SCENE ANALYSIS USING SCALE INVARIANT FEATURE EXTRACTION AND PROBABILISTIC MODELING

Yao Shen, M.S.

Dissertation Prepared for the Degree of

DOCTOR OF PHILOSOPHY

UNIVERSITY OF NORTH TEXAS

August 2011

APPROVED:

Bill Buckles, Major Professor
Parthasarathy Guturu,Co-major Professor
Kamesh Namuduri, Committee Member
Xinrong Li, Committee Member
Barrett Bryant, Chair of the Department of Computer Science and Engineering
Costas Tsatsoulis, Dean of the College of Engineering
James D. Meernik, Acting Dean of the Toulouse Graduate School

Shen, Yao. *Scene Analysis Using Scale Invariant Feature Extraction and Probabilistic Modeling*. Doctor of Philosophy (Computer Science and Engineering), August 2011, 98 pp., 11 tables, 27 figures, 119 reference items.

Conventional pattern recognition systems have two components: feature analysis and pattern classification. For any object in an image, features could be considered as the major characteristic of the object either for object recognition or object tracking purpose. Features extracted from a training image, can be used to identify the object when attempting to locate the object in a test image containing many other objects. To perform reliable scene analysis, it is important that the features extracted from the training image are detectable even under changes in image scale, noise and illumination. Scale invariant feature has wide applications such as image classification, object recognition and object tracking in the image processing area. In this thesis, color feature and SIFT (scale invariant feature transform) are considered to be scale invariant feature. The classification, recognition and tracking result were evaluated with novel evaluation criterion and compared with some existing methods. I also studied different types of scale invariant feature for the purpose of solving scene analysis problems. I propose probabilistic models as the foundation of analysis scene scenario of images. In order to differential the content of image, I develop novel algorithms for the adaptive combination for multiple features extracted from images. I demonstrate the performance of the developed algorithm on several scene analysis tasks, including object tracking, video stabilization, medical video segmentation and scene classification.

ACKNOWLEDGEMENTS

I would like to thank my advisors Dr. Bill Buckles and Dr. Parthasarathy Guturu for their continuous guidance during my doctoral work. It has been a great privilege to work with and learn from them. I would also like to thank Kamesh Namuduri and Dr. Xinrong Li for providing good suggestions for my research and work. I am grateful to the members of my doctoral committee for their time and comments to improve this dissertation.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

1.1 Feature Extraction

Scale invariant feature extraction is an essential part for the problem of scene analysis. The robust object feature should act as the signature which could help us to identify, locate and track the object in the image.

1.1.1 Color Feature Extraction

The color features have been widely used in many computer vision areas because of their invariance to rotation and scaling, and robustness to occlusions. The R,G and B alone is not enough in all circumstances, thus, HSV(hue-saturation-value) space representation should also be used. Moreover, rgb space has been considered more reliable when the illumination changes. Therefore, in my method, they are derived from 3 color spaces: i) RGB color space directly obtainable from the image, ii) HSV model that is important from the standpoint of human color perception, and iii) normalized rgb color spaces in which the r, g and b values of an image pixel are obtained by scaling the corresponding pixel R, G and B values with $R+G+B\neq0$;naturally, they lie in the range [0, 1]. Since many computer vision problems need to be robust against intensity variations, I do not include the V-component (intensity) of the HSV color space in my color feature set. Because the summation of r,g,b is equal to 1,it is not necessary to include b in my feature space. Thus, my color feature set turns out to be an 7-tuple {R, G, B, H, S, r, g}.

1.1.2 Texture Feature Extraction

Since the color features are not reliable when the illumination changes, texture features are sometimes used in conjunction with color features for better performance. Texture feature can be extracted by using several descriptors: co-occurrence matrices, Law's texture measures, wavelet packets and Gabor filters. Here, I will choose Gabor filters because they can easily be tuned to different scales and orientations of texture as necessary. The energy distribution in the frequency domain of a Gabor filtered image is used to identify textures. Through the Gabor filter, in its general form, is obtained by multiplication of a complex sinusoid and a Gaussian function, I use the following simpler version in which a simple harmonic function replaces the complex sinusoid:

$$G(x, y, f, \theta) = \exp\left\{-\frac{1}{2} * \left[\left(\frac{x'}{sx'}\right)^2 + \left(\frac{y'}{sy'}\right)^2\right]\right\} * \cos\left(2 * pi * f * x'\right) \quad (1)$$

where

$$x' = x * \cos(\theta) + y * \sin(\theta) \quad (2)$$

and

$$y' = y * \cos(\theta) - x * \sin(\theta) \quad (3)$$

θ represents the orientation of the normal to the parallel stripes of a Gabor function, f is the frequency , sx' and sy' denote the variances along x and y axis, respectively.

The Gabor features from an image $I(x, y)$ can be extracted via the convolution

$$\psi(x, y, f, \theta) = G(x, y, f, \theta) * I(x, y) \quad (4)$$

The global feature can be obtained by summing over the response of an image

$$\phi(f, \theta) = \sum_x \sum_y \psi(x, y, f, \theta) \quad (5)$$

Global feature $\phi(f, \theta)$ can be considered as a histogram of the responses in the image with different frequencies f and orientations θ. This global feature is also translation invariant, since it is formed by adding the responses in whole image.

Since edges appear in high frequency, a properly selected frequency is needed to represent the histogram of lines in different orientations. Therefore, I can construct the feature vector as

$$\phi = \{\phi(f, \theta_0), \phi(f, \theta_1), \cdots, \phi(f, \theta_{n-1})\} \quad (6)$$

where

$$\theta_k = \frac{k\pi}{n}, \ k = \{0, \cdots, n-1\} \quad (7)$$

By normalizing this feature vector, a scaling invariant feature vector can be constructed, because the response magnitudes is affected by the scale of the object, but the ratio between magnitudes will not be changed for different scale.

Moreover, the normalization also adds the illumination invariance to the feature vector.

1.1.3 SIFT Feature Extraction

The motivation for the choice of SIFT features for this application stems from the prior evidence that the SIFT features of the grey-level intensity images have been used quite successfully in the bag-of-feature approaches to general scene and object categorization (see e.g. [13]). The SIFT has been designed for extracting highly discriminative local image features that are invariant to image scaling and rotation, and partially invariant to changes in illumination and viewpoint. The key-points in SIFT are derived by considering the extrema by convolving the DOG (difference of Gaussians) filter at multiple scale with an input image I(x; y) as follows:

$$\mathrm{D}(\mathrm{x},\mathrm{y},\sigma) = (\mathrm{G}(\mathrm{x},\mathrm{y},\mathrm{k}\sigma) - \mathrm{G}(\mathrm{x},\mathrm{y},\sigma) * \mathrm{I}(\mathrm{x},\mathrm{y}) \tag{8}$$

$$\approx (\mathrm{k}-1)\sigma^2\nabla^2\mathrm{G}(\mathrm{x},\mathrm{y}) * \mathrm{I}(\mathrm{x},\mathrm{y}) \tag{9}$$

where

$$\mathrm{D}(\mathrm{x},\mathrm{y},\sigma) = \frac{1}{2\pi\sigma^2}\mathrm{e}^{-(\mathrm{x}^2+\mathrm{y}^2)/2\sigma^2} \tag{10}$$

In the above equation, x and y are image pixel coordinates, σ is the standard deviation of the Gaussian filter, and k is a constant, which may be chosen as $\sqrt{2}$. Since convolution an image with Gaussians with different values of σ produces different levels of smoothing, σ may be considered as a scale parameter defining the DOG filier outputs. The keypoints of the images are then selected to be those with pixel values that are the extrema (maxima or minima) among the eight neighboring pixels at the same scale (σ value) and the nine pixels in each of the corresponding 3x3 pixel windows in the adjacent scales as shown in Fig. 1.1. Due to their selection from DOG filter outputs at

multiple scales, the keypoints turn out to be scale invariant. Once the key-points have been identified, the gradient directions of pixels in the vicinity of each key-point are separated into 36-bins, where each bin covers a 10^{o} range, and each sample in the bin is weighted by the corresponding gradient magnitude. One or more orientations corresponding to the bins with the highest bin value or within 80% of the highest value are assigned to a key point. Computation of key-point orientation(s) is folloId by the computation of key-point descriptors. For this, a 16x16 pixel window around each key-point is selected and segmented into 16 4x4 sub-windows as shown in Fig. 1.2(a). The pixel gradients in each sub-window are then accumulated with their magnitude values into 8 bins as shown in Fig. 1.2(b). Since there are 16 sub-windows and 8 bins per window, each key-point descriptor turns out to be 128-dimensional feature vector. Rotation invariance is achieved by using the pixel gradient directions relative to the key-point orientation(s) rather than the absolute direction values. However, since a key-point may occasionally have more than one orientation, its descriptors could also be multiple.

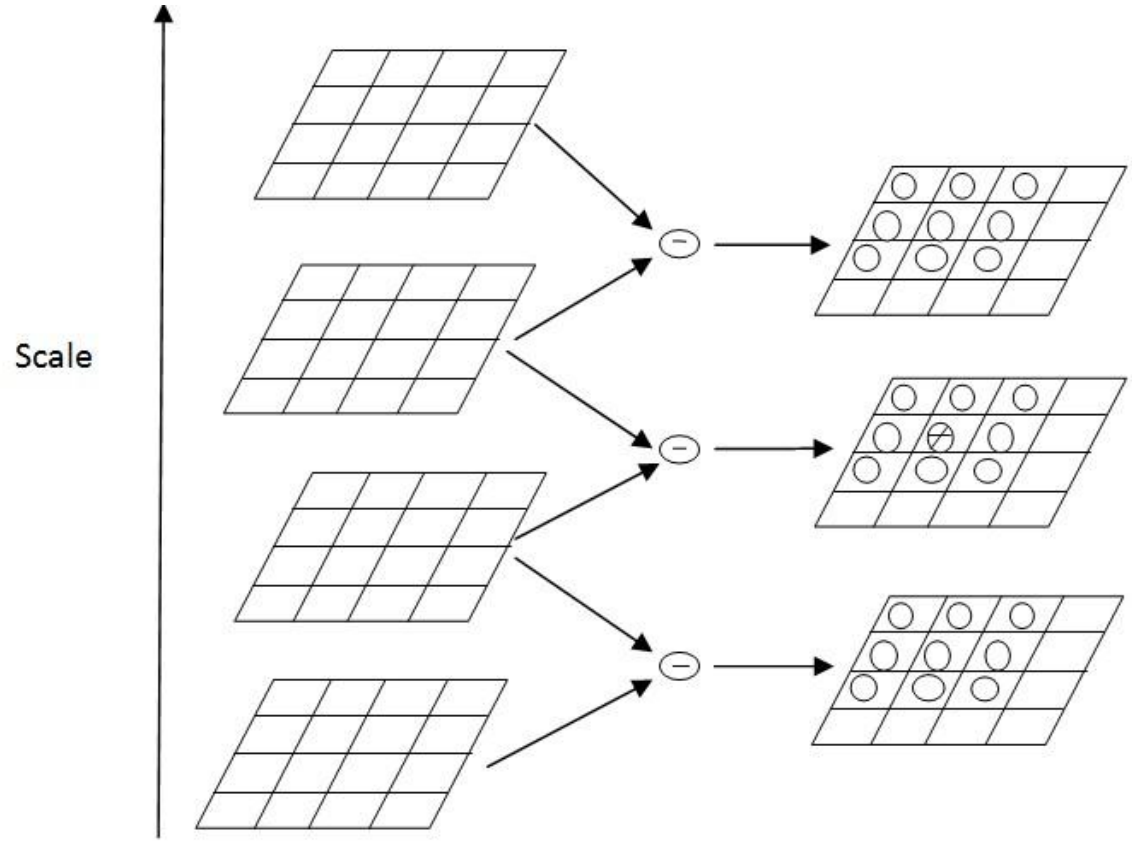
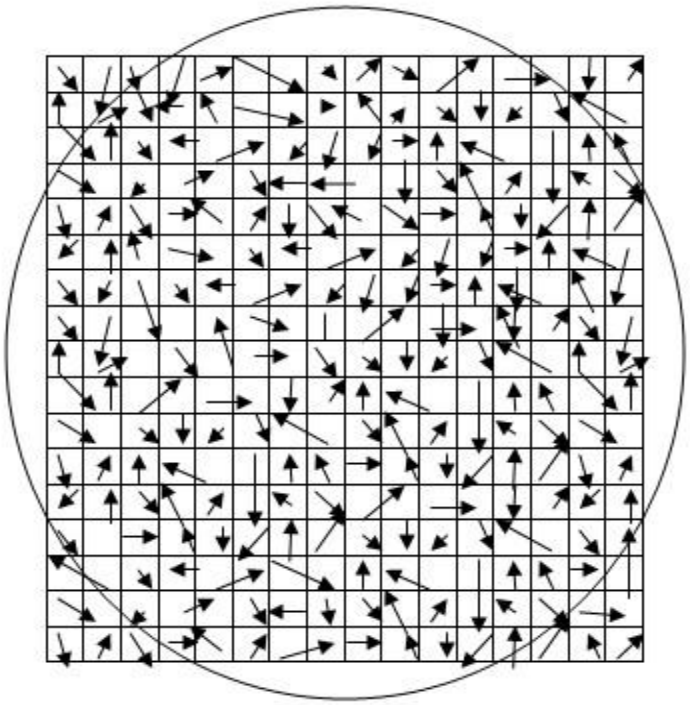
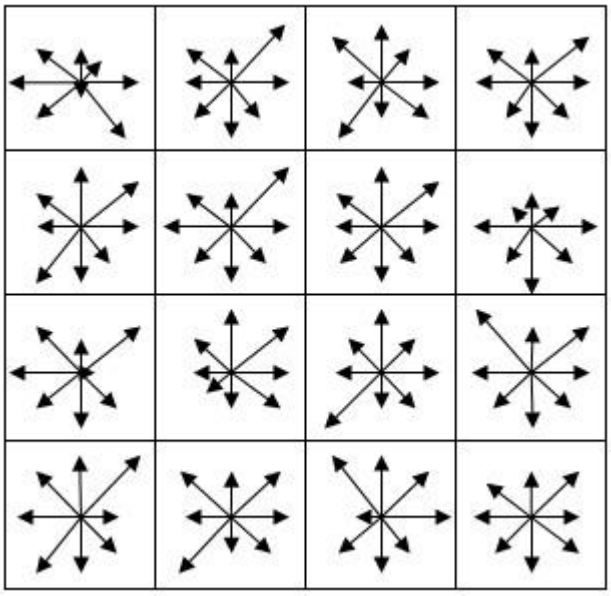


Figure 1.1. Convolution an image with Gaussians with different scale values. The key-points of the images are selected to be those with pixel values that are the extrema (maxima or minima) among the eight neighboring pixels at the same scale (σ value) and the nine pixels in each of the corresponding 3x3 pixel windows in the adjacent scales



(a) Image Gradients



(b) Keypoint Descriptor

Figure 1.2. SIFT feature extraction

1.2 Probabilistic Analysis

1.2.1 Particle Filter

Traditionally, motion estimation of current frame is computed either with its previous adjacent frame or with a predefined reference frame. Unfortunately, both have their drawbacks. Motion estimation based on the previous adjacent frame suffers from the accumulative error problem while a single reference frame based approach may lead to extraction of unreliable feature points in a long sequence video. Thus, to cope with the limitations of the above methods, I propose a novel adaptive particle filter algorithm to refine the motion estimation. This algorithm takes over the initial motion parameters between adjacent frames computed by RANSAC algorithm and minimizes the motion estimation errors with respect to the reference frame in the parameter correction step, thus, solving both the accumulative error and unreliable feature problems simultaneously.

