國立交通大學

電機與控制工程研究所

碩士論文

使用空間與顏色特徵的平均移動演算法
於物件大小與方位追蹤

A New Spatial-Color Mean-Shift Object Tracking Algorithm with Scale and Orientation Estimation

研究生：　阮崇維

指導教授：　胡竹生 博士

中華民國九十六年六月

使用空間與顏色特徵的平均移動演算法

於物件大小與方位追蹤

研究生：阮崇維

指導教授：胡竹生 博士

國立交通大學

電機與控制工程研究所碩士班

摘要

本論文中發展了一個以空間和顏色為基礎的平均移動演算法。其中以空間中顏色分佈的相對資訊和顏色的特徵來定義物件的模型，並以新的相似度函數發展出新的平均移動演算法來做物件追蹤，為了要使物件追蹤的效果更穩健，針對不同的特徵做了實驗並選出使追蹤效果最好的顏色特徵，接著並在演算法中加入了以背景資訊而建立的權重值，使得演算法具有更好的穩定性。而為了解決在物件追蹤中常遇到的物件大小與方位的問題，我們使用了主成分分析的方法來估測物件的方位，並以主成分分析所延伸而來的演算法來估計物件的大小，而此方法確實可以自動更新物件的大小與方位。在最後的實驗中則可以看出此追蹤演算法可以解決部份遮蔽和物件變形的問題，且在複雜背景下仍具有良好的即時追蹤效能。

A New Spatial-Color Mean-Shift Object Tracking Algorithm with Scale and Orientation Estimation

Student： Chung-Wei Juan Advisor： Prof. Jwu-Sheng Hu

Institute of Electrical and Control Engineering
National Chiao-Tung University

ABSTRACT

In this thesis, we propose the new mean-shift tracking algorithms based on a new similarity measure function. The joint spatial-color feature is used as our basic model elements. The target image is modeled with the kernel density estimation and we use the concept of expectation of the estimated kernel density to develop the new similarity measure functions. With these new similarity measure functions, two new similarity-based mean-shift tracking algorithms were derived. To enhance the robustness, we add the weighted-background information to the proposed mean-shift tracking algorithm. In order to solve the deformation problem, the principal component analysis method is used to update the orientation of the tracking object, and a simple method is elaborated to monitor the scale of the object. The results of the experiments show that the new similarity-based tracking algorithms are real-time and can track the moving object correctly, and update the orientation and scale of the object automatically.

誌謝

首先感謝胡竹生老師兩年來辛勤的指導，在老師的實驗室中接觸了各式各樣的東西，學到了相當多專業上的知識，並完成了碩士學位。謝謝黃育綸老師在 TI DSP 大獎賽期間所給予的指導，也感謝老師不時的關心以及在人生方向上的鼓勵與建議。謝謝可愛的蔡中庸老師，當了兩年蔡媽的計概助教，也和老師一起吃了不少好吃的素食。另外謝謝口試委員莊仁輝老師、周志成老師、張文中老師的建議，讓學生更加了解想要做一份好的論文究竟還需要些什麼。

謝謝實驗室的成員， 謝謝永遠只能說實話的永融不時在人生方向與學業間的建議與指教，謝謝宗敏學長在學業期間關於影像方面的的指導，謝謝興哥的人生共勉，謝謝一起打拼的楷祥、弘齡、鳥蕙、恒嘉、螞蟻、佩靜、朱木、鏗元、劉大人、啟揚、趴趴、俊宇的肚子、瓊文、阿吉、楨。

謝謝多年的室友兼好友爽比油(俊瑋)，感謝你的畢業禮物。謝謝我最好的朋友姚明(宇晨)。謝謝廖公(子揚)不時解答我影像處理上的問題，謝謝吁吁吁(志軒)一個禮拜一次的海賊王，謝謝賓拉登常常在大半夜找我去吃宵夜，也謝謝老涂、天婦羅和他們實驗室的 NND，謝謝信元學長、嘉豪學長、賓哥的鼓勵與指導，祝願你們能早日脫離博班苦海。

謝謝大四畢業那年一起在墾丁當解說員的夥伴，欣民、孝旭、舒帆、怡伶、雅瑜、婕瑜、妍伶、彥廷、佩君、嘉玲、建炘、志原、紅生、致融、彭大哥、秋樺姐、怡慧姐、婷葳姐、華珍姐、翠玲姐，和你們在一起帶團的 2 個月大概是我在交大這六年中最快樂的回憶。

最後要謝謝家人的支持以及活動中心公用琴房中那台較破舊的鋼琴，在我研究所後期無數難過與沮喪的日子裡，碰碰你往往都可以讓我暫時忘掉一切，也謝謝陪我度過 6 個酷暑的電風扇，希望你在永融家仍能一如往昔。

Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Motivation and Objective

In related research of the computer vision, the object tracking is an important issue in many computer vision applications. The object tracking can be applied on the surveillance system which can capture the person of unknown identity and notifies related persons immediately. Perceptual interfaces also require the tracking system to capture where the user is. A good tracking system makes driving more secure and assists the driver to handle the situation of navigation. Furthermore, robot system, augmented reality, digital home, and object-based video compression all depend on the object tracking system.

Up to now, there is not a robust object tracking system which can be applied under all kinds of different circumstances. The object tracking system is always developed for specific situations. For example, V. Parameswaran et al. [3] proposed a tunable representation for tracking encoding appearance and geometry but failed for deformation, and F. Porikli et al. [2] presented a method for the low-frame-rate video in which objects have fast motion but failed under the huge variation of illumination.

In general, the tracking system is easily influenced by many factors. An insufficient target representation of tracking system could easily create confusion between the target and the background. Huge variations of illumination always make the appearance of target be different from that of model. Occlusion problem results in an incomplete target representation and makes the tracking fail. Moreover, the computer can not judge the same target with different scale size at the scene automatically. With these problems and related applications, how to track the moving object robustly is an important and interesting research issue.

1.2 Literature Review

In this thesis, we propose an algorithm based on the mean-shift tracking algorithm proposed in [1]. The advantages of the mean-shift tracker include fast operation, robustness and invariance to a large class of object deformations. A large number of related research followed [1] to develop various related aspects such as feature spaces [4] [5], spatial information [6] [7], shape adaptation [8] [9], etc.

In visual tracking, object representation is an important issue because it can describe the correlation between the appearance and the state of the object. An appropriate object representation is more robust and makes the target model more distinguished from the background, and achieves a better tracking result. In [1], D. Comaniciu et al. used the spatial kernels with the pixels which are weighted by a radially symmetric normalized distance from the object center, together with color histograms, to represent blob-alike color objects, and the representation of target make mean-shift tracking more efficient. Radially symmetric kernel preserves representation of the distance of a pixel from the center even the object has a large set of transformations, but this approach only contains the color information of the target and the spatial information is discarded. As shown in Figure 1.2-1, the tracker fails because the rectangular block being tracked overlapped with another block of the same color distribution but inverted spatial distribution of colors.

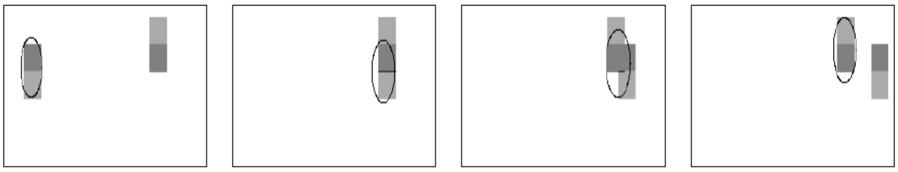


Figure 1.2-1 : Similar color distribution blocks tracking sequence. (The figure is obtained from [3])

Furthermore, V. Parameswaran et al. [3] proposed the tunable kernels for tracking, which simultaneously encodes appearance and geometry that enable the use of mean-shift iterations for tracking. A method was presented to modulate the feature histogram of the target that uses a set of spatial kernels with different bandwidths to encode the spatial information. This method shows how one could learn the optimal set of bandwidths to use the captured data for the case of pedestrians walking upright. This approach indeed can solve the problem of similar color distribution blocks with different spatial configuration, but it just works for some cases, such as walking upright.

Another problem in the visual tracking is how to track the scale of object. In [1], the mean-shift algorithm is run several times, using the current and scaled window sizes. For each different window size, the similarity measure Bhattacharyya coefficient is computed to be compared, and the window size yielding the largest Bhattacharyya coefficient, i.e. the most similar distribution, is chosen as the new current scale. V. Parameswaran et al. [3], S. Birchfield et al. [7] and F. Porikli et al. [10] use the similar variation method to solve the scale problem, but this method is unstable, and easily make the tracker lose the target.

R. Collins [4] extended the mean-shift tracker by adapting T. Lindeberg’s theory [11] of feature scale selection based on local maxima of differential scale-space filters. This method uses blob tracking and a scale kernel to accurately capture the target’s variation in scale. But in the paper the detailed iteration method was not described. Furthermore, an EM-like algorithm [9] is provided to estimate the shape of the local mode. This approach estimates simultaneously the position of the local mode and uses the covariance matrix to describe the approximate shape of object. But this paper also does not illustrate how to decide the scale size from the covariance matrix and other details about implementation.

Q. Zhao et al. [6] and H. Zhang et al. [8] propose the methods to solve the problem of rotation and translation. H. Zhang et al. [8] proposed a method which represents the object by a kernel-based model, which offers more accurate spatial-spectral description than general blob models. Q. Zhao et al. [6] proposed the color correlogram method to use the correlation of colors to solve the related problem. But these methods are not suitable for the complex background situation.

Most papers in literature provide methods for specific applications. This thesis extends the traditional mean-shift tracking algorithm and will propose a new mean-shift based method to improve the arbitrary object tracking problem, and try to estimate the scale and orientation of target.

1.3 Thesis Subject and Contribution

The subject of this thesis can be divided into two parts. The first past is to develop the new spatial-color mean-shift trackers for the purpose of capturing the target more accurately than the traditional mean-shift tracker. The second part is to develop a method for solving the scale and orientation problem which always appears in computer vision.

In the first part, the new spatial-color mean-shift object tracking algorithms are presented, thus the trackers can track the target consistently. The tracking algorithms combine the spatial information and color feature to represent the model more robustly, and use the new similarity measure functions to obtain the iterative mean-shift procedure. Some other extension methods and algorithms are used to improve the performance of these new trackers, such as different color feature space and weighted-background information.

In the second part, this thesis uses principle component analysis method to estimate the scale and orientation of the tracking target. The principle component analysis method can be extended from the tracking algorithm proposed above because the spatial-color mean-shift object tracking algorithms and the principle component analysis method both use the spatial information and weighted-background information.

The proposed spatial-color mean-shift object tracking algorithms are implemented and the experiment results show that the new methods are more robust than the traditional mean-shift tracking algorithm, and can improve the scale and orientation problems.

1.4 Outlines of Thesis

The remainder of this thesis is organized as follows.

Chapter 2: the traditional mean-shift tracking algorithm is reviewed, including how to represent the target model, the traditional similarity measure Bhattacharyya coefficient, how to derive the traditional mean-shift tracker, and the summary of total mean-shift tracking algorithm procedure.

Chapter 3: at first, two recent papers are reviewed, and the similar concept of these two papers is extended to develop the new spatial-color mean-shift tracking algorithms. To improve the new trackers and make them more robust, some extensions of the basic algorithm is discussed and applied. Finally, the algorithm for solving scale and orientation is presented and the total algorithm is summarized at the end of this chapter.

Chapter 4: the experiment results are presented according to the developing steps of algorithms in chapter 3. Some real image sequences and figures are presented, and the experiment results are discussed.

Chapter 5: the conclusion of this thesis and the possible improvement in the future is presented in this chapter.

Chapter 2. Traditional Mean-Shift Tracking Algorithm

2.1 Introduction

Mean shift tracking algorithm [1] is a template base image tracking algorithm. The main concept of mean-shift tracking is to find the candidate which is the most similar with target image by mean-shift iterations. The principle of mean-shift is to compare the color distribution of candidate region with the color distribution of the model, and to compute the similarity measure, Bhattacharyya coefficient, to observe the variation of gradient of candidate to find the mean-shift vector. Further, mean-shift finds the most similar region or the most possible area of the candidate. In later sections, we will introduce the derivation and principle of the traditional mean-shift tracking algorithm.

2.2 Target Representation

Mean-shift is a template based algorithm, so we must find a feature to represent our target model. In general, we always choose the color p.d.f. as our reference model. We consider the center of target model as location **0** and the candidate is defined at location **y**. Further, we define the target model as **q** and the target candidate as $\mathbf{p}(\mathbf{y})$. In practice, the image data are classified to *m*-bin histograms in order to reduce the computational complexity. Thus we define the target model as

$$\hat{\mathbf{q}} = \{\hat{q}_u\}_{u=1,\dots,m} \qquad \sum_{u=1}^{m} \hat{q}_u = 1 \tag{2-1}$$

and the target candidate as

$$\hat{\mathbf{p}}(\mathbf{y}) = \{\hat{p}_u(\mathbf{y})\}_{u=1,\dots,m} \qquad \sum_{u=1}^{m} \hat{p}_u = 1 \tag{2-2}$$

Although, the histogram is not the best nonparametric density estimate [16], it is simple and sufficient for traditional mean-shift algorithm.

2.2.1 Model Representation

We need to capture the character to form a p.d.f. from the target model image with the first step of mean-shift tracking algorithm. Let $\{\mathbf{x}_i^*\}_{i=1\dots n}$ represent the pixel locations of the region which we want to track in the target model, and we consider the center of target model as location **0**. We define the function $b : R^2 \rightarrow \{1,\dots,m\}$ as color index, and the value of function $b(\mathbf{x}_i^*)$ is the index of its bin in the quantized feature space of pixel $\mathbf{x}_i^*$. The probability of the feature $u = 1,\dots,m$ of the target model is then defined as

$$\hat{q}_u = C\sum_{i=1}^{n} k(\left\|\mathbf{x}_i^*\right\|^2)\delta[b(\mathbf{x}_i^*) - u] \tag{2-3}$$

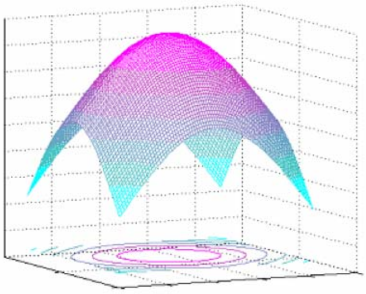
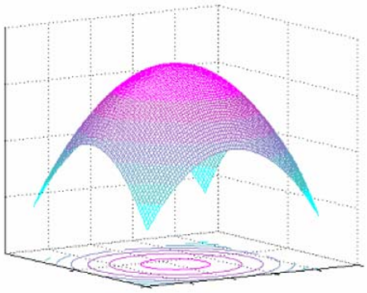
where δ is the Kronecher delta function, C is the normalization constant computed for condition $\sum_{u=1}^{m} \hat{q}_u = 1$, so we can obtain

$$C = \frac{1}{\sum_{i=1}^{n} k(\left\|\mathbf{x}_i^*\right\|^2)} \tag{2-4}$$

since the summation of delta functions is equal to one for $u = 1,\dots,m$.

In (2-3) and (2-4), $k(\left\|\mathbf{x}_i^*\right\|^2)$ is a convex and monotonic decreasing kernel function which contains the highest weight at the center and smaller weights to pixels farther from the center. In general, the pixel near the center of the target model region is more important than the pixel near the periphery. In some situations, the periphery of the target is covered by some obstacles, and the weights improve the robustness of the tracking result because the peripheral pixels are less significant. D.W. Scott [16] and D. Comaniciu et al. [17] mention two functions which are normal function (Gaussian function) and Epanechnikov function are more suitable to be the kernel function of mean-shift tracking algorithm. We list some information about these two functions in Table 2.2-1.

Table 2.2-1: Two weight kernel functions.

Function Name	Definition	Sketch with $d = 2$
Normal Function (Gaussian Function)	$K_N(\mathbf{x}) = \begin{cases} \frac{1}{(2\pi)^{d/2}} \exp(-\frac{1}{2}\|\mathbf{x}\|^2), & \mathit{if}\ \|\mathbf{x}\| < 1 \\ 0, & \mathit{otherwise.} \end{cases}$	
Epanechnikov Function	$K_E(\mathbf{x}) = \begin{cases} \frac{1}{2C_d}(d+2)(1-\|\mathbf{x}\|^2), & \mathit{if}\ \|\mathbf{x}\| < 1 \\ 0, & \mathit{otherwise.} \end{cases}$	
d: dimension of space (in our 2D image case, $d = 2$) C_d: the volume of the unit d-Dimension sphere (in our 2D image case, $C_d = \pi$)		

2.2.2 Candidate Representation

Now we define the p.d.f. of candidate in mean-shift tracking algorithm. Let $\{\mathbf{x}_i\}_{i=1,\ldots,n_h}$ represent the pixel locations of the region in the target candidate, which centered at $\mathbf{y}$ in the current frame. As the same in 2.2.1, we define $b(\mathbf{x})$ as color index of its bin in the quantized feature space of pixel $\mathbf{x}_i$. The probability of the feature $u = 1,\ldots,m$ of the target candidate is then defined as

$$\hat{p}_u(\mathbf{y}) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2\right)\delta[b(\mathbf{x}_i)-u] \qquad (2\text{-}5)$$

where $k(x)$ is the same kernel function with target model, h is bandwidth which defines the region size of the candidate, δ is the Kronecher delta function, C_h is the normalization constant computed for condition $\sum_{u=1}^{m} \hat{p}_u = 1$, so we can obtain

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2\right)} \qquad (2\text{-}6)$$

Note that C_h does not depend on $\mathbf{y}$, since the pixel locations $\mathbf{x}_i$ are organized in a regular lattice and $\mathbf{y}$ is one of the lattice nodes [1]. Therefore, C_h can be pre-calculated for a given kernel and different values of h.

2.3 Similarity Based on Bhattacharyya Coefficient

The similarity measure function is to compare the similarity between the target candidate and the target model to find the most similar region. There are various similarity measure functions to be used for different target representations. A differentiable kernel function yields a differentiable similarity function and efficient gradient-based optimizations procedures can be used for finding its local maximum which is the most possible region which we want to track.

In traditional mean-shift tracking algorithm, Bhattacharyya coefficient is used as the similarity measure function. First, the similarity function is defined as a distance among model and candidate, and the distance between two discrete distributions as

$$d(\mathbf{y}) = \sqrt{1-\rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}]} \tag{2-7}$$

and then ρ is chosen as Bhattacharyya coefficient between candidate $\hat{\mathbf{p}}$ and model $\hat{\mathbf{q}}$.

$$\hat{\rho}(\mathbf{y}) \equiv \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{y})\hat{q}_u} \tag{2-8}$$

The concept of Bhattacharyya coefficient is the cosine of the angle between the *m*-dimensional unit vectors $(\sqrt{\hat{\rho}_1}, \ldots, \sqrt{\hat{\rho}_m})^T$ and $(\sqrt{\hat{q}_1}, \ldots, \sqrt{\hat{q}_m})^T$, and it is an efficient and divergence-type for statistical measure.

With a different point of view, Bhattacharyya coefficient can be considered as the error estimation of two distributions. Figure 2.3-1 shows that we can obtain the best classification according to the vertical line of point A, and we can get the smallest error which is the yellow region. The larger error about classification results in larger Bhattacharyya coefficient and represents high similarity, and smaller error results in smaller Bhattacharyya coefficient and represents low similarity.

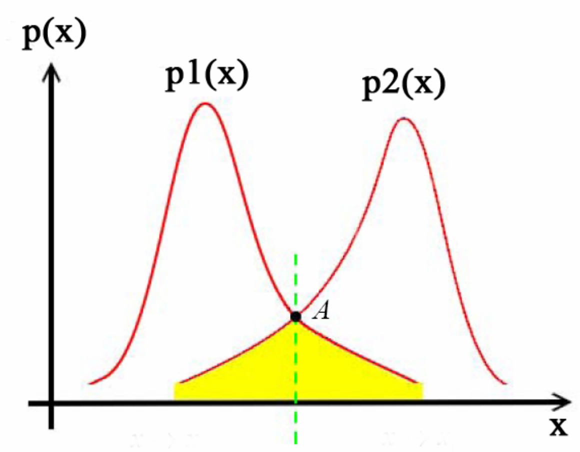


Figure 2.3-1 : Illustration of classification of two distributions.

2.4 Traditional Mean-Shift Tracker

Minimizing (2-7) is equivalent to maximizing the sample estimation of the

Bhattacharyya coefficient (2-8). Using Taylor expansion to expand (2-8) around the values $\hat{\rho}_u(\mathbf{y})$ at $\mathbf{y}=\hat{\mathbf{y}}_0$ which is the location of the target in previous frame. We start to find the new target location in current frame. The linear approximation is obtained as

$$\begin{aligned}
\rho[\hat{\mathbf{p}}(\mathbf{y}),\hat{\mathbf{q}}] &= \sum_{u=1}^{m}\sqrt{\hat{\rho}_u(\hat{\mathbf{y}}_0)\hat{q}_u} \\
&= \sum_{n=0}^{\infty}\frac{\left.\dfrac{\partial^n \rho[\hat{\mathbf{p}}(\mathbf{y}),\hat{\mathbf{q}}]}{\partial\hat{\rho}_u(\mathbf{y})^n}\right|_{\mathbf{y}=\mathbf{y}_0}}{n!}[\hat{\rho}_u(\hat{\mathbf{y}})-\hat{\rho}_u(\hat{\mathbf{y}}_0)]^n \\
&= \rho[\hat{\mathbf{p}}(\mathbf{y}),\hat{\mathbf{q}}]\Big|_{\mathbf{y}=\mathbf{y}_0} + \left.\frac{\partial\rho[\hat{\mathbf{p}}(\mathbf{y}),\hat{\mathbf{q}}]}{\partial\hat{\rho}_u(\mathbf{y})}\right|_{\mathbf{y}=\mathbf{y}_0}[\hat{\rho}_u(\hat{\mathbf{y}})-\hat{\rho}_u(\hat{\mathbf{y}}_0)] \\
&\quad + \frac{\left.\dfrac{\partial^2 \rho[\hat{\mathbf{p}}(\mathbf{y}),\hat{\mathbf{q}}]}{\partial\hat{\rho}_u(\mathbf{y})^2}\right|_{\mathbf{y}=\mathbf{y}_0}}{2!}[\hat{\rho}_u(\hat{\mathbf{y}})-\hat{\rho}_u(\hat{\mathbf{y}}_0)]^2+\ldots\ldots \\
&\approx \rho[\hat{\mathbf{p}}(\mathbf{y}),\hat{\mathbf{q}}]\Big|_{\mathbf{y}=\mathbf{y}_0} + \left.\frac{\partial\rho[\hat{\mathbf{p}}(\mathbf{y}),\hat{\mathbf{q}}]}{\partial\hat{\rho}_u(\mathbf{y})}\right|_{\mathbf{y}=\mathbf{y}_0}[\hat{\rho}_u(\hat{\mathbf{y}})-\hat{\rho}_u(\hat{\mathbf{y}}_0)] \\
&= \sum_{u=1}^{m}\sqrt{\hat{\rho}_u(\mathbf{y})\hat{q}_u}\Bigg|_{\mathbf{y}=\mathbf{y}_0} + \sum_{u=1}^{m}\frac{1}{2\sqrt{\hat{\rho}_u(\mathbf{y})\hat{q}_u}}\Bigg|_{\mathbf{y}=\mathbf{y}_0}\cdot\hat{q}_u\cdot[\hat{\rho}_u(\hat{\mathbf{y}})-\hat{\rho}_u(\hat{\mathbf{y}}_0)] \\
&= \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{\rho}_u(\hat{\mathbf{y}}_0)\hat{q}_u}+\frac{1}{2}\sum_{u=1}^{m}\hat{\rho}_u(\mathbf{y})\sqrt{\frac{\hat{q}_u}{\hat{\rho}_u(\mathbf{y}_0)}} \qquad \text{(2-9)}
\end{aligned}$$

The approximation is established under the assumption of the target not moving drastically from previous location $\hat{\mathbf{y}}_0$ to current location $\mathbf{y}$, and this condition is always satisfactory between consecutive image frames. Substituting (2-5) into (2-9), we can obtain

$$\rho(\mathbf{y})=\rho[\hat{\mathbf{p}}(\mathbf{y}),\hat{\mathbf{q}}]\approx\frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{\rho}_u(\hat{\mathbf{y}}_0)\hat{q}_u}+\frac{1}{2}\sum_{u=1}^{m}C_h\sum_{i=1}^{n_h}k\left(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2\right)\delta[b(\mathbf{x}_i)-u]\sqrt{\frac{\hat{q}_u}{\hat{\rho}_u(\mathbf{y}_0)}}$$

$$=\frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{\rho}_u(\hat{\mathbf{y}}_0)\hat{q}_u}+\frac{C_h}{2}\sum_{i=1}^{n_h}w_ik(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2) \tag{2-10}$$

where

$$w_i=\sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{\rho}_u(\hat{\mathbf{y}}_0)}}\delta[b(\mathbf{x}_i)-u] \tag{2-11}$$

The objective is to find the maximum of Bhattacharyya coefficient $\rho(\mathbf{y})$. Because $\rho(\mathbf{y})$ is independent of **y**, the term of $\frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{\rho}_u(\hat{\mathbf{y}}_0)\hat{q}_u}$ in (2-10) does not affect the value of $\rho(\mathbf{y})$, and $\rho(\mathbf{y})$ is only influenced by

$$f(\mathbf{y})=\frac{C_h}{2}\sum_{i=1}^{n_h}w_ik(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2) \tag{2-12}$$

Further, using gradient-based optimizations procedure with (2-12), we obtain

$$\nabla f(\mathbf{y})=\frac{C_h}{2h^2}\sum_{i=1}^{n_h}(\mathbf{y}-\mathbf{x}_i)w_ik'(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2) \tag{2-13}$$

Letting $g(x)=-k'(x)$, we obtain

$$\nabla f(\mathbf{y})=\frac{C_h}{2h^2}\sum_{i=1}^{n_h}(\mathbf{x}_i-\mathbf{y})w_ig(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2)$$

$$=\frac{C_h}{2h^2}\left[\sum\nolimits_{i=1}^{n_h}g(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2)\right]\times\left[\frac{\sum\nolimits_{i=1}^{n_h}\mathbf{x}_iw_ig(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2)}{\sum\nolimits_{i=1}^{n_h}g(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2)}-\mathbf{y}\right] \tag{2-14}$$

We can separate (2-14) into two parts. The first term is proportional to the density estimation at **y** and $\sum\nolimits_{i=1}^{n_h}g(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2)$ is assumed as a positive number [18], and let (2-14) be equal to **0**.

$$\nabla f(\mathbf{y})=\mathbf{0} \tag{2-15}$$

The second term can be obtained as the mean shift vector.

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i w_i g\left(\left\| \frac{\hat{\mathbf{y}}_0 - \mathbf{x}_i}{h} \right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\| \frac{\hat{\mathbf{y}}_0 - \mathbf{x}_i}{h} \right\|^2\right)} \tag{2-16}$$

The mean shift vector always points toward the direction of maximum increase in the density. In this procedure, we can find the local maximum of the density by (2-16), and the kernel region can recursively moved from current location $\hat{\mathbf{y}}_0$ to the new location $\hat{\mathbf{y}}_1$.

2.5 Mean-Shift Tracking Algorithm Procedure

The complete traditional mean-shift tracking algorithm is presented as Figure 2.5-1.

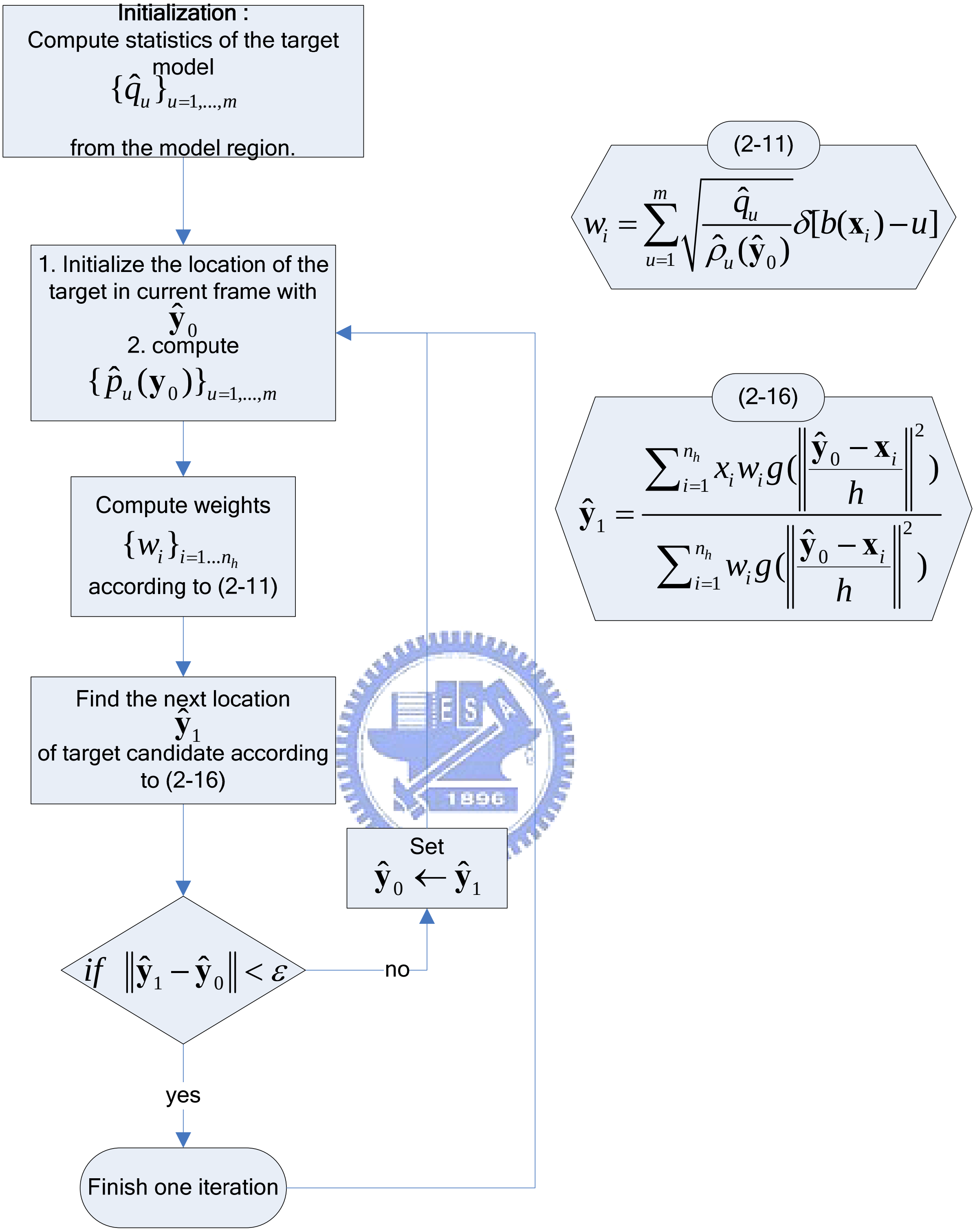

Initialization :
Compute statistics of the target model
$\{\hat{q}_u\}_{u=1,\dots,m}$
from the model region.
1. Initialize the location of the target in current frame with
$\hat{\mathbf{y}}_0$
2. compute
$\{\hat{p}_u(\hat{\mathbf{y}}_0)\}_{u=1,\dots,m}$
Compute weights
$\{w_i\}_{i=1\dots n_h}$
according to (2-11)
Find the next location
$\hat{\mathbf{y}}_1$
of target candidate according to (2-16)
Set
$\hat{\mathbf{y}}_0 \leftarrow \hat{\mathbf{y}}_1$
$if \ \left\|\hat{\mathbf{y}}_1 - \hat{\mathbf{y}}_0\right\| < \varepsilon$
no
yes
Finish one iteration
(2-11)
$w_i = \sum_{u=1}^{m} \sqrt{\frac{\hat{q}_u}{\hat{\rho}_u(\hat{\mathbf{y}}_0)}} \delta[b(\mathbf{x}_i) - u]$
(2-16)
$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{\hat{\mathbf{y}}_0 - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\hat{\mathbf{y}}_0 - \mathbf{x}_i}{h}\right\|^2\right)}$

Figure 2.5-1 : Traditional mean-shift tracking algorithm procedure.

Chapter 3. Spatial-Color Mean-Shift Object Tracking Algorithm

3.1 Introduction

In this chapter, we will introduce two papers [7][12] with the spatiogram and the new similarity measure. With the spatiogram of [7], we will extend the original model to a new model with spatial and color feature information, and then use resembling method as [12] to get two different similarity measures. We will derive the iterative mean-shift tracking algorithms from the similarity measure functions. And then we will discuss the two different color features and select better color feature space by experiment test. To improve the robustness, we will take account of the background information and add the background-weighted parameter to the new mean-shift algorithms. In the final step, we will discuss the scale problem and try to use the principal component analysis method to solve it. In conclusion, we will give a summary and list the complete new mean-shift algorithm procedures.

3.2 Model Definition

In traditional mean-shift tracking algorithm, color histogram is used as the target representation. Color histogram discards all spatial information and uses the concept of color distribution to represent the target. This foundation technique is used to develop several tracking systems [2] [4] [10] which show that color histogram is robustness about deformation of the tracked object. But in some circumstances, spatial information is important and advantageous for different interference. In this chapter, we want to build the target with color and spatial information.

3.2.1 Paper Survey about Spatiogram

Recently, S. Birchfield et al. [7] proposed the concept of each histogram bin which contains the mean and covariance information of the locations of the pixels which belong to that bin. This idea involves the spatially weighted by the mean and covariance of the locations of the pixels and not only the color information as traditional method. They call this concept a spatial histogram, or spatiogram. The model of spatiogram of an image *I* can be represented as follows.

$$h_I(b) = \langle n_b, \boldsymbol{\mu}_b, \Sigma_b \rangle \qquad b = 1, \ldots, B \tag{3-1}$$

where n_b is the number of pixels whose values belong to the *b*-th bin, $\boldsymbol{\mu}_b$ is mean vector of locations of all pixels which belongs to the *b*-th bin (i.e. the 2D coordinates), and Σ_b is covariance matrix of locations of all pixels which belongs to the *b*-th bin (symmetric matrix), and the pixels in the image can be classified to *B* bins.

The spatiogram captures the spatial information of the general histogram bins, but the traditional color histogram only gets the color distribution information. For instance, Figure 3.2-1 illustrates the difference between the spatiogram and traditional histogram. There are three different poses of a person’s head in the first row. For each person’s head to compute the histogram and spatiogram first, if we want to rebuild the original image from the computed histogram, the second row shows that we only can get the disorderly image which barely contains the color information. However, the image rebuilt form the computed spatiogram reveals the relationships about the color as shown in the third row. This paper uses the spatiogram and the general Bhattacharyya coefficient to derive a mean shift procedure algorithm and improve the tracking result when being compared with histogram method. The experiment results demonstrate that spatial information captures a firmer description of the target to improve robustness in tracking.

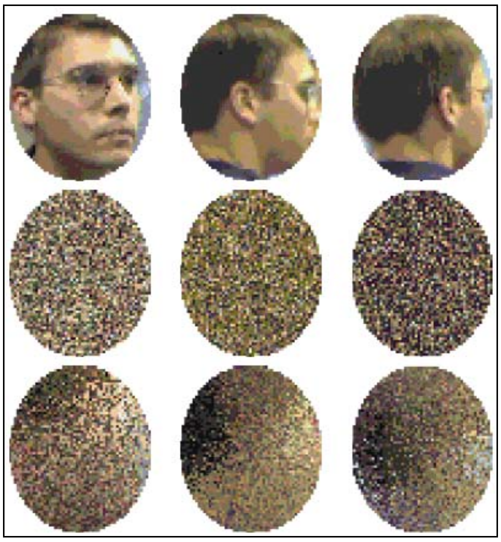


Figure 3.2-1 : Three different poses of a person's head (top), images generated from the computed histogram (middle), images generated from the computed spatiogram (bottom). (The figure is obtained from [7].)

