotemporal MRF for hierarchical segmentation of face features in video. *Image Processing, IEEE Transactions on, 13*(1), 63-71.

Liu, Z., Yang, J., & Peng, N. S. (2005). An efficient face segmentation algorithm based on binary partition tree. *Signal Processing: Image Communication, 20*(4), 295-314.

Lu, Y., & Payandeh, S. (2008). Cooperative hybrid multi-camera tracking for people surveillance. *Electrical and Computer Engineering, Canadian Journal of, 33*(3), 145-152.

Math Forum. (2003, Online accessed 15February-2010). Points within an Ellipse. from http://mathforum.org/library/drmath/view/63045.html,

Math Open Reference. (2008, Online accessed 15-February-2010). Foci of an Ellipse. from http://www.mathopenref.com/ellipsefoci.html

Matsuyam, T., Hiura, S., Wada, T., Muease, K., & Toshioka, A. (2000). Dynamic memory: architecture for real time integration of visualperception, camera action, and network communication. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2*, 728-735.

McKenna, S. J., Jabri, S., Duric, Z., Rosenfeld, A., & Wechsler, H. (2000). Tracking groups of people. *Computer Vision and Image Understanding, 80*(1), 42-56.

Micheloni, C., & Foresti, G. L. (2005). *Zoom on target while tracking.* Paper presented at the Image Processing, 2005. ICIP 2005. IEEE International Conference on.

Micheloni, C., & Foresti, G. L. (2006). Real-time image processing for active monitoring of wide areas. *Journal of Visual Communication and Image Representation, 17*(3), 589-604.

Murray, D., & Basu, A. (1994). Motion tracking with an active camera. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 16*(5), 449-459.

Papageorgiou, C. P., Oren, M., & Poggio, T. (1998). *A general framework for object detection.* Paper presented at the Computer Vision, 1998. Sixth International Conference on.

Pentland, A., Moghaddam, B., & Starner, T. (1994, 21-23 June 1994). *View-based and modular eigenspaces for face recognition.* Paper presented at the Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA.

Phimoltares, S., Lursinsap, C., & Chamnongthai, K. (2002). *Locating essential facial features using neural visual model.* Paper presented at the Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on.

Phimoltares, S., Lursinsap, C., & Chamnongthai, K. (2007). Face detection and facial feature localization without considering the appearance of image context. *Image and Vision Computing, 25*(5), 741-753.

Rasmussen, C., & Hager, G. D. (2001). Probabilistic data association methods for tracking complex visual objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 23*(6), 560-576.

Roh, M.-C., Kim, T.-Y., Park, J., & Lee, S.-W. (2007). Accurate object contour tracking based on boundary edge selection. *Pattern Recognition, 40*(3), 931-943.

Rosales, R., & Sclaroff, S. (1999, 23-25 June 1999). *3D trajectory recovery for tracking multiple objects and trajectory guided recognition of actions.* Paper presented at the Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Fort Collins, CO.

Sangkeun, L., & Hayes, M. H. (2004, 24-27 Oct. 2004). *A simple and fast color-based human face detection scheme for content-based indexing and retrieval.* Paper presented at the 2004 International Conference on Image Processing (ICIP), Singapore.

Sangkyu, K., Abidi, B., & Abidi, M. (2004). *Integration of color and shape for detecting and tracking security breaches in airports.* Paper presented at the Security Technology, 2004. 38th Annual 2004 International Carnahan Conference on.

Sangkyu, K., Joonki, P., Andreas, K., Besma, A., & Mongi, A. A. (2003). Real-time tracking using PTZ cameras. *Proc. of SPIE 6th International Conference of Quality Control by Artificial Vision, 5132*, 103-111.

Sangoh, J. (Online accessed 15-February-2010). Histogram-Based Color Image Retrieval. from http://scien.stanford.edu/class/psych221/projects/02/sojeong/

Schreiber, D. (2008). Generalizing the lucas-kanade algorithm for histogram-based tracking. *pattern recognition letters, 29*(7), 852-861.

Schweitzer, H., Bell, J., & Wu, F. (2002). Very Fast Template Matching. In *Computer Vision — ECCV 2002* (pp. 145-148).

Ser-Nam, L., Elgammal, A., & Davis, L. S. (2003). *Image-based pan-tilt camera control in a multi-camera surveillance environment*. Paper presented at the Proceedings of the 2003 International Conference on Multimedia and Expo - Volume 2.

Shafique, K., & Shah, M. (2003, 13-16 Oct. 2003). *A non-iterative greedy algorithm for multi-frame point correspondence.* Paper presented at the ICCV 2003: 9th International Conference on Computer Vision, Nice, France.

Shaohua, Z., Chellappa, R., & Moghaddam, B. (2003, 6-9 July 2003). *Adaptive visual tracking and recognition using particle filters.* Paper presented at the 2003 IEEE International Conference on Multimedia and Expo, Baltimore, MD.

Shi, J., & Tomasi, C. (1994). *Good features to track.* Paper presented at the Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on.

Shoushtarian, B., & Bez, H. E. (2005). A practical adaptive approach for dynamic background subtraction using an invariant colour model and object tracking. *Pattern Recognition Letters, 26*(1), 5-26.

Sim, T., Sukthankar, R., Mullin, M., & Baluja, S. (2000, 28-30 March 2000). *Memory-based face recognition for visitor identification.* Paper presented at the Proceedings of the Fourth International Conference on Automatic Face and Gesture Recognition, Grenoble, France.

Sommerlade, E., & Reid, I. (2008). *Information-theoretic active scene exploration.* Paper presented at the Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.