Kalman Filters are widely used to estimate the global motion vector due to its efficiency and accuracy. However, as indicated earlier, the traditional Kalman Filter is limited by its linear system and Gaussian noise assumptions. Since in most real applications, a shaky camera cannot be modeled as a linear system nor may process and measurement noise be Gaussian. The PF [4] provides greater efficiency and extreme flexibility in solving non-linear and non-Gaussian problems (including even multi-modal probability density functions) by using the concept of important sampling wherein the intractable integrals in the optimal Bayesian solution for estimating the current state from the past observations, are replaced by discrete sums of weighted samples drawn from the posterior distribution. Mathematically, global motion estimation

is the prediction of the system state S_t with state transition and observation models as follows:

$$S_t = f_t(S_{t-1}, U_t) \tag{11}$$

$$Y_t = h_t(S_t, V_t) \tag{12}$$

Here, Y_t denotes the observation parameter set, S_t the system state, U_t the process noise, and V_t the measurement noise, all at time step t. Now, under the assumptions of first order Markov model for state transition and conditional dependence of Y_t exclusively on S_t, motion estimation problem may be formulated as a recursive Bayesian estimation problem to compute the posterior probability $p(S_t|Y_{1:t-1})$ where $Y_{1:t-1}$ denotes the observation parameter sets from start to the time step $t-1$. The PF is a simple approach to the complex problem of estimating $p(S_t|Y_{1:t-1})$ by using a set of $P_t = \{S_t^i; w_t^i\}_{i=1,\ldots N}$ of N weighted particles S_t^i (with their weights w_t^i summing up to unity) drawn from the so called proposal (or importance) distribution $\pi(S_t)$, which is an approximation of $p(S_t|Y_{1:t-1})$. With these weighted samples, $p(S_t|Y_{1:t})$ may be approximated as follows:

$$p(S_t|Y_{1:t-1}) \approx \sum_{i=1}^{N} \delta(S_t - S_t^i) \tag{13}$$

In this equation δ refers to the Dirac delta function. It has been shown in [1] that the weights w_t^i for the above process called Sample Importance Sampling (SIS) can be computed using the iterative formula:

$$w_t^i = w_{t-1}^i \frac{p(Y_t|S_t^i)p(S_t^i|S_{t-1}^i)}{q(S_t^i|S_{1:t-1}^i, Y_{1:t-1})} \tag{14}$$

The popular choice for $q(S_t^i|S_{1:t-1}^i, Y_{1:t-1})$ is $p(S_t^i|S_{t-1}^i)$ possibly because it yields a simple Weight iteration formula:

$$w_t^i = w_{t-1}^i p(Y_t|S_t^i) \tag{15}$$

1.2.2 Probabilistic Latent Semantic Analysis

Probabilistic Latent Semantic Analysis (pLSA) is a generative model based on sound statistical foundation for addressing the challenging problem of semantic content analysis of natural language or text documents. Mathematically, the problem here is to analyze each of M documents $\mathrm{d_j} \in \mathrm{D} = \{\mathrm{d_1}, \ldots, \mathrm{d_M}\}$ containing some words $\mathrm{w_i} \in \mathrm{W} = \{\mathrm{w_1}, \ldots, \mathrm{w_N}\}$, where N is the total number of words in a vocabulary W. The analysis process is to find the latent aspects or classes (or simply topics, such as sports, politics, etc.) $\mathrm{z_k} \in \mathrm{Z} = \{\mathrm{z_1}, \ldots, \mathrm{z_K}\}$ in the documents. Since some words (called polysemous words) could have different meanings in different documents depending upon their latent aspects, it is not possible to decipher the semantic contents of documents simply from the words they contain. Analysis is further complicated by the fact that a document $\mathrm{d_j}$ could be a mixture of latent aspects. The power of pLSA stems from its approach to decomposition of a document into a mixture of latent aspects defined by a multinomial

distribution over the words in the vocabulary. In the present context, d_j corresponds to a WCE video frame, and D, to the total video. The z_ks are the parts (e.g. small intestine) of digestive tract captured by the video frame, and w_is are visterms or quantized feature vectors, obtained by the vocabulary building process in section 2. Since the video is to be segmented into four parts corresponding to the four regions of the digestive tract, the value of K here is 4.

With the above mapping of my problem variables onto the generic pLSA problem variables, I can proceed with the Hofmann's pLSA formulation of my problem. Suppose now that each video frame d_j is a mixture of latent aspects, defined by the multinomial distribution$P(z_k|d_j)$. Let $P(w_i|z_k)$ be the multinomial distribution for aspect z_k. The joint probability between a word (visterm) and image can now be defined by the symmetric mixture model as follows:

$$P(w_i, d_j) = \sum_{k=1}^{K} P(z_k)P(w_i|z_k)P(d_j|z_k) \quad (16)$$

This model is said to be symmetric because the probability $P(z_k|d_j)$ that a document d_j contains latent class z_k is assumed to be the same as the probability $P(d_j|z_k)$ that the latent class z_k influences the document d_j.

Since the aspect attribution is latent and hence not observable, an expectation maximization (EM) algorithm could be used for estimation of the parameters $P(z_k), P(w_i|z_k)$ and $P(d_j|z_k)$ iteratively from the observed data by maximizing the following log-likelihood function:

$$L = \sum_{i=1}^{M} \sum_{j=1}^{N} n(w_i, d_j) \log P(w_i, d_j) \quad (17)$$

Where $n(w_i, d_j)$ denotes the times the visterm w_i occurred in an image d_j .

The two steps of the EM algorithm are the following:

In the E-step, the conditional probability distribution of the latent aspect z_k is computed as follows:

$$P(z_k | w_i, d_j) = \frac{P(z_k) P(w_i | z_k) P(d_j | z_k)}{\sum_{l=1}^{L} P(z_l) P(w_i | z_l) P(d_j | z_l)} \quad (18)$$

In the M-step, the topic probability $P(z_k)$, the visterm probability $P(w_i | z_k)$ and the image probability $P(d_j | z_k)$ are updated according to the new expected value $P(z_k | w_i, d_j)$.

$$P(z_k) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} n(w_i, d_j) P(z_k | w_i, d_j)}{\sum_{i=1}^{M} \sum_{j=1}^{N} n(w_i, d_j)} \quad (19)$$

$$P(w_i | z_k) = \frac{\sum_{j=1}^{N} n(w_i, d_j) P(z_k | w_i, d_j)}{\sum_{m=1}^{M} \sum_{j=1}^{N} n(w_m, d_j) P(z_k | w_m, d_j)} \quad (20)$$

$$P(d_j | z_k) = \frac{\sum_{i=1}^{M} n(w_i, d_j) P(z_k | w_i, d_j)}{\sum_{m=1}^{M} \sum_{j=1}^{N} n(w_m, d_j) P(z_k | w_m, d_j)} \quad (21)$$

The EM algorithm starts with random values (that sum to 1 over the respective variable ranges) for $P(z_k), P(w_i | z_k)$ and $P(d_j | z_k)$. After each EM iteration defined by the equations (5) through (8), the likelihood value L defined by the equation (3) is computed, and if it is better than the previously computed maximum value, the new value of L

replaces the old one. The whole problem configuration, particularly the $P(d_j|z_k)$ values are also saved. After a number of iterations when the algorithm converges, and no further improvements in L could be obtained, the most dominant aspect of each document d_j is ascertained to be z_k where $P(d_j|z_l)$is maximum for l=k.

CHAPTER 2

OBJECT TRACKING

In the last few years, there has been much work applying target recognition and tracking for traffic monitoring. The status of traffic flow can be determined by tracking and counting the number of vehicles existing in the scene. Some traffic violation and traffic accident can also be detected by the estimation of the vehicle direction and speed using vehicle identification and tracking mechanism.

Historically, target detection has been performed on single images or static imagery [1, 2]. More recently, however, video streams have been exploited for target detection [3, 4, 5]. Many methods like these, are computationally expensive and are inapplicable to real-time applications, or require specialized hardware to operate in the real-time domain. However, methods such as Pfinder [4] and Beymer et al [6] are designed to extract targets in real-time. Recognizing and tracking objects in images taken by moving camera is much more challenging than traditional tracking with a stationary camera. In [7][8][9], authors tried to identify the objects using their symmetry information. In [10][11][12], optical flow method is developed to locate and track the object.

Tracking with a single camera in 2D space is used in a wide range of civilian and military applications. Consequently, researchers have concentrated on various tracking algorithms specific to different applications. Kalman filter (KF) [35] is an efficient algorithm to estimate the state of a dynamical system, which is assumed to be a linear system operating in Gaussian noise [26]. The extended Kalman filter (EKF) [36] addresses the limitations of KF by using non-linear functions and non-Gaussian noise

models. Since the non-linear functions are applied only to the sample means iteratively, the EKF may diverge quickly if either the initial state estimation or the process model is incorrect. Isard and Blake [28] proposed a more powerful particle filter (PF) tracker, which is considered to be both efficient and flexible in solving non-linear and non-Gaussian problems [37, 38].

2.1 Feature Extraction and Matching of Regions based on Feature Distribution

Efficacy of a tracking algorithm depends on the accuracy with which the supposedly corresponding object regions in two consecutive frames (called as the reference and target frames in the sequel) could be matched based on the selected features. The feature set in my case consists of 13 features (6 texture and 7 color based features) all normalized to lie in the range [0, 1]. It is Ill known from pattern recognition theory that larger number of features, particularly with high correlations, does not necessarily result in better recognition (matching) accuracy. On the other hand, higher dimensional feature vectors cause what is called curse of dimensionality according to Bellman and result in poor classifier performance. Hence, a feature extractor (FE) that reduces the dimensionality of the feature space is essential for good pattern matching. A requirement for this FE is that the extracted features contain substantial information for discrimination between the foreground and background pixels. It is possible to construct a FE satisfying this requirement from two scatter matrices: i) the within-class scatter matrix S_w which is the covariance variance of the feature vectors of the two (foreground and background) classes of observed pixels centered around their respective class means, and ii) the between-class scatter matrix S_w which the covariance matrix of the class mean vectors centered around the overall sample mean μ.

Now, if $A = {S_w}^{-1}S_b$, the known feature extractor (called the Fisher's linear discriminated function) is given by the dominant few eigenvectors of the matrix e in the eigenvalue equation:

$$Ae = \Lambda e \tag{22}$$

In the above, Λ is the eigenvalue matrix of the system. The pixels in the reference and target frame observation windows can be separated into foreground and background pixel classes depending upon whether they are on the periphery or in the interior of the observation windows. The two sample mean vectors μ_f and μ_g for the foreground and the background classes can be computed from the feature vectors of the corresponding pixels. The matrix S_w can then be constructed using the pattern vector samples with their respective class means removed (subtracted) from them. The matrix S_b can also be constructed from 2 samples: $(\mu_f - \mu)\,(\mu_f - \mu)^T$ and $(\mu_b - \mu)\,(\mu_b - \mu)^T$, where T denotes the transpose of the column matrices. Since, as shown in, the rank of a sample covariance matrix is at most equal to $N_s - 1$ where N_s is the number of sample used for construction of the sample matrix, the rank A and consequently the number of its significant eigenvalues is usually determined by the rank of S_b. Thus, in my case, it is enough to determine just first three dominant eigenvectors of A, and use them to linearly translate the 13-dimensional feature vectors in the two observation windows into 3-dimensional feature vectors.

For computational efficiency, a faster dominant eigenvector computation method rather than a full-fledged eigenvalue analysis is used in my implementation.

Next step in tracking is to compute a similarity measure for matching the images of the object in the two frames. This measure needs to be computed separately for each one of the three feature vector components in the reduced feature space, and aggregated. I present the aggregation (or feature fusion) procedure later in section 4, but discuss here the procedure for computation of a similarity measure based on the distributions of the values for a specific feature vector component in the regions being matched. The first step in this procedure, which is common to all the 3 components, is to quantize the possible feature values in the regions into L levels. Let $\mathrm{L}(\mathrm{x_i})$ be the function that maps the feature value of the pixel located at $\mathrm{x_i}$ to a level $\mathrm{l} \in [1, \mathrm{L}]$. Now the discrete feature probability distribution $\mathrm{p_t}(\mathrm{x}) = \{\mathrm{p_t^l}(\mathrm{x})\}_{\mathrm{l}=1\ldots\mathrm{L}}$ for pixels within a bounding ellipse, that is, an elliptic region centered at x and supposedly enclosing the object under focus fully, may be obtained by computing the individual components as follows:

$$\mathrm{p_t^l}(\mathrm{x}) = \mathrm{C}\sum_{\mathrm{i}=1}^{\mathrm{N_R}} \mathcal{K}\left(\frac{\|\mathrm{x}-\mathrm{x_i}\|}{\mathrm{h}}\right)\delta\left(\mathrm{L}(\mathrm{x_i}) - \mathrm{l}\right) \tag{23}$$

where $\mathrm{x_i}$ is the location of a pixel within the ellipse, $\mathrm{N_R}$ is the number of such pixels, $\mathcal{K}$ is a kernel function to be discussed shortly, δ is the Kronecker delta function, $\|.\|$ is the Euclidean norm, and h and C are the normalization constants that ensure the argument of the function $\mathcal{K}$ and the sum, respectively, to be within unity. A choice of h that is adaptable to changes in scale (i.e., size of the ellipse) is $\mathrm{h} =$

$\sqrt{a^2 + b^2}$ where a and b are half lengths of the major and minor axes of the ellipse. An obvious choice for the constant C is as follows:

$$C = \frac{1}{\sum_{i=1}^{N_R} \mathcal{K}\left(\frac{\|x - x_i\|}{h}\right)} \tag{24}$$

Now, finally, the choice of the kernel function is based on the idea that the feature information of the pixels farther away from the center of the ellipse should be given lesser Weight because they are more likely to be background pixels. This consideration yields the following simple kernel function:

$$\begin{aligned} \mathcal{K}(r) &= 1 - r^2 \qquad \text{if} \quad r < 1 \\ &= 0 \qquad\quad \text{if} \quad r \geq 1 \end{aligned} \tag{25}$$

The normalizing constant h in the equation (24) ensures that the value of radial distance r is less than 1 for all the pixels within the ellipse.