3.2.2 A Joint Spatial-Color Feature Model

As shown in Figure 3.2-2, if cyan and blue belong to the same bin, these two blocks have the same spatiogram, but they have different color patterns. To keep the robustness of color description of the spatiogram, we extend the spatiogram and define our joint spatial-color model as

$$h_I(b) = \langle n_b, \boldsymbol{\mu}_{P,b}, \Sigma_{P,b}, \boldsymbol{\mu}_{C,b}, \Sigma_{C,b} \rangle \qquad b = 1,\ldots,B \tag{3-2}$$

where n_b , $\boldsymbol{\mu}_{P,b}$, $\Sigma_{P,b}$ are the same as spatiogram which S. Birchfield et al. proposed, and are respectively the number of pixels, the mean vector of locations, and covariance matrix of locations of pixels which belong to the *b*-th bin. In (3-2), we add two elements. $\boldsymbol{\mu}_{C,b}$ is mean vector of color feature with *d* color channels of all pixels which belong to the *b*-th bin (for example, in RGB color space, *d*=3 and $\boldsymbol{\mu}_{C,b} = (R_b, G_b, B_b)$). $\Sigma_{C,b}$ is covariance matrix of color feature with *d* color channels of all pixels which belong to the *b*-th bin, and the pixels in the image can be classified to *B* bins. We choose RGB channels as the color feature first, so $\boldsymbol{\mu}_{C,b}$ is a 3D vector, $\Sigma_{C,b}$ is a symmetric matrix, and we will discuss another more robust color feature in 3.5.

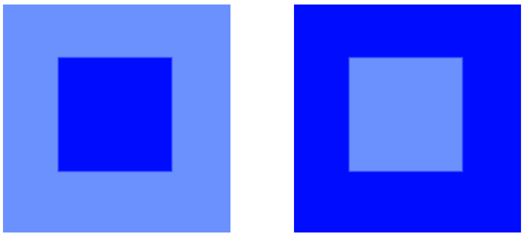


Figure 3.2-2 : Illustration of the same spatial information with different color distribution for one bin.

3.3 Paper Survey about New Similarity Measure

The significance of mean shift algorithm is to find the local maximum of the similarity measure between the image model and the candidate. In general, the similarity measure can be derived to a mean shift algorithm and we use the iterative result to track the candidate location. The most general similarity measures used in image tracking are the Bhattacharyya coefficient and the Kullback-Leibler divergence. For the spatial-color model, we want to find a simple similar measure function to obtain the mean-shift algorithm.

Lately, C. Yang et al. [12] proposed a new simple symmetric similarity function between kernel density estimates of the template and candidate distributions in a joint spatial-feature space and then presented an iterative tracking algorithm. This paper denotes model image as $I_x = \{\mathbf{x}_i, u_i\}_{i=1,\ldots,M}$ and candidate image as $I_y = \{\mathbf{y}_j, v_j\}_{j=1,\ldots,N}$, where $\mathbf{x}_i$ and $\mathbf{y}_j$ are locations of pixels, u_i and v_j belong to the feature space. This paper describes the target feature distribution in the joint spatial-feature space, and uses estimated kernel density function to model the p.d.f. of the object in the model image as

$$\hat{p}_x(\mathbf{x},u)=\frac{1}{N}\sum_{i=1}^{N} w\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{\sigma}\right\|^2\right)k\left(\left\|\frac{u-u_i}{h}\right\|^2\right) \tag{3-3}$$

where σ and h are the bandwidths in spatial and feature space. We can also regard (3-3) as a spatially weighted function *w* with Gaussian Mixture Model of feature space *k*.

Finally, this paper uses expectation of the estimated kernel density function between the model I_x and candidate I_y in the joint feature-spatial space as similarity measure

$$J(I_x,I_y)=\frac{1}{N}\sum_{j=1}^{M}\hat{p}_x(\mathbf{y}_j,v_j) \tag{3-4}$$

$$=\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M} w\left(\left\|\frac{\mathbf{x}_i-\mathbf{y}_j}{\sigma}\right\|^2\right)k\left(\left\|\frac{u_i-v_j}{h}\right\|^2\right) \tag{3-5}$$

The paper then uses (3-5) to derive a similarity-based mean-shift tracking algorithm. The experiment results show that it is more accurate and the number of iterations is less than the traditional Bhattacharyya coefficient method. This main concept of the new similarity function is based on the expectation of all pixels over the model and candidate.

3.4 Spatial-Color Mean-Shift Object Tracking Algorithm

With the spatial-color feature and the concept of expectation, we develop two different tracking algorithms. The detailed statement and demonstration will be derived as follows.

3.4.1 Kernel Density Estimation of the Model Image

We start to derive the first similarity-based mean-shift tracker. First of all, we use model (3-2) and set the image model as the estimated kernel density function.

$$\begin{aligned} p(\mathbf{x},u) &= \frac{1}{M}\sum_{i=1}^{M} \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{P,b(i)})^T(\Sigma_{P,b(i)})^{-1}(\mathbf{x}-\boldsymbol{\mu}_{P,b(i)})\right)}{2\pi\left|\Sigma_{P,b(i)}\right|^{1/2}} \frac{\exp\left(-\frac{1}{2}(\mathbf{c}_\mathbf{x}-\boldsymbol{\mu}_{C,b(i)})^T(\Sigma_{C,b(i)})^{-1}(\mathbf{c}_\mathbf{x}-\boldsymbol{\mu}_{C,b(i)})\right)}{(2\pi)^{3/2}\left|\Sigma_{C,b(i)}\right|^{1/2}}\delta[u-b(i)] \\ &\triangleq \frac{1}{M}\sum_{i=1}^{M} K_P(\mathbf{x}-\boldsymbol{\mu}_{P,b(i)},\Sigma_{P,b(i)})K_C(\mathbf{c}_\mathbf{x}-\boldsymbol{\mu}_{C,b(i)},\Sigma_{C,b(i)})\delta[u-b(i)] \end{aligned} \tag{3-6}$$

where $b(i)$ is the color bin which pixel i belongs to. K_P and K_C are multivariate Gaussian kernel functions. We use the delta function whose role is the Gaussian function in (3-3), the difference of these two concepts is that the Gaussian function contains the smoothed component but the delta function does not. We can also regard K_P and K_C as the spatially weighted and color-feature weighted function.

3.4.2 Similarity Measure Function

Similar with the concept of the expectation of the estimated kernel density as (3-5), we can get a new similarity measure function among the model $I_x = \{\mathbf{x}_i, u_i\}_{i=1,\ldots M}$ and candidate $I_y = \{\mathbf{y}_j, v_j\}_{j=1,\ldots,N}$ as

$$\begin{aligned} J(I_x, I_y) = J(\mathbf{y}) &= \frac{1}{N}\sum_{j=1}^{N} p(\mathbf{y}_j, v_j) \\ &= \frac{1}{NM}\sum_{j=1}^{N}\sum_{i=1}^{M} K_P(\mathbf{y}_j-\boldsymbol{\mu}_{P,b(i)},\Sigma_{P,b(i)})K_C(\mathbf{c}_{\mathbf{y}_j}-\boldsymbol{\mu}_{C,b(i)},\Sigma_{C,b(i)})\delta[v_j-b(i)] \end{aligned} \tag{3-7}$$

As shown in Figure 3.4-1, if there is no deformation between candidate and target, and the distance of motion is not large between frames, we can consider the motion of object of two frames as a pure translation. Under these assumptions, the center of target with respect to the mean of location of the *b*-th bin in the model is in proportion to the center of candidate with respect to the mean of location of the *b*-th bin in the candidate image. So we can obtain

$$\boldsymbol{\mu}_{P,b(i)} - \mathbf{x} = \boldsymbol{\mu}_{P,b(j)} - \mathbf{y} \tag{3-8}$$

$$\boldsymbol{\mu}_{P,b(i)} = \boldsymbol{\mu}_{P,b(j)} - \mathbf{y} + \mathbf{x} \tag{3-9}$$

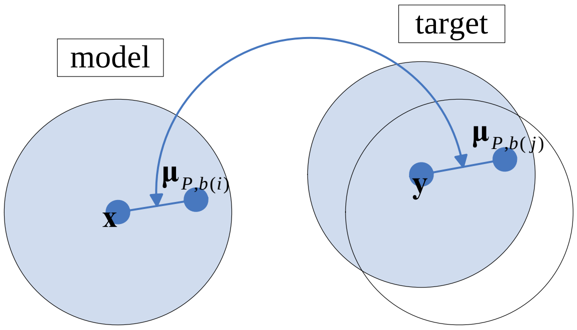


Figure 3.4-1 : Illustration of pure translation.

Substitute (3-9) into (3-7), we can obtain the new similarity measure function as follows.

$$J(\mathbf{y}) = \frac{1}{NM}\sum_{j=1}^{N}\sum_{i=1}^{M} K_P(\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)} + \mathbf{y} - \mathbf{x}, \Sigma_{P,b(i)}) K_C(\mathbf{c}_{\mathbf{y}_j} - \boldsymbol{\mu}_{C,b(i)}, \Sigma_{C,b(i)}) \delta[v_j - b(i)] \tag{3-10}$$

3.4.3 Spatial-Color Mean-Shift Tracker

Similar with traditional Bhattacharyya coefficient method, we want to find the maximum value of the similarity measure to get the best candidate, so we let the gradient of the similarity function with respect to the vector $\mathbf{y}$ be equal to $\mathbf{0}$.

$$\nabla J(\mathbf{y}) = \mathbf{0}$$

$$\Rightarrow \frac{1}{NM}\sum_{j=1}^{N}\sum_{i=1}^{M} \text{-}(\Sigma_{P,b(i)})^{-1}(\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)} + \mathbf{y} - \mathbf{x}) K_P K_C \delta[v_j - b(i)] = \mathbf{0}$$

$$\Rightarrow \left\{\sum_{j=1}^{N}\sum_{i=1}^{M} (\Sigma_{P,b(i)})^{-1} K_P K_C \delta[v_j - b(i)]\right\}(\mathbf{y} - \mathbf{x}) = \sum_{j=1}^{N}\sum_{i=1}^{M} (\Sigma_{P,b(i)})^{-1}(\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)}) K_P K_C \delta[v_j - b(i)]$$

$$\mathbf{y} - \mathbf{x} = \left\{\sum_{j=1}^{N}\sum_{i=1}^{M} (\Sigma_{P,b(i)})^{-1} K_P K_C \delta[v_j - b(i)]\right\}^{-1} \left\{\sum_{j=1}^{N}\sum_{i=1}^{M} (\Sigma_{P,b(i)})^{-1}(\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)}) K_P K_C \delta[v_j - b(i)]\right\} \tag{3-11}$$

(3-11) is the mean shift vector and also an iterative function with respect to $\mathbf{y}$, and we rewrite (3-11) as

$$\mathbf{y}_{new}=\left\{\sum_{j=1}^{N}\sum_{i=1}^{M}(\Sigma_{P,b(i)})^{-1}K_P K_C\delta[v_j-b(i)]\right\}^{-1}\left\{\sum_{j=1}^{N}\sum_{i=1}^{M}(\Sigma_{P,b(i)})^{-1}(\mathbf{y}_j-\boldsymbol{\mu}_{P,b(j)})K_P K_C\delta[v_j-b(i)]\right\}+\mathbf{x}$$

(3-12)

where

$$K_P(\mathbf{y}_j-\boldsymbol{\mu}_{P,b(j)}+\mathbf{y}_{old}-\mathbf{x},\Sigma_{P,b(i)})$$
$$=\frac{\exp\left(-\frac{1}{2}(\mathbf{y}_j-\boldsymbol{\mu}_{P,b(j)}+\mathbf{y}_{old}-\mathbf{x})^T(\Sigma_{P,b(i)})^{-1}(\mathbf{y}_j-\boldsymbol{\mu}_{P,b(j)}+\mathbf{y}_{old}-\mathbf{x})\right)}{2\pi\left|\Sigma_{P,b(i)}\right|^{1/2}} \tag{3-13}$$

$$K_C(\mathbf{c}_{\mathbf{y}_j}-\boldsymbol{\mu}_{C,b(i)},\Sigma_{C,b(i)})$$
$$=\frac{\exp\left(-\frac{1}{2}(\mathbf{c}_{\mathbf{y}_j}-\boldsymbol{\mu}_{C,b(i)})^T(\Sigma_{C,b(i)})^{-1}(\mathbf{c}_{\mathbf{y}_j}-\boldsymbol{\mu}_{C,b(i)})\right)}{(2\pi)^{3/2}\left|\Sigma_{C,b(i)}\right|^{1/2}} \tag{3-14}$$

and $\mathbf{y}_{new}$ is the new position of the target which we want to track and $\mathbf{y}_{old}$ is the current position.

3.4.4 Another Derivation of the New Mean-Shift Tracker

Now we want to use another method to derive the second similarity-based mean-shift tracker. As the kernel density estimation model (3-6) which we defined in 3.4.1, if we replace $\mathbf{x}$ by $\mathbf{x}_i$, and $\mathbf{c}_{\mathbf{x}}$ by $\mathbf{c}_i$ in (3-6), we can get a new kernel density estimation function as

$$p(u)=\frac{1}{M}\sum_{i=1}^{M}K_P(\mathbf{x}_i-\boldsymbol{\mu}_{P,b(i)},\Sigma_{P,b(i)})K_C(\mathbf{c}_i-\boldsymbol{\mu}_{C,b(i)},\Sigma_{C,b(i)})\delta[u-b(i)] \tag{3-15}$$

where

$$K_P(\mathbf{x}_i-\boldsymbol{\mu}_{P,b(i)},\Sigma_{P,b(i)})=\frac{\exp\left(-\frac{1}{2}(\mathbf{x}_i-\boldsymbol{\mu}_{P,b(i)})^T(\Sigma_{P,b(i)})^{-1}(\mathbf{x}_i-\boldsymbol{\mu}_{P,b(i)})\right)}{2\pi\left|\Sigma_{P,b(i)}\right|^{1/2}} \tag{3-16}$$

$$K_C(\mathbf{c}_i-\boldsymbol{\mu}_{C,b(i)},\Sigma_{C,b(i)}) = \frac{\exp\left(-\frac{1}{2}(\mathbf{c}_i-\boldsymbol{\mu}_{C,b(i)})^T(\Sigma_{C,b(i)})^{-1}(\mathbf{c}_i-\boldsymbol{\mu}_{C,b(i)})\right)}{(2\pi)^{3/2}\left|\Sigma_{C,b(i)}\right|^{1/2}} \tag{3-17}$$

K_P and K_C are also the spatially weighted and color-feature weighted functions, but these two weighted functions are depend on the image model.

With similar concept of the expectation of the estimated kernel density used in 3.4.2, we define another new similarity measure function between the model $I_x=\{\mathbf{x}_i,u_i\}_{i=1,\dots,M}$ and candidate $I_y=\{\mathbf{y}_j,v_j\}_{j=1,\dots,N}$ as

$$\begin{aligned} J(I_x,I_y)=J(\mathbf{y}) &= \frac{1}{N}\sum_{j=1}^{N}G(\mathbf{y}-\mathbf{y}_j)p(v_j) \\ &= \frac{1}{NM}\sum_{j=1}^{N}\sum_{i=1}^{M}G(\mathbf{y}-\mathbf{y}_j)K_P(\mathbf{x}_i-\boldsymbol{\mu}_{P,b(i)},\Sigma_{P,b(i)})K_C(\mathbf{c}_i-\boldsymbol{\mu}_{C,b(i)},\Sigma_{C,b(i)})\delta[v_j-b(i)] \end{aligned} \tag{3-18}$$

where $\mathbf{y}$ is the center of the candidate image, $G(\mathbf{y}-\mathbf{y}_j)$ is a weighted function which is spatially weighted depends on the candidate image. (3-18) is another new similarity measure function which we proposed.

Now we let the gradient of the similarity function with respect to the vector $\mathbf{y}$ be equal to $\mathbf{0}$ to find the maximum value of the similarity measure to obtain the best candidate.

$$\nabla J(\mathbf{y})=\mathbf{0} \tag{3-19}$$

$$\Rightarrow \frac{1}{NM}\sum_{j=1}^{N}\sum_{i=1}^{M}(\mathbf{y}-\mathbf{y}_j)G'(\mathbf{y}-\mathbf{y}_j)K_PK_C\delta[v_j-b(i)]=\mathbf{0}$$

$$\Rightarrow \mathbf{y}\sum_{j=1}^{N}\sum_{i=1}^{M}G'(\mathbf{y}-\mathbf{y}_j)K_PK_C\delta[v_j-b(i)]=\sum_{j=1}^{N}\sum_{i=1}^{M}\mathbf{y}_jG'(\mathbf{y}-\mathbf{y}_j)K_PK_C\delta[v_j-b(i)]$$

$$\Rightarrow \mathbf{y} = \frac{\sum_{j=1}^{N}\sum_{i=1}^{M}\mathbf{y}_j G'(\mathbf{y}-\mathbf{y}_j)K_P K_C \delta[v_j - b(i)]}{\sum_{j=1}^{N}\sum_{i=1}^{M} G'(\mathbf{y}-\mathbf{y}_j)K_P K_C \delta[v_j - b(i)]} \tag{3-20}$$

So we obtain (3-20) which is another iterative mean shift vector and we rewrite (3-20) as

$$\mathbf{y}_{new} = \frac{\sum_{j=1}^{N}\sum_{i=1}^{M}\mathbf{y}_j G'(\mathbf{y}_{old}-\mathbf{y}_j)K_P K_C \delta[v_j - b(i)]}{\sum_{j=1}^{N}\sum_{i=1}^{M} G'(\mathbf{y}_{old}-\mathbf{y}_j)K_P K_C \delta[v_j - b(i)]} \tag{3-21}$$

where $\mathbf{y}_{new}$ is the new position of the target and $\mathbf{y}_{old}$ is the current position. (3-21) contains the spatially weighted term $G'(\mathbf{y}_{old}-\mathbf{y}_j)$, and we choose function G as the Epanechnikov kernel function as

$$K(\mathbf{x}) = \begin{cases} \frac{1}{2C_d}(d+2)(1-\|\mathbf{x}\|^2), & if\ \|\mathbf{x}\|<1 \\ 0, & otherwise \end{cases} \tag{3-22}$$

where d is the dimension of space, C_d is the volume of the unit d-Dimension sphere.

Letting $K(\mathbf{x}) = k(\|\mathbf{x}\|^2)$, we obtain

$$k(x) = \begin{cases} \frac{1}{2C_d}(d+2)(1-x), & if\ x<1 \\ 0, & otherwise \end{cases} \tag{3-23}$$

In image case, $d=2$, so $C_d = \pi$ and we obtain

$$k(x) = \begin{cases} \frac{1}{2\pi}(2+2)(1-x) = \frac{2}{\pi}(1-x), & if\ x<1 \\ 0, & otherwise \end{cases} \tag{3-24}$$

Letting $G(x) = k(x)$, we obtain

$$G'(x) = k'(x) = -\frac{2}{\pi} \tag{3-25}$$

which is a constant. The result is easy to compute and simpler, and this is the reason why we choose weighted function G as Epanechnikov kernel function. Finally, by substituting (3-25) into (3-21), we can get the second similarity-based mean-shift algorithm as follows.

$$\mathbf{y}_{new}=\frac{\sum_{j=1}^{N}\sum_{i=1}^{M}\mathbf{y}_{j}K_{P}K_{C}\delta[v_{j}-b(i)]}{\sum_{j=1}^{N}\sum_{i=1}^{M}K_{P}K_{C}\delta[v_{j}-b(i)]} \tag{3-26}$$

(3-26) interprets that the object tracking algorithm is an iterative procedure which moves from current position $\mathbf{y}_{old}$ to the new position $\mathbf{y}_{new}$.

3.4.5 Spatial-Color Mean-Shift Tracking Procedure

we have found the new spatial-color mean-shift tracking algorithms, single object tracking can be summarized as Figure 3.4-2 and Figure 3.4-3.

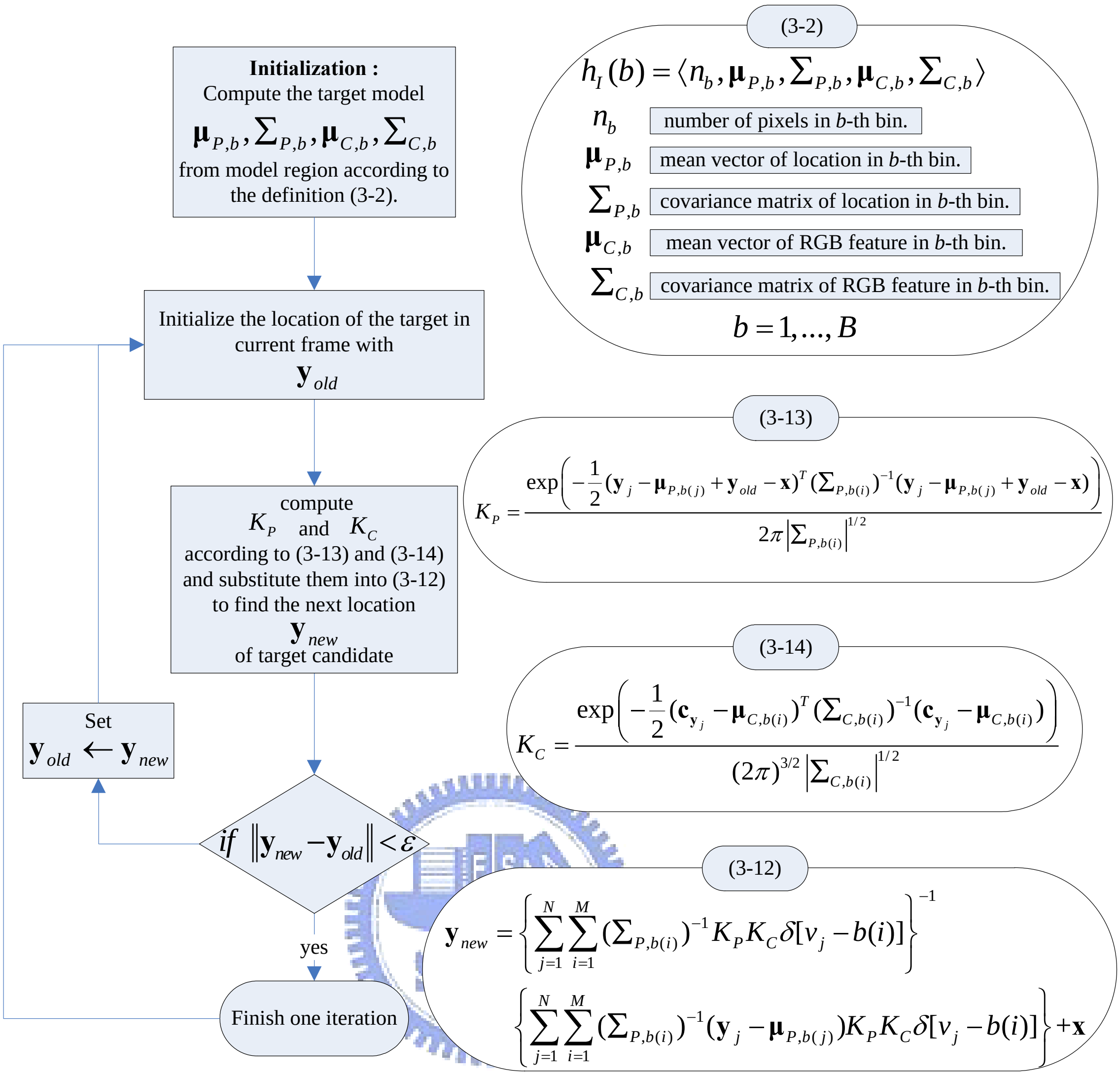

Initialization :
Compute the target model
$\boldsymbol{\mu}_{P,b}, \Sigma_{P,b}, \boldsymbol{\mu}_{C,b}, \Sigma_{C,b}$
from model region according to the definition (3-2).
Initialize the location of the target in current frame with
$\mathbf{y}_{old}$
compute
K_P and K_C
according to (3-13) and (3-14) and substitute them into (3-12) to find the next location
$\mathbf{y}_{new}$
of target candidate
Set
$\mathbf{y}_{old} \leftarrow \mathbf{y}_{new}$
$if \ \|\mathbf{y}_{new} - \mathbf{y}_{old}\| < \varepsilon$
yes
Finish one iteration
(3-2)
$h_I(b) = \langle n_b, \boldsymbol{\mu}_{P,b}, \Sigma_{P,b}, \boldsymbol{\mu}_{C,b}, \Sigma_{C,b} \rangle$
n_b number of pixels in *b*-th bin.
$\boldsymbol{\mu}_{P,b}$ mean vector of location in *b*-th bin.
$\Sigma_{P,b}$ covariance matrix of location in *b*-th bin.
$\boldsymbol{\mu}_{C,b}$ mean vector of RGB feature in *b*-th bin.
$\Sigma_{C,b}$ covariance matrix of RGB feature in *b*-th bin.
$b = 1, \ldots, B$
(3-13)
$K_P = \frac{\exp\left(-\frac{1}{2}(\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)} + \mathbf{y}_{old} - \mathbf{x})^T (\Sigma_{P,b(i)})^{-1} (\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)} + \mathbf{y}_{old} - \mathbf{x})\right)}{2\pi |\Sigma_{P,b(i)}|^{1/2}}$
(3-14)
$K_C = \frac{\exp\left(-\frac{1}{2}(\mathbf{c}_{\mathbf{y}_j} - \boldsymbol{\mu}_{C,b(i)})^T (\Sigma_{C,b(i)})^{-1} (\mathbf{c}_{\mathbf{y}_j} - \boldsymbol{\mu}_{C,b(i)})\right)}{(2\pi)^{3/2} |\Sigma_{C,b(i)}|^{1/2}}$
(3-12)
$\mathbf{y}_{new} = \left\{ \sum_{j=1}^{N} \sum_{i=1}^{M} (\Sigma_{P,b(i)})^{-1} K_P K_C \delta[v_j - b(i)] \right\}^{-1}$
$\left\{ \sum_{j=1}^{N} \sum_{i=1}^{M} (\Sigma_{P,b(i)})^{-1} (\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)}) K_P K_C \delta[v_j - b(i)] \right\} + \mathbf{x}$

Figure 3.4-2 : Spatial-color mean-shift tracking procedure of the first tracker.

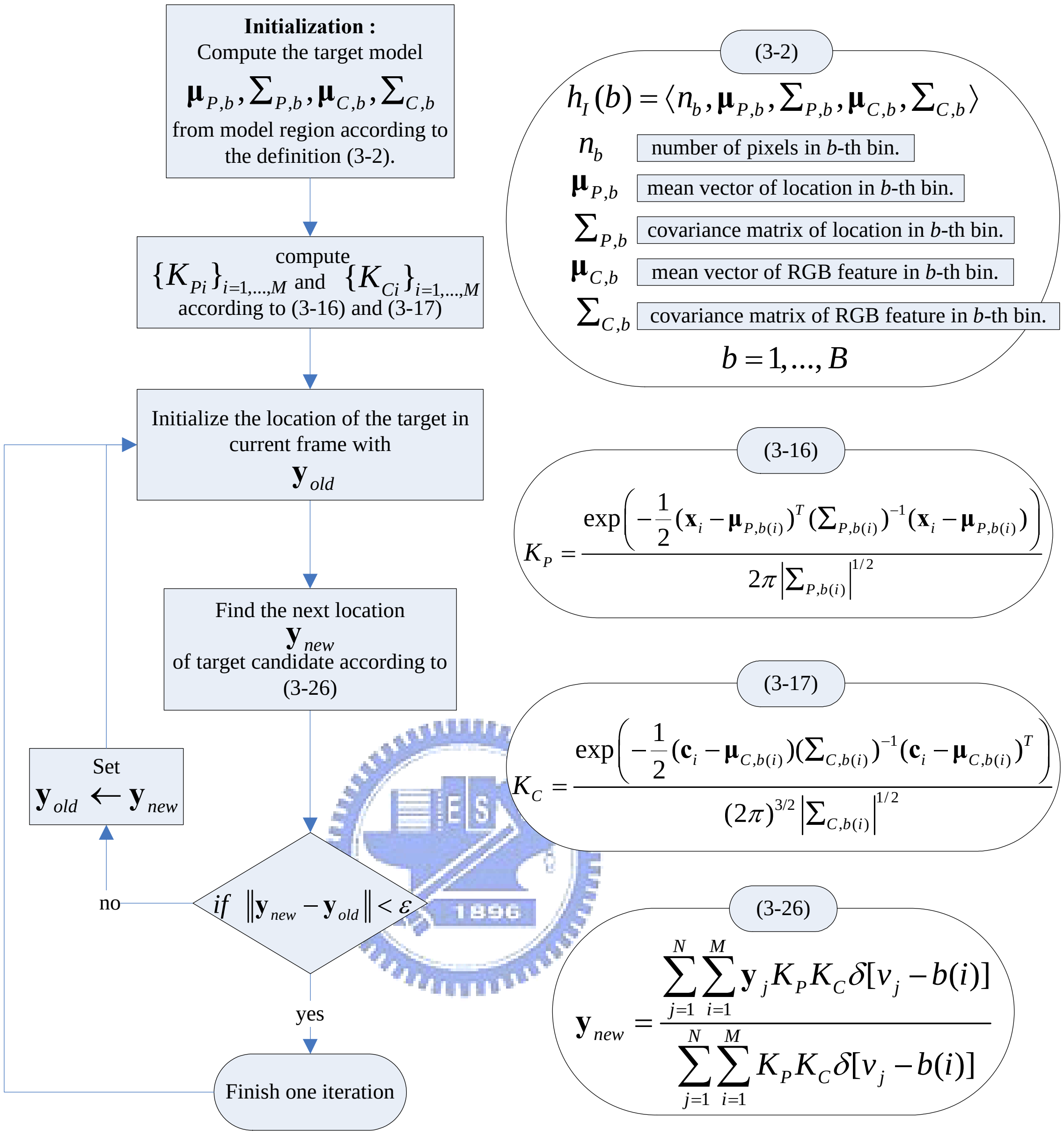

Initialization :
Compute the target model
$\boldsymbol{\mu}_{P,b}, \Sigma_{P,b}, \boldsymbol{\mu}_{C,b}, \Sigma_{C,b}$
from model region according to the definition (3-2).
compute
$\{K_{Pi}\}_{i=1,\ldots,M}$ and $\{K_{Ci}\}_{i=1,\ldots,M}$
according to (3-16) and (3-17)
Initialize the location of the target in current frame with
$\mathbf{y}_{old}$
Find the next location
$\mathbf{y}_{new}$
of target candidate according to (3-26)
$if \ \left\|\mathbf{y}_{new} - \mathbf{y}_{old}\right\| < \varepsilon$
no
Set
$\mathbf{y}_{old} \leftarrow \mathbf{y}_{new}$
yes
Finish one iteration
(3-2)
$h_I(b) = \langle n_b, \boldsymbol{\mu}_{P,b}, \Sigma_{P,b}, \boldsymbol{\mu}_{C,b}, \Sigma_{C,b} \rangle$
n_b number of pixels in b-th bin.
$\boldsymbol{\mu}_{P,b}$ mean vector of location in b-th bin.
$\Sigma_{P,b}$ covariance matrix of location in b-th bin.
$\boldsymbol{\mu}_{C,b}$ mean vector of RGB feature in b-th bin.
$\Sigma_{C,b}$ covariance matrix of RGB feature in b-th bin.
$b = 1, \ldots, B$
(3-16)
$K_P = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_{P,b(i)})^T (\Sigma_{P,b(i)})^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_{P,b(i)})\right)}{2\pi \left|\Sigma_{P,b(i)}\right|^{1/2}}$
(3-17)
$K_C = \frac{\exp\left(-\frac{1}{2}(\mathbf{c}_i - \boldsymbol{\mu}_{C,b(i)})(\Sigma_{C,b(i)})^{-1} (\mathbf{c}_i - \boldsymbol{\mu}_{C,b(i)})^T\right)}{(2\pi)^{3/2} \left|\Sigma_{C,b(i)}\right|^{1/2}}$
(3-26)
$\mathbf{y}_{new} = \frac{\sum_{j=1}^{N}\sum_{i=1}^{M} \mathbf{y}_j K_P K_C \delta[v_j - b(i)]}{\sum_{j=1}^{N}\sum_{i=1}^{M} K_P K_C \delta[v_j - b(i)]}$

Figure 3.4-3 : Spatial-color mean-shift tracking procedure of the second tracker.

3.5 Choice of the Color Feature Space

In 3.2.2, we choose color space (R,G,B) as our color feature, so $\boldsymbol{\mu}_{C,b}$ is the 3-dimension mean vector of values of (R,G,B) and $\Sigma_{C,b}$ is the covariance matrix of (R,G,B). The color space (R,G,B) is easily influenced by illumination that affects our tracking results greatly. So we take account of the normalized color space (r,g,b) which is formed independently from varying lighting levels. The normalized color space (r,g,b) is defined as

$$r=\frac{R}{(R+G+B)},\quad g=\frac{G}{(R+G+B)},\quad b=\frac{B}{(R+G+B)} \qquad (3\text{-}27)$$

The covariance matrix of the normalized color space (r,g,b) is near singular because the definition (3-27), so we choose (r,g) as the color feature space. Chapter 4 will show that the experiment results of (r,g) is more robust about the variation of illumination than that of (R,G,B).

3.6 Background-Weighted Information

In many tracking applications, background information is an important issue. Exactly representing the target model is a difficult subject, and the system is always confused by the foreground feature with the background feature because the foreground always contains the background information. The proposed tracking method is based on the similarity between the target and the candidate; therefore, how to represent the foreground model is very important. Further, the improper representation of the foreground may concern with the scale and orientation selection algorithm, and obtain inappropriate scale. In this chapter, we derive a simple weighted-background representation and add this approach to the spatial-color mean-shift trackers which we proposed before.

Let $N_{F,b}$ as the normalized histogram of the foreground of the *b*-th bin ($\sum_b N_{O,b}=1$), and $N_{O,b}$ as the normalized histogram of the background of the *b*-th bin ($\sum_b N_{F,b}=1$). The histogram of background is computed in the region around the foreground (target). We define weights as

$$W_b = \begin{cases} \frac{N_{F,b}}{N_{O,b}} \times \max(\frac{N_{F,1}}{N_{O,1}}, \frac{N_{F,2}}{N_{O,2}}, \ldots, \frac{N_{F,B}}{N_{O,B}}), & if \quad N_{O,b} \neq 0 \\ 1, & if \ \ N_{F,b} \neq 0 \ \ and \ \ N_{O,b} = 0 \\ 0, & otherwise \end{cases} \tag{3-28}$$

The weights transformation diminishes the effect of features which contribute more to the background than to the foreground.