Sony corporation. (2005, Online accessed 15-February-2010). Network Camera- SNC-RZ50N, SNC-RZ50P. from http://www.sony.ca/ip/brochures/ip_cameras/SNCRZ50N%20Brochure.pdf

Sony corporation. (2005, Online accessed 15-February-2010). SNC-RZ25N/P CGI command manual. from http://www.cs.unc.edu/Research/stc/FAQs/Cameras_Lenses/Sony/New%20SNC-RZ30N_CGI%20Manual%202.00EN.pdf

Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on., 2*, 252 Vol. 252.

Sung, K. K., & Poggio, T. (1998). Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(1), 39-51.

Tao, H., Sawhney, H. S., & Kumar, R. (2002). Object tracking with Bayesian estimation of dynamic layer representations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24*(1), 75-89.

Torabi, A., Bilodeau, G.-A., Levesque, M., Langlois, J. M., Pablo, L., & Lionel, C. (2008). *Measuring an Animal Body Temperature in Thermographic Video Using Particle Filter Tracking.* Paper presented at the Proceedings of the 4th International Symposium on Advances in Visual Computing, Lecture Notes in Computer Science: Advances in Visual Computing, Las Vegas, NV.

Tordoff, B. J., & Murray, D. W. (2007). A method of reactive zoom control from uncertainty in tracking. *Comput. Vis. Image Underst., 105*(2), 131-144.

Tu, P., Wheeler, F., Krahnstoever, N., Sebastian, T., Rittscher, J., Liu, X., et al. (2007). Surveillance video analytics for large camera networks. *SPIE NewsroomIntelligent video research focuses on automatic camera calibration and person detection, tracking, and reacquisition.*

Vaswani, N., Roy Chowdhury, A., & Chellappa, R. (2003, 18-20 June 2003). *Activity recognition using the dynamics of the configuration of interacting objects.* Paper presented at the CVPR 2003: Computer Vision and Pattern Recognition Conference, Madison, WI.

Veenman, C. J., Reinders, M. J. T., & Backer, E. (2001). Resolving motion correspondence for densely moving points. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(1), 54-72.

Venkatesh babu, R., Perez, P., & Bouthemy, P. (2007). Robust tracking with motion estimation and local Kernel-based color modeling. *image and vision computing, 25*(8), 1205-1216.

Viola, P., & J. Jones, M. (2004). Robust Real-Time Face Detection. *Int. J. Comput. Vision, 57*(2), 137-154.

Viola, P., & Jones, M. (2001). *Robust Real-time Object Detection.* Paper presented at the International Journal of Computer Vision.

Walker, K. N., Cootes, T. F., & Taylor, C. J. (1998, 14-16 April 1998). *Locating salient facial features using image invariants.* Paper presented at the Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan.

Weber, M. (1999, Online accessed 8-March-2010). Frontal Face Caltech Image DataSet. *California Institute of Technology*, from http://www.vision.caltech.edu/html-files/archive.html

Wikipedia. (2008, Online accessed 15-February-2010). Von Luschan's chromatic scale -- Wikipedia, The Free Encyclopedia. from http://en.wikipedia.org/w/index.php?title=Von_Luschan%27s_chromatic_scale&oldid=249213206

Wikipedia. (2010, Online accessed 17-May-2010). Walking -- Wikipedia, The Free Encyclopedia. from http://en.wikipedia.org/wiki/Walking#cite_note-1

Xiaopeng, J., Zhiqiang, W., & Yewei, F. (2006). Effective vehicle detection technique for traffic surveillance systems. *Journal of Visual Communication and Image Representation, 17*(3), 647-658.

Yan, L., & Payandeh, S. (2008). *Cooperative hybrid multi-camera tracking for people surveillance.* Paper presented at the Electrical and Computer Engineering, 2008. CCECE 2008. Canadian Conference on.

Yao, Y., Abidi, B., & Abidi, M. (2006). 3D Target Scale Estimation and Motion Segmentation for Size Preserving Tracking in PTZ Video. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 130-136.

Yezzi, A., Zollei, L., & Kapur, T. (2001, 9-10 Dec. 2001). *A variational framework for joint segmentation and registration.* Paper presented at the Workshop on Mathematical Methods in Biomedical Image Analysis, Kauai, HI.

Yi, Y., Besma, A., & Mongi, A. (2009). 3D Target Scale Estimation and Target Feature Separation for Size Preserving Tracking in PTZ Video. *Int. J. Comput. Vision, 82*(3), 244-263.

Yilmaz, A., Xin, L., & Shah, M. (2004). Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 26*(11), 1531-1536.

Yin, F., Makris, D., & Velastin, S. A. (2007). Performance evaluation of object tracking algorithms. *IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance(PETS).*

Youshan, Q., Weijian, T., & Yingcai, L. (2003). The moving target detecting based on the parallelable discontinuous frame difference optical flow field integrated with Gray intensity analysis. *Acta Photonica Sinica, 32*(2), 182-186.

Yuan, L., Haizhou, A., Yamashita, T., Shihong, L., & Kawade, M. (2008). Tracking in Low Frame Rate Video: A Cascade Particle Filter with Discriminative Observers of Different Life Spans. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 30*(10), 1728-1740.

Yunqiang, C., Yong, R., & Huang, T. S. (2001). *JPDAF based HMM for real-time contour tracking.* Paper presented at the Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on.

Zhang, S.-C., & Liu, Z.-Q. (2005). A robust, real-time ellipse detector. *Pattern Recognition, 38*(2), 273-287.

Zhaowen, W., Xiaokang, Y., Yi, X., & Songyu, Y. (2009). CamShift guided particle filter for visual tracking. *Pattern Recogn. Lett., 30*(4), 407-413.