Once the feature probability distributions for the feature indexed by m in the feature vector is obtained using the equation (24) for two regions located at x_{tgt} and x_{ref} the target and reference frames, respectively, the dissimilarity (or distance) between the two distributions may be computed as using the Bhattacharyya distance defined as follows:

$$d_m = \sqrt{1 - \rho(p_t(x_{tgt}), p_t(x_{ref}))}$$

$$= \sqrt{1 - \sum_{i=1}^{L} \sqrt{p_t^l(x_{tgt}) p_t^l(x_{ref})}} \quad (26)$$

The sum term yielded by the function ρ in the equation (26) is known as the Bhattacharyya coefficient. It is a measure of similarity between the distributions.

2.2 Unscented Particle Filter for Effective Tracking and Feature Fusion

Tracking, in general, is a problem of estimating the current system state (in my case, vector of parameters characterizing object position, scaling, rotation, etc.) using the knowledge of observation parameters (in my case, the image features) from start to the previous time step. Thus, tracking could be solved by using particle filter framework for the estimation of the object position internationally given the initial position in the first frame.

As mentioned above, the popular choice for $q(S_t^i | S_{t-1}^i, Y_{1:t-1})$ is $p(S_t^i | S_{t-1}^i)$ possibly because it yields a simple Weight iteration formula: $w_t^i = w_{t-1}^i \, p(Y_t | S_t^i)$.

The main drawback of this choice, however, is that it does not incorporate the most recent observation Y_t and hence produces inaccurate estimates. MerI et al. have shown that the accuracy of tracking with the PF algorithm can be improved successively by using the KF, EKF and UKF algorithms, respectively, for proposal generation. Specifically, the proposal distribution for each particle may be chosen as

$$q(S | S_{1:t-1}^k, Y_{1:t}) = \mathcal{N}(\bar{S}_t^k, \Sigma_t^k) \quad (27)$$

where $\bar{S}_t^k$ and Σ_t^k are the mean and covariance matrix of S_t^k, computed using the UKF algorithm, and $\mathcal{N}$ is the Gaussian distribution. The superscript k here refers to the

index of the feature vector component based on which the particle generated. In my scheme, I generate a number of particles for each feature in the observed feature vector. To address the problem of degeneracy (stochastically increasing variance of the importance weights resulting in all but one particle having negligible weights) in the SIS algorithm of PF, I adopt, on observation of significant degeneracy, the standard Sample Importance Resampling (SIR) procedure wherein the weights of N_s unequally weighted particles are all reset to $w_t^i = 1/N_s$.

An effective algorithm for fusion of information from multiple disparate features is a significant component of a good tracker. If I somehow have information regarding the relative figures of merit for individual features on a dynamic basis during tracking, it will not be difficult to combine, using a weighted scheme, the similarity measures obtained as indicated in section 3 for these features . However, automatic evaluation of the figures of merit for the features is really a challenging task. Fortunately, in the PF framework, the likelihoods of the features can be considered as their figures of merit in the latest state. Therefore, it is intuitive to design the feature combination method as an integral part of the UPF method. In order to do this, I just need to simply build the likelihood by integrating all the likelihoods of features in the latest state as follows:

$$p(Y_t|S_t) = \sum_{m=1}^{M} \beta_{m,t}\, p(Y_t^m|S_t) \tag{28}$$

where $p(Y_t^m|S_t)$ is the likelihood for feature m in state S_t at time step t,and M is the total number of features. The mixture coefficients $\beta_{m,t}$ related to the M features are normalized so as to sum up to unity. The likelihood $p(Y_t^m|S_t)$ may be computed as

follows by using the Bhattacharya distance d_m(defined in section 3) between the feature distributions of the regions (under comparison) from the two (reference and target) frames:

$$p_m(Y_t|S_t) = e^{-(d_m/\sigma)^2} \tag{29}$$

The scaling constant σ in the above equation is so chosen as to ensure $p(Y_t|S_t) \leq 1$. The mixture coefficient the values proportionate to the percentage of particles (associated with the corresponding feature that survived resampling. This choice is intuitive because larger rates of rejection of the particles associated with a feature suggest that the feature is less effective.

2.3 Occlusion Handling with Single Camera

Adaptive feature selection method can deal with most cases of occlusions. However, for the extremely severe occlusion, due to the lack of the detected features of the object in the predicted area, the tracking may lose the target. Thus, I will introduce a novel algorithm to deal with the mistracking problem caused by severe occlusion.

For the estimation of the target position, the parameters of the state s can be represented as

$$P = \{C_x(n), V_x(n), C_y(n), V_y(n), A(n)\} \tag{30}$$

where $(C_x(n),\ C_y(n))$, $(V_x(n),\ V_y(n))$ and $A(n)\}$ represent the position, velocity and area of the object in the nth frame, respectively.

The target position in the n+1 frame can be estimated as

$$(C_x'^{(n+1)}, C_y'(n+1)) = (C_x(n) + V_x(n), C_y(n) + V_y(n)) \quad (31)$$

Object in frame n matches that in frame $n+1$ if following rule is satisfied:

$$|A(n) - A(n+1)| < TA \quad (32)$$

$$\sqrt{\left(C_x'(n+1) - C_x(n+1)\right)^2 + (C_y'(n+1) - C_y(n+1))^2} < TC \quad (33)$$

where TA and TC are the thresholds of the area and position which should not be exceeded for a matching.

In the occlusion detection step, if the object in frame n matches that in frame $n-1$, I set $Mb(n) = 1$;otherwise, $Mb(n) = 0$. If the object in frame n matches that in frame $n+1$, I set $Mf(n) = 1$;otherwise, $Mf(n) = 0$.

Detection of occlusion: If the object in frame $n-1$ cannot match that in frame n, I check if it is in the exiting region. If so, it is assumed that the object moves out of the scene. Otherwise, it is assumed that occlusion occurs. Once the occlusion is detected, I will turn off the particle filter prediction process, instead, I just estimate the position of the object using its average velocity in its most recent t frames. Meanwhile, I generate the maximum number of samples with largest variance in particle filter in order to capture the object in each frame once it appears again. When the object is detected

again, instead of using the constant velocity to update the position of the object, I use the particle filter algorithm again to estimate its location.

2.4 Multi-view Fusion Algorithm

In an effective algorithm for information fusion from multiple independent cameras, particles generated in 3D space can be evaluated based upon the performance of projected particles in 2D space, and the information from different views can be integrated using a weighted average scheme. In other words, If I somehow have information regarding the relative figures of merit for individual camera on a dynamic basis during tracking, the similarity measures obtained as indicated in section 3 for these cameras can be used to further estimate the relative figures of merit of particle prediction in 3D space . However, automatic evaluation of the figures of merit for each particle from different views is really a challenging task. Fortunately, in the PF framework, the likelihoods of the particles can be considered as their figures of merit in the latest state. Therefore, it is intuitive to design the multi-view combination method as an integral part of the PF method.

In the previous work, for the purpose of fusion multi-view information, researchers calculated the product of likelihoods from all the view planes as the fusion result for 3D position evaluation. However, in some circumstances, especially in the occlusion scenario, an excellent 3D position prediction from different camera views may be affected drastically by just a single view which has a very low likelihood value. Thus, I determine to incorporate the concept of camera reliability into my likelihood fusion framework.

In the PF, to address the problem of degeneracy (stochastically increasing variance of the importance weights resulting in all but one particle having negligible weights), the standard Sample Importance Resampling (SIR) procedure wherein the weights of N_s unequally weighted particles are all reset to $w_t^i = 1/N_s$ is a crucial step. Resampling procedure is applied if the number of effective particles is below a certain threshold:

$$N_{eff,t} = {1}/{\sum_{i=1}^{N}(w_t^i)^2}$$

$$< N_{thr} \tag{34}$$

Inspired by the resampling step, I take the number of effective particles as the measurement of reliability of cameras by defining reliability value as $K_{m,t} = \frac{N_{eff,t}^m}{N}$, since larger number of effective particles means more fitness between my proposed model and real world situation. Hence the overall likelihood of particles in 3D space can be combined as follows:

$$p(Y_t|S_t) = \sum_{m=1}^{M} \widetilde{K}_{m,t} p(Y_t^m|S_t) \tag{35}$$

where $\widetilde{K}_{m,t}$ is the normalized value of $K_{m,t}$ based on M. $p(Y_t^m|S_t)$ is the likelihood for camera m in state S_t at time step t,and M is the total number of cameras. The likelihood $p(Y_t^m|S_t)$ may be computed as follows by using the Bhattacharya distance d_m(defined in equation(26)) between the feature distributions of the regions (under comparison) from the two (reference and target) frames in m cameras:

$$p(Y_t^m|S_t) = e^{-(d_m/\sigma)^2} \quad (36)$$

The scaling constant σ in the above equation is so chosen as to ensure $p(Y_t|S_t) \leq 1$.

Based on the assumption that the reliability of each camera can be measured by the number of samples survive in the resampling procedure, this algorithm gives more Weight to the camera which is in a better view during the tracking. Suppose due to occlusion or template drift, cameras lose their targets will be assigned low Weight, since rare samples can survive in resampling, thus the tracking performance will not be affected severely by these unreliable cameras.

2.5 Experiment Results

2.5.1 Tracking Accuracy Analysis with Single Camera

In my experimental study, I used two separate experiments to establish the occlusion tolerance and tracking accuracy of my algorithm. In both the experiments, I chose 6 quantization levels for each one of the 3 features extracted as indicated in section 3. The number of particles has been set to 800. This choice is based on the consideration that no further improvement in tracking performance could be obtained for higher values.