Now we add the weighted-background information to the mean-shift trackers developed in 3.4 and re-derive the revised weighted spatial-color mean-shift as follows. We add the weights to (3-7) and (3-18), and obtain

$$J(I_x, I_y) = J(\mathbf{y}) = \frac{1}{N}\sum_{j=1}^{N} W_{b(j)}\, p(\mathbf{y}_j, v_j) \tag{3-29}$$

$$J(I_x, I_y) = J(\mathbf{y}) = \frac{1}{N}\sum_{j=1}^{N} W_{b(j)}\, \mathrm{G}(\mathbf{y}\text{-}\mathbf{y}_{\mathrm{j}})\, p(v_j) \tag{3-30}$$

By similar derivation in 3.4, we can obtain the final spatial-color mean-shift tracker functions which contain the weighted-background information from (3-29) and (3-30).

$$\mathbf{y}_{new} = \left\{\sum_{j=1}^{N}\sum_{i=1}^{M}(\Sigma_{P,b(i)})^{-1} W_{b(j)} K_P K_C \delta[v_j - b(i)]\right\}^{-1} \left\{\sum_{j=1}^{N}\sum_{i=1}^{M}(\Sigma_{P,b(i)})^{-1}(\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)}) W_{b(j)} K_P K_C \delta[v_j - b(i)]\right\} + \mathbf{x} \tag{3-31}$$

$$\mathbf{y}_{new} = \frac{\sum_{j=1}^{N}\sum_{i=1}^{M} \mathbf{y}_{\mathrm{j}} W_{b(j)} K_P K_C \delta[v_j - b(i)]}{\sum_{\mathrm{j}=1}^{\mathrm{N}}\sum_{i=1}^{M} W_{b(j)} K_P K_C \delta[v_j - b(i)]} \tag{3-32}$$

3.7 Update of Scale and Orientation

In computer vision and image processing, the object always changes its scale when it is away from the camera or toward the camera. In the situation of zoom in and zoom out of camera, the size of object body is also different between image frames. As shown in Figure 3.7-1, if the object size is smaller than tracking window, it will contain many background pixels as well as the foreground object pixels. This problem causes wrong tracking result with noisy background pixels when a histogram computed within the window is compared to a model histogram describing the appearance of the foreground object. If the object size is larger than tracking window, it will cause the tracker to become more easily distracted by background clutter.

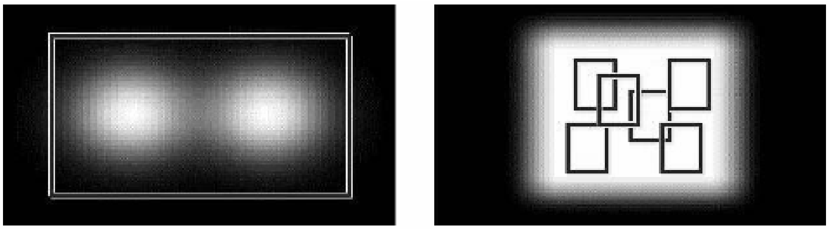


Figure 3.7-1 : Illustration of scale problem. (The figure is obtained from [4])

The orientation problem is similar with the scale problem. A fixed window may not contain all regions of the tracked object if it appears the variation of orientation and results in the failure of tracking. In the later section, we will use part of principal component analysis method to solve these two problems.

3.7.1 Introduction of Principal Component Analysis

Principal component analysis (PCA) is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate which is the first principal component, the second greatest variance on the second coordinate, and so on.

Assume the sample covariance matrix of standardized matrix $\mathbf{X}$ ($\mathbf{X} \in R^P$) to be

$$\mathbf{R} = \frac{1}{N-1}\mathbf{X}^T\mathbf{X} \tag{3-33}$$

The principal component analysis problem can be derived to be as the eigen-equation problem [13].

$$\mathbf{Rv} = \lambda \mathbf{v} \qquad (3\text{-}34)$$

By solving this eigen-equation, we can obtain eigen-values $\{\lambda_i\}_{i=1,\ldots,P}$ and eigen-vectors $\{\mathbf{v}_i\}_{i=1,\ldots,P}$, respectively. The largest eigen-vector is the largest principal component which is the direction that makes variance of the projected data to be maximum, and the smallest principal component is the direction that makes that variance minimum as shown in Figure 3.7-2. In *2*-dimension image data case, by this method we can obtain two eigen-values and two eigen-vectors which represent the orthogonal axes of data, respectively.

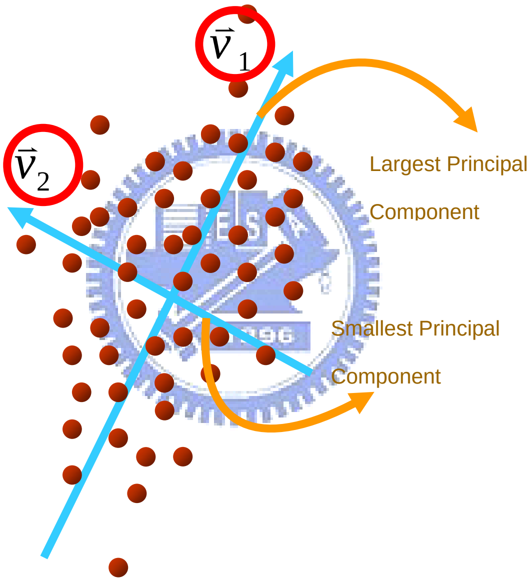


Figure 3.7-2 : Illustration of principal component analysis.

3.7.2 Orientation Selection by Principal Component Analysis

We can get the orientation of the total sample data by the concept of PCA method by previous section. Because we have computed some information about the image data location, we can use these data to get the total covariance matrix of total data for reducing the computation. In this section, we want to derive the covariance matrix of total image data from the elements which we defined in 3.2.2. Above all, we review the definition of some elements of model which we defined in (3-2) as follows. $\boldsymbol{\mu}_{P,b}$ is mean vector of locations of pixels which belong to the *b*-th bin, $\Sigma_{P,b}$ is covariance matrix of location of pixels of the *b*-th bin, and *B* is the bin number which we classified.

$$\boldsymbol{\mu}_{P,b} = \frac{1}{N_b}\sum_{i\in b}\mathbf{x}_i \tag{3-35}$$

$$\Sigma_{P,b} = \sum_{i\in b}(\mathbf{x}_i - \boldsymbol{\mu}_{P,b})(\mathbf{x}_i - \boldsymbol{\mu}_{P,b})^T \tag{3-36}$$

And we define several new elements as follows. $\boldsymbol{\mu}_T$ is the total mean vector of the locations of all pixels in the target, Σ_W is the within-class covariance matrix of the *B* bins, Σ_B is the between-class covariance matrix of the *B* bins, and Σ_T is the total covariance matrix of locations of all data.

$$\boldsymbol{\mu}_{\mathrm{T}} = \frac{1}{N}\sum_{b}N_b\boldsymbol{\mu}_{P,b} \tag{3-37}$$

$$\Sigma_W = \sum_{b}\sum_{i\in b}(\mathbf{x}_i - \boldsymbol{\mu}_{P,b})(\mathbf{x}_i - \boldsymbol{\mu}_{P,b})^T \tag{3-38}$$

$$\Sigma_B = \sum_{b}N_b(\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\mathrm{T}})(\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\mathrm{T}})^T \tag{3-39}$$

$$\Sigma_T = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu}_T)(\mathbf{x}_i - \boldsymbol{\mu}_T)^T \tag{3-40}$$

Decomposing Σ_T, we get some derivation results.

$$\begin{aligned}
\Sigma_T &= \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu}_T)(\mathbf{x}_i - \boldsymbol{\mu}_T)^T \\
&= \frac{1}{N-1}\sum_{b}\sum_{i\in b}(\mathbf{x}_i - \boldsymbol{\mu}_{P,b} + \boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_T)(\mathbf{x}_i - \boldsymbol{\mu}_{P,b} + \boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_T)^T \\
&= \frac{1}{N-1}\sum_{b}\sum_{i\in b}(\mathbf{x}_i - \boldsymbol{\mu}_{P,b})(\mathbf{x}_i - \boldsymbol{\mu}_{P,b})^T + \frac{1}{N-1}\sum_{b}\sum_{i\in b}(\mathbf{x}_i - \boldsymbol{\mu}_{P,b})(\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_T)^T \\
&\quad + \frac{1}{N-1}\sum_{b}\sum_{i\in b}(\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_T)(\mathbf{x}_i - \boldsymbol{\mu}_{P,b})^T + \frac{1}{N-1}\sum_{b}\sum_{i\in b}(\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_T)(\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_T)^T
\end{aligned}$$

Because

$$\begin{aligned}\sum_{b}\sum_{i\in b}(\boldsymbol{\mu}_{P,b}-\boldsymbol{\mu}_{T})(\mathbf{x}_i-\boldsymbol{\mu}_{P,b})^T &= \sum_{b}(\boldsymbol{\mu}_{P,b}-\boldsymbol{\mu}_{T})\sum_{i\in b}(\mathbf{x}_i-\boldsymbol{\mu}_{P,b})^T \\ &= \sum_{b}(\boldsymbol{\mu}_{P,b}-\boldsymbol{\mu}_{T})(N_b\boldsymbol{\mu}_{P,b}-N_b\boldsymbol{\mu}_{P,b})^T \\ &= \mathbf{0}\end{aligned}$$

, we can obtain

$$\begin{aligned}\Sigma_T &= \frac{1}{N-1}\sum_{b}\sum_{i\in b}(\mathbf{x}_i-\boldsymbol{\mu}_{P,b})(\mathbf{x}_i-\boldsymbol{\mu}_{P,b})^T+\frac{1}{N-1}\sum_{b}\sum_{i\in b}(\boldsymbol{\mu}_{P,b}-\boldsymbol{\mu}_{T})(\boldsymbol{\mu}_{P,b}-\boldsymbol{\mu}_{T})^T \\ &= \Sigma_W+\Sigma_B\end{aligned} \tag{3-41}$$

Therefore, we can get the total covariance matrix of all image data from the elements of model which we have defined, and we substitute Σ_T into **R** in (3-34) as

$$\Sigma_T\,\mathbf{v}=\lambda\mathbf{v} \tag{3-42}$$

By solving this eigen-equation, we can get two eigen-vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ with respect to the largest principal component and smallest principal component, respectively. If we use ellipse as the region of the target, the largest principal component represents the long axis and the smallest principal component represents the short axis as shown in Figure 3.7-2.

3.7.3 Adding Weighted-Background Information

In 3.6, we have discussed the influence of background information about the scale and orientation selection. For improving robustness and accurate of scale and orientation selection, we add the weighted background information to (3-41), and obtained the total covariance matrix with weighted-background information.

$$
\begin{aligned}
\Sigma_T^{'} &= \frac{1}{\sum_{i=1}^{N} W_{b(i)}} \sum_{i=1}^{N} W_{b(i)} (\mathbf{x}_i - \boldsymbol{\mu}_T)(\mathbf{x}_i - \boldsymbol{\mu}_T)^T \\
&= \frac{1}{\sum_{i=1}^{N} W_{b(i)}} \sum_{b} \sum_{i \in b} W_{b(i)} (\mathbf{x}_i - \boldsymbol{\mu}_{P,b})(\mathbf{x}_i - \boldsymbol{\mu}_{P,b})^T + \frac{1}{\sum_{i=1}^{N} W_{b(i)}} \sum_{b} \sum_{i \in b} W_{b(i)} (\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_T)(\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_T)^T \\
&= \Sigma_W^{'} + \Sigma_B^{'}
\end{aligned}
$$

(3-43)

We change (3-42) as $\Sigma_T^{'} \mathbf{v} = \lambda \mathbf{v}$ and solve eigen-vectors again, and we obtain the more accurate direction of long axis and short axis.

3.7.4 Scale Selection

By the total covariance matrix, we can get orientation of the distribution of target image data by the axes of ellipse, but we can not obtain the length of axes. Now we want to know the relation between total covariance matrix and two axes.

We consider a uniform ellipse distribution, and assume probability of this ellipse is $\frac{1}{\pi ab}$. Now we compute the variances along two axes.

$$
\begin{aligned}
\sigma_{xx} &= \iint_R (x-0)^2 p(x) dx \\
&= \iint_{R^{'}} a^2 u^2 \frac{1}{\pi ab} du dv \\
&= \frac{1}{\pi} \int_0^{2\pi} \int_0^1 r^3 a^2 \cos^2 \theta dr d\theta \\
&= \frac{a^2}{4}
\end{aligned}
\qquad (3\text{-}44)
$$

$$
\begin{aligned}
\sigma_{yy} &= \iint_R (y-0)^2 p(y) dy \\
&= \iint_{R^{'}} b^2 v^2 \frac{1}{\pi ab} du dv \\
&= \frac{1}{\pi} \int_0^{2\pi} \int_0^1 r^3 b^2 \sin^2 \theta dr d\theta \\
&= \frac{b^2}{4}
\end{aligned}
\qquad (3\text{-}45)
$$

where σ_{xx} and σ_{yy} are elements of total covariance matrix $\Sigma_T^{'} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix}$, and we can obtain

$$\begin{aligned} a &= 2\sqrt{\sigma_{xx}} = 2\sigma_x \\ b &= 2\sqrt{\sigma_{yy}} = 2\sigma_y \end{aligned} \tag{3-46}$$

The values of two axes are about double of variances along the long axis and short axis.

3.8 Summary

In the previous section, we obtain the spatial-color mean-shift trackers, now we summarize these concepts as Figure 3.8-1 and Figure 3.8-2.

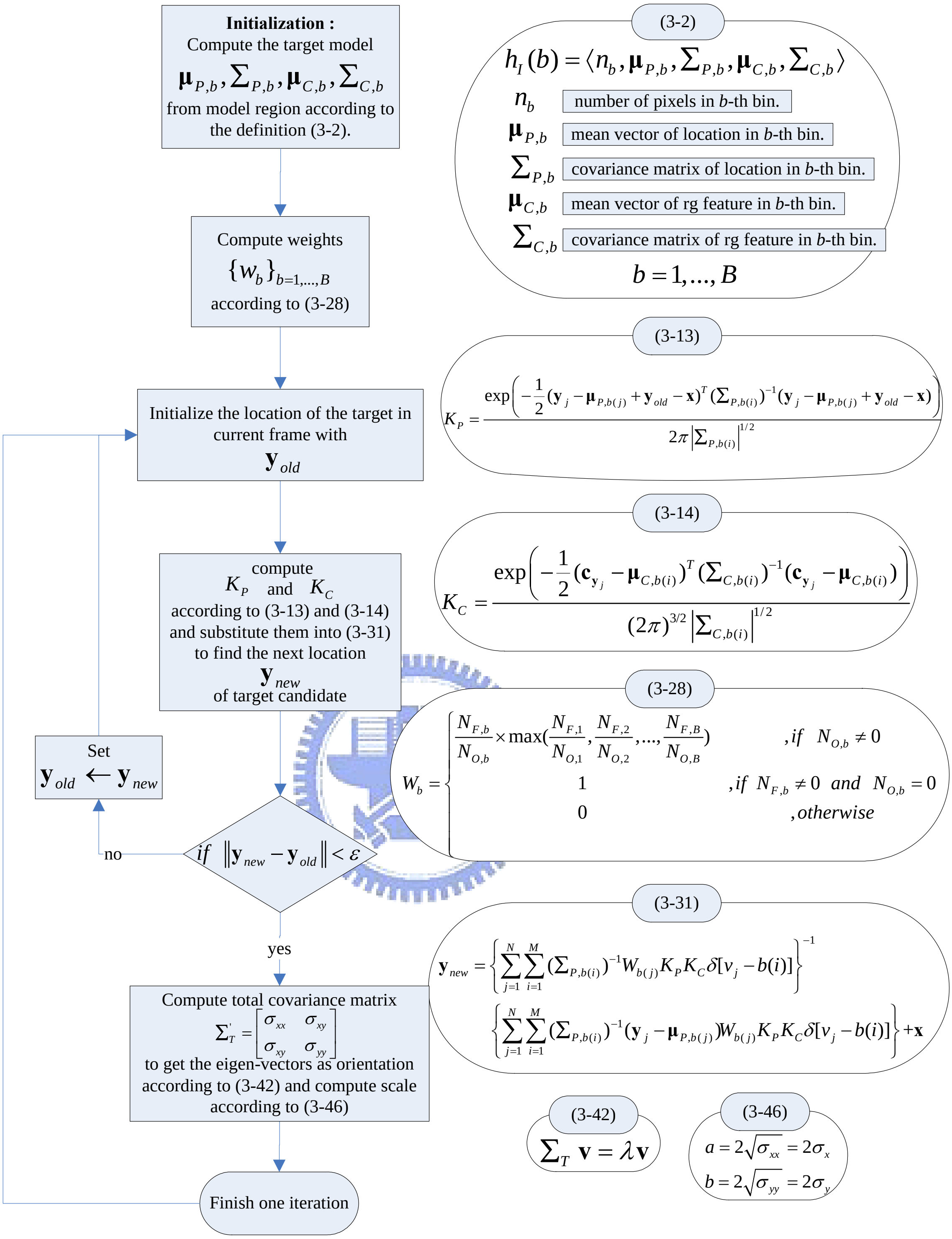

Initialization :
Compute the target model
$\boldsymbol{\mu}_{P,b}, \Sigma_{P,b}, \boldsymbol{\mu}_{C,b}, \Sigma_{C,b}$
from model region according to the definition (3-2).
Compute weights
$\{w_b\}_{b=1,...,B}$
according to (3-28)
Initialize the location of the target in current frame with
$\mathbf{y}_{old}$
compute
K_P and K_C
according to (3-13) and (3-14) and substitute them into (3-31) to find the next location
$\mathbf{y}_{new}$
of target candidate
Set
$\mathbf{y}_{old} \leftarrow \mathbf{y}_{new}$
no
$if \ \|\mathbf{y}_{new} - \mathbf{y}_{old}\| < \varepsilon$
yes
Compute total covariance matrix
$\Sigma_T' = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix}$
to get the eigen-vectors as orientation according to (3-42) and compute scale according to (3-46)
Finish one iteration
(3-2)
$h_I(b) = \langle n_b, \boldsymbol{\mu}_{P,b}, \Sigma_{P,b}, \boldsymbol{\mu}_{C,b}, \Sigma_{C,b} \rangle$
n_b number of pixels in b-th bin.
$\boldsymbol{\mu}_{P,b}$ mean vector of location in b-th bin.
$\Sigma_{P,b}$ covariance matrix of location in b-th bin.
$\boldsymbol{\mu}_{C,b}$ mean vector of rg feature in b-th bin.
$\Sigma_{C,b}$ covariance matrix of rg feature in b-th bin.
$b = 1, ..., B$
(3-13)
$K_P = \frac{\exp\left(-\frac{1}{2}(\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)} + \mathbf{y}_{old} - \mathbf{x})^T (\Sigma_{P,b(i)})^{-1} (\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)} + \mathbf{y}_{old} - \mathbf{x})\right)}{2\pi |\Sigma_{P,b(i)}|^{1/2}}$
(3-14)
$K_C = \frac{\exp\left(-\frac{1}{2}(\mathbf{c}_{\mathbf{y}_j} - \boldsymbol{\mu}_{C,b(i)})^T (\Sigma_{C,b(i)})^{-1} (\mathbf{c}_{\mathbf{y}_j} - \boldsymbol{\mu}_{C,b(i)})\right)}{(2\pi)^{3/2} |\Sigma_{C,b(i)}|^{1/2}}$
(3-28)
$W_b = \begin{cases} \frac{N_{F,b}}{N_{O,b}} \times \max(\frac{N_{F,1}}{N_{O,1}}, \frac{N_{F,2}}{N_{O,2}}, ..., \frac{N_{F,B}}{N_{O,B}}) & ,if \ N_{O,b} \neq 0 \\ 1 & ,if \ N_{F,b} \neq 0 \ and \ N_{O,b} = 0 \\ 0 & ,otherwise \end{cases}$
(3-31)
$\mathbf{y}_{new} = \left\{ \sum_{j=1}^{N} \sum_{i=1}^{M} (\Sigma_{P,b(i)})^{-1} W_{b(j)} K_P K_C \delta[v_j - b(i)] \right\}^{-1} \left\{ \sum_{j=1}^{N} \sum_{i=1}^{M} (\Sigma_{P,b(i)})^{-1} (\mathbf{y}_j - \boldsymbol{\mu}_{P,b(j)}) W_{b(j)} K_P K_C \delta[v_j - b(i)] \right\} + \mathbf{x}$
(3-42)
$\Sigma_T \mathbf{v} = \lambda \mathbf{v}$
(3-46)
$a = 2\sqrt{\sigma_{xx}} = 2\sigma_x$
$b = 2\sqrt{\sigma_{yy}} = 2\sigma_y$

Figure 3.8-1 : Complete spatial-color mean-shift tracking procedure of the first tracker.

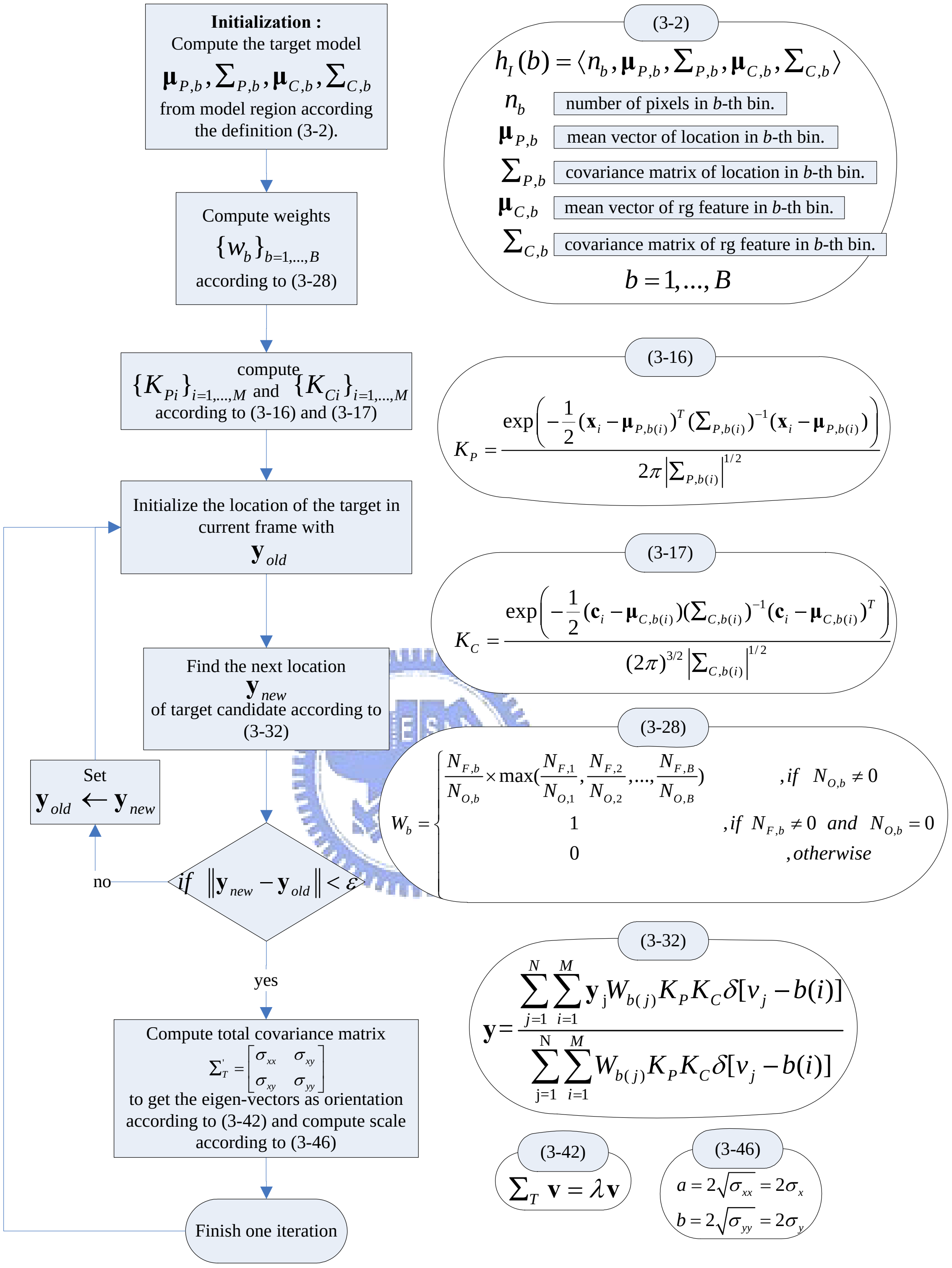

Initialization :
Compute the target model
$\boldsymbol{\mu}_{P,b}, \Sigma_{P,b}, \boldsymbol{\mu}_{C,b}, \Sigma_{C,b}$
from model region according
the definition (3-2).
Compute weights
$\{w_b\}_{b=1,\dots,B}$
according to (3-28)
compute
$\{K_{Pi}\}_{i=1,\dots,M}$ and $\{K_{Ci}\}_{i=1,\dots,M}$
according to (3-16) and (3-17)
Initialize the location of the target in current frame with
$\mathbf{y}_{old}$
Find the next location
$\mathbf{y}_{new}$
of target candidate according to (3-32)
Set
$\mathbf{y}_{old} \leftarrow \mathbf{y}_{new}$
$if \ \left\| \mathbf{y}_{new} - \mathbf{y}_{old} \right\| < \varepsilon$
no
yes
Compute total covariance matrix
$\Sigma_T = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix}$
to get the eigen-vectors as orientation according to (3-42) and compute scale according to (3-46)
Finish one iteration
(3-2)
$h_I(b) = \langle n_b, \boldsymbol{\mu}_{P,b}, \Sigma_{P,b}, \boldsymbol{\mu}_{C,b}, \Sigma_{C,b} \rangle$
n_b number of pixels in *b*-th bin.
$\boldsymbol{\mu}_{P,b}$ mean vector of location in *b*-th bin.
$\Sigma_{P,b}$ covariance matrix of location in *b*-th bin.
$\boldsymbol{\mu}_{C,b}$ mean vector of rg feature in *b*-th bin.
$\Sigma_{C,b}$ covariance matrix of rg feature in *b*-th bin.
$b = 1, \dots, B$
(3-16)
$K_P = \dfrac{\exp\left(-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_{P,b(i)})^T (\Sigma_{P,b(i)})^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_{P,b(i)})\right)}{2\pi \left|\Sigma_{P,b(i)}\right|^{1/2}}$
(3-17)
$K_C = \dfrac{\exp\left(-\frac{1}{2}(\mathbf{c}_i - \boldsymbol{\mu}_{C,b(i)})(\Sigma_{C,b(i)})^{-1} (\mathbf{c}_i - \boldsymbol{\mu}_{C,b(i)})^T\right)}{(2\pi)^{3/2} \left|\Sigma_{C,b(i)}\right|^{1/2}}$
(3-28)
$W_b = \begin{cases} \dfrac{N_{F,b}}{N_{O,b}} \times \max(\dfrac{N_{F,1}}{N_{O,1}}, \dfrac{N_{F,2}}{N_{O,2}}, \dots, \dfrac{N_{F,B}}{N_{O,B}}) & ,if \ \ N_{O,b} \neq 0 \\ 1 & ,if \ \ N_{F,b} \neq 0 \ \ and \ \ N_{O,b} = 0 \\ 0 & ,otherwise \end{cases}$
(3-32)
$\mathbf{y} = \dfrac{\sum_{j=1}^{N} \sum_{i=1}^{M} \mathbf{y}_j W_{b(j)} K_P K_C \delta[v_j - b(i)]}{\sum_{j=1}^{N} \sum_{i=1}^{M} W_{b(j)} K_P K_C \delta[v_j - b(i)]}$
(3-42)
$\Sigma_T \ \mathbf{v} = \lambda \mathbf{v}$
(3-46)
$a = 2\sqrt{\sigma_{xx}} = 2\sigma_x$
$b = 2\sqrt{\sigma_{yy}} = 2\sigma_y$

Figure 3.8-2 : Complete spatial-color mean-shift tracking procedure of the second tracker.

Chapter 4.　Experiment Results

4.1　Experiment Illustration

The proposed spatial-color mean-shift tracking algorithms have been implemented in C and tested on a 2.8GHz Pentium 4 PC with 1GB memory. We divide the color histograms into 512 bins, i.e. the *B* of (3-2) is equal to 512.　In the first part, we show our experiment results with the steps of what we developed our final trackers in chapter 3 in order, and present the tracking results with single scale experiment. We use the face sequence for face tracking, the cup sequence with complex appearance in complex background, and the walking girl sequence which is obtained from [14] with partial occlusions. In the second part, we present the experiment results with the boy walking sequence and surveillance sequence. The first sequence is the person away from the camera and toward the camera with huge variation of scale. The second sequence which is obtained from the CAVIAR database [15] illustrates the problem of huge deformation. The image size of face sequence, cup sequence, walking girl sequence, and walking boy sequence are 320x240, and the image size of surveillance sequence is 352x288. The tracking window sizes of face sequence, cup sequence, walking girl sequence are 59x82, 50x65, and 27x98, respectively.

We define (3-12) and the extension part as tracker 1, and (3-26) and the extension part as tracker 2. In the later section, we will compare the proposed tracker 1 and tracker 2 with the traditional mean-shift tracker, i.e. (2-16), and the general scale adaptation method with plus or minus 10 percent [1].

About the experiment results, we show part of the real tracking sequence, the distance error figure, and iteration num figure. We define the correct location of the object every 10 frames by hand in advance, and use these data to analysis the tracking results. We discard the distance error which is larger than 50 pixels that shows the tracker loses the target. The iteration number is the frequency of tracker finding the target in that frame in the iterative procedure. Finally, the computing time of the proposed trackers will be discussed.

4.2 Spatial-Color Mean-Shift Trackers with RGB Feature

In this section, we present the experiment results of trackers with RGB color feature that we proposed in 3.4, and we define (3-12) as tracker 1 and (3-26) as tracker 2.

4.2.1 Face Sequence

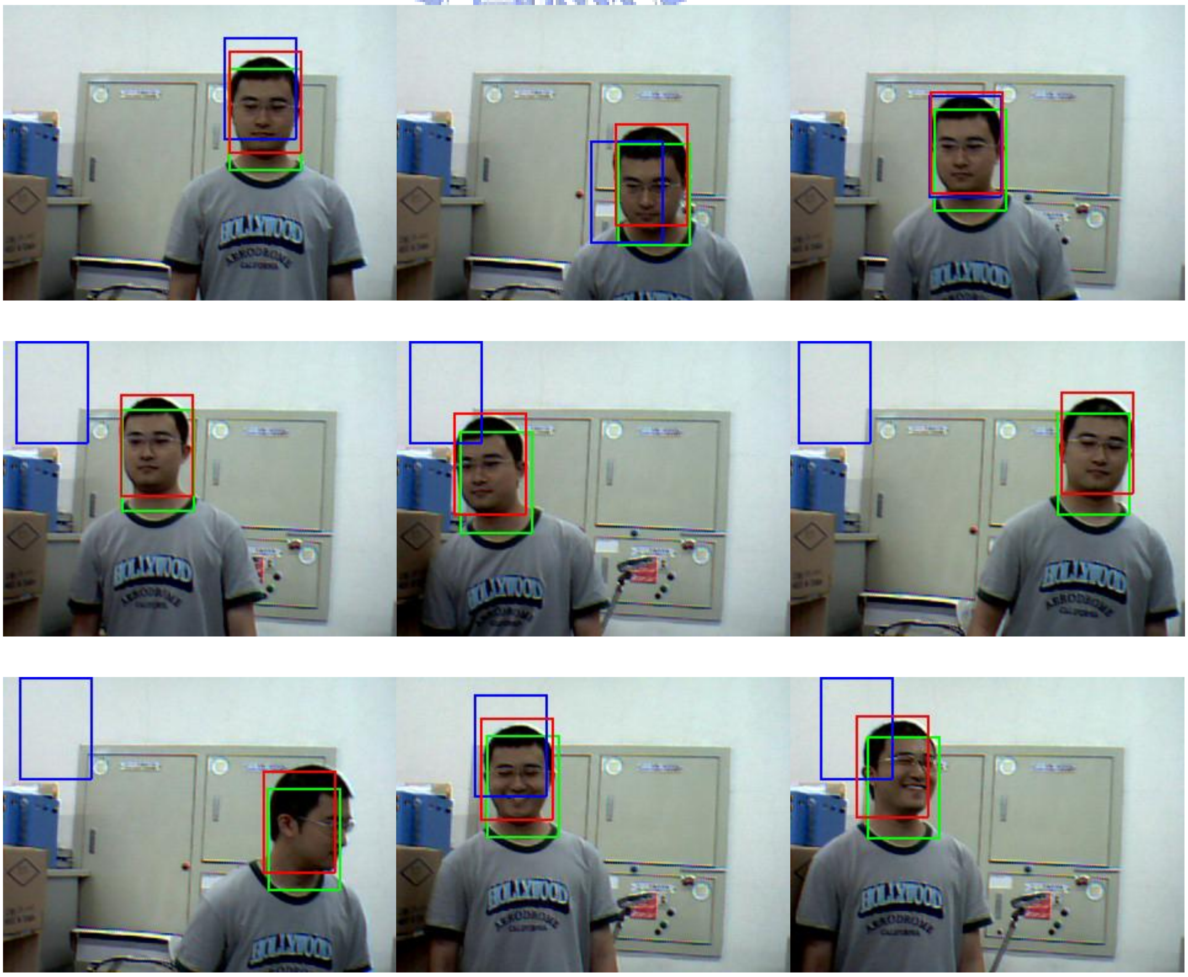


Figure 4.2-1 : Face tracking results of spatial-color mean-shift trackers proposed in 3.4. Shown are frames 33, 93, 117, 126, 183, 256, 271, 455, 766. (red: tracker 1, blue: tracker 2, green: traditional mean-shift tracker)

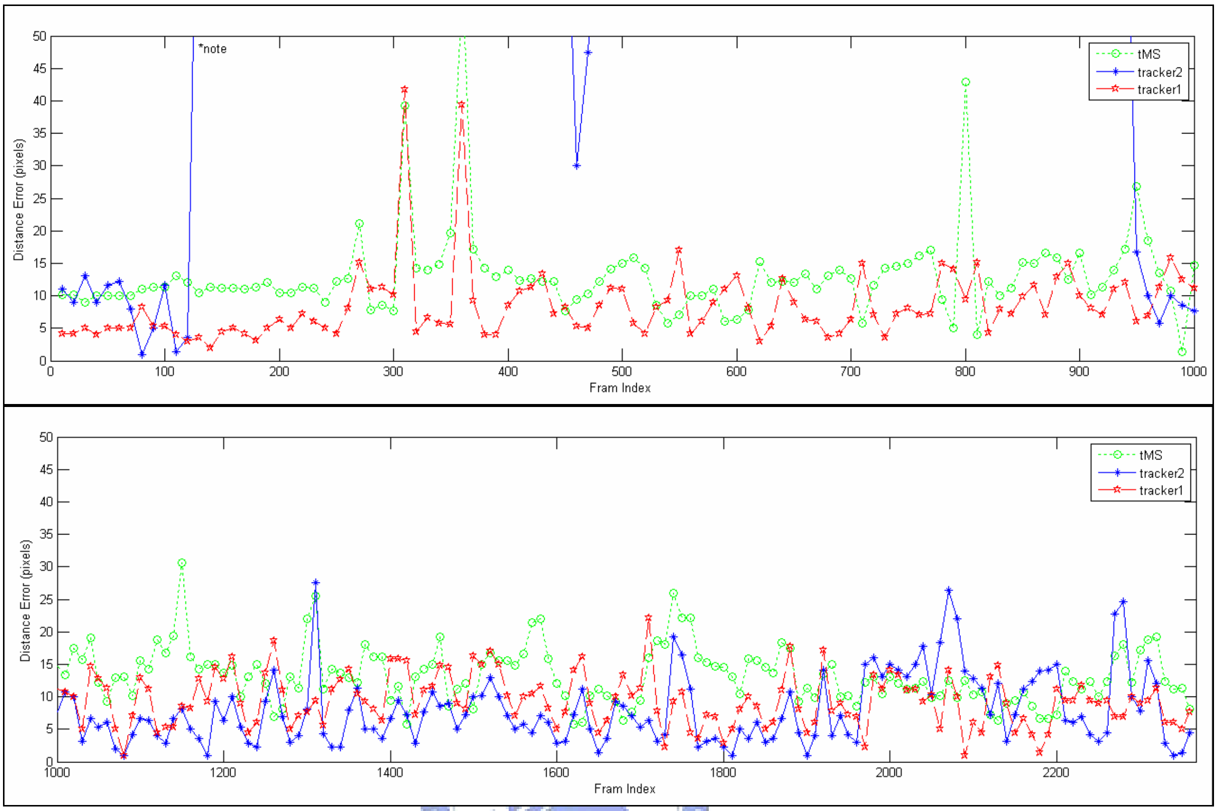


Figure 4.2-2 : Distance error of face sequence of spatial-color mean-shift tracker1 and tracker 2 proposed in 3.4 that is compared with traditional mean-shift tracker. (*note: we only consider the distance error which is smaller than 50 pixels)

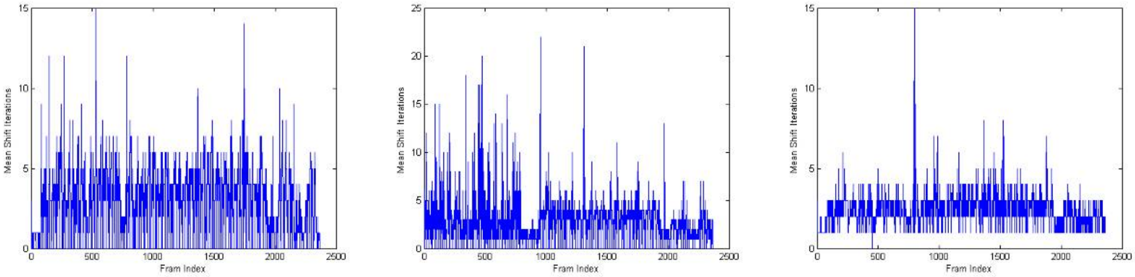


Figure 4.2-3 : Iteration number of face tracking sequence. (left: tracker1, middle: tracker2, right: traditional mean-shift tracker)

At about 120th frame, tracker 1 loses the face and captures the target again at about 950th frame. In the situation of face being captured of three trackers, the distance errors of tracker 1 and tracker 2 are always smaller than those of traditional mean-shift tracker. The average of iteration number of traditional mean-shift tracker is smaller than the other trackers. Up to now, the tracker 2 which we developed is not robust and more unstable than the traditional mean-shift tracker, but the tracker 1 is better about accurately tracking.

4.2.2 Cup Sequence

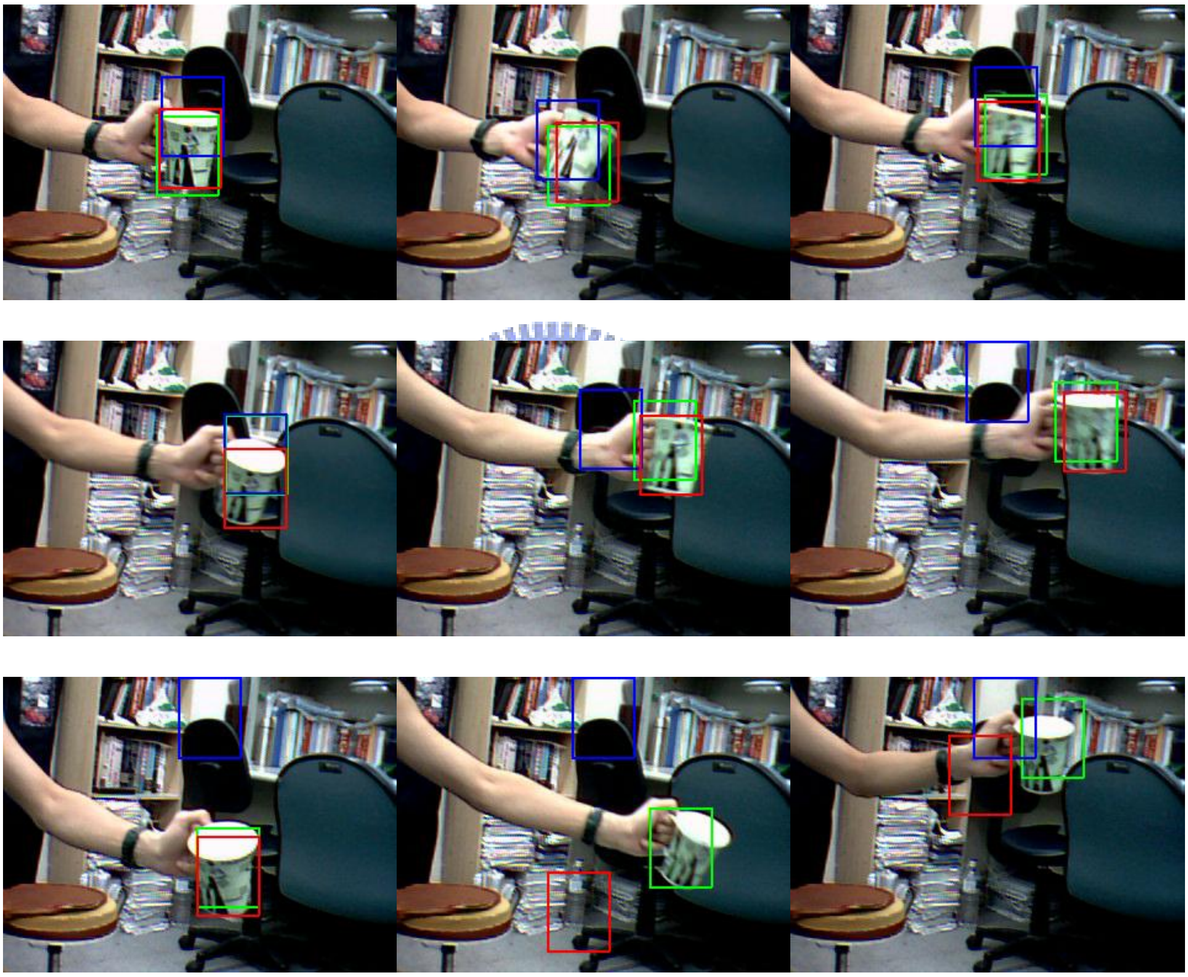


Figure 4.2-4 : Cup tracking results of spatial-color mean-shift trackers proposed in 3.4. Shown are frames 4, 45, 63, 69, 81, 105, 166, 243, 364. (red: tracker 1, blue: tracker 2, green: traditional mean-shift tracker)

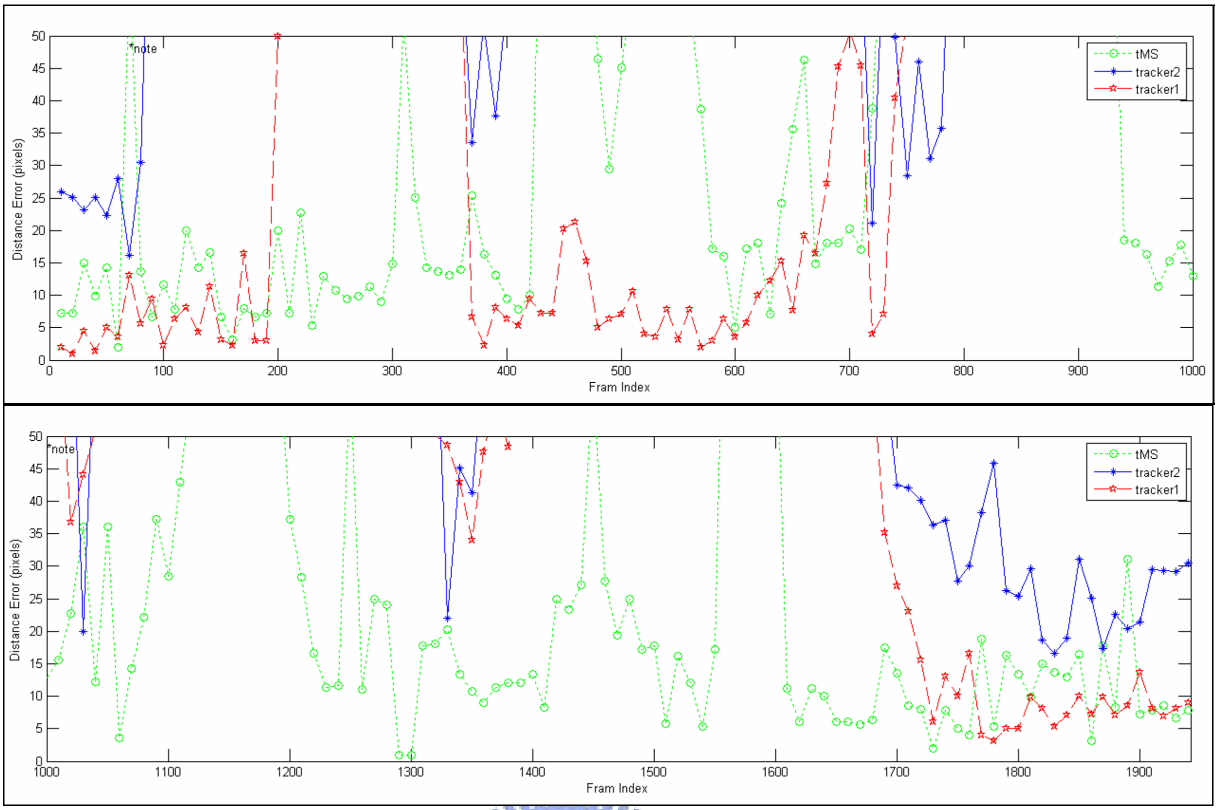


Figure 4.2-5 : Distance error of cup sequence of spatial-color mean-shift tracker1 and tracker 2 proposed in 3.4 that is compared with traditional mean-shift tracker. (*note: we only consider the distance error which is smaller than 50 pixels)

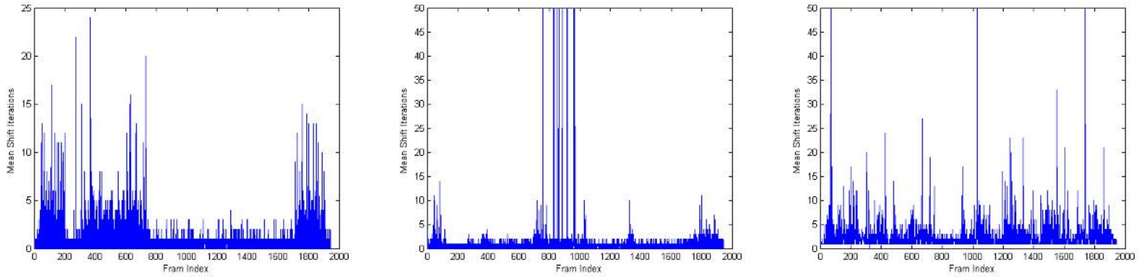


Figure 4.2-6 : Iteration number of cup sequence. (left: tracker1, middle: tracker2, right: traditional mean-shift tracker)

At most frames, the tracker 1 and tracker 2 lose the target, and the mean-shift tracker has weakly capturing. Because the background of this scene is very complex and the appearance of cup which we want to track is also complex, the trackers easily track the background object. The tracker 1 and tracker 2 contain the spatial information, so the trackers easily capture the background region which involves the similar spatial information when the cup is swayed. The traditional mean-shift tracker only contains the color distribution information, so it is easily affected by the complex background information and can not accurately track the target.

4.2.3 Walking Girl Sequence

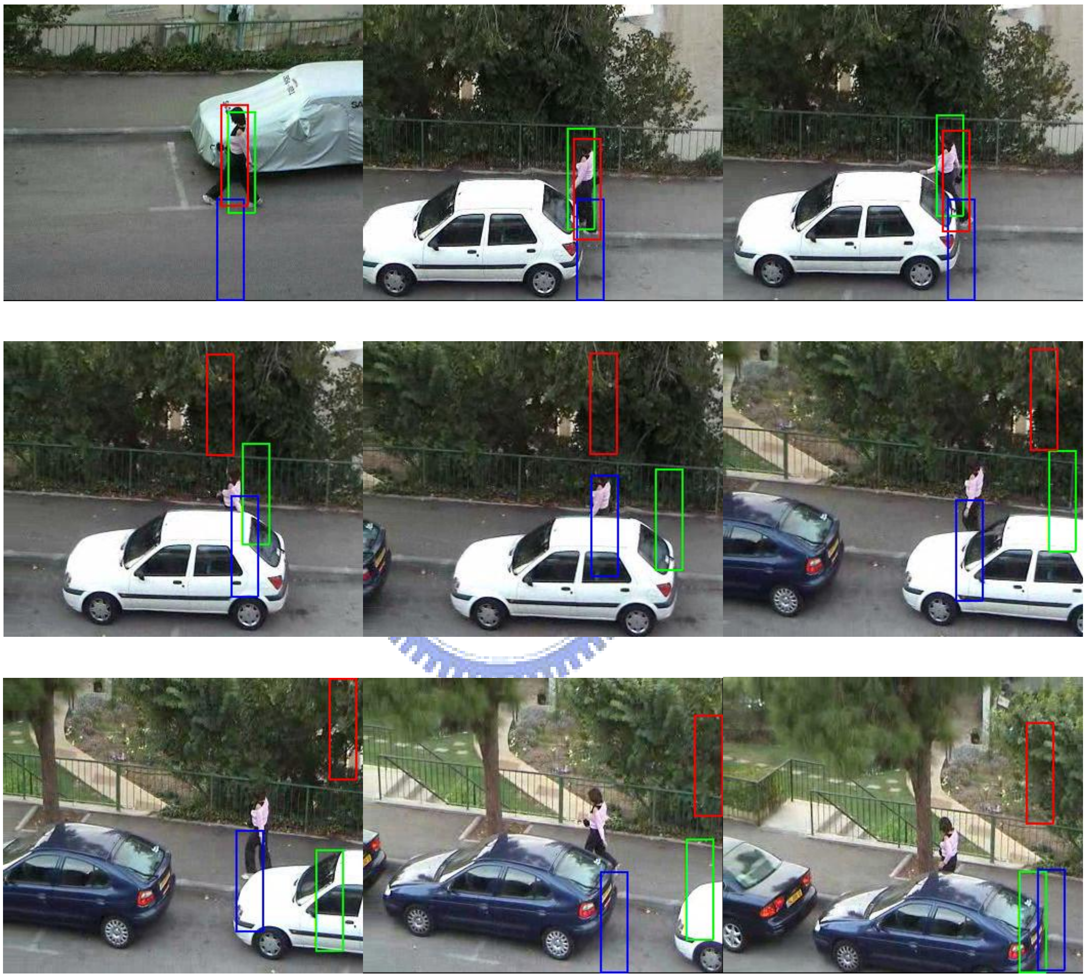


Figure 4.2-7 : Walking girl tracking results of spatial-color mean-shift trackers proposed in 3.4. Shown are frames 28, 106, 111, 124, 130, 153, 166, 196, .220. (red: tracker 1, blue: tracker 2, green: traditional mean-shift tracker)

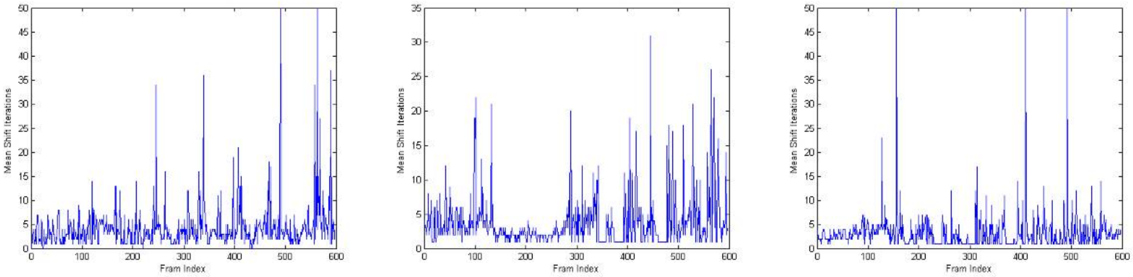


Figure 4.2-8 : Iteration numbers of walking girl sequence. (left: tracker1, middle: tracker2, right: traditional mean-shift tracker)

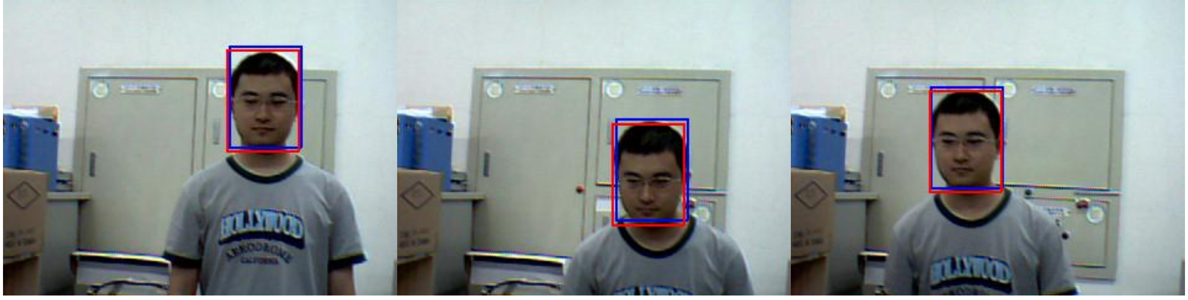
The walking girl sequence contains the problem of variation of illumination and partial occlusion. The variation of illumination from darker to bright and all trackers are not robust with this situation. At 111th frame, part of girl has be covered by the car and the tracker 1 and traditional mean-shift tracker still track the girl, but the tracker 2 loses her. The trackers are not better enough.

4.3 Spatial-Color Mean-Shift Trackers with Normalized Feature

In order to reduce the influence of the slight variation of illumination, the normalized feature rg is used in 3.5. Similar with 4.2, the tracker 1 is defined as (3-12) with rg feature and the tracker 2 is defined as (3-26) with rg feature.

4.3.1 Face Sequence

◆ Tracker 1



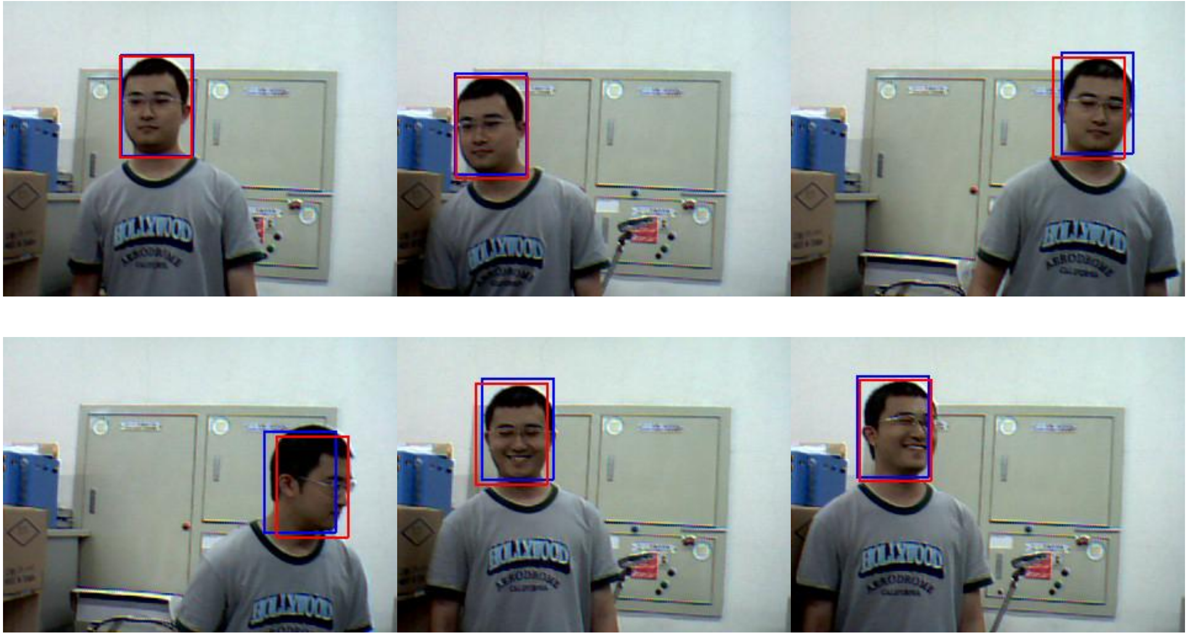


Figure 4.3-1 : Face tracking results of spatial-color mean-shift tracker 1 with rg feature in 3.5. Shown are frames 33, 93, 117, 126, 183, 256, 271, 455, 766. (red: tracker 1 with rg feature, blue: tracker 1 with RGB feature)

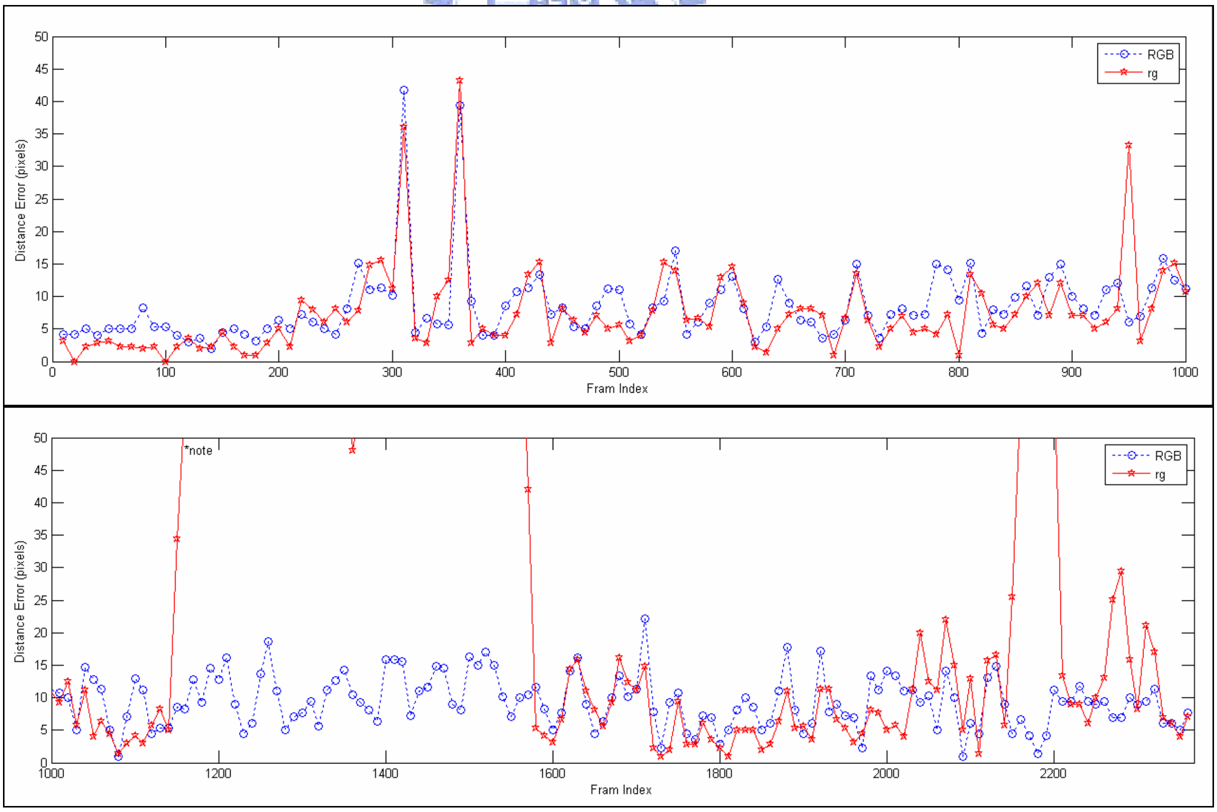


Figure 4.3-2 : Distance error of face sequence of spatial-color mean-shift tracker1 with rg feature that is compared with that with RGB feature in 3.5. (*note: we only consider the distance error which is smaller than 50 pixels)

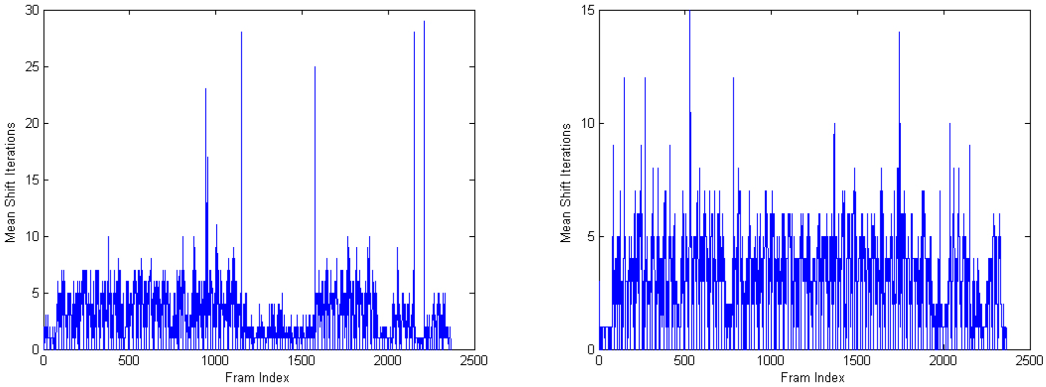
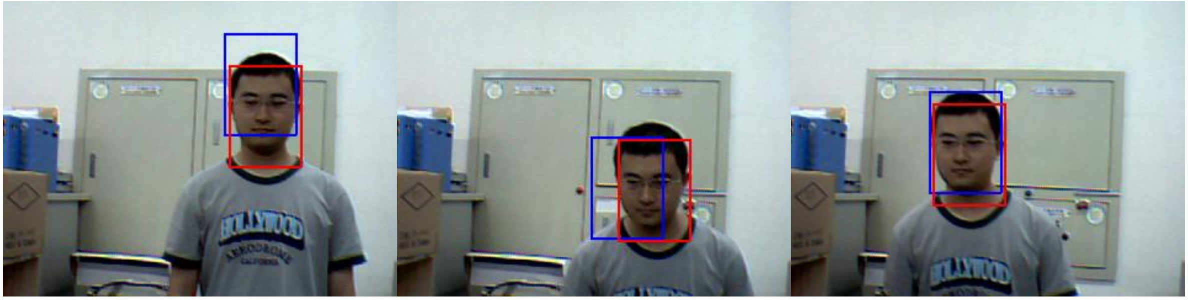


Figure 4.3-3 : Iteration numbers of face sequence. (left: tracker1 with rg feature, right: tracker 1 with RGB feature)

As shown in Figure 4.3-2, the tracker 1 with rg feature loses the target at about 1190th frame because at the left top of the scene there is a box which has similar appearance with face. In the situation of the face being tracked, the distance errors of tracker 1 with rg feature are more accurate than those of tracker 1 with RGB feature, and the average of iteration number of tracker 1 with rg feature is smaller than that of tracker 1 with RGB feature. Changing feature space to the normalized feature space can speed up the performance of tracker.

◆ Tracker 2



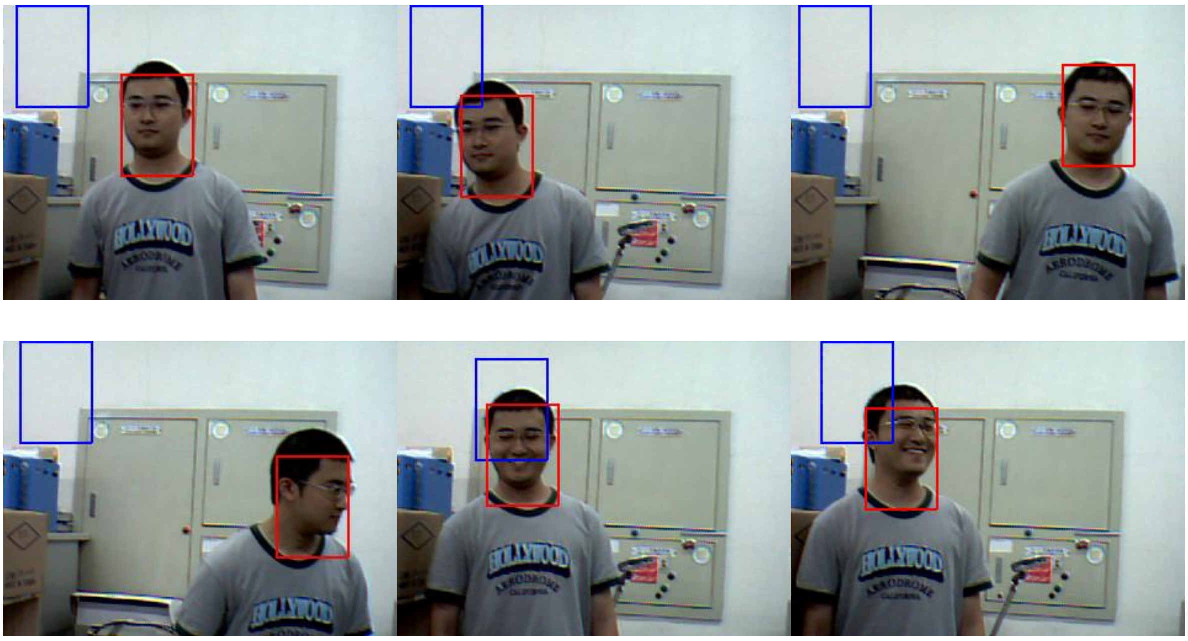


Figure 4.3-4 : Face tracking results of spatial-color mean-shift tracker 2 with rg feature in 3.5. Shown are frames 33, 93, 117, 126, 183, 256, 271, 455, 766. (red: tracker 2 with rg feature, blue: tracker 2 with RGB feature)

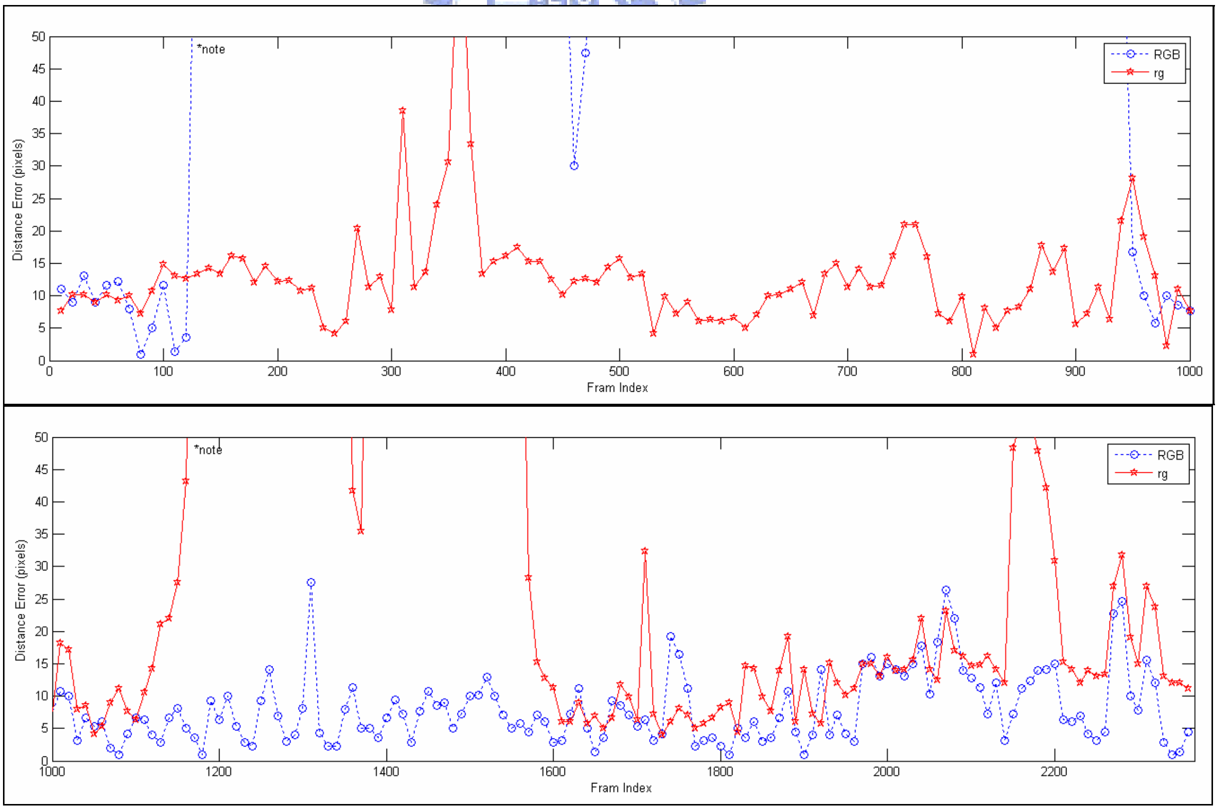


Figure 4.3-5 : Distance error of face sequence of spatial-color mean-shift tracker2 with rg feature that is compared with that with RGB feature in 3.5. (*note: we only consider the distance error which is smaller than 50 pixels)

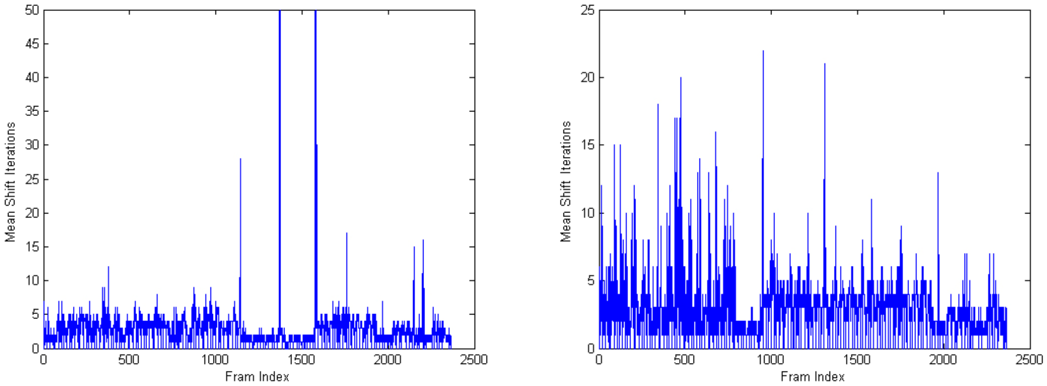
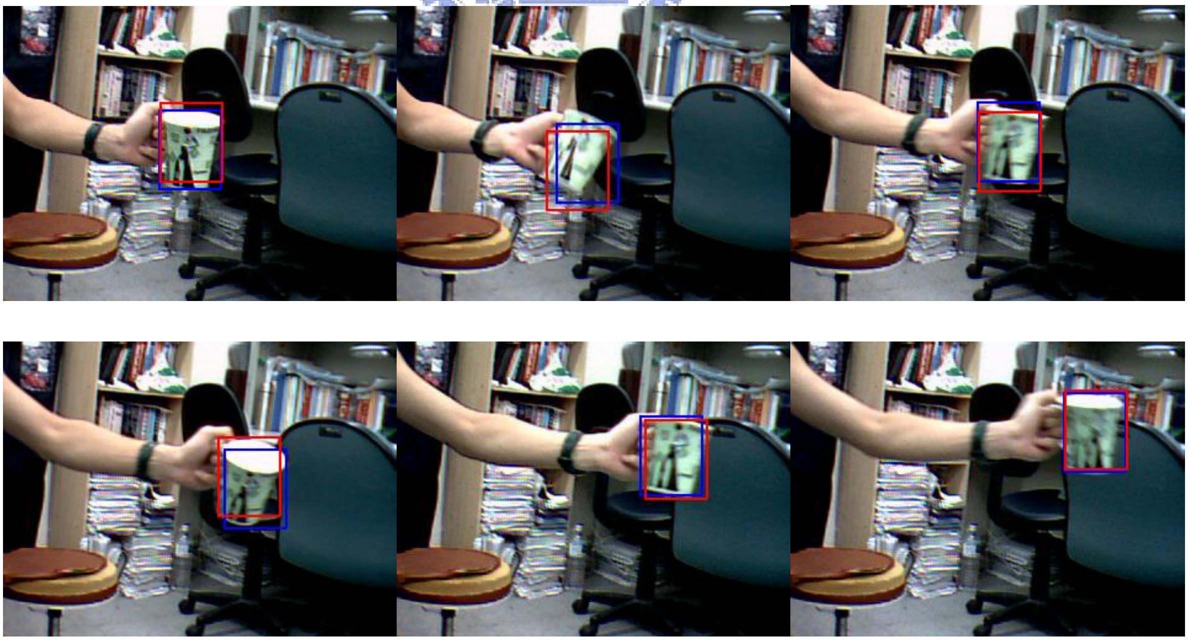


Figure 4.3-6 : Iteration numbers of face sequence. (left: tracker2 with rg feature, right: tracker 2 with RGB feature)

Figure 4.3-5 shows that the tracker 2 with rg feature makes tracking more 'workable' than the tracker 2 with RGB feature. The performance of tracker 2 with rg feature is better than that of tracker 1 with RGB feature.