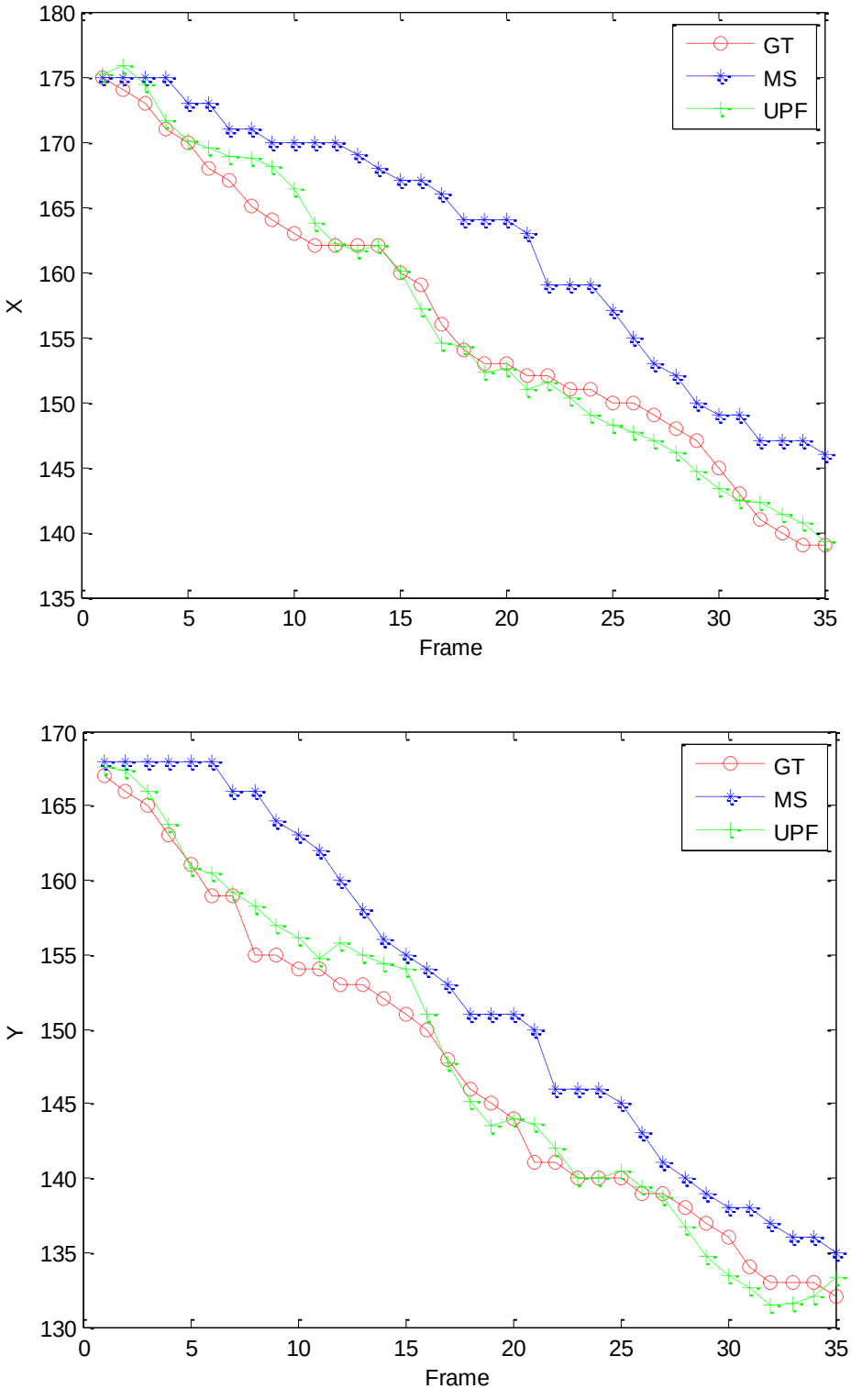


Figure 2.1. Comparison between UPF and MS method

Figure 2.2 Tracking results of frame 5 without occlusion handling

Figure 2.3 Tracking results of frame 15 without occlusion handling

Figure 2.4 Tracking results of frame 26 without occlusion handling

Figure 2.5 Tracking results of frame 40 without occlusion handling

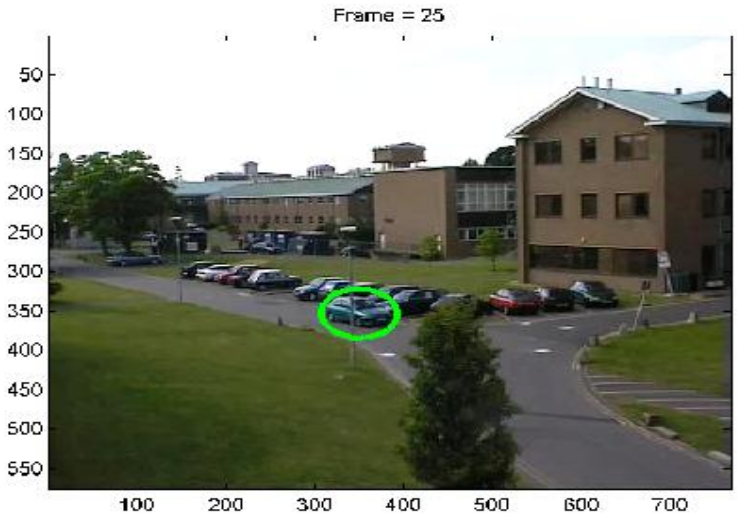


Figure 2.6 Tracking results of frame 25 with occlusion handling

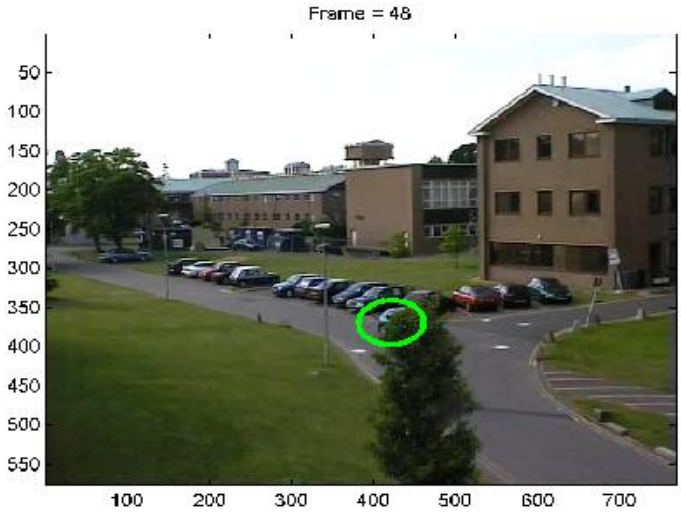


Figure 2.7 Tracking results of frame 48 with occlusion handling

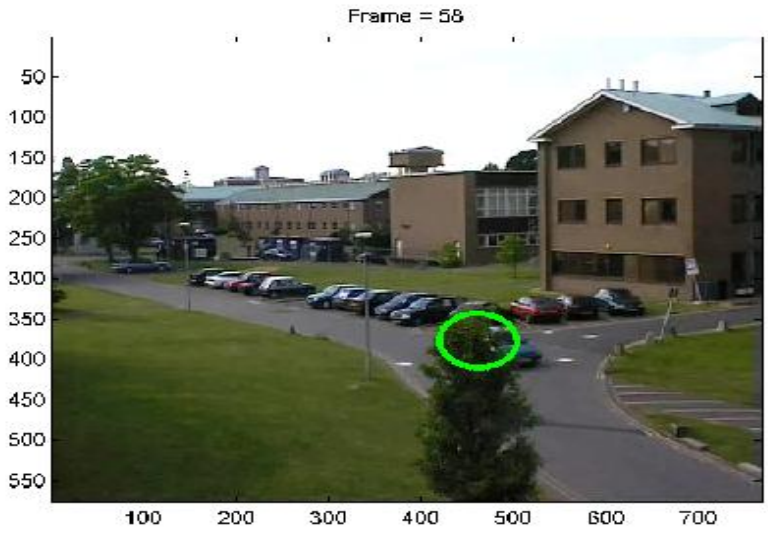


Figure 2.8 Tracking results of frame 48 with occlusion handling

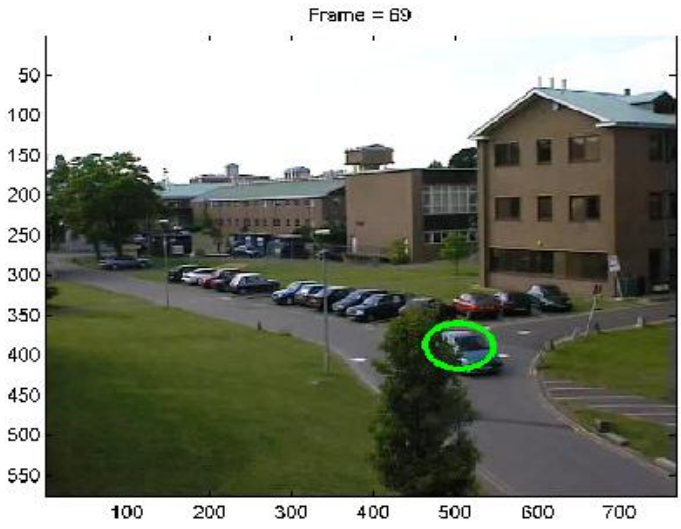


Figure 2.9 Tracking results of frame 69 with occlusion handling

2.5.2 Tracking Performance with Two Cameras

I test my proposed method using the data from Performance Evaluation of Tracking and Surveillance (PETS) workshops where videos are taken with two calibrated cameras. The camera calibration information is made available with the dataset.

2.5.3 Occlusion Tolerance Test

I compare my proposed multi-view fusion method with the independent camera tracking method. Using independent camera method, the tracking results in frame 48 and frame 58 indicate that when object is under fully occlusion, the tracker drifts away from the object, since color feature is not reliable any more. In the experiment result of my proposed multiple camera fusion method, I can observe that object is still tracked correctly, though complete occlusion occurs. The number of particles is set to 800, based on the consideration that no further improvement in tracking performance could be obtained for higher values.

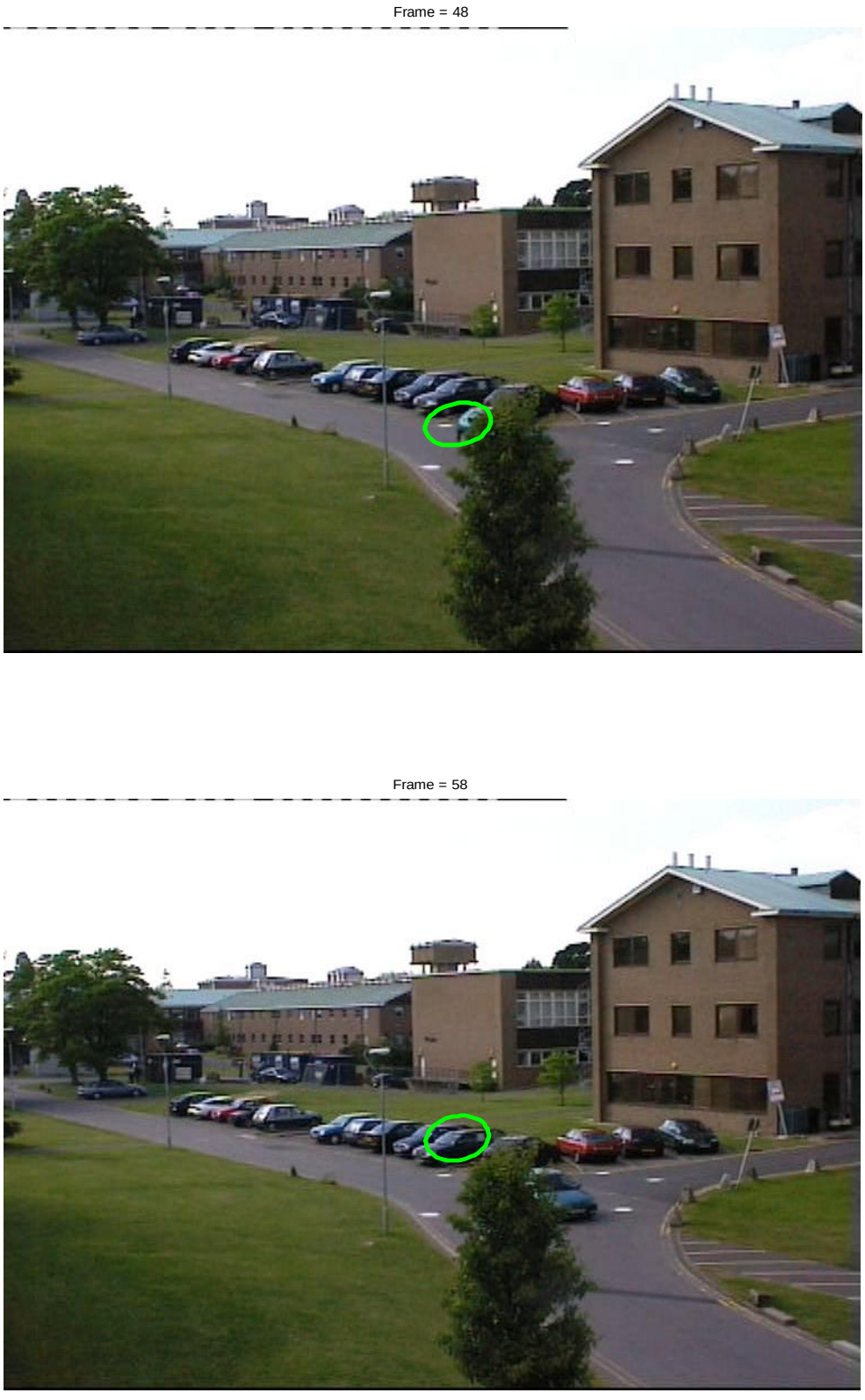
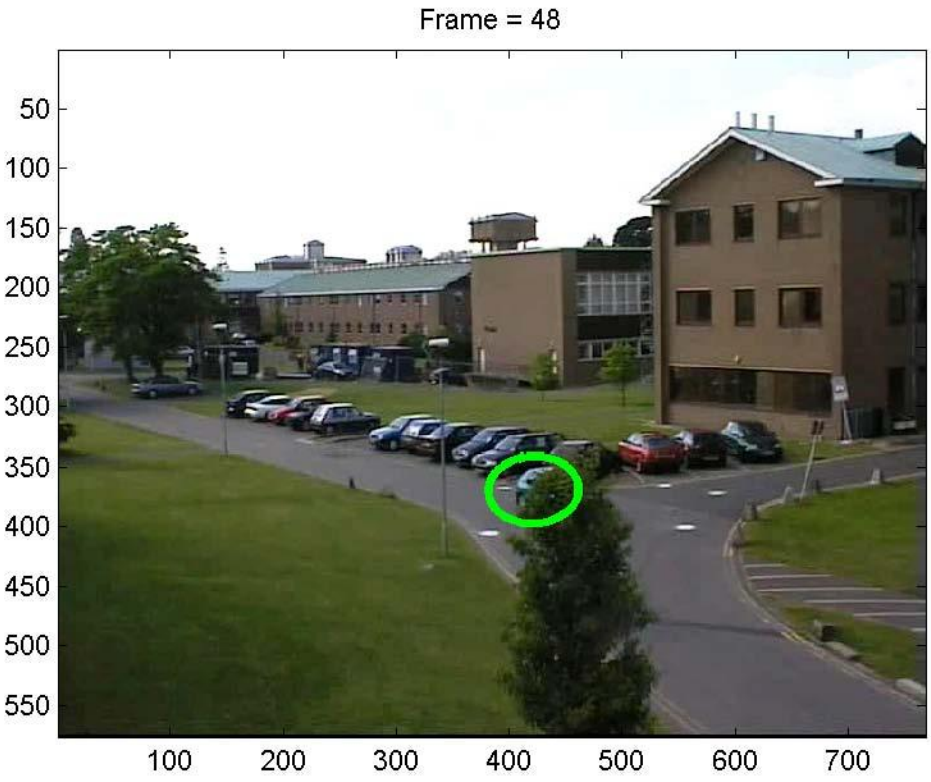
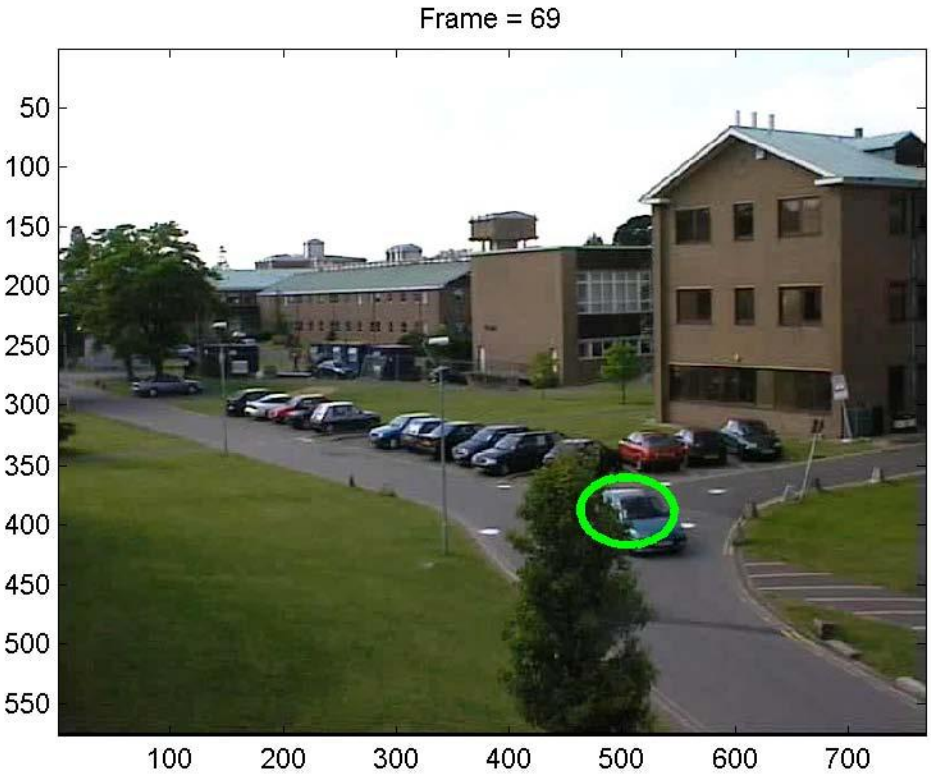