4.3.2 Cup Sequence

◆ Tracker 1



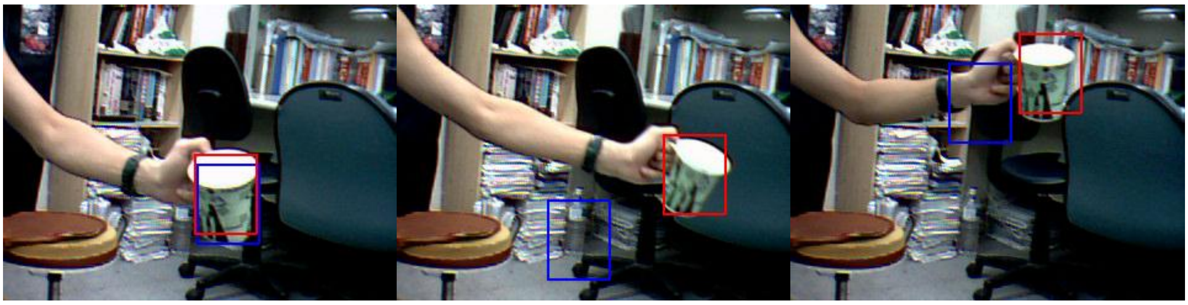


Figure 4.3-7 : Cup tracking results of spatial-color mean-shift tracker 1 with rg feature in 3.5. Shown are frames 4, 45, 63, 69, 81, 105, 166, 243, 364. (red: tracker 1 with rg feature, blue: tracker 1 with RGB feature)

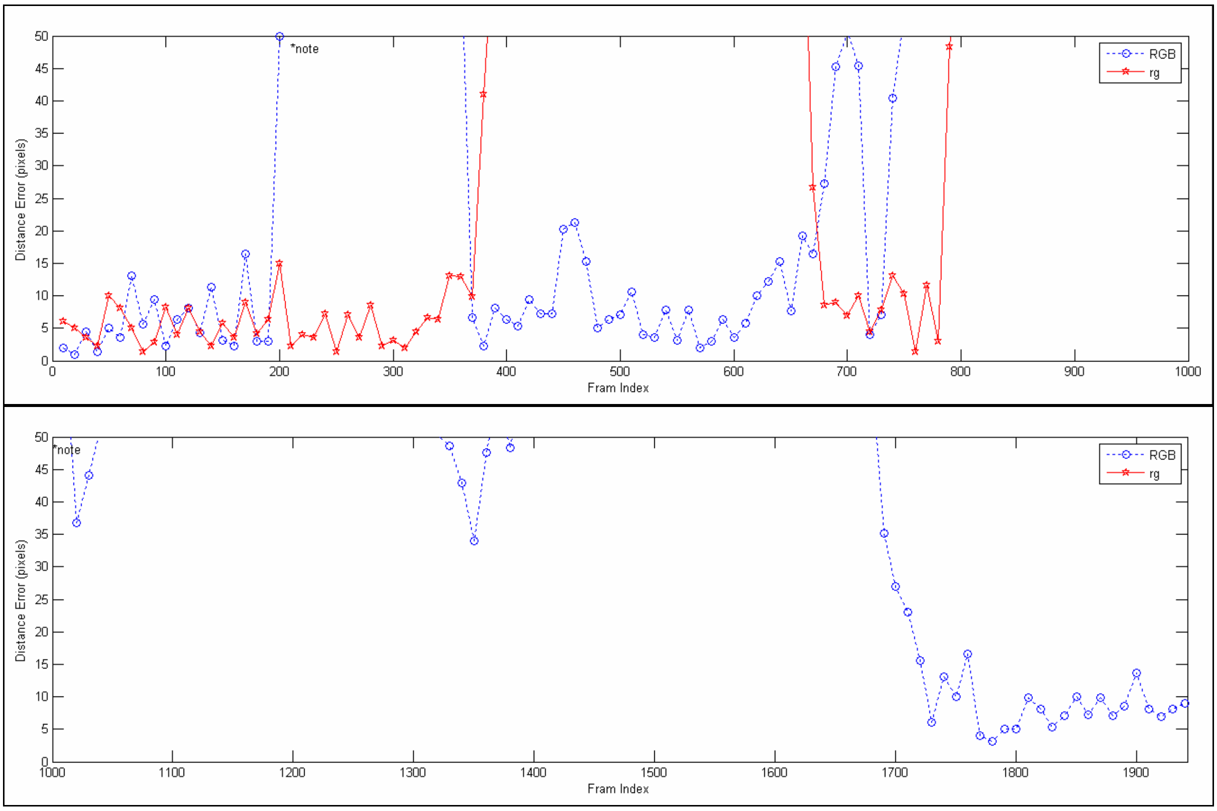


Figure 4.3-8 : Distance error of cup sequence of spatial-color mean-shift tracker 1 with rg feature that is compared with that with RGB feature in 3.5. (*note: we only consider the distance error which is smaller than 50 pixels)

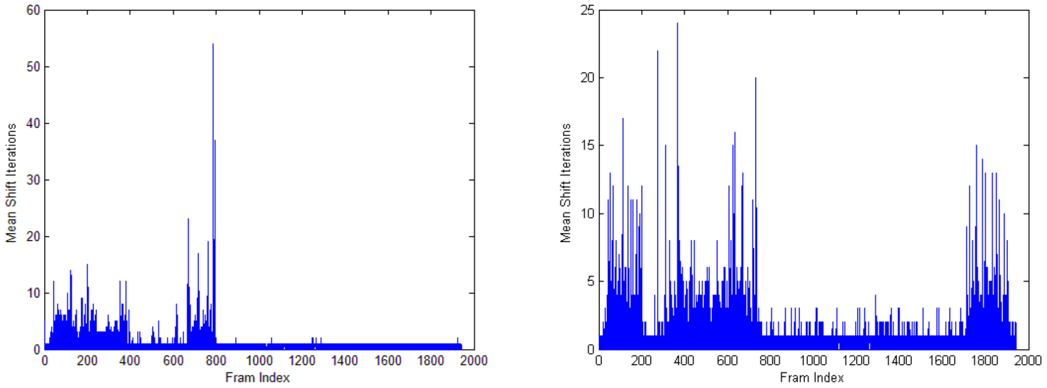
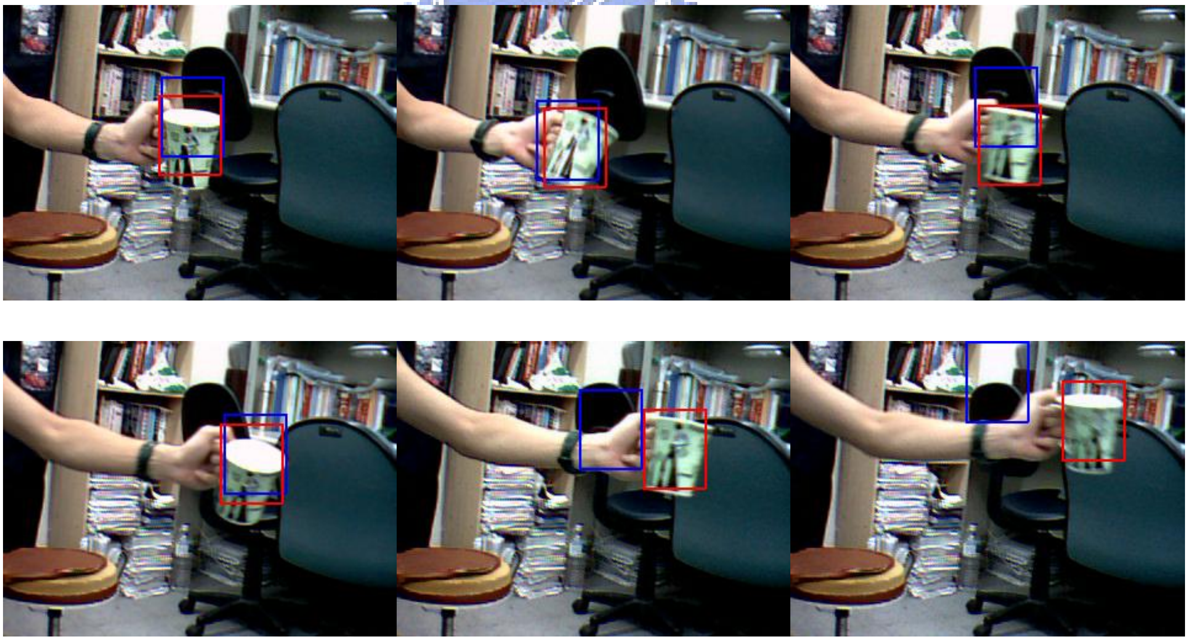


Figure 4.3-9 : Iteration number of cup sequence. (left: tracker 1 with rg feature, right: tracker 1 with RGB feature)

Change of the feature space with the cup sequence does not improve the tracking performance obviously. Because the variation of illumination is not large, the results of using rg feature are as well as that of using RGB feature.

◆ Tracker 2



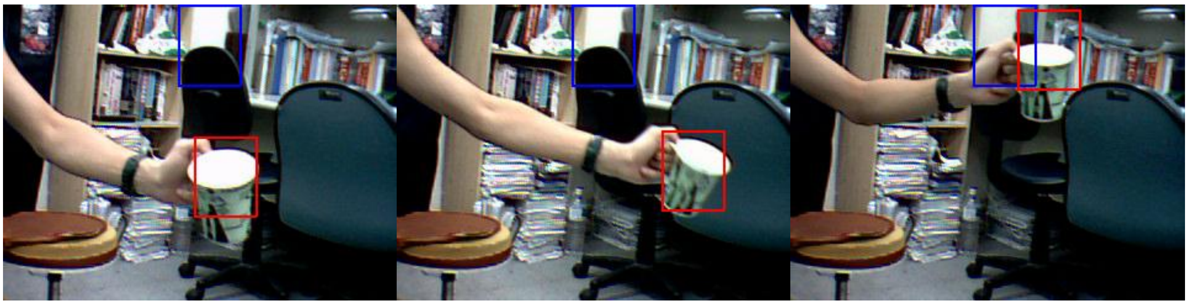


Figure 4.3-10 : Cup tracking results of spatial-color mean-shift tracker 2 with rg feature in 3.5. Shown are frames 4, 45, 63, 69, 81, 105, 166, 243, 364. (red: tracker 2 with rg feature, blue: tracker 2 with RGB feature)

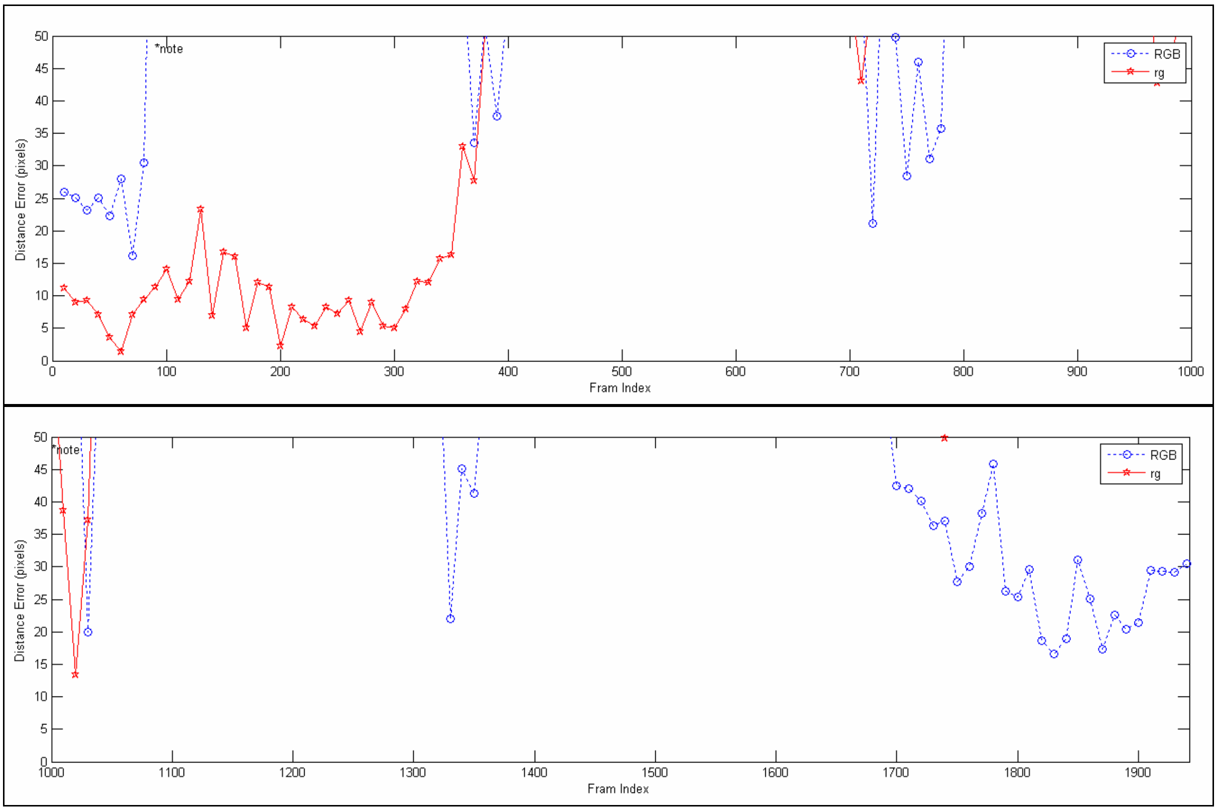


Figure 4.3-11 : Distance error of cup sequence of spatial-color mean-shift tracker 2 with rg feature that is compared with that with RGB feature in 3.5. (*note: we only consider the distance error which is smaller than 50 pixels)

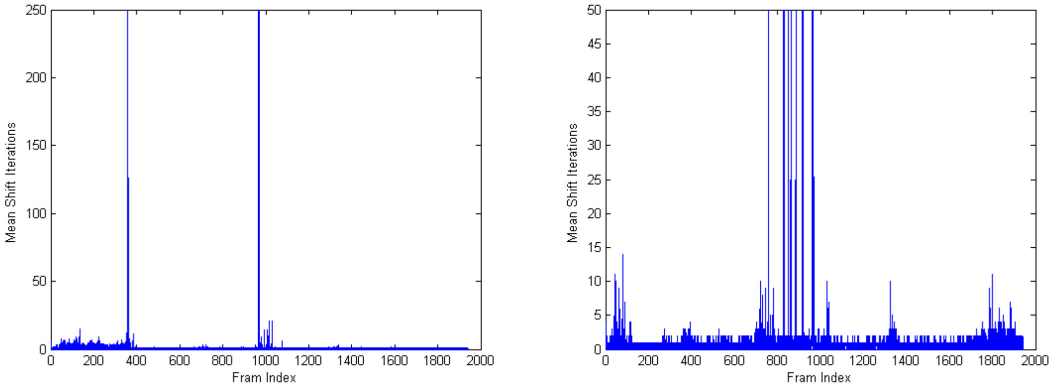
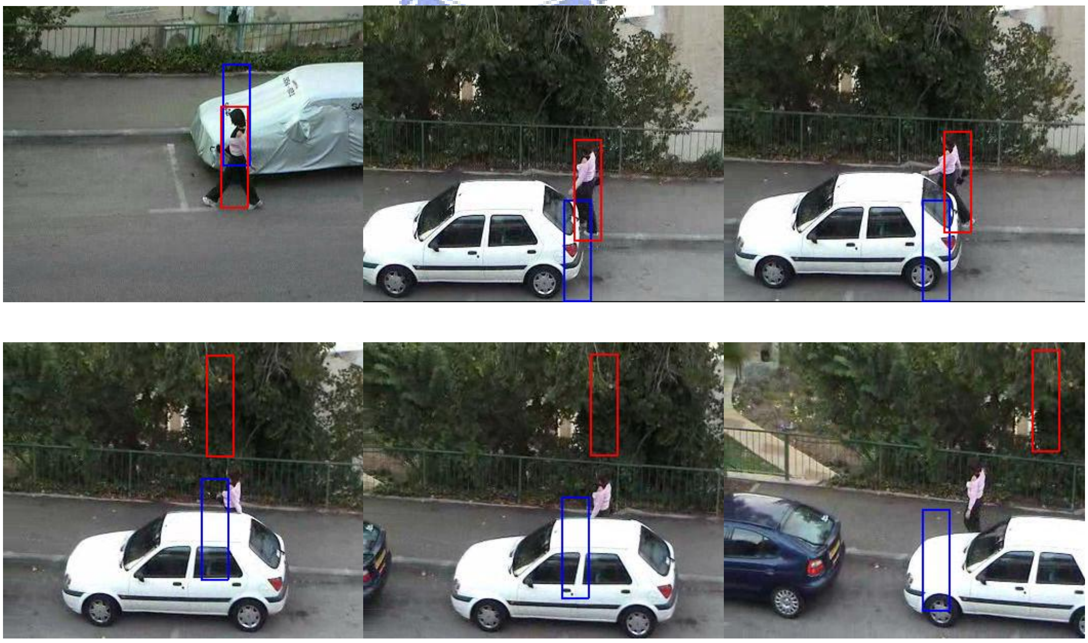


Figure 4.3-12 : Iteration numbers of cup sequence. (left: tracker 2 with rg feature, right: tracker 2 with RGB feature)

As shown in Figure 4.3-11, the tracker 2 with rg feature captures the target and tracks more accurately than the tracker 1 from 1st frame to about 350th frame. But looking at the overall distance errors, similar with the tracker 1 which uses the rg feature, the tracker 2 using the rg feature does not make tracking performance better.

4.3.3 Walking Girl Sequences

- Tracker 1



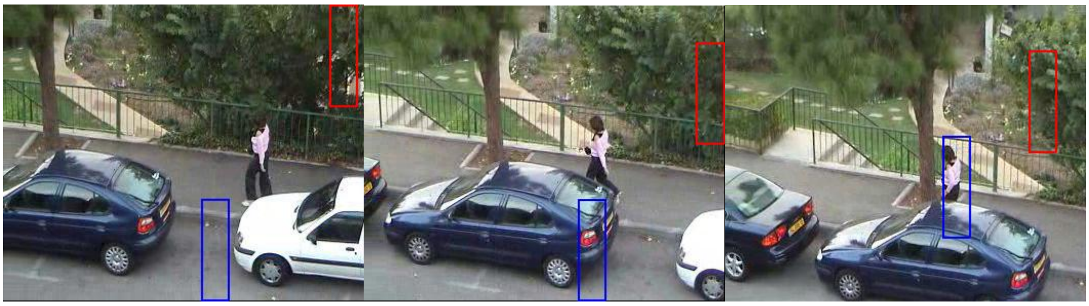


Figure 4.3-13 : Walking girl tracking results of spatial-color mean-shift tracker 1 with rg feature in 3.5. Shown are frames 28, 106, 111, 124, 130, 153, 166, 196, .220. (red: tracker 1 with rg feature, blue: tracker 1 with RGB feature)

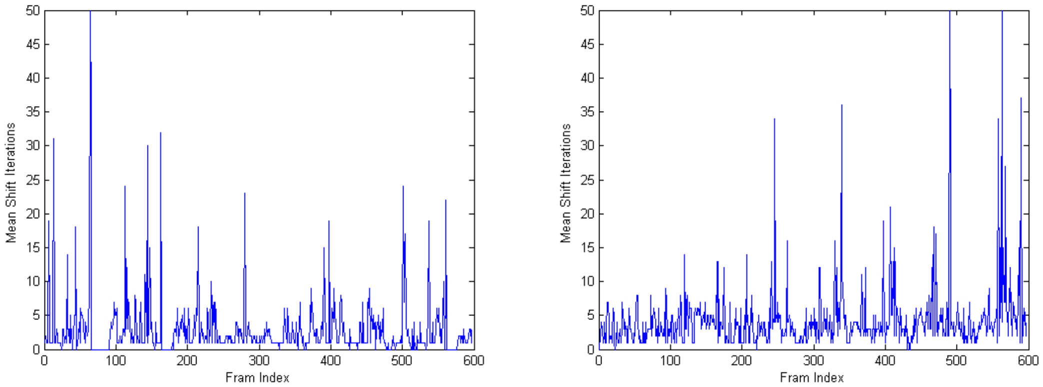
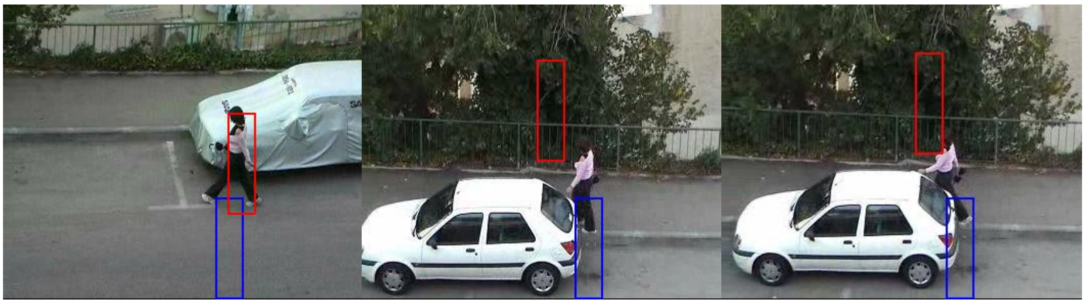


Figure 4.3-14 : Iteration numbers of walking girl sequence. (left: tracker 1 with rg feature, right: tracker 1 with RGB feature)

The tracker 1 with rg feature captures the girl from 1st frame to 111th frame, but loses it because the huge variation of illumination. The rg feature is not good enough to solve the huge variation of illumination.

- Tracker 2



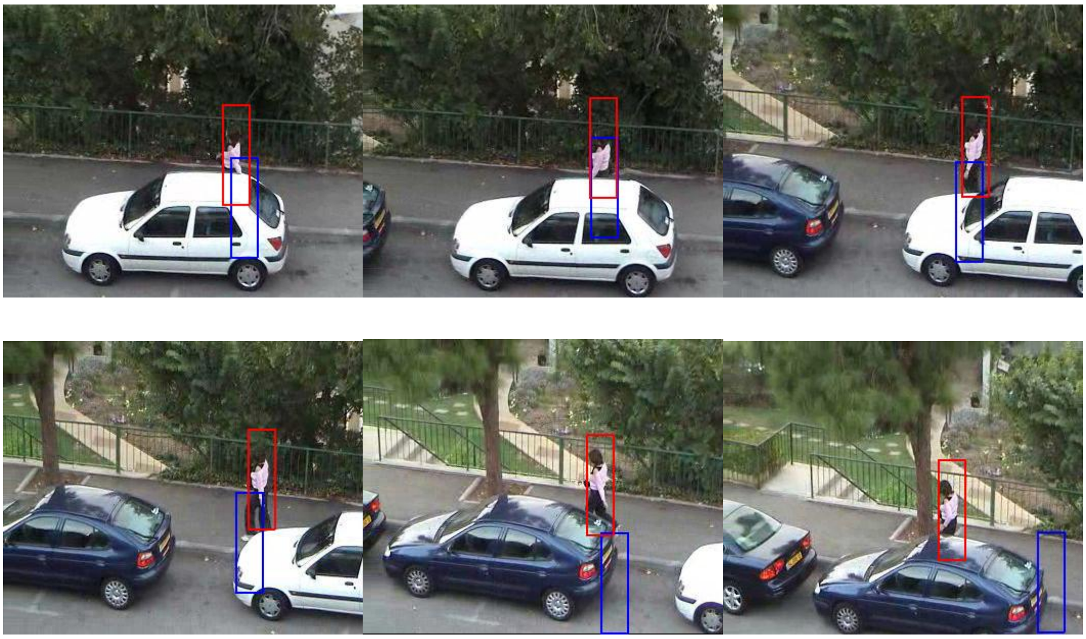


Figure 4.3-15 : Walking girl tracking results of spatial-color mean-shift tracker 2 with rg feature in 3.5. Shown are frames 28, 106, 111, 124, 130, 153, 166, 196, .220. (red: tracker 2 with rg feature, blue: tracker 2 with RGB feature)

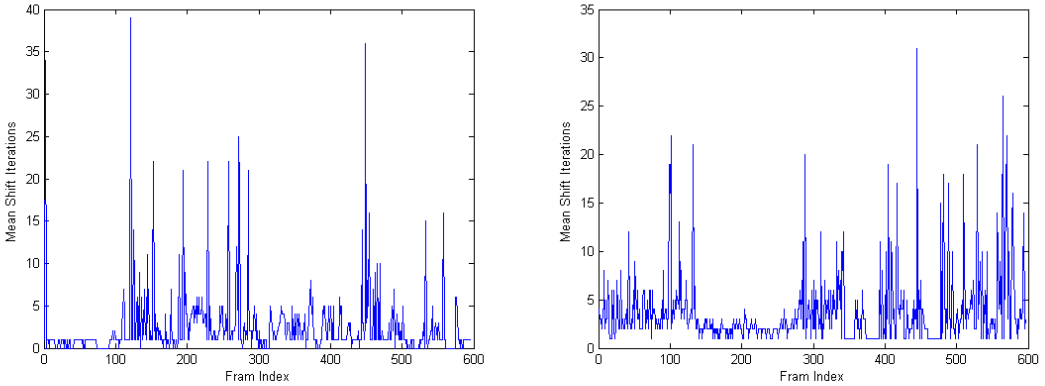


Figure 4.3-16 : Iteration numbers of walking girl sequence. (left: tracker 2 with rg feature, right: tracker 2 with RGB feature)

The tracker 2 with rg feature captures the girl at most frames, but loses the girl at about 70th frame because of the huge variation of illumination. Tracker 2 with rg feature is better than tracker 2 with RGB feature, and the total performance of tracker 2 with rg feature is better than that of tracker 1 with RGB feature, obviously.

4.4 Spatial-Color Mean-Shift Trackers with Normalized Feature

and Weighted Information

Considering the background information, we add the weighted-background information to the trackers which we developed before. In this section, the tracker 1 is defined as (3-31) and the tracker 2 is defined as (3-32).

4.4.1 Face Sequence

- Tracker 1

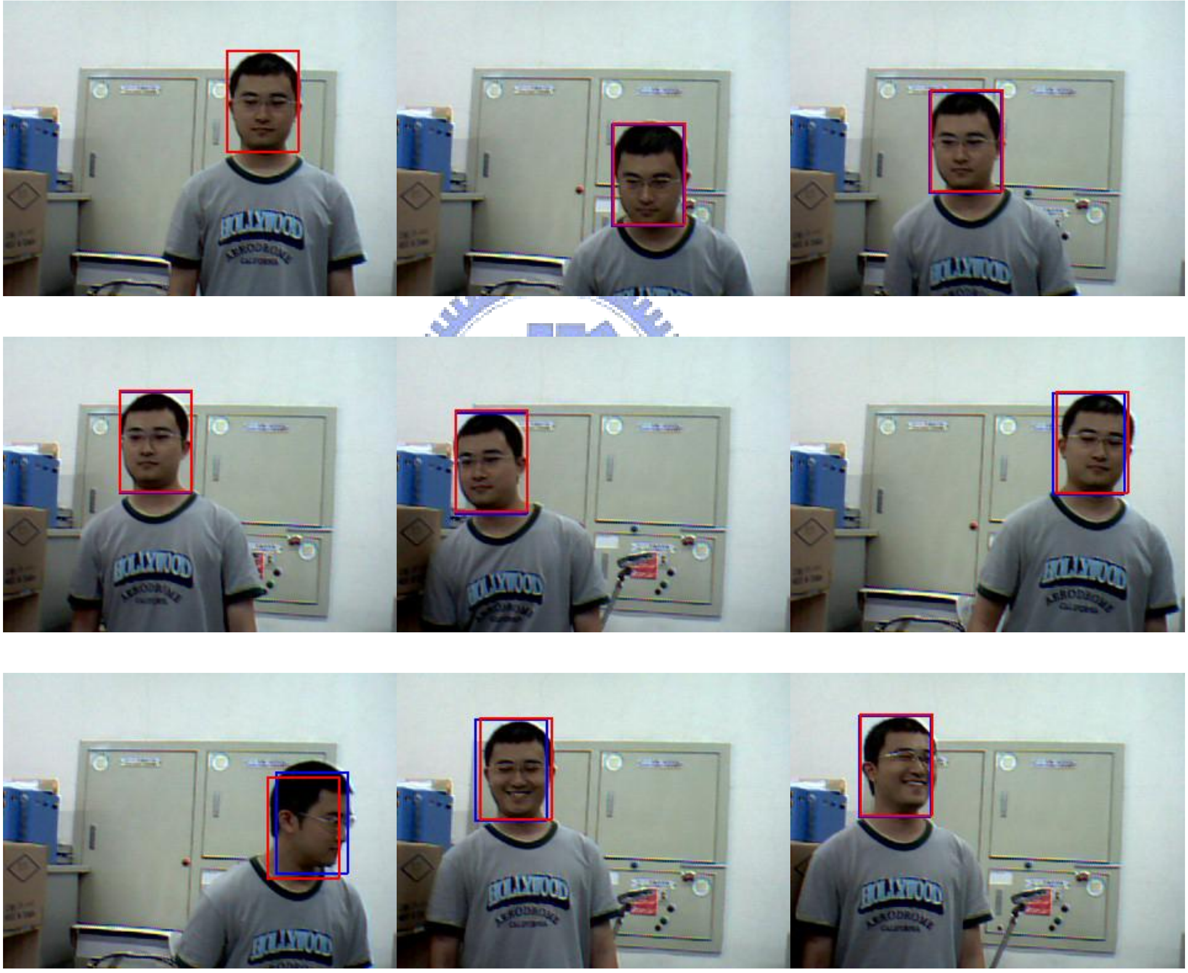


Figure 4.4-1 : Face tracking results of spatial-color mean-shift tracker 1 with rg feature and weighted-background information in 3.6. Shown are frames 33, 93, 117, 126, 183, 256, 271, 455, 766. (red: tracker 1 with rg feature and weighted-background information, blue: tracker 1 with rg feature only)

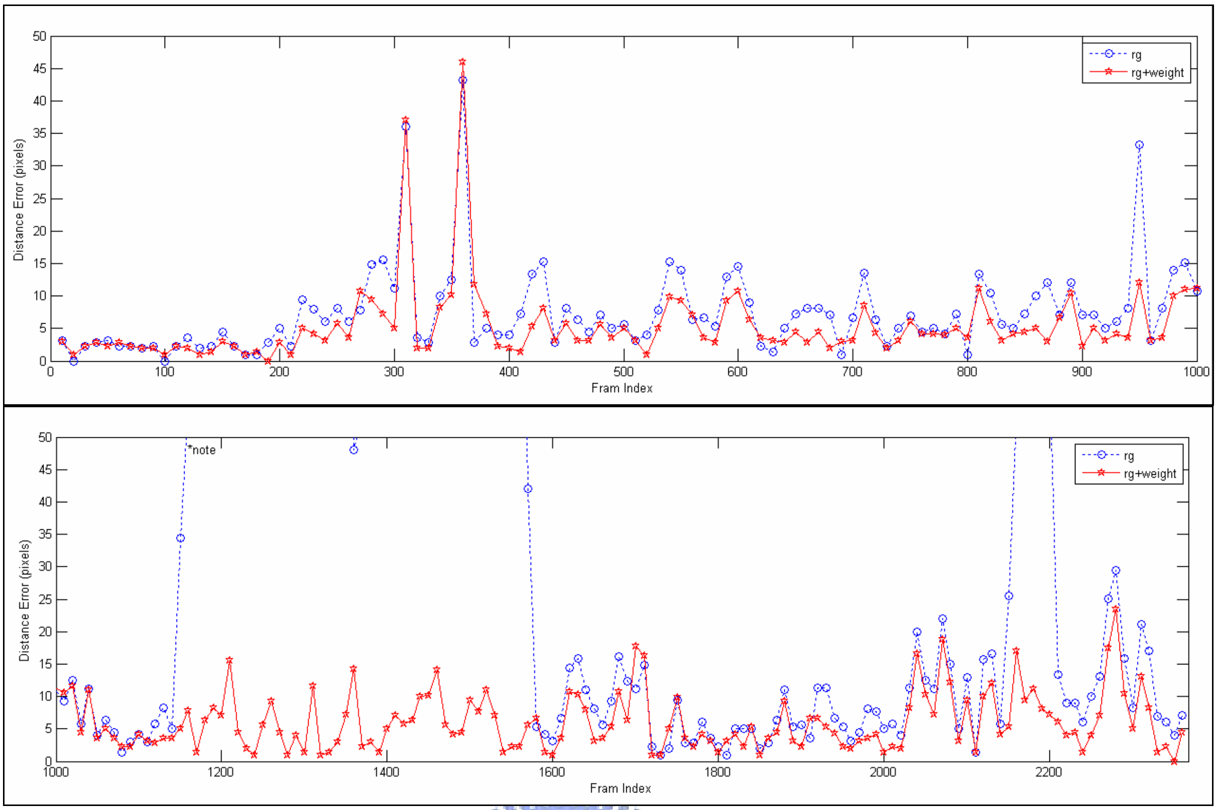


Figure 4.4-2 : Distance error of face sequence of spatial-color mean-shift tracker 1 with rg feature and weighted-background information in 3.6 that is compared with that with rg feature only. (*note: we only consider the distance error which is smaller than 50 pixels)

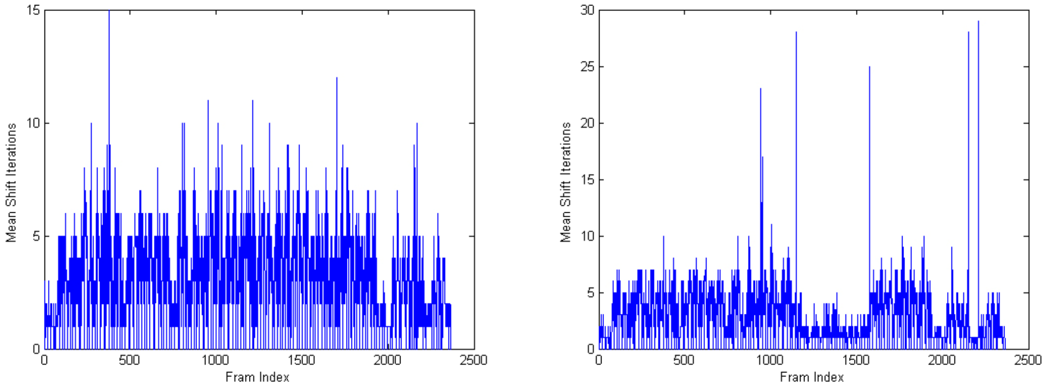


Figure 4.4-3 : Iteration number of face sequence. (left: tracker 1 with rg feature and weighted-background information, right: tracker 2 with rg feature only)

Figure 4.4-2 shows that the performance of tracker 1 is more accurate than that of tracker 1 without the weighted-background information, and tracker 1 captures the target at all times. The average of iteration number of tracker 1 is larger than that of tracker 1 without weighted-background information, but the difference between these two trackers is not large. In conclusion, the tracker 1 about the face tracking is much better.

- Tracker 2

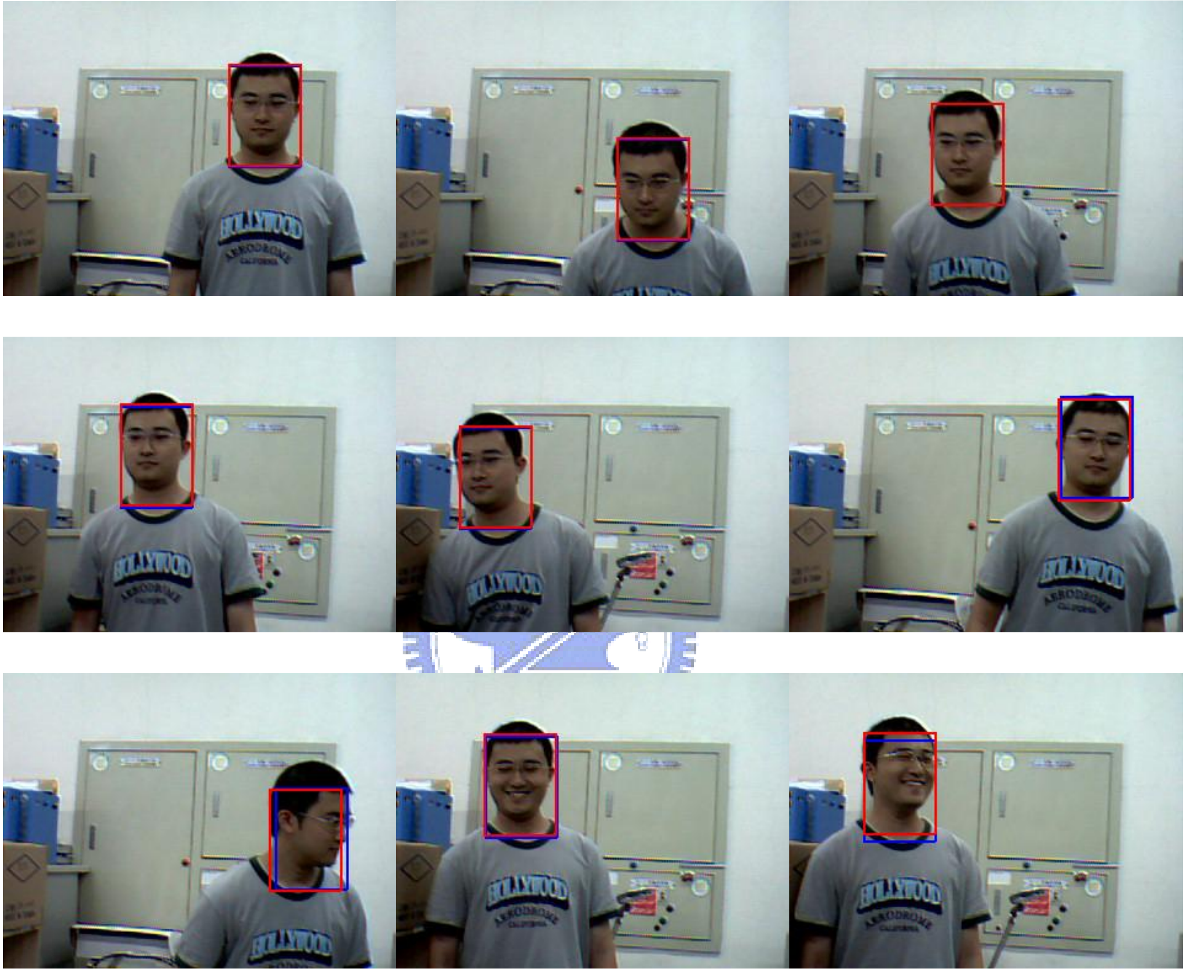


Figure 4.4-4 : Face tracking results of spatial-color mean-shift tracker 2 with rg feature and weighted-background information in 3.6. Shown are frames 33, 93, 117, 126, 183, 256, 271, 455, 766. (red: tracker 2 with rg feature and weighted-background information, blue: tracker 2 with rg feature only)

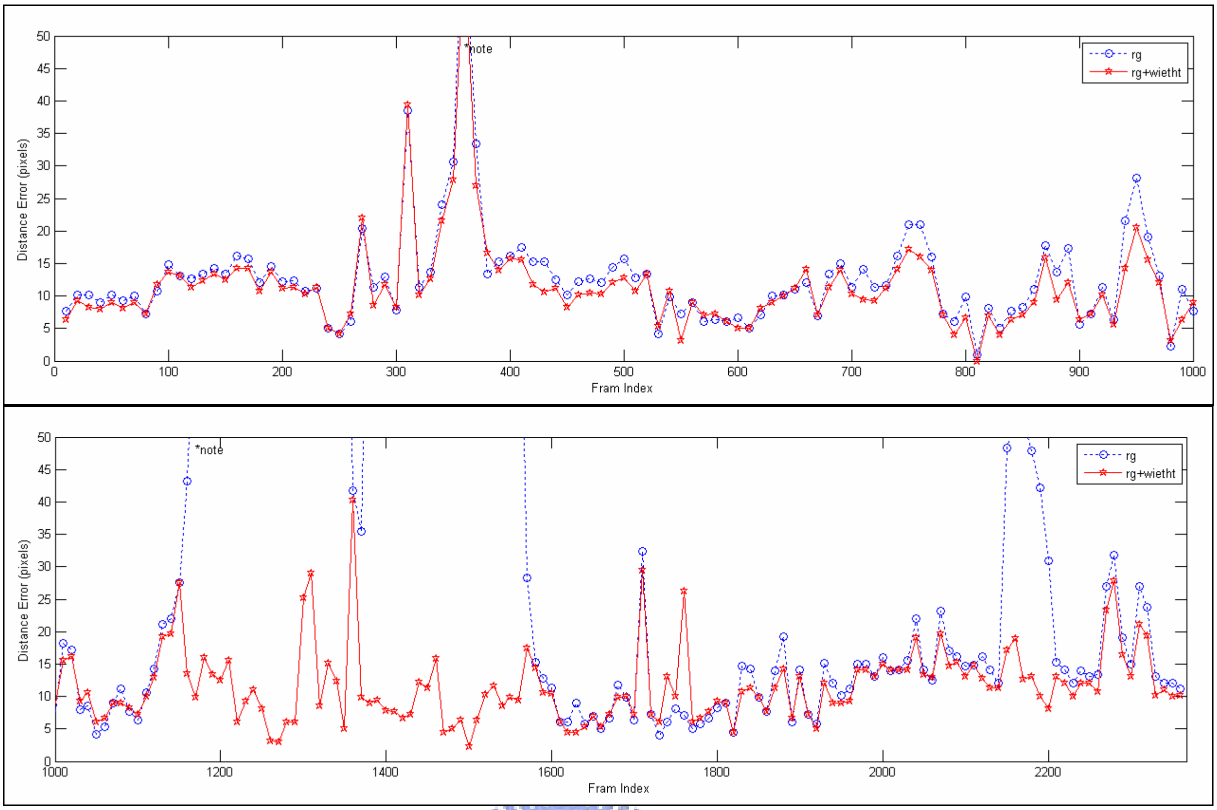


Figure 4.4-5 : Distance error of face tracking sequence of spatial-color mean-shift tracker 2 with rg feature and weighted-background information in 3.6 that is compared with that with rg feature only. (*note: we only consider the distance error which is smaller than 50 pixels)

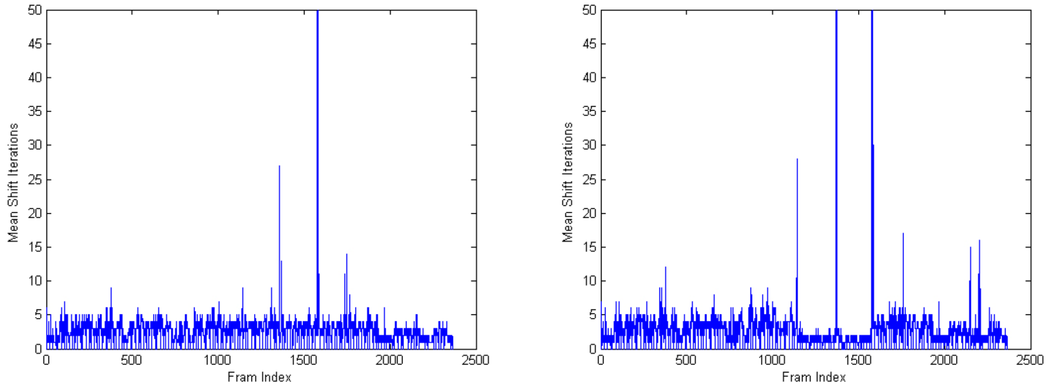


Figure 4.4-6 : Iteration numbers of face sequence. (left: tracker 1 with rg feature and weighted-background information, right: tracker 2 with rg feature only)

With the experiment result of tracker 2, Figure 4.4-5 also shows that the tracking results of tracker 2 is much better when adding the weighted-background information. The location of tracking is more accurate, and the iteration number of tracker 2 is about the same as the tracker 2 without the weighted-background information as shown in Figure 4.4-6.

◆ Tracker 1, Tracker 2, Traditional Mean-Shift Tracker

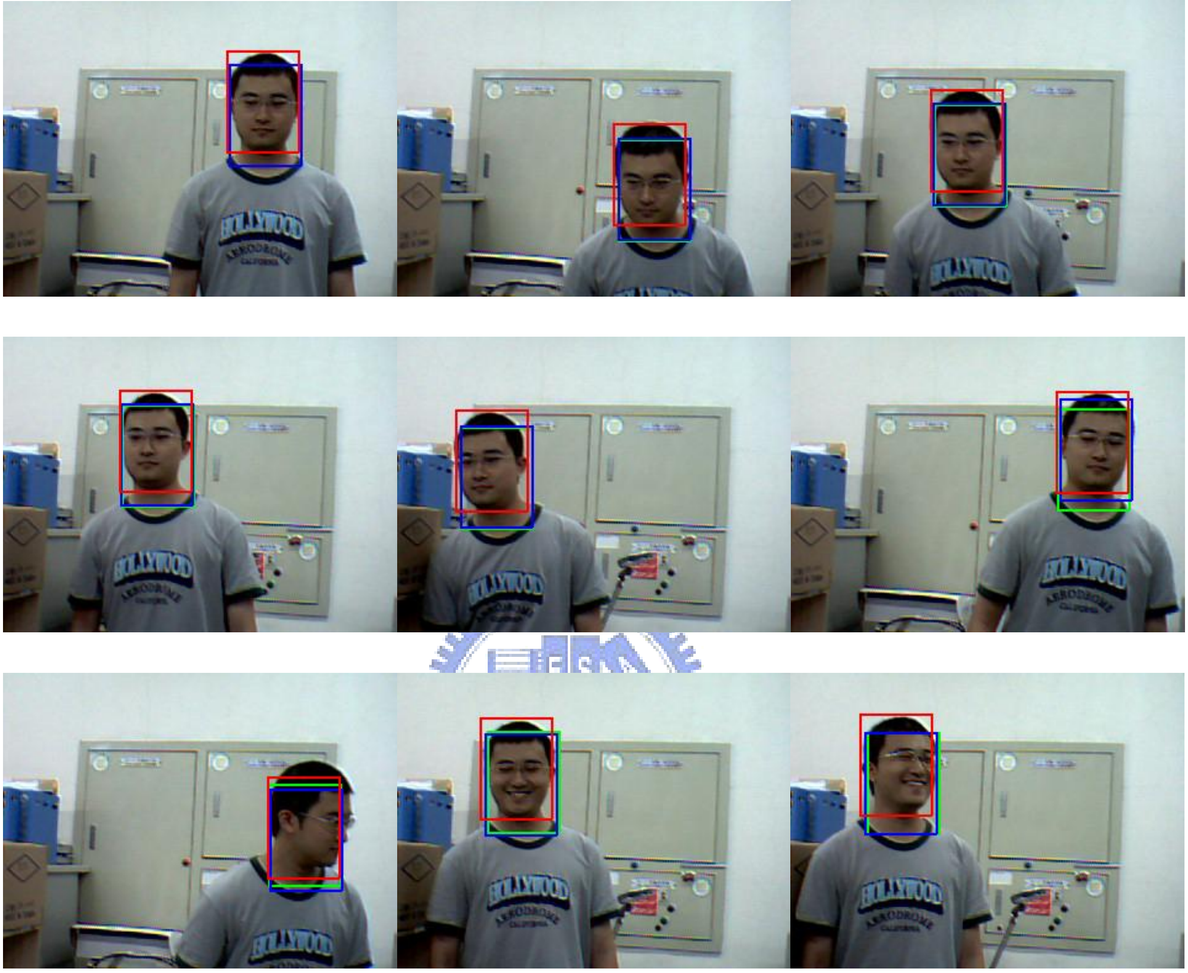


Figure 4.4-7 : Face tracking results of spatial-color mean-shift trackers with rg feature and weighted-background information in 3.6. Shown are frames 33, 93, 117, 126, 183, 256, 271, 455, 766. (red: tracker 1 with rg feature and weighted-background information, blue: tracker 2 with rg feature and weighted-background information, green: traditional mean-shift tracker)

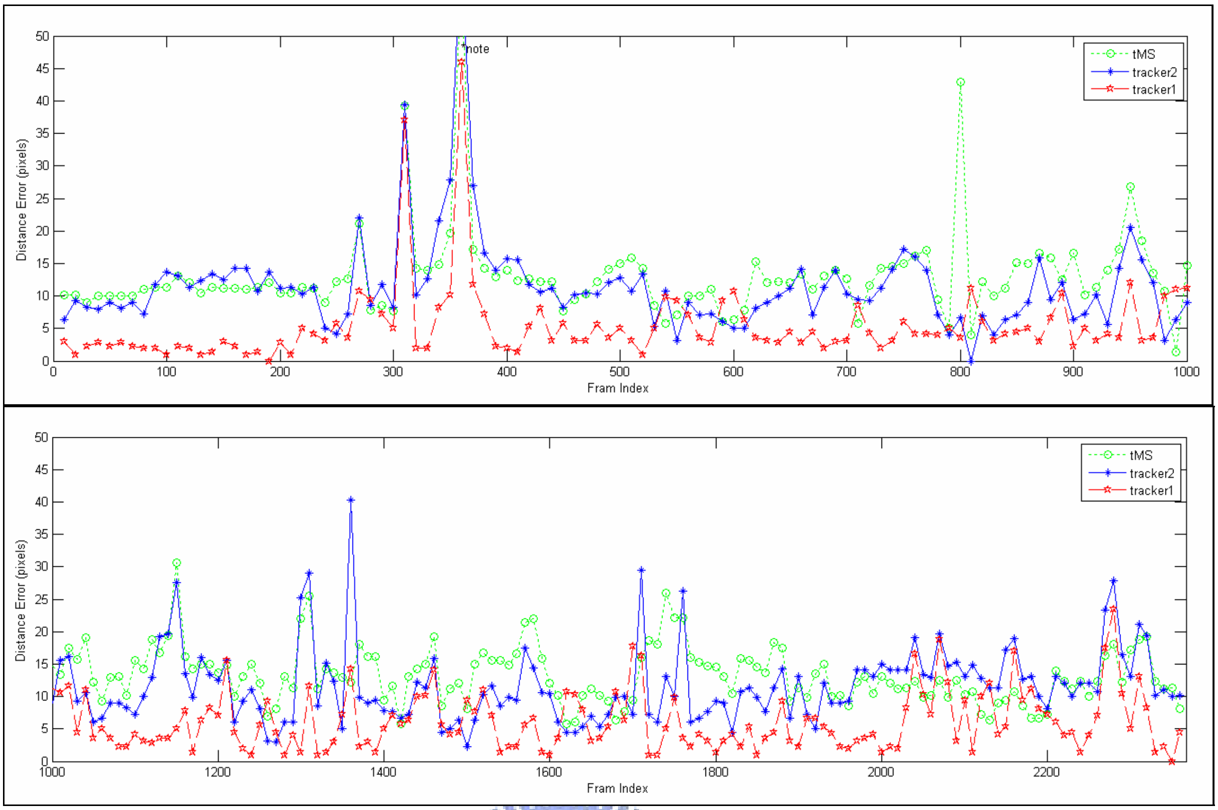


Figure 4.4-8 : Distance error of face sequence of spatial-color mean-shift trackers with rg feature and weighted-background information in 3.6. (*note: we only consider the distance error which is smaller than 50 pixels)

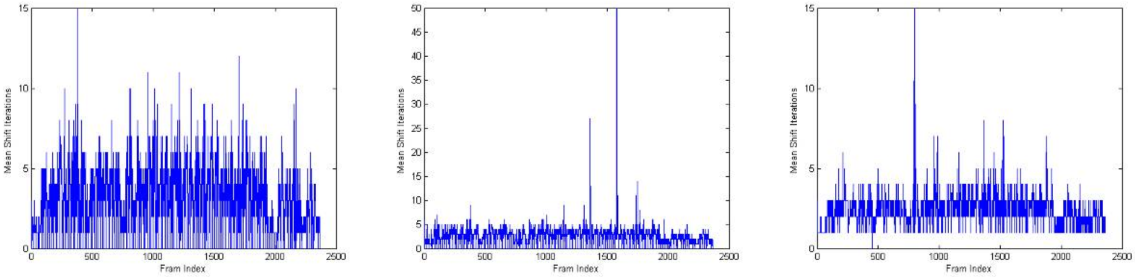


Figure 4.4-9 : Iteration number of face sequence. (left: tracker 1 with rg feature and weighted-background information, middle: tracker 2 with rg feature and weighted-background information, right: traditional mean-shift tracker)

We can see that the tracker 1 is the best tracker from the real face sequence as shown in Figure 4.4-7. The distance error of tracking results of the tracker 1, tracker 2, and the traditional mean-shift tracker are compared in Figure 4.4-8. In these three trackers, the tracking locations of tracker 1 are the most accurate than those of tracker

2 and than those of traditional mean-shift tracker. The iteration numbers of three trackers are about the same. Summary of all, the spatial-color mean-shift tracker 1 is the best, and the spatial-color mean-shift tracker 2 is better than the traditional mean-shift tracker.

4.4.2 Cup Sequence

◆ Tracker 1

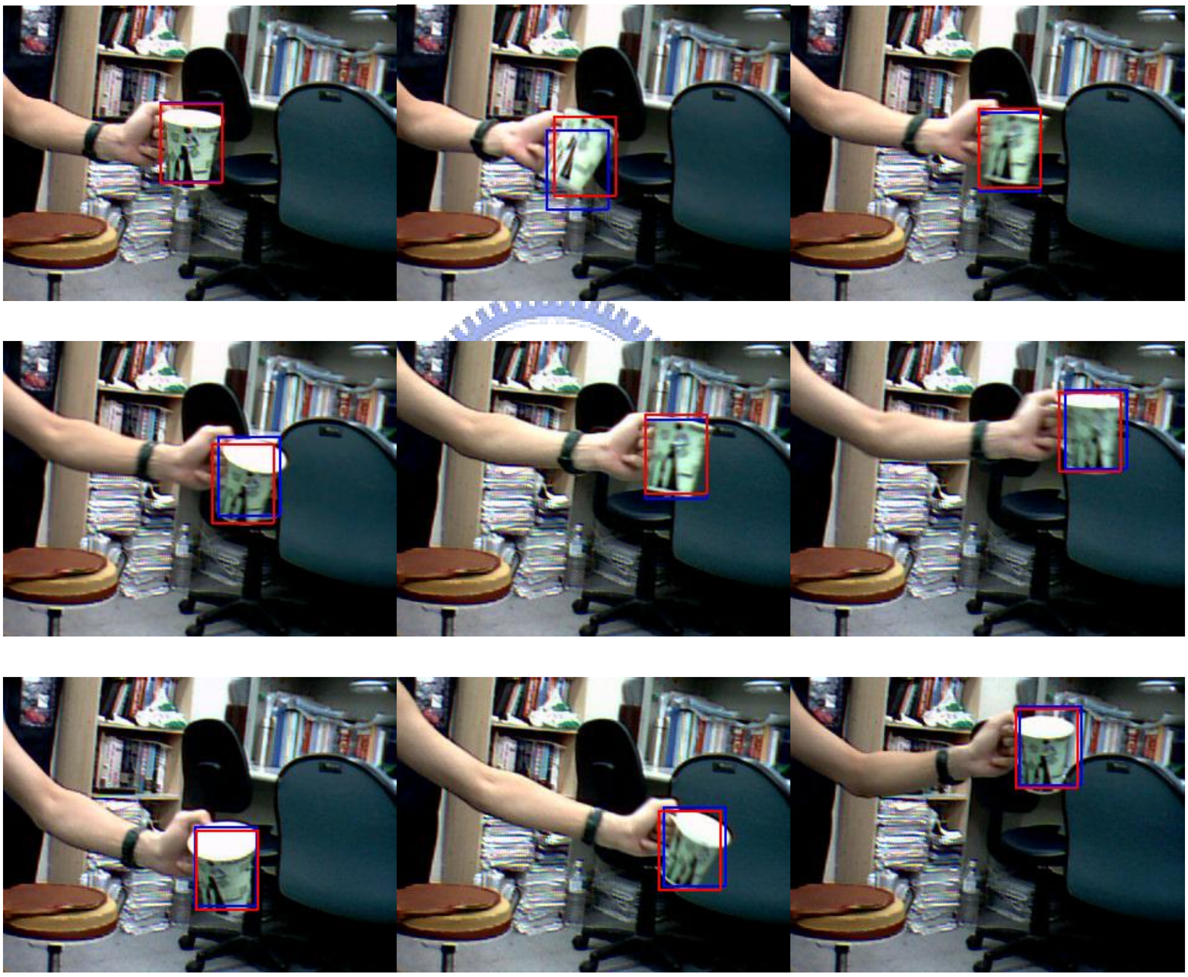


Figure 4.4-10 : Cup tracking results of spatial-color mean-shift tracker 1 with rg feature and weighted-background information in 3.6. Shown are frames 4, 45, 63, 69, 81, 105, 166, 243, 364. (red: tracker 1 with rg feature and weighted-background information, blue: tracker 1 with rg feature only)

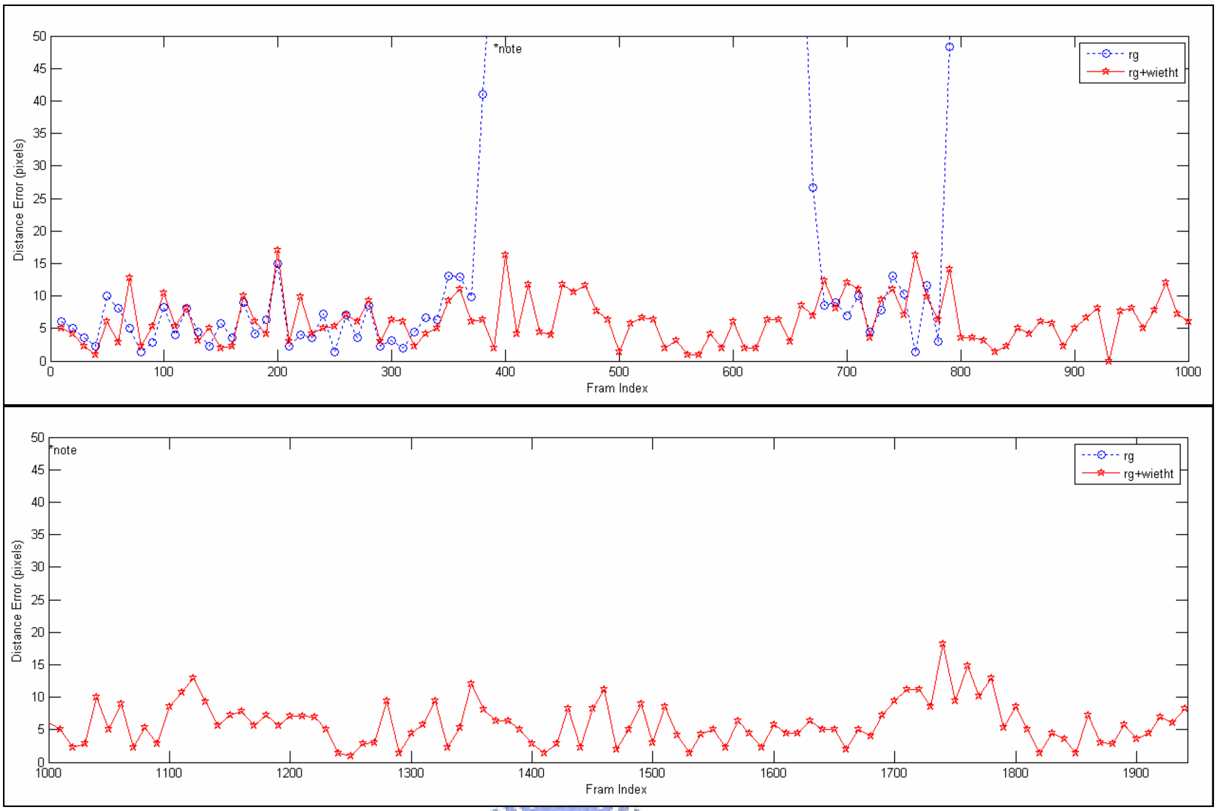


Figure 4.4-11 : Distance error of cup sequence of spatial-color mean-shift trackers with rg feature and weighted-background information in 3.6. (*note: we only consider the distance error which is smaller than 50 pixels)

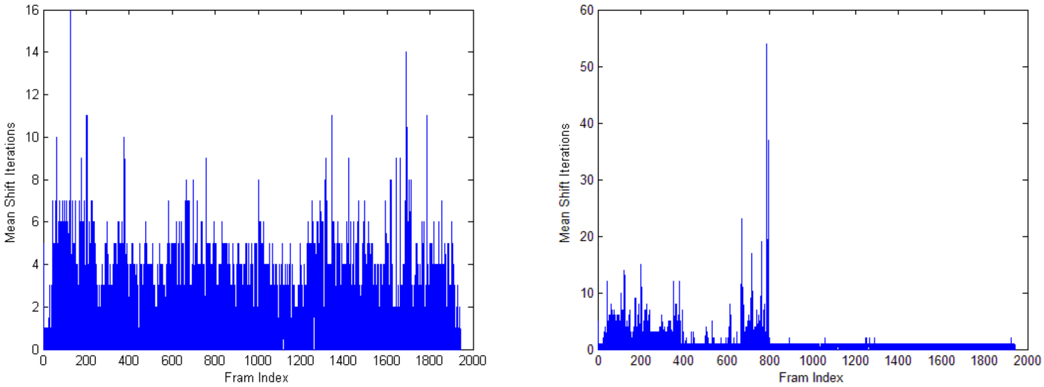


Figure 4.4-12 : Iteration number of cup sequence. (left: tracker 1 with rg feature and weighted-background information, right: tracker 1 with rg feature only)

With this sequence of complex background and complex appearance of target, the performance of the tracker 1 is strongly improved, obviously. The tracker 1 captures the target at all times when the tracker 1 without weighted-background information always loses the target.

- Tracker 2

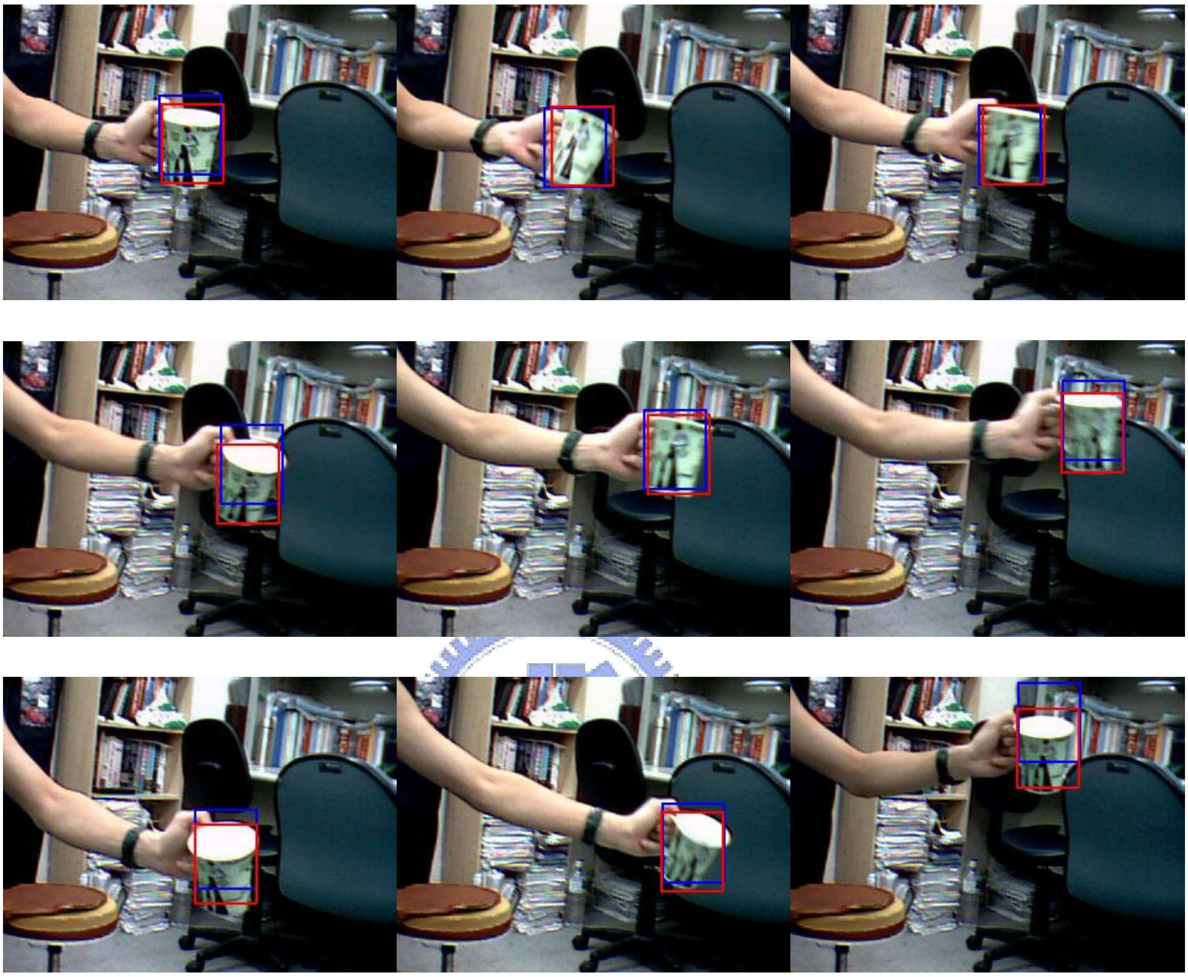
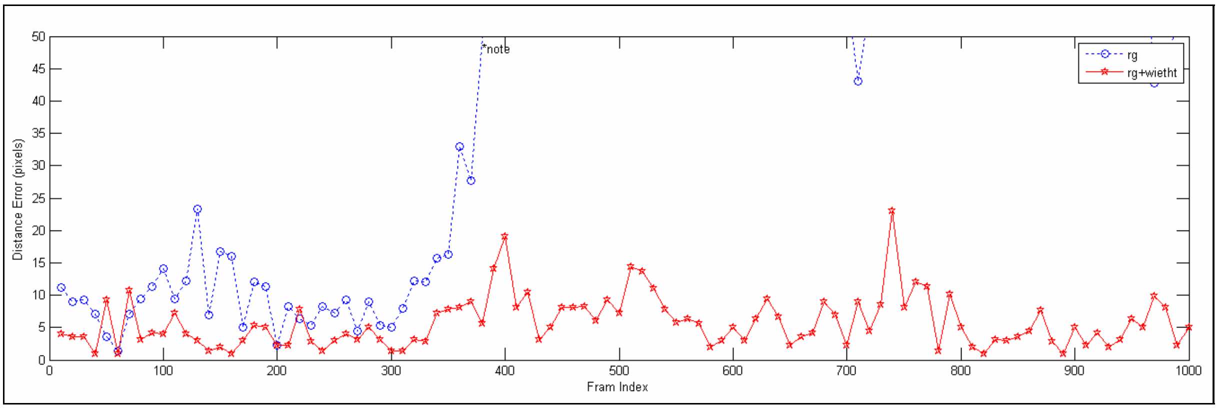


Figure 4.4-13 : Cup tracking results of spatial-color mean-shift tracker 2 with rg feature and weighted-background information in 3.6. Shown are frames 4, 45, 63, 69, 81, 105, 166, 243, 364. (red: tracker 2 with rg feature and weighted-background information, blue: tracker 2 with rg feature only)



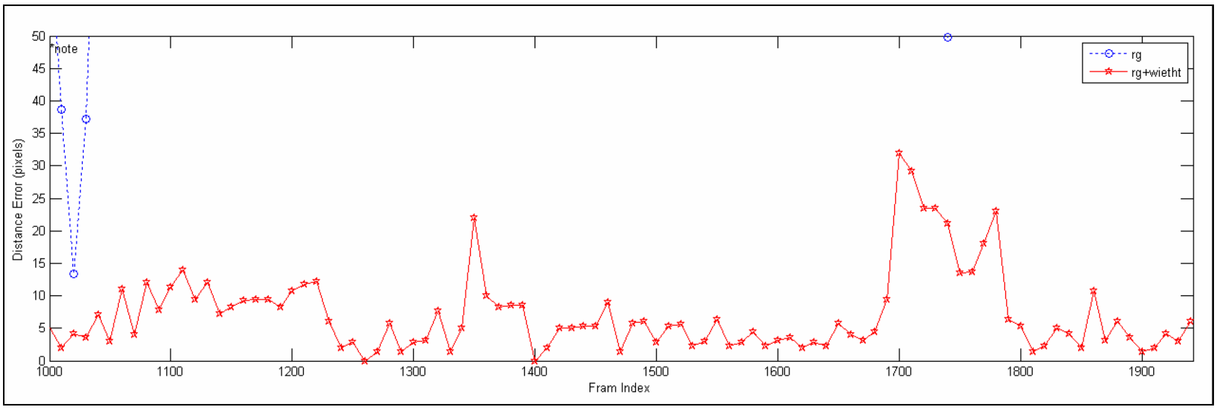


Figure 4.4-14 : Distance error of cup sequence of spatial-color mean-shift tracker with rg feature and weighted-background information in 3.6. (*note: we only consider the distance error which is smaller than 50 pixels)

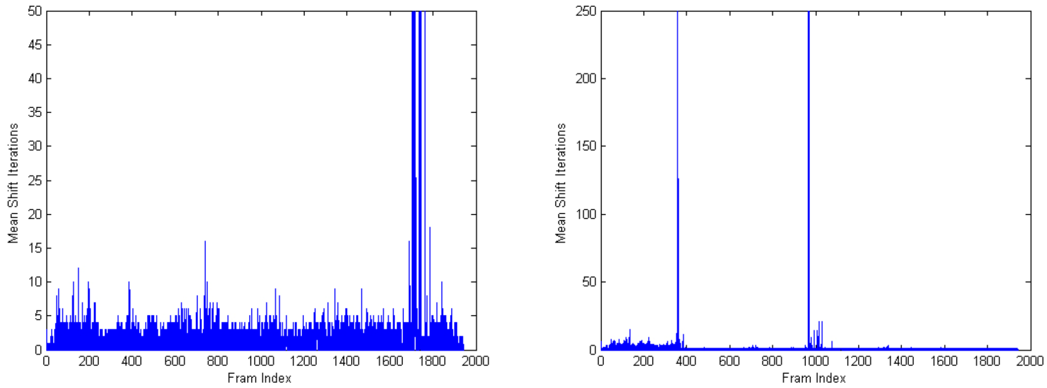
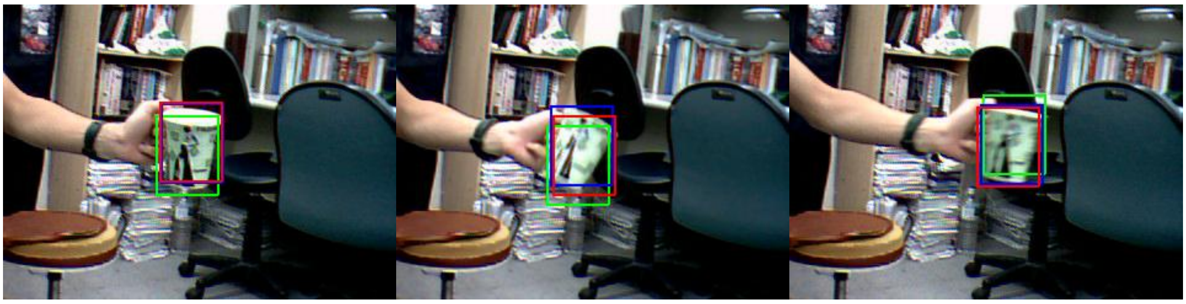


Figure 4.4-15 : Iteration numbers of cup sequence. (left: tracker 2 with rg feature and weighted-background information, right: tracker 2 with rg feature only)

Similar with tracker 2, the tracker is improved after adding weighted-background information and captures the target at all times.