Figure 2.10 Tracking using independent camera

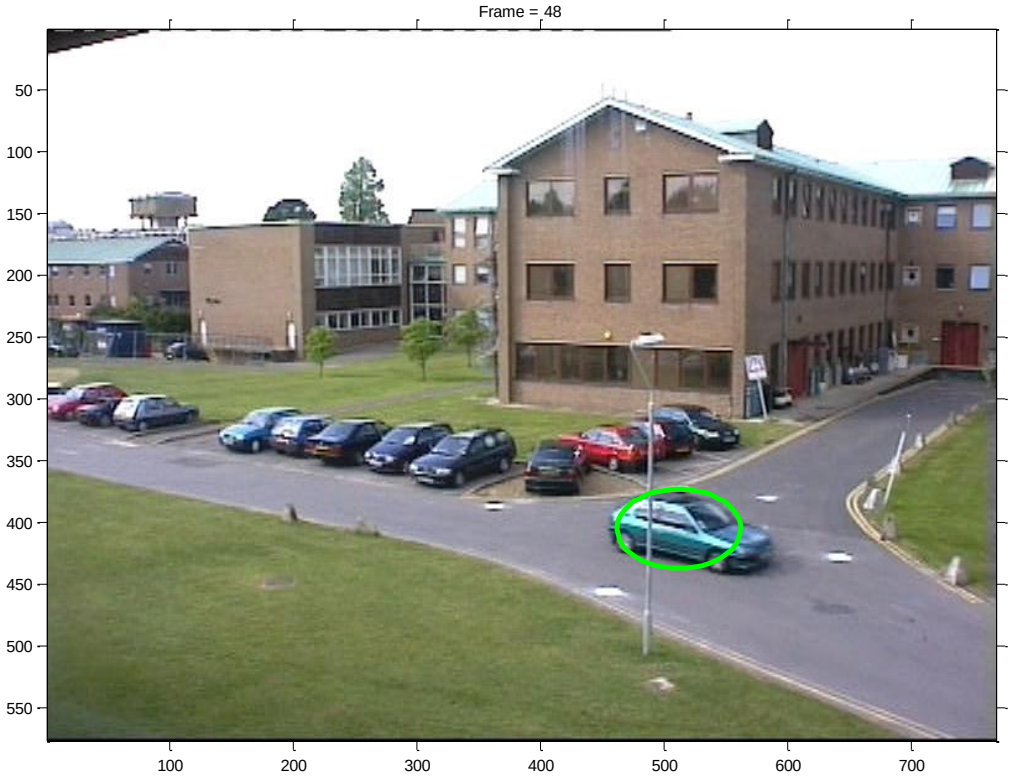


(a) camera 1 frame 48

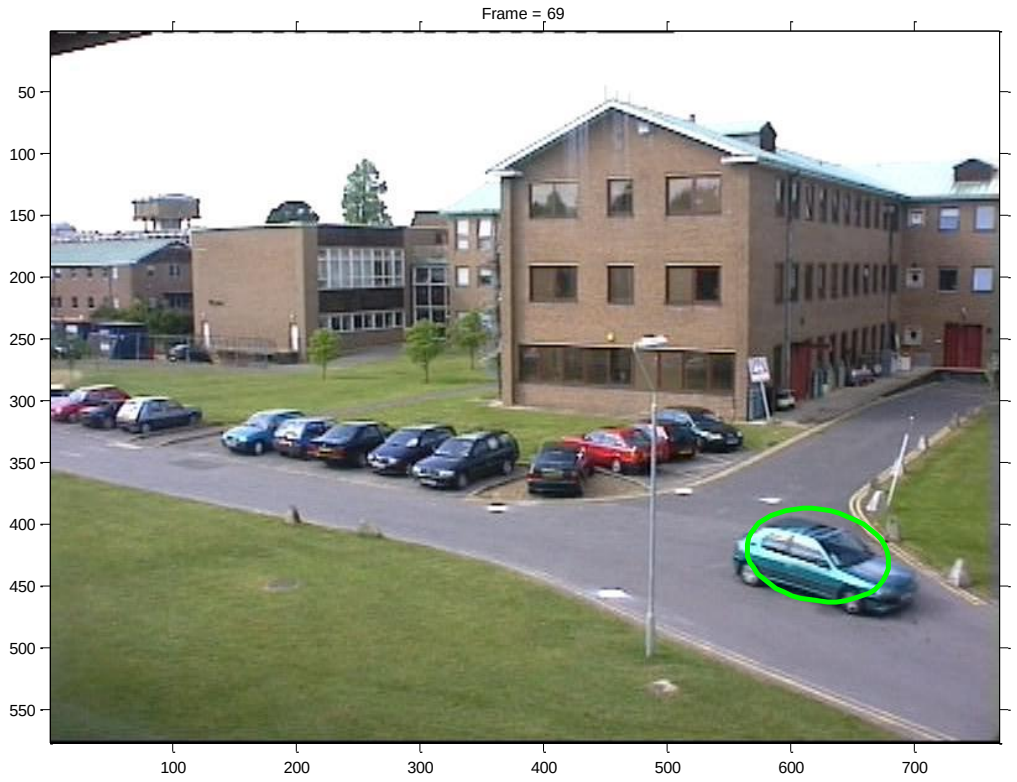


(b) camera 1 frame 69

Figure 2.11 Tracking using my multi-view fusion algorithm for camera 1**.**



(a) camera 2 frame 48



(b) camera 2 frame 69

Figure 2.12 Tracking using my multi-view fusion algorithm for camera 2**.**

2.5.4 Tracking Accuracy for Two Cameras

I am using videos, in which no occlusion problem occurs, as my accuracy test videos. By comparing the mean square root error of object position with the ground truth provided with the datasets, I can observe that my proposed multi-view fusion algorithm increase the accuracy of both cameras during tracking. The experiment is performed 50 times with 80 frames. Average error produced by the tracking algorithm on all sequences is referred to as the overall error which is measured by pixel. Type 1

represents cameras working independently, while type2 represents cameras working under my fusion method.

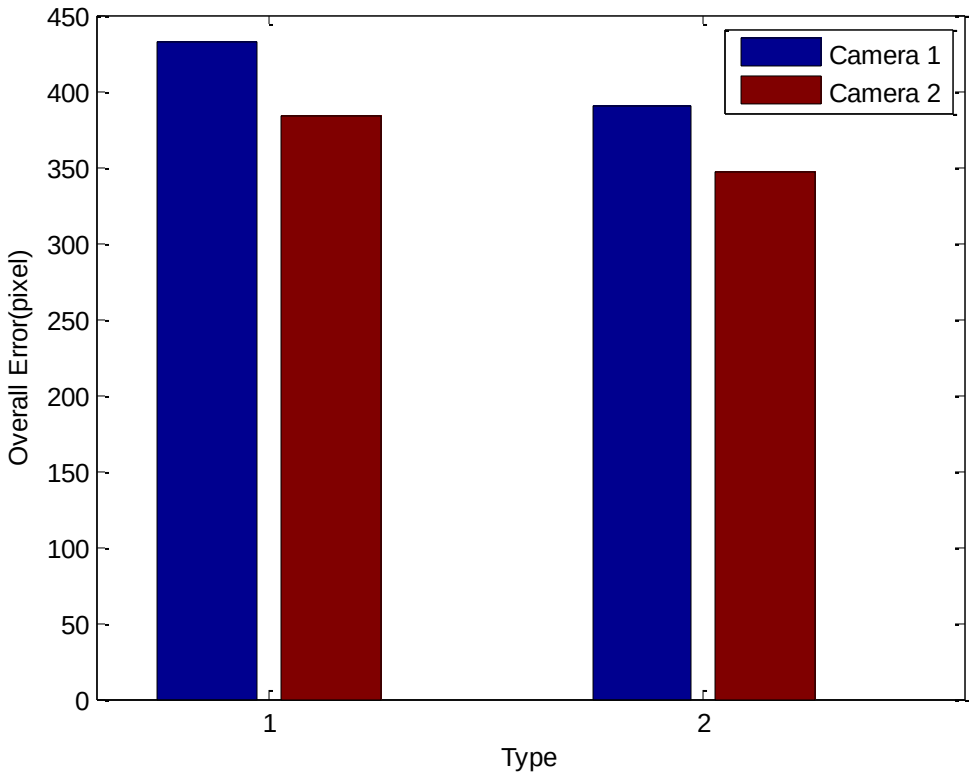


Figure 2.13 Tracking accuracy comparison between independent cameras and cooperated cameras. Type 1 represents the independent cameras while type 2 represents the camera fusion results of my algorithm.

CHAPTER 3

VIDEO STABILIZATION

Since hand-held video cameras become more and more popular, video stabilization techniques have attracted great interest recently. The main technique for video stabilization is to remove the unwanted vibrations for each frame of the video. Based on this technique, there are mainly two approach: first one is the hardware based method which tries to physically avoid camera shakes by adjusting camera motion sensors once unwanted motion is detected. Though this approach performs Ill in real life applications, the video systems adopting this approach turn out to be very expensive because of the need for sophisticated sensors that measure camera shakes accurately. Another method is image processing software based methods, usually known as digital image stabilization methods, are much less expensive. In this paper, I am focusing on the software based approach. The proposed method uses the particle filter approach for an accurate estimation of undesired motion of the camera with the SIFT feature extraction.

A digital image stabilization system essentially consists of two modules: global motion estimation module and the motion correction module. An accurate estimation of global motion is important because motion estimation accuracy directly affects the motion correction performance during the stabilization phase. Many methods proposed in the current literature aim to estimate the global motion accurately. Erturk [40] introduces a global motion estimation method by estimating the motions of four sub-images located at the corners of the image. This method is proved to be efficient and accurate, but is of limited applicability because of the assumption that foreground

objects are more likely located at the center of the image and hence less likely to be cropped in these four local images at the corners. Xu [41] proposes a local motion estimation method with circular blocks which are considered to be rotation invariant. In [42], corner features are extracted and tracked in order to estimate global motion. However, these features are not robust with respect to different kind of image transformations such as scaling and rotation. Hence, shift invariant feature transform (SIFT) features, which are considered to be invariant to image scaling and rotation [43], are being widely used in the latest methods for global motion estimation [44]-[46]. However, SIFT is not very efficient due to its high dimensionality of feature vectors. Thus, in my algorithm, I introduce a novel PCA-SIFT feature based method, which is yet to be used in the video stabilization field. PCA-SIFT approach has been found to be more accurate and faster in matching key-points compared to the SIFT-based approach [47].

3.1 Algorithm and Method

For accurate estimation of the motion, I use the particle filter (PF) approach, which is considered to be a powerful algorithm to deal with non-linear systems operating under non-Gaussian noise. My adaptive PF algorithm, which is a further improvement over the classical PF, shows better performance in both accuracy and efficiency. For matching the object in two consecutive frames, I use a novel PCA-SIFT method. Mosaicking, a technique of using an earlier frame contents, is employed to fill in the voids in the background created while performing the transformation required for obtaining the stabilized output from the shaky image. In summary, the primary contributions of my work are: i) Extraction of the PCA-SIFT features to estimate global

motion, and ii) Use of adaptive particle filter algorithm to improve motion estimation results.

Different modules of my algorithm are depicted in Figure 3.1. First of all, PCA-SIFT based features are extracted and matching pairs are determined. Next, the RANSAC method is used to estimate the initial motion between the frames. Based on the previous matching pairs, RANSAC can iteratively compute the parameters of the motion model by detecting the inliers and outliers among these matching pairs. Subsequently, a newly developed adaptive PF algorithm is used to refine the initial motion parameters and motion compensation is performed. Finally, the missing areas in the compensated sequences are recovered by the mosaic method.[115]

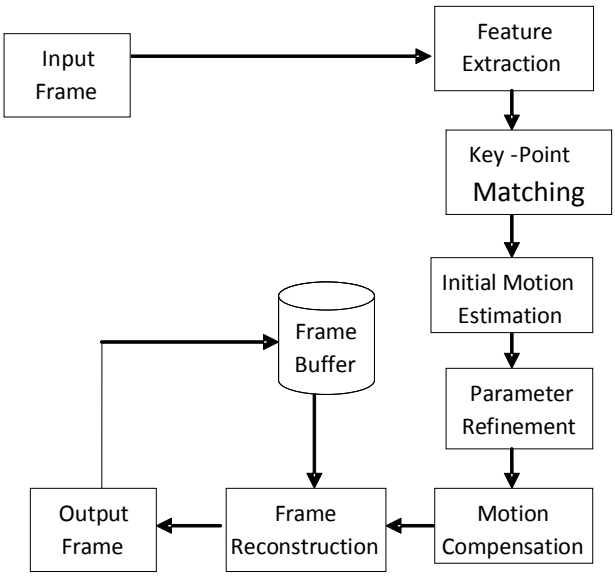


Figure 3.1. Key components of the proposed algorithm

3.1.1 Initial Motion Estimation Using RANSAC

Traditionally, the geometric transformation between two images can be described by a homograph which is a 3D model with eight unknown parameters. However, due to the complexity of homograph, I adopt a 2D affine transformation having only four unknown parameters. Suppose $P_1 = (x, y, 1)^T$ and $P_2 = (x', y', 1)^T$ to be the pixel location of corresponding points in consecutive video frames, the relationship between these two locations can be expressed by following transform:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} C_o \cos(\theta_o) & -C_o \sin(\theta_o) & Tx_o \\ C_o \sin(\theta_o) & C_o \cos(\theta_o) & Ty_o \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{37}$$

Here, Tx_o and Ty_o denote translation along x and y axis respectively. C_o and θ_o are scaling and rotation parameters in the image plane.

Although [9] states that the number of mismatches can be reduced by using PCA-SIFT compared with SIFT, small amount of mismatches still occur, and this may lead to unreliable prediction of motion estimation. Thus, a further check of matching errors is a significant part of the algorithm. I use the Ill known RANSAC algorithm for the purpose of both elimination of the outliers in the previous matching and estimation of the four unknown parameters of the above affine transform model. In each iteration of the RANSAC algorithm, minimal sample sets (MSSs) are randomly selected from the input dataset and the affine model parameters are computed using only the elements of this MSS, then RANSAC checks the entire dataset and decides the inliers or outliers depending upon whether an element of input data set fits the model or not. After K

iterations, the result that has minimal outliers is used as the initial value of the parameters of my affine transform model.[115]

3.1.2 Construction of Motion Model

The state vector of S_t can be represented as $S_t = [Tx, Ty, \theta, C]^T$ where these four elements represent translation along x, y axis, rotation and scaling parameters in affine transform model . My goal in the current problem is to estimate the global motion which can be considered as a cumulative motion of previous frame neighbors. The state transition equation in my PF model is given by:

$$S_{t+1} = AS_t + U_t \equiv$$

$$\begin{bmatrix} Tx \\ Ty \\ \theta \\ C \end{bmatrix}_{t+1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} Tx \\ Ty \\ \theta \\ C \end{bmatrix}_t + \begin{bmatrix} N(Tx_o, \sigma_x) \\ N(Ty_o, \sigma_y) \\ N(\theta_o, \sigma_\theta) \\ N(C_o, \sigma_C) \end{bmatrix} \quad (38)$$

Here A is the transition matrix and U_t is the process noise following the Gaussian distribution with the variances[σ_x , σ_y , σ_θ , σ_C] and the means $[Tx_o, Ty_o, \theta_o, C_o]$. The components of the mean vector are determined using the RANSAC procedure.

Note that, unlike other parameters, the cumulative result of scaling parameter is a product rather than a summation of the previous values. Hence, in the above transformation, I compute the scaling parameter just based on the initial value C_o as computed by RANSAC method without considering the prior state. Another reason for this kind of computation model is that scale changes tend to be small, thus, direct

estimation without sacrificing accuracy is possible. The RANSAC approach provides rough estimates of the motion parameters to the PF algorithm, and thereby saves the unnecessary computational cost involved in the generation of useless particle samples.[115]

3.1.3 SIFT-BMSE Cost Function

In the PF approach, each particle is weighted by a performance evaluation metric that measures the accuracy between the estimated particles and the measurements. In my paper, I develop a novel error evaluation method, I call it SIFT-BMSE (SIFT block mean square error).

Once the SIFT feature is extracted and RANSAC algorithm is applied to the frames, the location of each matching keypoints are obtained. Let (u_o, v_o) denote the position of one of the SIFT feature point in the image, the SIFT box function corresponding to this point is defined as:

$$b(u,v) = \begin{cases} 1 & \begin{array}{l} u_o - w/2 + 1 \le u \le u_o + w/2, \\ v_o - h/2 + 1 \le v \le v_o + h/2 \end{array} \\ 0 & otherwise \end{cases} \tag{39}$$

Where w and h denote the size of the SIFT box. Once the SIFT box function is applied for all the SIFT keypoints, a mask is obtained. The characteristic of this mask is that the moving foreground object is discarded by applying this mask into the image, since the outlier of the matching is already rejected in the RANSAC process. Let M^k be the mask extracted in the frame k. Let T_i^k be the global motion between frame k and $k+1$

estimated by the i th particle. Then MSE is computed between $M^k I^k$ and $(T_i^k M^k)(T_i^k I^{k+1})$. Another advantage of this method is that the computational cost of MSE based on the image after applying this mask is much loIr than that of the whole image.

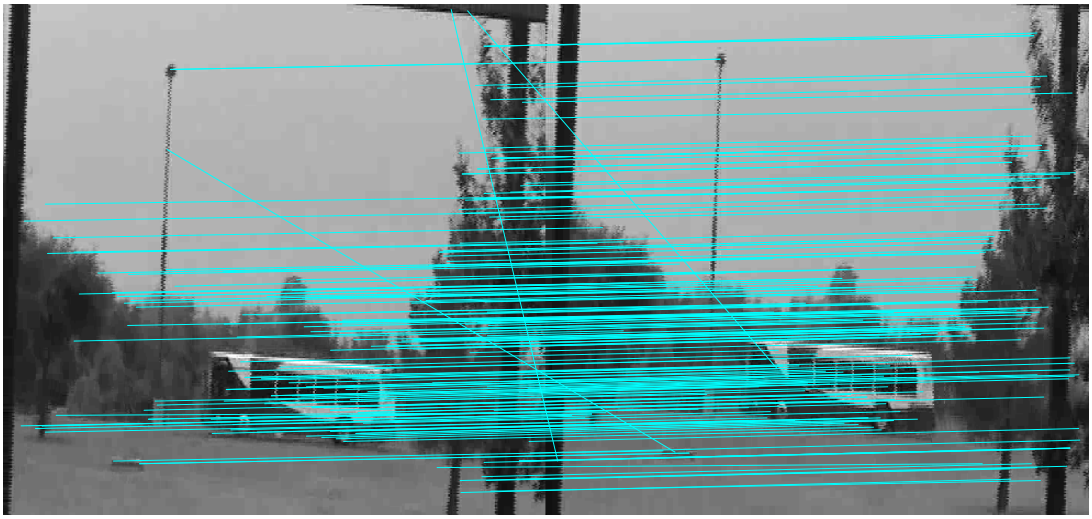


Figure 3.2 Keypoints extraction and matching before RANSAC

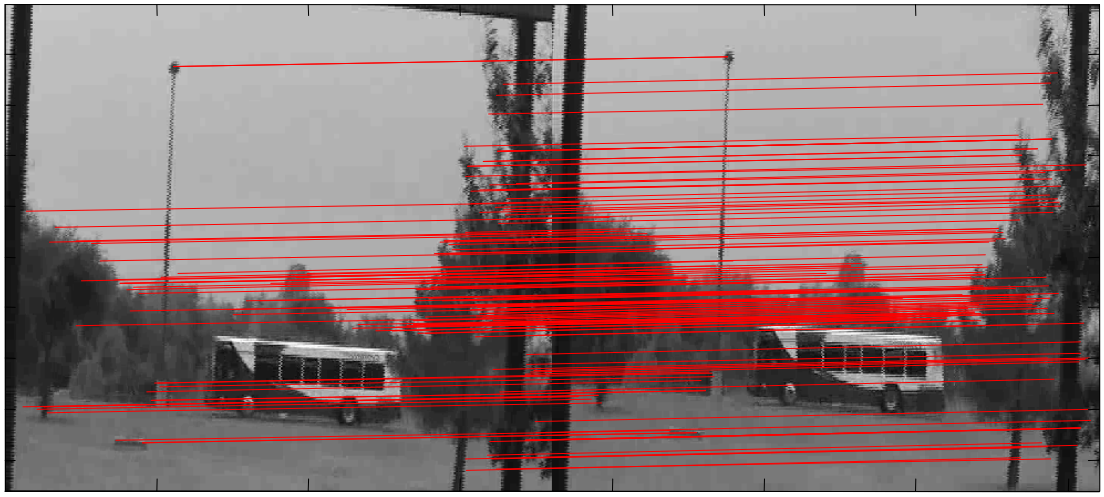


Figure 3.3 Keypoints extraction and matching after RANSAC

Figure 3.4 Block mask based on SIFT box function

Let d_i denote the total value of SIFT-BMSE estimated by using particle i. The likelihood $p(Y_t|S_t^i)$ of particle i is given by

$$p(Y_t|S_t^i) \propto e^{(-d_i/\sigma_M)^2} \tag{40}$$

Using this likelihood value in (6), the Weight w_t^i of particle i can be computed. Finally, all the weights $\{w_t^i\}_{i=1,\ldots N}$ are normalized so that they total to 1. [115]

3.1.4 Adaptive Model Noise and Particle Number

In real applications, the intensity of the motion in the images taken by shaking camera is unpredictable. Thus, the noise in the state transition model needs to be adaptive in order to handle with both slight and intensive motions. My PF algorithm achieves adaptivity by choosing the model noise variance and the number of particles based on the estimation performance. Now, if σ represents the noise variance, then, σ can be defined as $\sigma = R \cdot \sigma_o$, where σ_o is the basis noise variance and R is a function of the overall error d.

$$R = \max(\min(R_o \cdot d, R(\max)), R(\min)) \tag{41}$$

where $R(\min)$ and $R(\max)$ denote as loIr bound and upper bound respectively. R_o is the basis value of R. If the noise variance σ is large, more particles are needed. Hence, in my algorithm, the number of particles required for accurate motion estimates are determined by:

$$L = (L_o \cdot R_o)/R \tag{42}$$

where L_o is the basis number of particles.

Rough estimation of the model parameters as discussed in subsection II.B together with adaptive selection of the number of particles and noise variance used in the model as above not only improves the robustness and accuracy of the estimation, but also saves the computation cost by reducing the generation of unnecessary particles.

3.1.5 Frame Reconstruction

After each frame is compensated by using the above discussed motion estimates, the pixels near the frame boundary may be undefined, and this leads to unacceptable visual effects. Traditionally, many researchers either trim the undefined region or fill in the region with a constant value. This results in information loss and quality degradation especially in the case of frames that undergo large scale translations and rotations because of a jerky camera. To avoid this problem after compensation, information of neighboring frames can be borrowed to fill the undefined regions though sometimes undefined pixels may still exist. Intuitively, if I increase the number of neighboring frames, the number of pixels in undefined region will be reduced, but it is computationally intensive. In this paper, I apply mosaic method to reconstruct undefined regions using previous stabilized full-frames, since the undefined region of these frames are already determined. However, this may occasionally result in a carry-forward information (information from previous frames) across a considerable number of frames.[115]

3.2 Video Stabilization Results

I tested the proposed algorithm on two video sequences. Original video sequences are taken by a digital video camera that undergoes translation and rotation during the video shooting.

1) Outdoor Scenario: the video was captured with a moving camera with 100 frames. Since there is a bus moving across the scene, the problem turns out to be challenging. The results are shown in Figure 3.5. The original frames are presented in the second row and the results after compensation are depicted in the third row. With

the help of the red crosses that mark the center of the frame, I can observe that the output sequences remove the unwanted motion of camera.

2) Indoor Scenario: I record the indoor video with 120 frames of a moving mouse using a camera under severe rotation and translation. This is a very difficult scenario due to the large moving object and the severe motion of the camera. In Figure 3.6, images in the first row of the figure are the reference frames, second row shows the compensation results in different frames, while third row are the frames after reconstruction using Mosaic. The visual quality of the results provides a qualitative assessment of the proposed method for video stabilization.

3.2.1 Performance Evaluation

The performance of my algorithm has been evaluated based on ITF (Interframe Transformation Fidelity) measure given by:

$$ITF = \frac{1}{N_{frame} - 1} \sum_{k=1}^{N_{frame}-1} PSNR(k) \tag{43}$$

where N_{frame} represents the number of video frames. $PSNR(k)$ is the Peak Signal-to-Noise Ratio which can be defined as:

$$PSNR(k) = 10 \log_{10} \frac{I_{\max}}{MSE(k)} \tag{44}$$

where $I_{\max}$ is the maximum pixel intensity and $MSE(k)$ is the Mean Square Error between consecutive frames.

The higher ITF values in Table 3.1 for my algorithm compared to the RANSAC algorithm (without PF) indicate that my method yields more accurate motion estimates. In Table 3.2, the comparison is performed between proposed SIFT-BMSE method and traditional MSE method, the results show that SIFT-BMSE results in relative high accuracy.

Table 3.1 ITF Values for RANSAC and My Algorithm

Sequence No.	Original ITF(dB)	RANSAC ITF(dB)	RANSAC+ Adaptive PF ITF(dB)
Sequence 1	18.64	20.98	21.11
Sequence 2	19.55	22.17	22.24

Table 3.2 ITF Values for Using SIFT-BMSE and Using Traditional MSE

Sequence No.	Original ITF(dB)	MSE ITF(dB)	SIFT-BMSE ITF(dB)
Sequence 1	18.64	21.02	22.11
Sequence 2	19.55	22.13	22.32

3.2.2 Computational Complexity

According to the algorithm proposed in section II, the computational complexity depends on many factors such as the number of particles, the number of iterations in RANSAC and the compensation process. Thus it is difficult to give a quantitive measurement of computational complexity. In this paper, I evaluate the computational complexity in terms of computation time. The simulation is performed in a laptop with a Pentium4 2.8 GHz CPU without any hardware acceleration.

The simulation results are summarized in Table 3.3 and Table 3.4. The simulation is performed 10 times in two sequences with 100 frames and 120 frames respectively. I test the proposed adaptive Particle Filter algorithm with the traditional Particle Filter in Table 3.5, the results show that the frames per second (FPS) value of adaptive PF is relative high. In Table 3.6, the higher FPS value of SIFT-BMSE indicates that my proposed SIFT-BMSE method runs faster than traditional MSE method.