- Tracker 1, Tracker 2, Traditional Mean-Shift Tracker



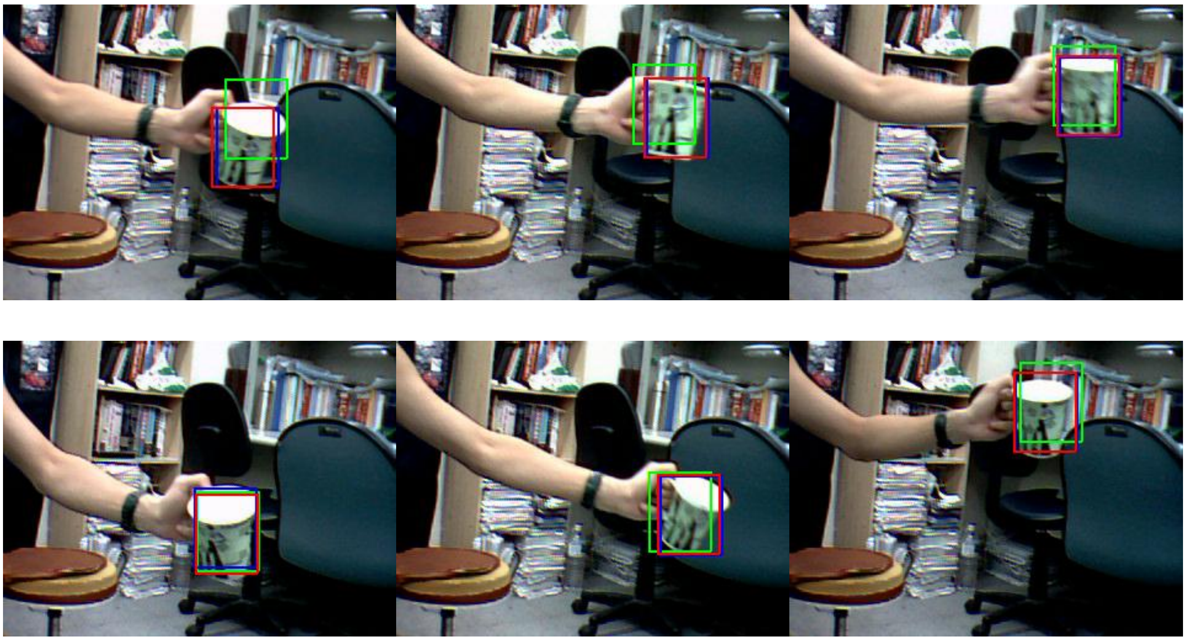


Figure 4.4-16 : Cup tracking results of spatial-color mean-shift trackers with rg feature and weighted-background information in 3.6. Shown are frames 4, 45, 63, 69, 81, 105, 166, 243, 364. (red: tracker 1 with rg feature and weighted-background information, blue: tracker 2 with rg feature and weighted-background information, green: traditional mean-shift tracker)

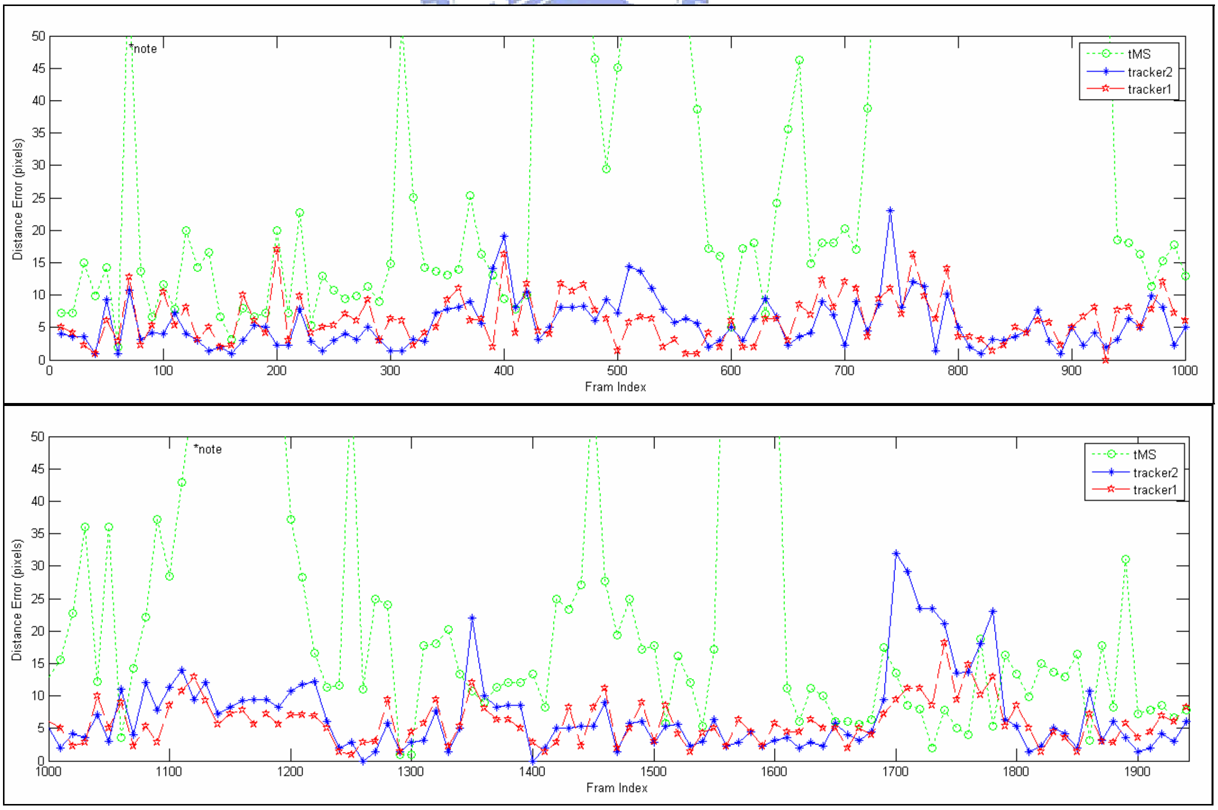


Figure 4.4-17 : Distance error of cup sequence of spatial-color mean-shift trackers with rg feature and

weighted-background information in 3.6. (*note: we only consider the distance error which is smaller than 50 pixels)

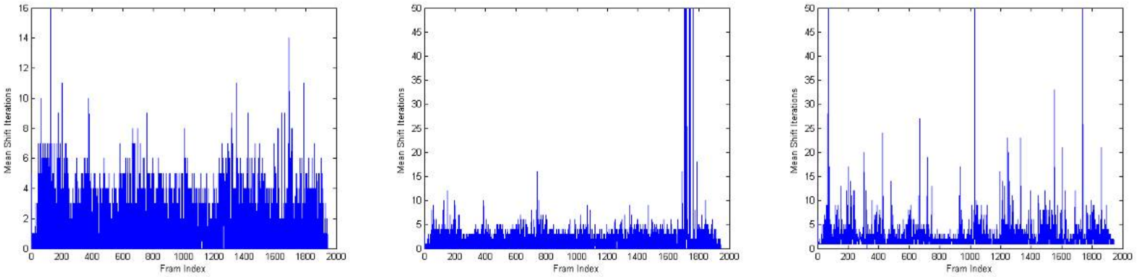
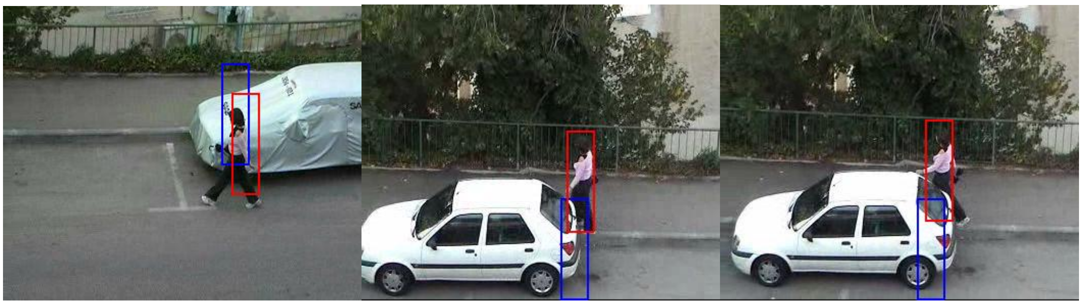


Figure 4.4-18 : Iteration number of cup sequence. (left: tracker 1 with rg feature and weighted-background information, middle: tracker 2 with rg feature and weighted-background information, right: traditional mean-shift tracker)

As shown in Figure 4.4-16, the all trackers capture the target at all times, but the tracking locations of the tracker 1 and tracker 2 are more accurate than the traditional mean-shift tracker in the real image sequences. Figure 4.4-17 shows the distance errors of three trackers; the errors of tracker 1 and tracker 2 are about the same and are always smaller than those of traditional mean-shift tracker. The iteration numbers of the proposed trackers are quite similar with the traditional one as shown in Figure 4.4-18. To sum up, the trackers which we developed up to this point is already better than the traditional one.

4.4.3 Walking Girl Sequence

◆ Tracker 1



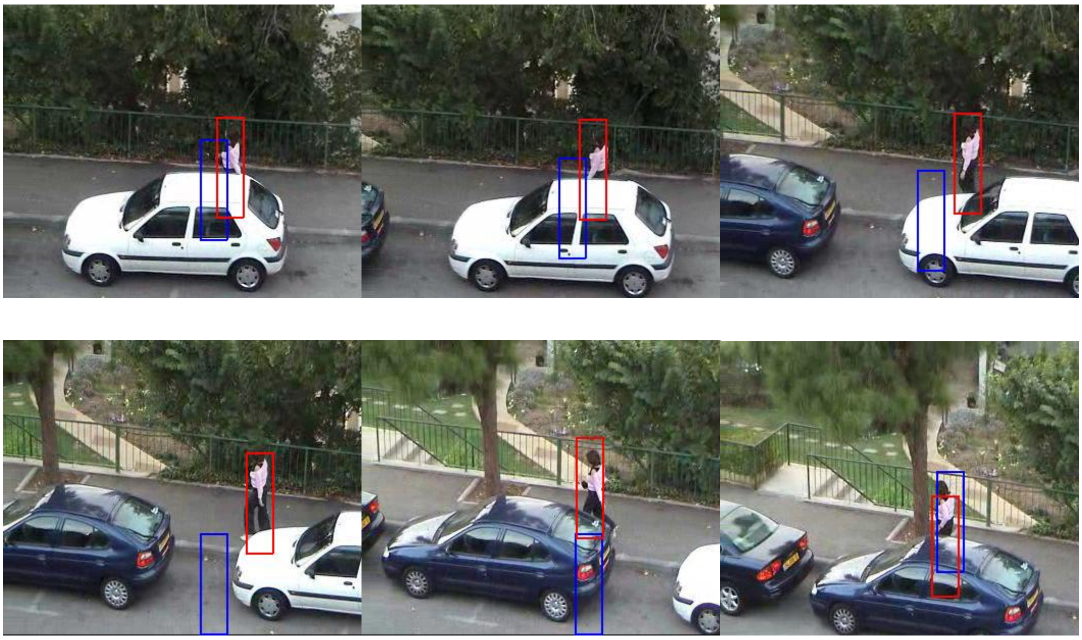


Figure 4.4-19 : Walking girl tracking results of spatial-color mean-shift tracker 1 with rg feature and weighted-background in 3.6. Shown are frames 28, 106, 111, 124, 130, 153, 166, 196, .220. (red: tracker 1 with rg feature and weighted-background, blue: tracker 1 with rg feature only)

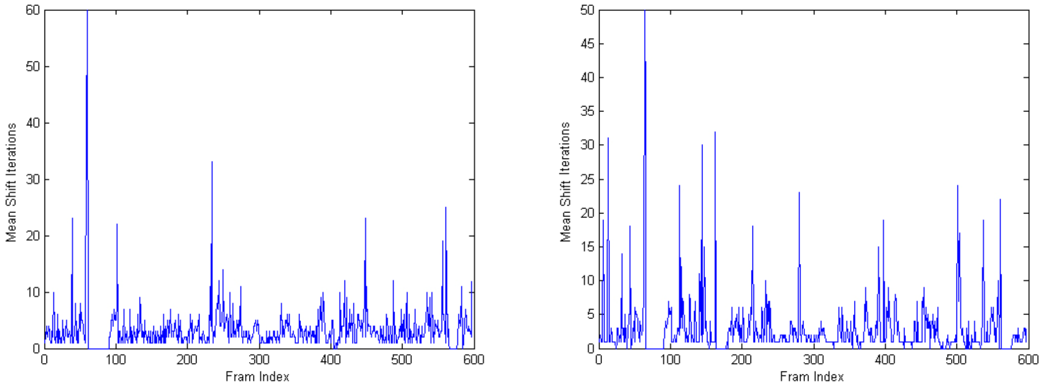


Figure 4.4-20 : Iteration numbers of walking girl sequence. (left: tracker 1 with rg feature and weighted-background, right: tracker 1 with rg feature only)

Under these circumstances of the variation of illumination and partial occlusion, Figure 4.4-19 shows that the tracker 1 captures the target when tracker 1 without weighted-background always fail. When the target has been covered by the cars from 106th frame to 220th frame, the tracker 1 still captures the target.

- ◆ Tracker 2

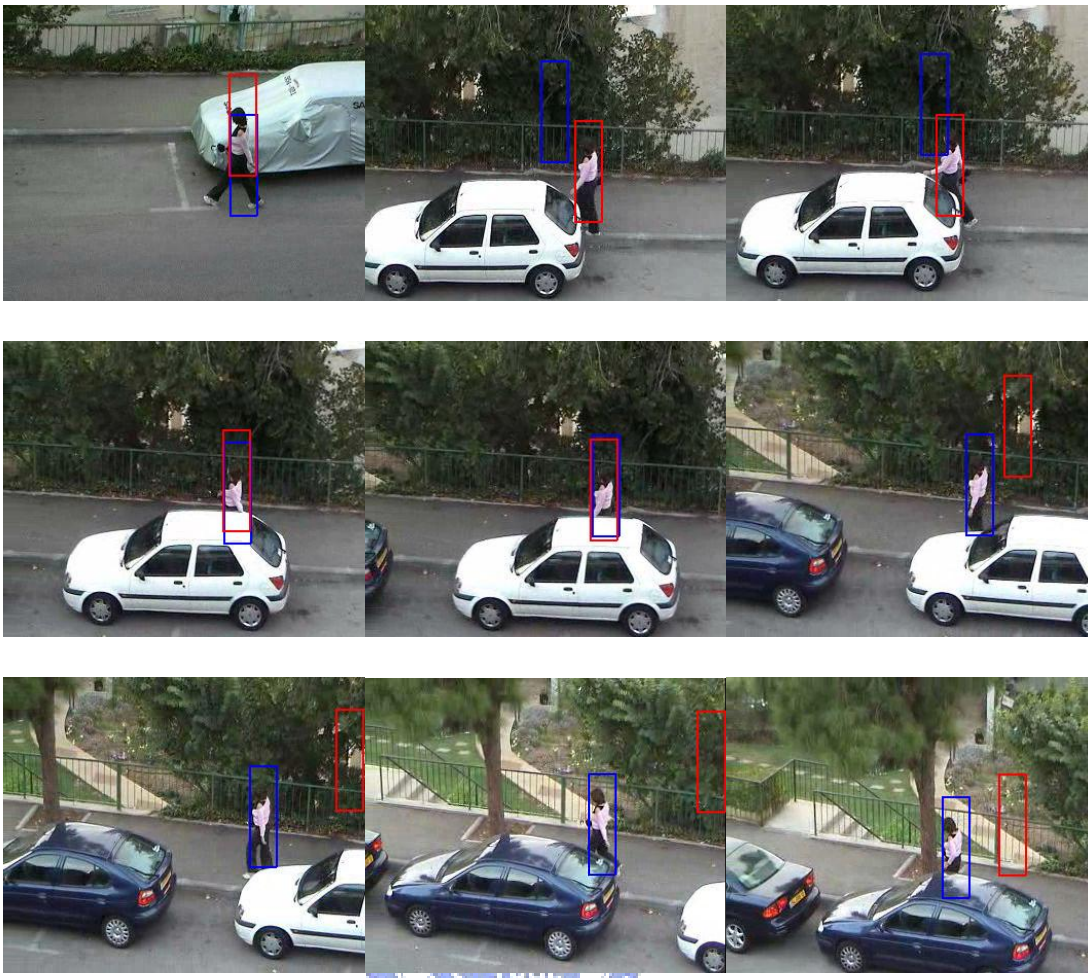


Figure 4.4-21 : Walking girl tracking results of spatial-color mean-shift tracker 2 with rg feature and weighted-background in 3.6. Shown are frames 28, 106, 111, 124, 130, 153, 166, 196, .220. (red: tracker 2 with rg feature and weighted-background information, blue: tracker 2 with rg feature only)

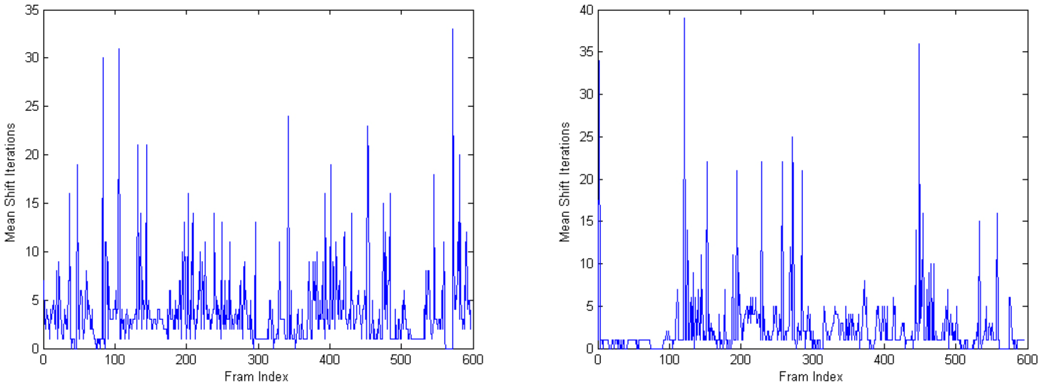


Figure 4.4-22 : Iteration numbers of walking girl sequence. (left: tracker 2 with rg feature and weighted-background, right: tracker 2 with rg feature only)

Figure 4.4-21 shows that the tracker 2 fails during the tracking process. At 153rd frame, the tracker 2 fails and captures the background object because the appearance of the region which tracker 2 tracks is very similar with the girl model which we want to track. The tracker 2 uses the spatial information and the spatial information of the background region at 153rd frame has very similar part, and this is the reason why tracker 2 fails.

- Tracker 1, Tracker 2, Traditional Mean-Shift Tracker

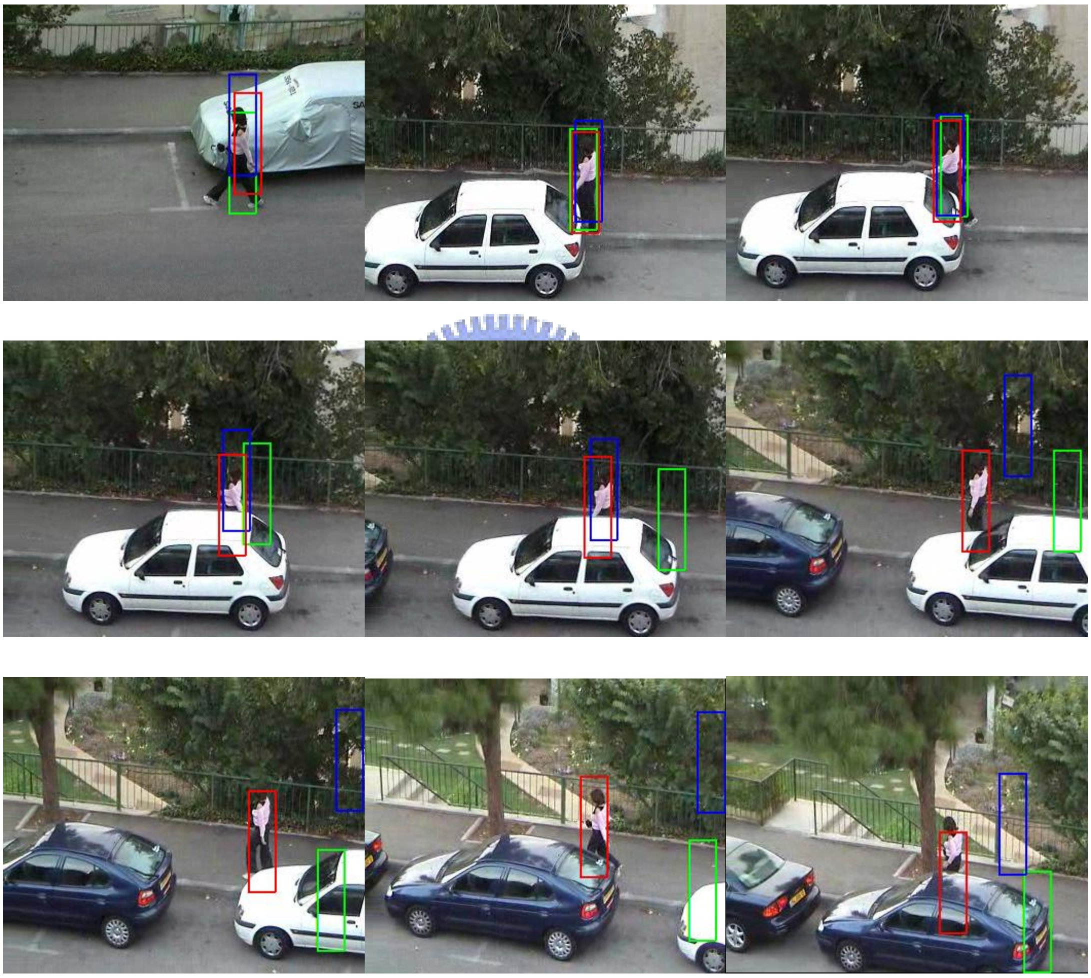


Figure 4.4-23 : Walking girl tracking results of spatial-color mean-shift trackers with rg feature and weighted-background in 3.6. Shown are frames 28, 106, 111, 124, 130, 153, 166, 196, .220. (red: tracker 1 with rg feature and weighted-background, blue: tracker 2 with rg feature and weighted-background, green: traditional mean-shift tracker)

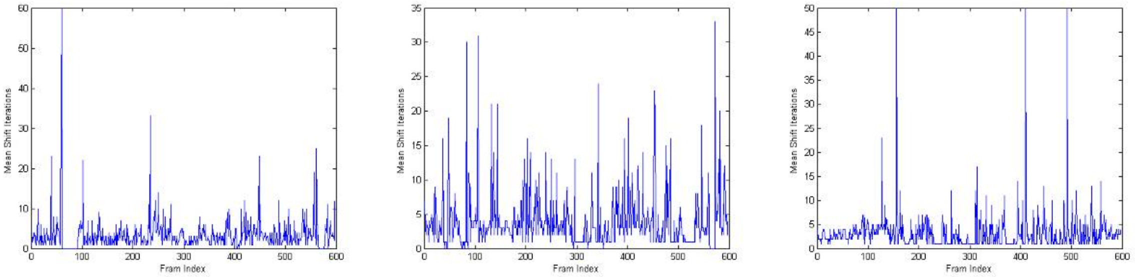


Figure 4.4-24 : Iteration numbers of walking girl sequence. (left: tracker 1 with rg feature and weighted-background, middle: tracker 2 with rg feature and weighted-background, right: traditional mean-shift tracker)

All trackers are placed in Figure 4.4-23 to be compared the performance and results. To sum up, the tracker 1 always captures the target girl under the circumstances of the variation of illumination and partial occlusion, but the tracker 2 and traditional mean-shift fail in the tracking process. The spatial-color mean-shift tracker 1 is the best tracker which we developed.

4.5 Spatial-Color Mean-Shift Trackers with Scale and Orientation

In this section, the scale and orientation method in 3.7 is applied. The procedure of Figure 3.8-1 is defined as the tracker 1 and the procedure of Figure 3.8-2 is defined as the tracker 2.

4.5.1 Walking Person Sequence

- Tracker 1

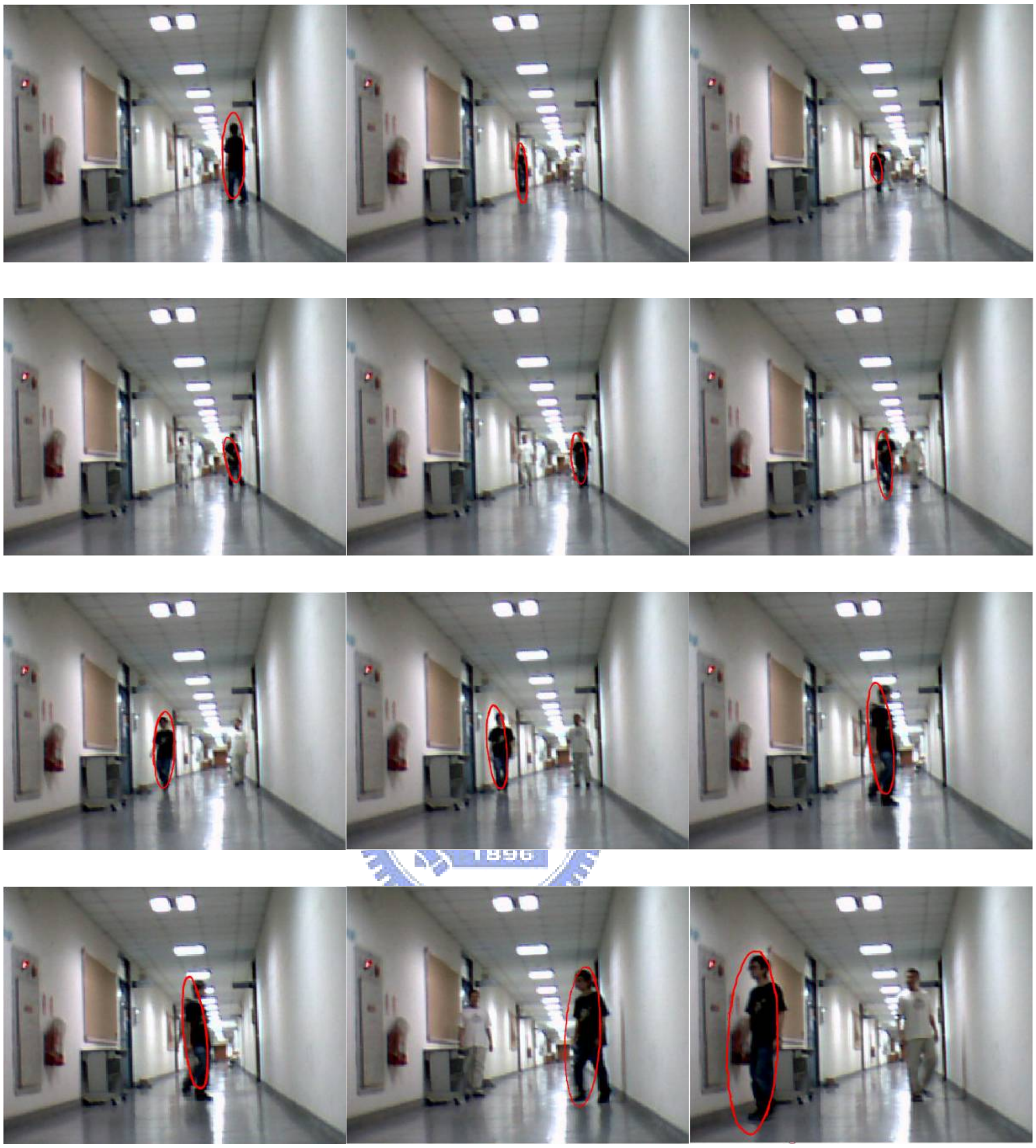


Figure 4.5-1 : Walking person tracking results of spatial-color mean-shift tracker1 with PCA scale method. Shown are frames 83, 358, 409, 494, 513, 598, 655, 689, 733, 854, 914, and 1000.

- Tracker 2

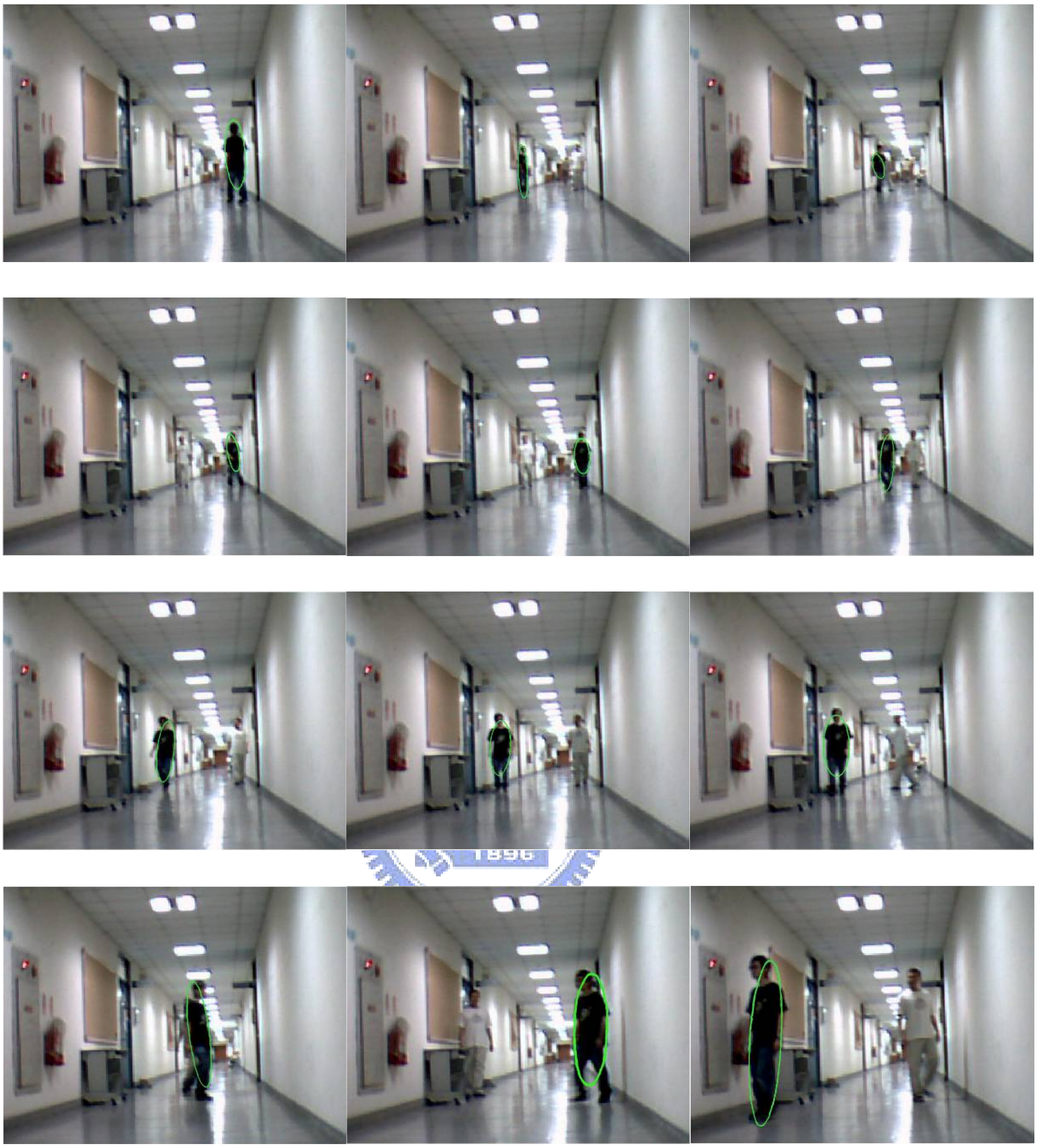


Figure 4.5-2 : Walking person tracking results of spatial-color mean-shift tracker2 with PCA scale method. Shown are frames 83, 358, 409, 494, 513, 598, 655, 689, 733, 854, 914, and 1000.

◆ Traditional Mean-Shift Tracker

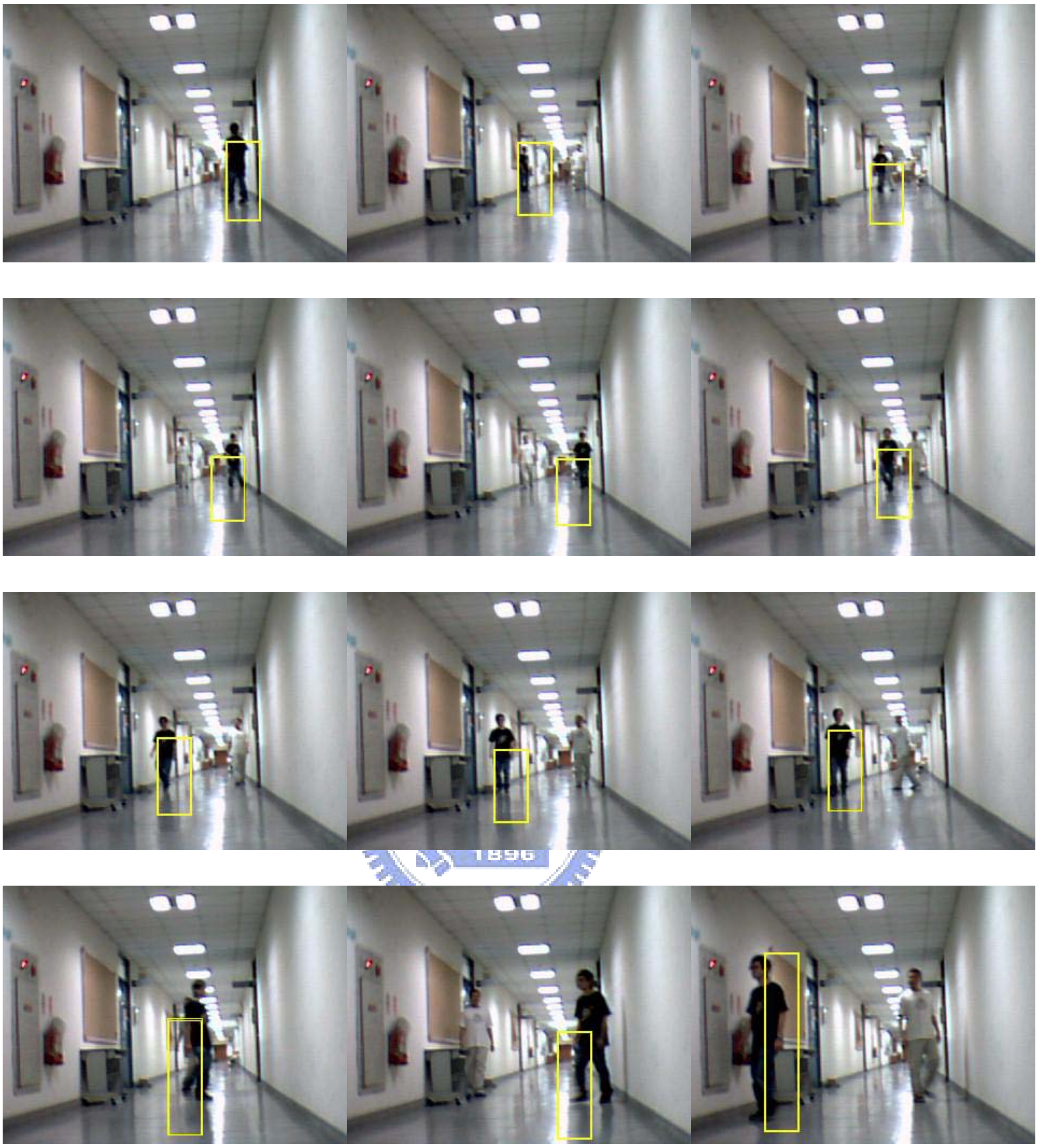


Figure 4.5-3 : Walking person tracking results of traditional mean-shift tracker with plus or minus 10 percent scale adaptation method. Shown are frames 83, 358, 409, 494, 513, 598, 655, 689, 733, 854, 914, and 1000.

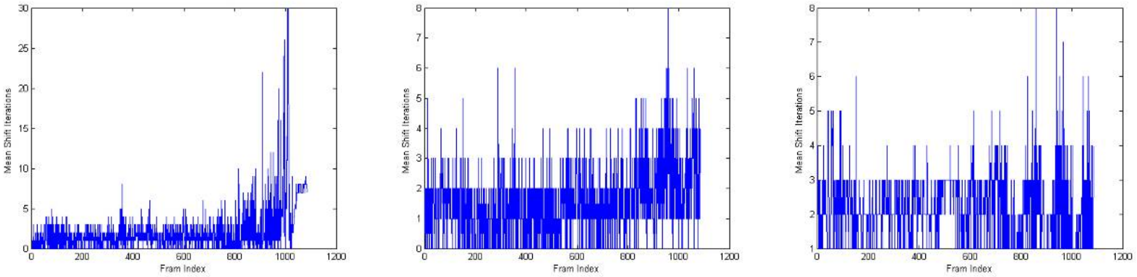


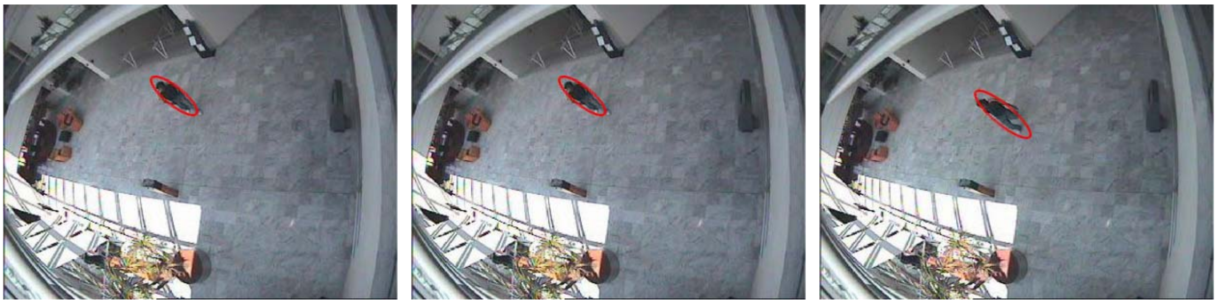
Figure 4.5-4 : Iteration number of walking person sequence. (left: tracker 1 with PCA scale method, middle: tracker 2 with PCA scale method, right: traditional mean-shift tracker with plus or minus 10 percent scale adaptation method)

Figure 4.5-3 shows the experiment results of traditional mean-shift tracker with plus or minus 10 percent scale adaptation method [1]. At each step of traditional mean-shift iteration, the mean-shift algorithm is run three times, once with current scale, and once with the window of plus or minus 10 percent of the current window size. The similarity measure Bhattacharyya coefficient is computed with different window sizes, and the window size yielding the largest Bhattacharyya coefficient is chosen as the current window size. In the tracking process, the tracker with traditional method can always capture the person, but the scale size is not accurate with the true one.

Figure 4.5-1 and Figure 4.5-2 are the spatial-color mean-shift trackers with PCA scale method which we proposed. With the person away from the camera and toward the camera, the two trackers capture the target at all times, and the scale size of the target is probably tracked, too. The proposed method is more robust and accurate than the traditional mean-shift method.

4.5.2 Surveillance Sequences

- ◆ Tracker 1



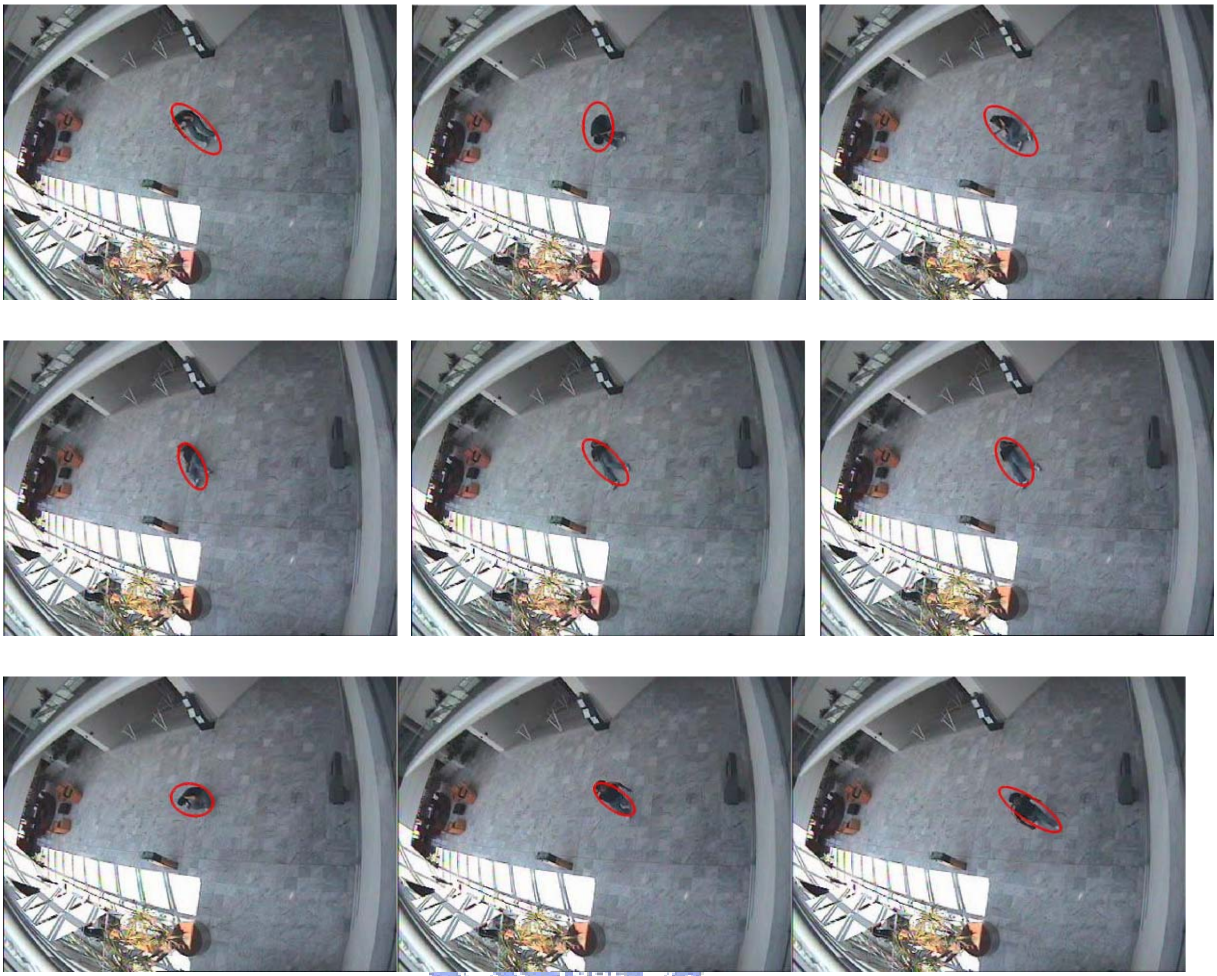
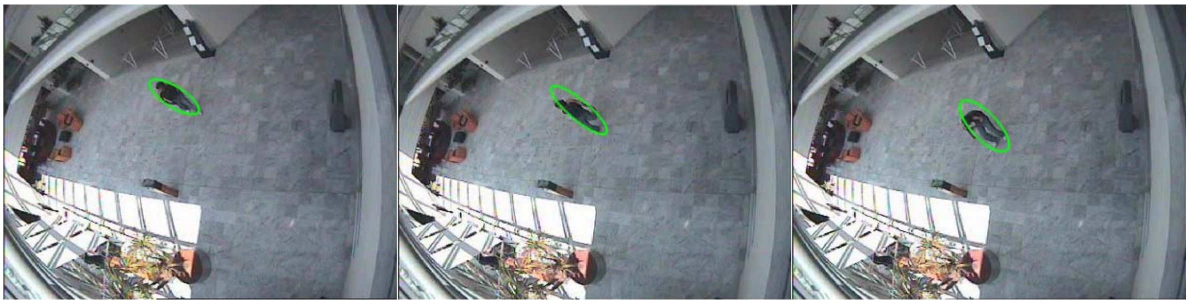


Figure 4.5-5 : Surveillance Tracking results of spatial-color mean-shift tracker1 with PCA scale method. Shown are frames 3, 16, 24, 32, 51, 59, 76, 127, 286, 318, 332, and 353.

- Tracker 2



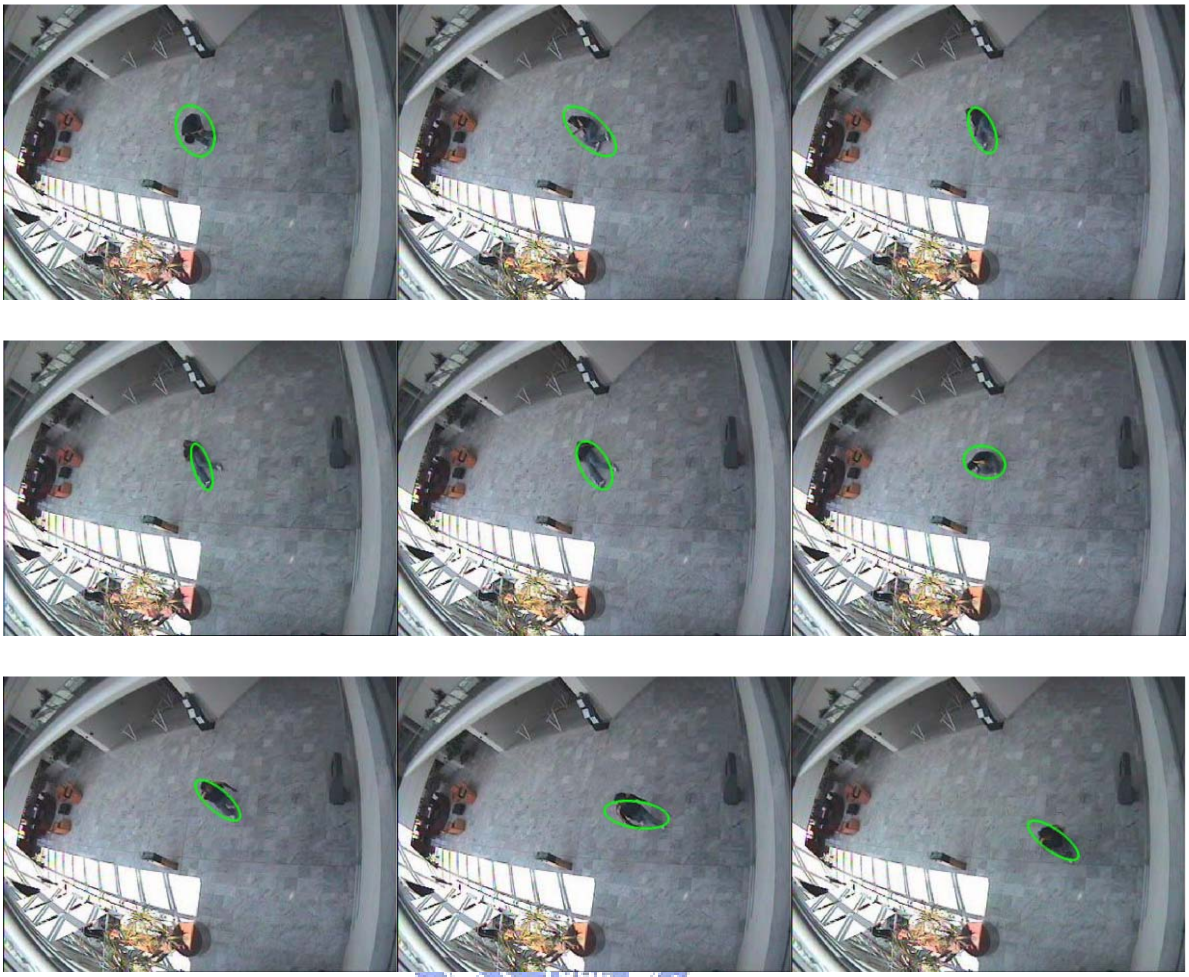
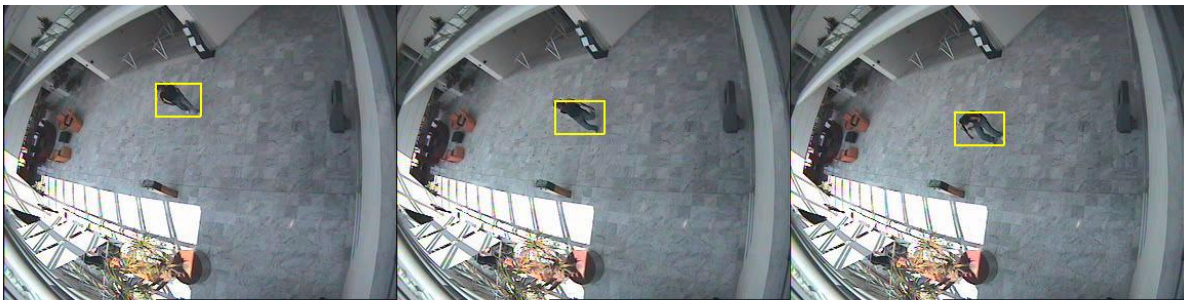


Figure 4.5-6 : Surveillance tracking results of spatial-color mean-shift tracker2 with PCA scale method. Shown are frames 3, 16, 24, 32, 51, 59, 76, 127, 286, 318, 332, and 353.

◆ Traditional Mean-Shift Tracker



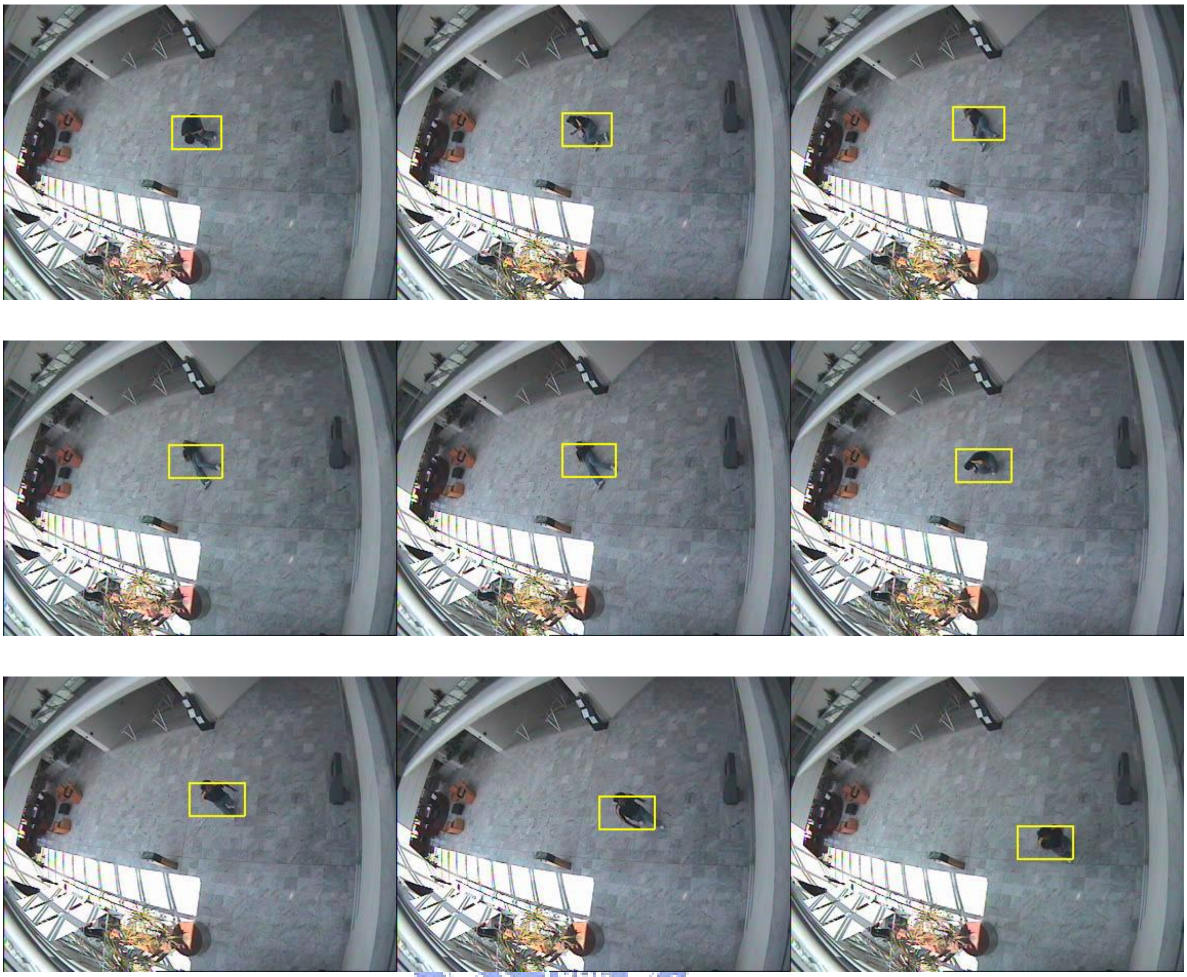


Figure 4.5-7 : Surveillance tracking results of traditional mean-shift tracker with plus or minus 10 percent scale adaptation method. Shown are frames 3, 16, 24, 32, 51, 59, 76, 127, 286, 318, 332, and 353.

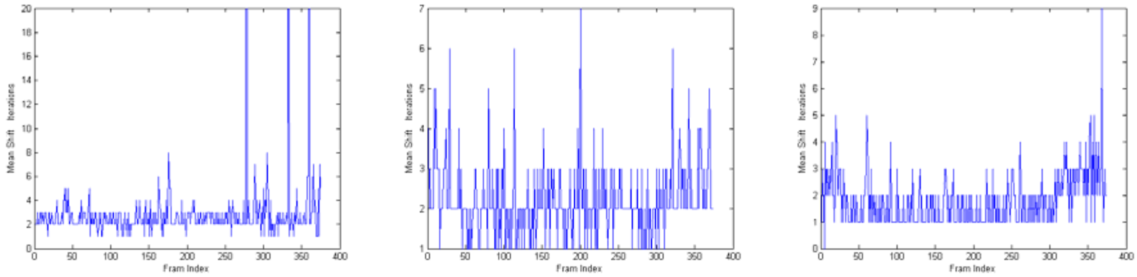


Figure 4.5-8 : Iteration number of surveillance sequence. (left: tracker 1 with PCA scale method, middle: tracker 2 with PCA scale method, right: traditional mean-shift tracker with plus or minus 10 percent scale adaptation method)

In surveillance sequence, a person walks, lies down, and finally stands up to keep walking. In these different actions, the target contains much deformation. Figure 4.5-5 and Figure 4.5-6 show that the trackers proposed in this thesis always track the target with the corresponding scale, orientation, and shape. Even though traditional mean-shift tracker captures the target at all times as shown in Figure 4.5-7, the scale size and orientation can not be fitted for the real one, the tracking window contains too many background information. By these sequences, the proposed trackers are more suitable for applying the surveillance applications than the traditional mean-shift tracker.

4.6 Performance Analysis

We have shown that the proposed trackers are more accurate than the traditional mean-shift tracker, and now we discuss the real time issue about the trackers. We separate the analysis into two parts. The first part is the preprocessing time of the model building, and the second part is the tracking time (iteration time).

The face sequence and cup sequence are used to test the performance of the proposed trackers. The models are built from the first image of these two sequences, and the preprocessing procedure is executed five times to obtain the average computing time. The tracking time of each iteration of the first 200 frames are shown in figures, and the average time of total frames is presented.

◆ Tracker 1

Table 4.6-1 shows the preprocessing time of tracker 1 with RGB feature and without weighted-background information according to the preprocessing procedure as shown in Figure 3.4-2, and Table 4.6-2 shows the preprocessing time of tracker 1 with rg feature and weighted-background information according to the preprocessing procedure as shown in Figure 3.8-1. The procedure of converting the RGB feature space to the rg feature space and computing the weighted-background information is added, so the preprocessing time as shown in Figure 3.8-1 is larger, but the average time is still small enough to let the tracking system be real time.

Table 4.6-1 : The preprocessing time of tracker 1 according to the procedure as shown in Figure 3.4-2.

	1	2	3	4	5	Average time
Face sequence	0.006384	0.006994	0.006821	0.006322	0.006766	0.006657
Cup sequence	0.005079	0.005184	0.005241	0.004497	0.005059	0.005012

Table 4.6-2 : The preprocessing time of tracker 1 according to the procedure as shown in Figure 3.8-1.

	1	2	3	4	5	Average time
Face sequence	0.030134	0.026233	0.027771	0.025795	0.029355	0.027858
Cup sequence	0.016165	0.015690	0.017430	0.018856	0.018387	0.017306

Figure 4.6-1 and Figure 4.6-2 show the iteration time of the first 200 frames of face sequence and cup sequence. The worst case is about 0.07 second for finishing one iteration of the face sequence, but the average time of total frames (about 2300 frames) is 0.035855 second (about 28 frames/sec). The average time of one iteration of total frames (about 1900 frames) of cup sequence is 0.017854 (about 56 frames/sec). The tracker 1 can achieve the standard of real time system.

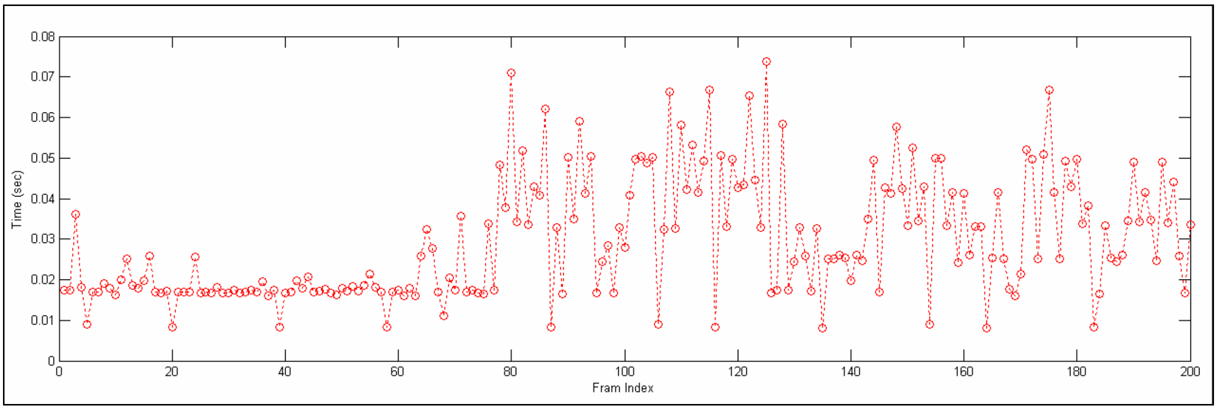


Figure 4.6-1 : The tracking time of the first 200 frames of face sequence of tracker 1.

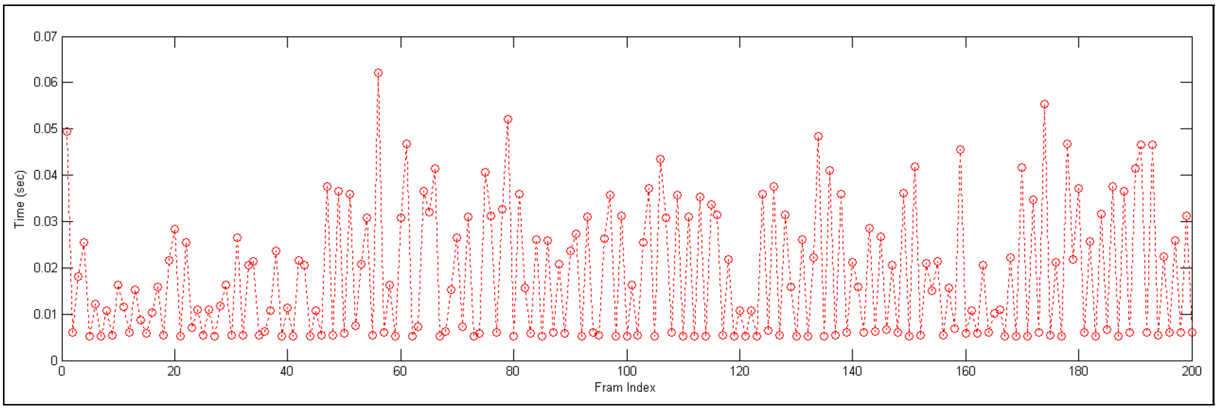


Figure 4.6-2 : The tracking time of the first 200 frames of cup sequence of tracker 1.

- ◆ Tracker 2

The same as tracker 1, the procedure of converting the RGB feature space to the rg feature space and computing the weighted-background information is added, so the preprocessing time as shown in Figure 3.8-2 is larger, but it still achieves the standard of real time system. The average time of preprocessing of tracker 2 is larger than that of tracker 1 because the preprocessing procedure of tracker 2 includes the computing of K_P and K_C.

Table 4.6-3 : The preprocessing time of tracker 2 according to the procedure as shown in Figure 3.4-3.

	1	2	3	4	5	Average time
Face sequence	0.023081	0.021131	0.021117	0.021489	0.023731	0.022110
Cup sequence	0.015975	0.015543	0.015505	0.015641	0.015808	0.015694

Table 4.6-4 : The preprocessing time of tracker 2 according to the procedure as shown in Figure 3.8-2.

	1	2	3	4	5	Average time
Face sequence	0.030712	0.032042	0.030352	0.032472	0.030917	0.031299
Cup sequence	0.020987	0.021176	0.021784	0.025734	0.022561	0.022448

Figure 4.6-3 and Figure 4.6-4 show the iteration time of the first 200 frames of face sequence and cup sequence. The average time of total frames (about 2300 frames) is 0.020670 second (about 48 frames/sec). The average time of one iteration of total frames (about 1900 frames) of cup sequence is 0.006608 (about 151 frames/sec). The tracking time of tracker 2 is smaller than that of tracker 1 because the procedure of tracking includes the computing of K_P and K_C, but the tracker 2 computes K_P and K_C in the procedure of preprocessing. The tracker 2 can achieve the standard of real time system.

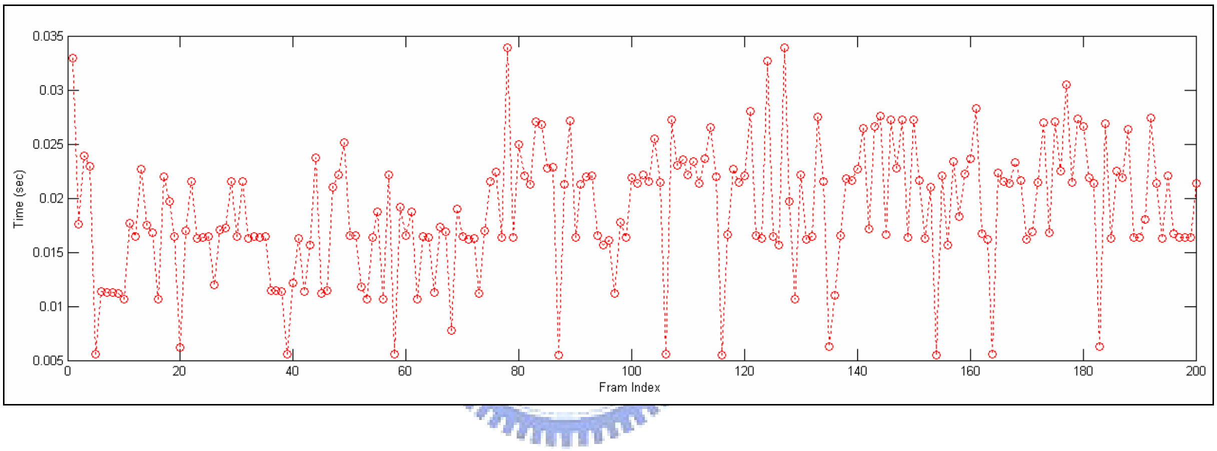


Figure 4.6-3 : The tracking time of the first 200 frames of face sequence of tracker 2.

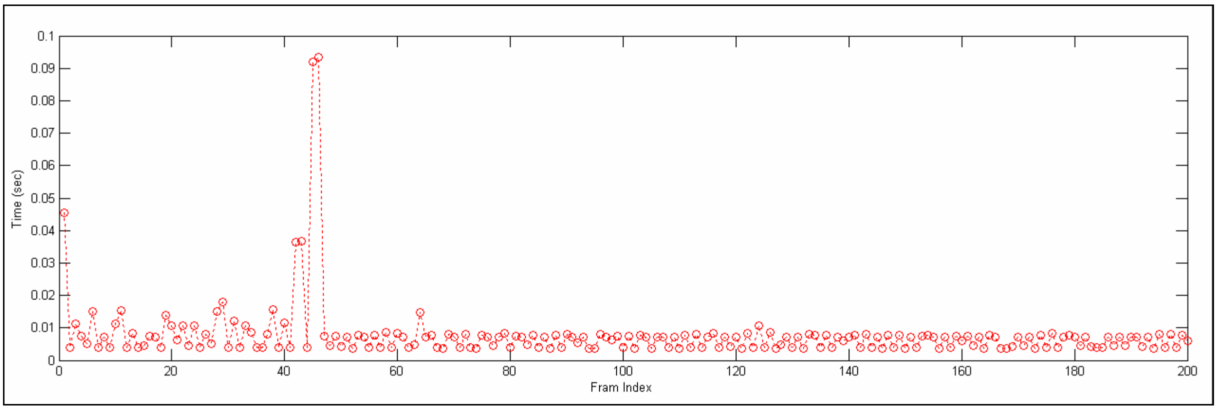


Figure 4.6-4 : The tracking time of the first 200 frames of cup sequence of tracker 2.

Chapter 5. Conclusion and Future Work

The spatial-color mean-shift object tracking algorithm is proposed in this thesis. Combining the spatial information with color feature makes the model contain more robust information. The new trackers can be derived from the new similarity measure of concept of the expectation of the estimated kernel density. Using the principal component analysis and the extension method, the scale size and the orientation of the target can be updated. The new iterative tracking algorithm can be summarized as Figure 3.8-1 and Figure 3.8-2.

The experiment results presented in chapter 4 show that the new trackers can track the target consistently, both in image location and in scale. The performance of tracking algorithm shown in Figure 3.8-1 is better than Figure 3.8-2, but these two new trackers are both much better than traditional mean-shift tracker under the different cases, such as face tracking, object tracking under complex background, and partial occlusion situation. The experiments results of scale and orientation show that the principal component analysis method is better than traditional scale adaptation method, and it can solve the problem of deformation. The performance analysis shows that the proposed trackers can achieve the standard of real-time system.

There are several areas for improvement. First, the issue of model update is not addressed in this thesis. Under what conditions the target histogram need to be updated is a difficult problem, because it requires one to detect whether an observed appearance change is due to the target changing appearance or a temporary occlusion. Second, the huge variation of illumination is another problem. The variation of illumination makes the appearance of target quite different from the original model, and it makes the trackers confused and tracking results fail.

References

[1] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 25(5):564-577, May 2003.

[2] F. Porikli, O. Tuzel, “Multi-kernel object tracking,” *IEEE International Conference on Multimedia and Expo*, pp. 1234-1237, 2005.

[3] V. Parameswaran, V. Ramesh, and I. Zoghlami, “Tunable kernels for tracking,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol.2, pp.2179-2186, 2006.

[4] R. Collins, “Mean-shift blob tracking through scale space”, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[5] K. She, G. Bebis, H. Gu, and R. Miller, “Vehicle tracking using on-line fusion of color and shape features”, *Proc. IEEE Conf. on Intelligent Transportation Systems*, 2004.

[6] Q. Zhao and H. Tao, “Object tracking using color correlogram,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[7] S. Birchfield and S. Rangarajan, “Spatiograms versus histograms for region-based tracking,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[8] H. Zhang, Z. Huang, W. Huang, and L. Li, “Kernel-based method for tracking objects with rotation and translation,” *Proc. International Conf. on Pattern Recognition*, 2004.

[9] Z. Zivkovic and B. Krose, “An em-like algorithm for color-histogram-based object tracking,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[10] F. Porikli and O. Tuzel, “Object tracking in low-frame-rate video,” *Proc. PIE/EI—Image and Video Communication and Processing*, San Jose, CA, 2005.

[11] T. Lindeberg, “Feature detection with automatic scale selection,” *International Journal of Computer Vision*, vol. 30(2), pp.79-116, November 1998.

[12] C. Yang, R. Duraiswami, and L. Davis, “Efficient mean-shift tracking via a new similarity measure,” *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 176-183, 2005.

[13] T. Hastie, R. Tibshirani, andJ. Friedman, “*The elements of statistical learning*,” Springer, 2001.

[14] http://www.cs.technion.ac.il/~amita/fragtrack/fragtrack.htm

[15] http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/

[16] D.W. Scott, “Multivariate Density Estimation,” New York: Wiley, pp.24-26, 1992.

[17] D. Comaniciu, V. Ramesh, and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” *Best Paper Award, IEEE Conf. Computer Vision and Pattern Recognition*, Hilton Head Island, South Carolina, vol.2, pp.142-149, 2000.

[18] D. Comaniciu, P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.24 n.5, pp.603-619, May 2002