Table 3.3 ITF Values for Only RANSAC And RANSAC Combined with Adaptive Particle Filter

Sequence No.	Original ITF(dB)	RANSAC ITF(dB)	RANSAC+ Adaptive PF ITF(dB)
Sequence 1	18.64	20.98	21.11
Sequence 2	19.55	22.17	22.32

Table 3.4 ITF Values for Using SIFT-BMSE and Using Traditional MSE

Sequence No.	Original ITF(dB)	MSE ITF(dB)	SIFT-BMSE ITF(dB)
Sequence 1	18.64	21.02	21.11
Sequence 2	19.55	22.13	22.32

Table 3.5 Comparison of Computational Speed between Traditional Particle Filter and Adaptive Particle Filter

Sequence No.	PF(FPS)	Adaptive PF(FPS)
Sequence 1	1.03	1.14
Sequence 2	1.12	1.25

Table 3.6 Comparison of Computational Speed for Using SIFT-BMSE Cost Function and without Using SIFT-BMSE Cost Function

Sequence No.	APF Without SIFT-SBME (FPS)	With SIFT-BMSE(FPS)
Sequence 1	1.08	1.14
Sequence 2	1.18	1.25

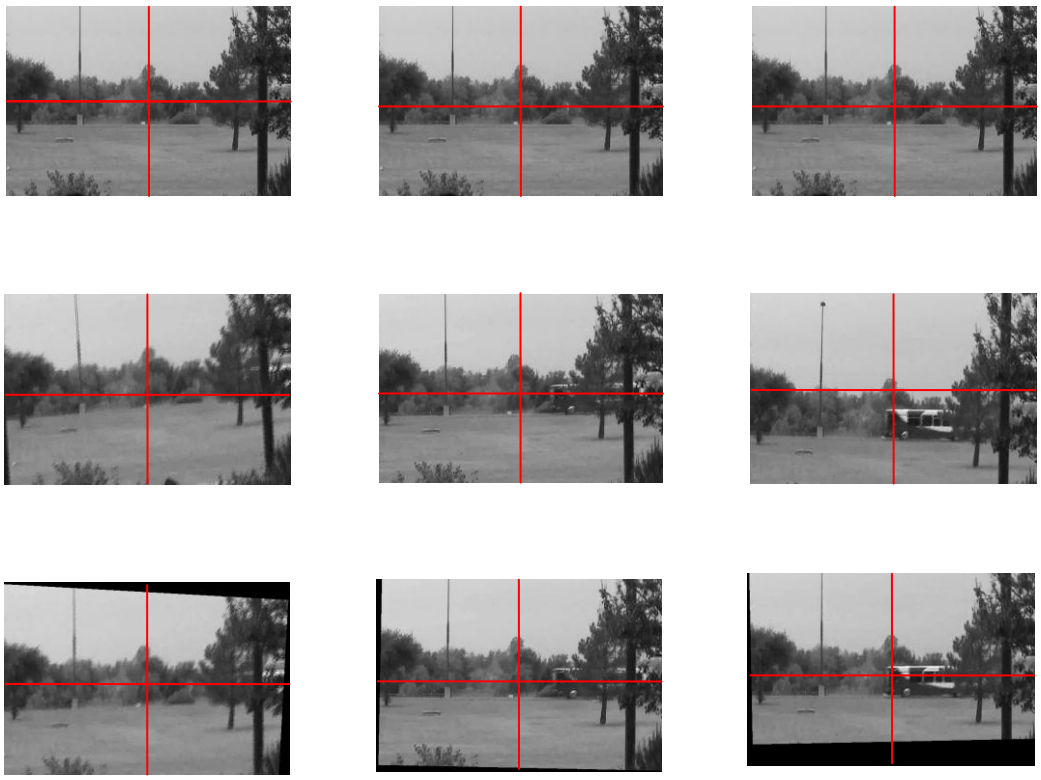


Figure 3.5 Results of stabilization for a outdoor image (First row depicts the reference frames, the second row, the original frames, and the third row, the final results after compensation using my algorithm).

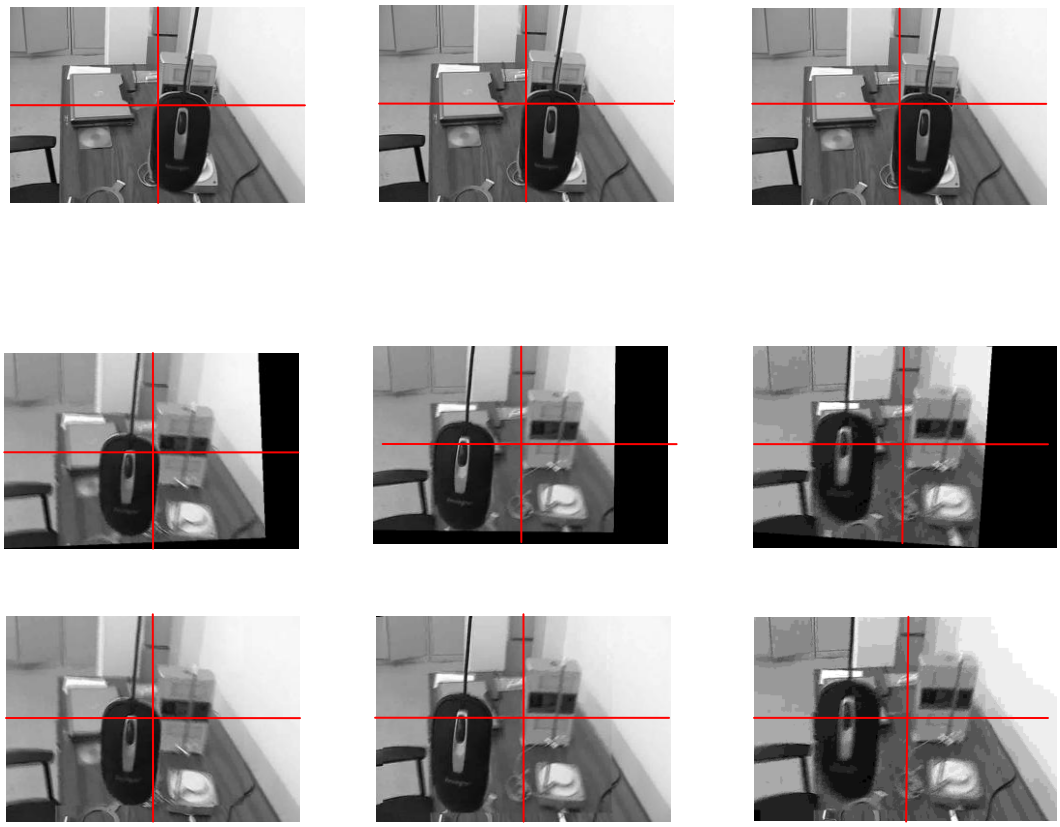


Figure 3.6 Results of stabilization for a computer mouse image (First row depicts the reference frames, the second row, the results after compensation using my algorithm, and the third row, the final results after reconstruction using Mosaicking).

CHAPTER 4

WIRELESS CAPSULE ENDOSCOPY VIDEO SEGMENTATION

Wireless capsule endoscopy (WCE) is a novel technology to record the videos of the parts of gastrointestinal tract that cannot be visualized through other types of endoscopy such as colonoscopy. In WCE, a patient swallows a pill sized capsule equipped with a tiny camera, which captures the videos of the digestive tract as the capsule propels through the tract by normal peristalsis. These videos are transmitted by a tiny wireless device attached to the capsule to a wireless receiver located outside the human body. There are about 5000 frames in each video with the frame ratio of 2 frames per second.

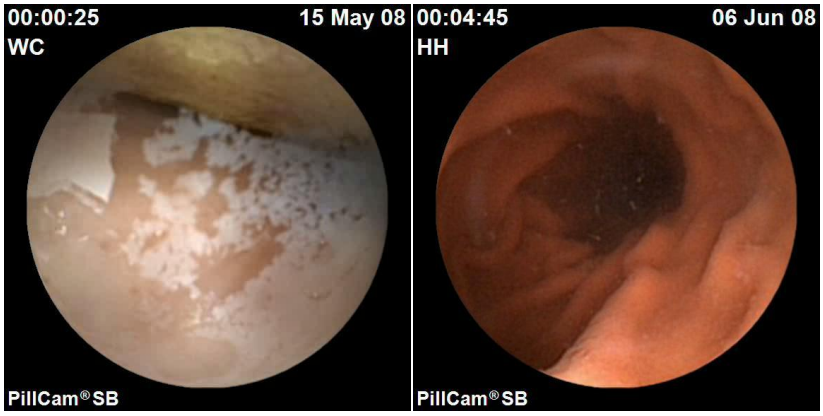
Since the capsule endoscopy is introduced in clinical area, it is proved to play an important role such as the detection of bleeding location [55], diagnosis of Crohn's disease [56] and ceiliac disease [57]. However, this method is man-power intensive since it takes over an hour for a trained specialist to examine one WCE video for abnormal conditions such as blood or ulcers [54]. To mitigate this problem, the endoscopy video frames are usually separated into four groups corresponding to the four parts of digestive tract: entrance, stomach, small intestine and large intestine. Figure 15 depicts the exemplary frames from these groups. Since some abnormal events only occur in particular parts of the digestive tract, such a grouping of video frames facilitates a clinician to focus on a small set of frames for the purpose of analysis. Thus segmentation of a WCE video into four groups is pivotal to quick analysis of the WCE data.

The problem of automatic WCE video segmentation (equivalently part-part boundary detection) has been addressed in the medical imaging literature from two fronts. The first set of papers are concerned with extraction of robust features such as color and texture features [58][59][60][61][62], whereas the second set of research articles focus on powerful classifier such as Bayesian classifier and Support Vector Machine (SVM) [63][64]. In [58], the Coimbra and Cunha use the scalable color and homogenous texture descriptors of MPEG-7 to segment the video. Boulougmya et al. [59] constitute their feature vectors by computing for the six color channels (R,G,B,H,S,V), the statistical measurements which are standard deviation , variance, skew, kurtosis, entropy, energy, inverse different moment, contrast, and covariance. Mackiewicz et al. [60] combine the color, texture and motion features of the sub-image region containing only visible tissue.

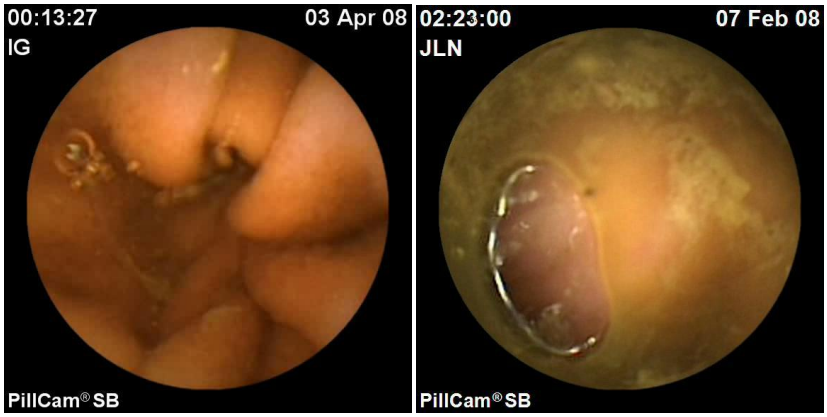
In [63], Spyridonos et al. employ an SVM with the gradient tensor feature of the image to detect the wrinkles that indicate the existence of contractions. Cunha et al. [64] compare the Bayesian classifier and SVM for the endoscopic video segmentation. Mackiewicz et al. use SVM within the framework of Hidden Markov Model (HMM), based on the assumption that the transition of between the four states (entrance, stomach, small intestine and large intestine) follows a certain probability distribution.

All the above approaches, however, employ supervised learning paradigm for WCE video segmentation. The main problem with these approaches is the presumption that a large database of correctly labeled samples is available for accurate training of the classifiers. But, WCE being a new technology, it may be

difficult to procure sufficiently large number of WCE videos. Further, because of the small differences in color and texture of the internal organs of different individuals, the labeled samples collected from the WCE of a person may not yield a good performance when applied to the video of another person. Hence, I propose in this paper a powerful unsupervised learning approach called probabilistic latent semantic analysis (pLSA) [75] to the WCE video segmentation. The pLSA employs bag of words model to analyze the semantic content of documents and segregate them based on the dominant topic (*e.g.* sports, politics) even though class (topic) information is latent (hidden). In the present image analysis context, local image features take the role of words in the linguistic analysis. Specifically, I perform image segmentation though pLSA with feature vectors extracted using scale invariant feature transform (SIFT) in lieu of words. I obtained better results by fusion of SIFT features with the color features.



(a) Entrance Image (b) Stomach Image



(c) Small Intestine Image (d) Large Intestine Image

Figure 4.1 Example of WCE images in different body part

4.1 Visual Word Extraction and Vocabulary Building

The first step in this multi-step process is to eliminate from the original WCE video frames a large black amount of background information and textural annotations. Due to the dome structure of the WCE camera, only the sub-image within the circular area around the center of each video frame can be considered as the region of interest (ROI). The area surrounding ROI is either out of focus or replete with irrelevant information, and hence can be safely removed.

4.1.1 Feature Extraction and Matching

Color provides more discriminatory information than simple intensities particularly in case of endoscopy image. Even though RGB (red, green, and blue) color space is simple and very common, HSI (hue, saturation, and intensity) features have been shown to have the best discriminatory potential for SIFT feature extraction among several perceptually relevant color spaces [79]. Also, previous work of Coimbra et al. [58] indicates that HSV (hue, saturation, and value) MPEG-7

scalable color descriptors yield best classification (segmentation) results for the WCE images. Hence, I compute in this work SIFT descriptors over all three channels of the HSV color model. Thus there will be a threefold increase in the dimensionality of the SIFT feature vector, and hence the key-point descriptors in color space would be having 3x128 (=384) components.

Since classification performance could also depend on not only the color space used but also the way color features extracted, I did not outright reject the RGB color space based on the counter evidence provided in [66] and [69]. I conducted my experiments also with the 384 component feature vectors obtained by a concatenation of the SIFT descriptors computed over the three channels of the RGB or 4 HSV color model.

4.1.2 Vocabulary Building

Final step of the process is building a vocabulary of visual words (or visterms) because my goal in this paper is to develop an unsupervised learning method based on the “bag of words” model used in the analysis of semantic content in text documents. However, the feature vector descriptors cannot be directly considered as visual words, simply because each component of the vector spans over the infinite set of real numbers whereas the words in a language are composed of characters belonging to a finite set of characters. A simple and obvious solution to this problem is to limit the number of possible feature vectors using vector quantization procedure. In this paper, I group the feature vectors extracted from some randomly chosen video frames into large (but finite) number of small clusters using the k-means clustering algorithm. It may be noted here that k is the size of the

vocabulary, and any feature vector can be uniquely mapped onto a specific word in the vocabulary depending upon which cluster mean is the closest to the feature vector under consideration. In Figure 4.2, the process of extraction of visual words from the sample video frames is depicted through Figure 4.2(a) to Figure 4.2(d). In my experimentation, I used vocabularies with up to 600 visual (code) words i.e. quantized vectors.

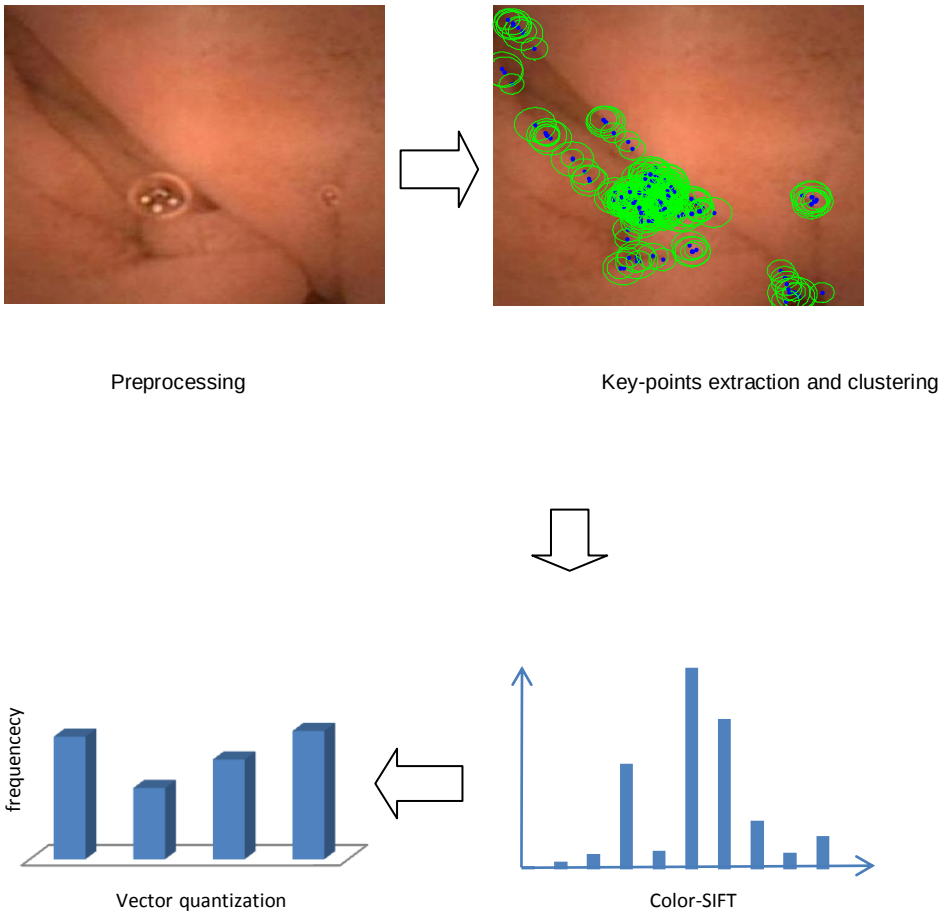


Figure 4.2 Visual word extraction and codebook construction

4.2 pLSA Model for Video Segmentation

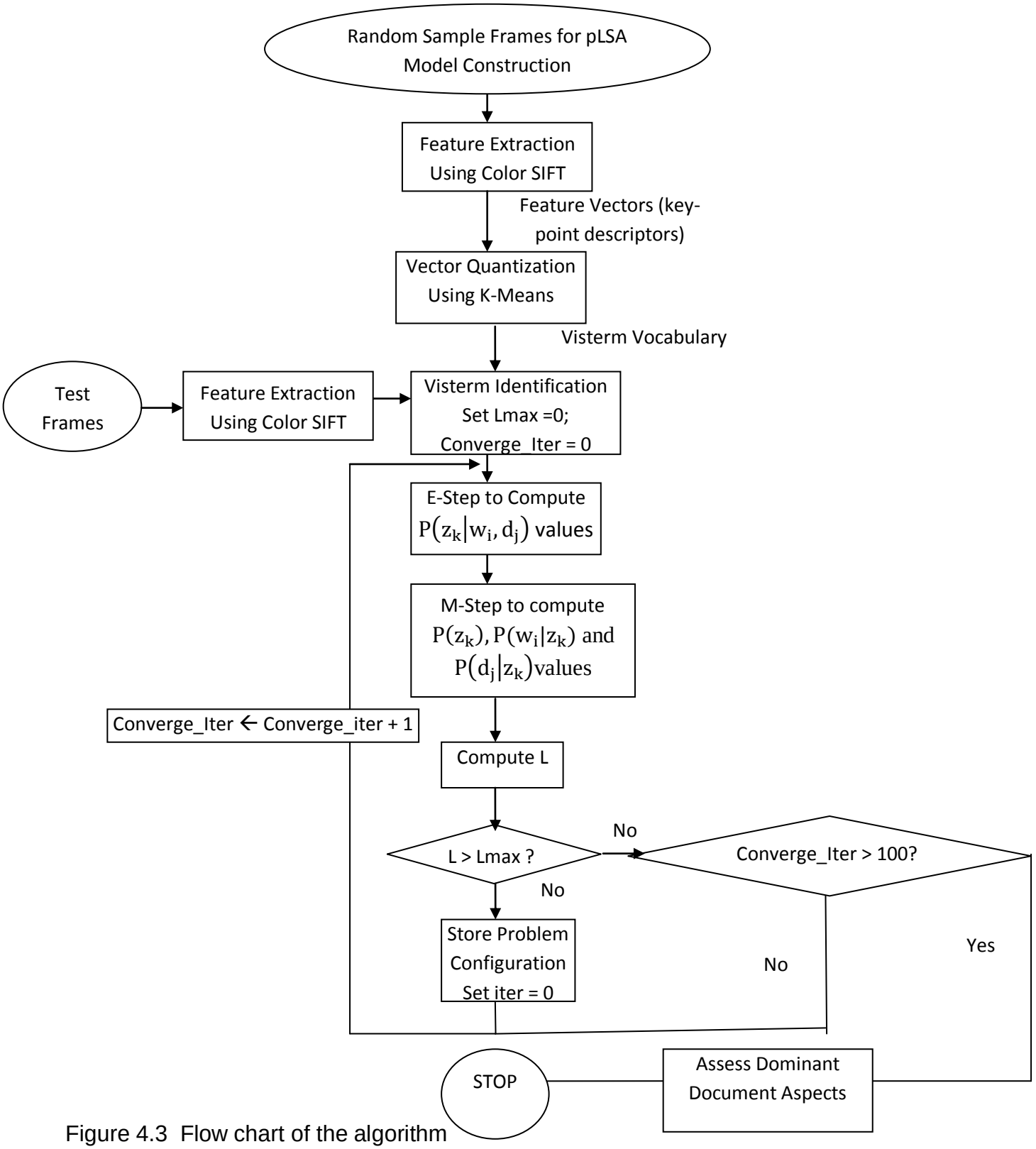


Figure 4.3 Flow chart of the algorithm

4.3 Experimental Results

For my experimentation, I used 10 annotated capsule endoscopy videos collected using the ``Given Imaging PillCam SB.'' Experienced clinicians annotated each video frame into four parts: entrance(P1), stomach (P2), small intestine (P3) and large intestine (P4) . I used 50% of the randomly sampled video frames for vocabulary building, and used the remaining 50% for testing the pLSA classifier.

Since classification depends both on the feature set and the type of classifier used, my first experimentation is on identification of the best classifier for the same (Grey-SIFT) features. My goal in this experiment is to compare my unsupervised approach with pLSA with Support Vector Machine (SVM), which established itself as the state-of-the-art classifier operating in supervised mode. Hence, I considered SVMs with both linear and non-linear (radial basis function) kernels. I used 3 one-against-one SVM classifier for the classification of 4 parts of human body after the extraction of the SIFT feature. However, the use of SIFT features with SVM (or any other classifier based on a supervised learning scheme) poses one problem. These classifiers consider each image frame as single holistic unit, and work on the feature vectors derived from individual frames. One way to construct the feature vectors for individual frames is to concatenate the feature vectors of the key-points of the frames that are raster scanned. But, since I can in general have different numbers of key-points per image, the frame feature vector sizes could be varying. To overcome this problem, I considered the size of the longest feature vector as the size of the typical feature vector, and padded all the shorter feature vectors with random numbers so as to make them constant sized vectors. I used 50% of the randomly

sampled video frames for training the classifiers, and used the remaining 50% for testing the pLSA classifier. Table 4.1 presents the results of classification.

	P1/P2	P2/P3	P3/P4
SIFT-SVM-lin	75.6	73.2	69.8
SIFT-SVM-rad	78.3	77.3	71.9
SIFT-pLSA	85.0	84.2	80.4

Table 4.1 The Comparison between Traditional SVM Classifier and pLSA Model based on the Local Feature

The results show that the entrance/stomach classifier and stomach/small intestine classifier perform better than the small intestine/large intestine classifier. The reason for that was the existence of the significant similarity of the tissues in the two most confusing parts. From the results, it is also clear that the SVM with radial kernel, as expected, performs better than that with the linear kernel. My proposed pLSA method yields much higher accuracy than either SVM classifier, but, in all fairness, it cannot be construed as the superiority of the pLSA method over the SVM. The earlier described approach for construction of the constant sized SIFT feature vectors for the SVM is rather ill-conceived, and this could be the reason for the inferior performance of SVM classifiers.

The goal of my second experiment is to make a comparative analysis of the traditional Gray-SIFT, RGB-SIFT and HSV-SIFT features with pLSA classifier for

endoscopy video segmentation. The results presented in Table 4.2 indicate the RGB-SIFT feature outperforms the other two features. Here P1/P2, P2/P3, and P3/P4 indicate the dichotomies entrance/stomach, stomach/small intestine, and small intestine/large intestine, respectively.

Table 4.2. The Comparison between Traditional Gray-SIFT And Color-SIFT

	P1/P2	P2/P3	P3/P4
Gray-SIFT	85.0	84.2	80.4
HSV-SIFT	94.4	92.5	91.0
RGB-SIFT	98.3	94.1	93.9

In view of the unsuitability of SIFT features for SVM classification, I compare in Table 4.3 the results my pLSA classifier with RGB-SIFT with the best classification results of Mackiewicz et al. presented in [8]. The authors in that paper obtained these results using the same two SVM classifiers but with feature constituted with following components: i) local binary pattern (LBP) histograms sorted into 343 ($=7^3$) bins over 3 color channels and compressed using principle component analysis (PCA), ii) motion feature 6-tuples extracted from 41 consecutive frames, transformed using discrete Fourier transform (DFT), and compressed using compressed using PCA, and iii) 32x32 bin HS (Hue and Saturation) histograms compressed using discrete cosine transform (DCT) and PCA.. From the Table 4.3, it is clear that the

classification results of the pLSA-RGB-SIFT algorithm are slightly shy of those for the state-of-the-art SVM classifiers. Still these results can be considered to be significant due to the fact that the pLSA classifier does not give the class label information as its input at any stage of the processing. These small discrepancies could have also been caused by the specific databases used for experimentation and the feature extraction process employed.

Table 4.3 The Comparison between PLSA Model based on the Local Feature and SVM Classification based on Multiple Features

	P1/P2	P2/P3	P3/P4
pLSA-RGB-SIFT	98.3	94.1	93.9
SVM-Radial	99.9	98.3	94.7
SVM-Linear	99.7	96.4	89.1

My last experiment is for testing an aspect peculiar to the pLSA method. Since the performance of the pLSA-RGB-SIFT algorithm could depend upon the size of the codebook (vocabulary), I wanted to study the size of my codebook (equivalently, the number of clusters formed in my k-means approach to vector quantization) on the accuracy of video segmentation. In this experiment, the size of codebook was varied from 200 to 600 in intervals of 100, and average classification accuracy over 100 runs with different sets of random samples for code book generation is plotted in Figure 4.4. From these results, I can observe that the entrance/ stomach, stomach/small intestine and small intestine/large intestine reach

their highest classification accuracy when the codebook size is 300, 400 and 500, respectively.

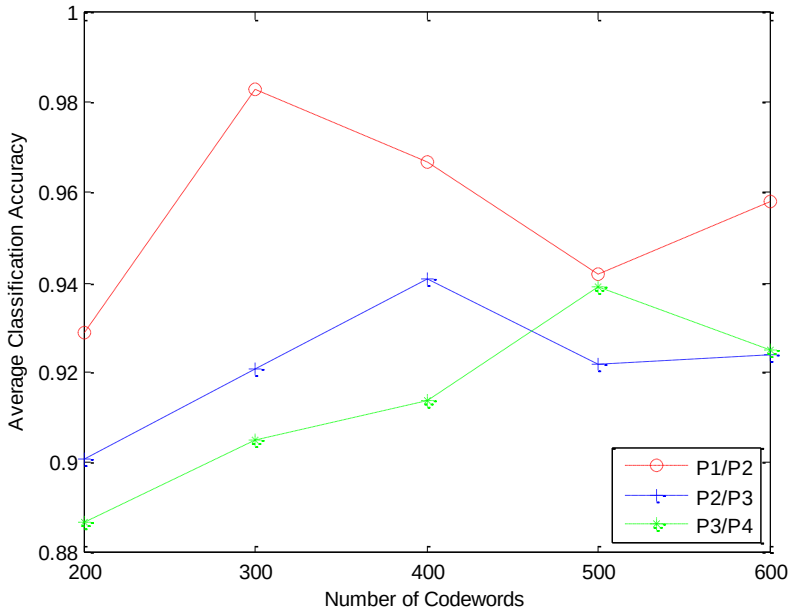


Figure 4.4 Classification accuracy based on different number of codewords

CHAPTER 5

SCENE CLASSIFICATION

The construction of the vocabulary is a crucial part in the task of scene classification. When the features of images are extracted, the vocabulary is built based on clustering algorithms using competitive learning scheme.

Hard competitive learning is one of the categories of typical competitive learning algorithms. It involves winner-take-all networks to determine the most responsive cell to a given input. The LBG algorithm [79] and k-means clustering algorithm [80] are typical hard competitive learning algorithms with the advantage of adaptive on-line update. Among the on-line methods variants with constant adaptation rate can be distinguished from variants with decreasing adaptation rates of different kinds.

However, many problems may occur in winner-take-all learning methods. One of the significant problems is the existence of dead units which never win for an input signal during the competition. The existence of these units is harmful since they take the resources of the network but do not contribute to the network.

This problem was addressed by frequency sensitive competitive learning (FSCL)[81], which was originally formulated to remedy the problem of under-utilization of parts of a codebook in vector quantization . By the introduction of the distortion which is the number of times that exemplar was the winner in the past, the highly winning exemplars were discouraged from attracting new inputs. Thus, eventually all units participate in the quantization of the data space.

Rival penalized competitive learning (RPCL) proposed by Xu et al[96] is developed based on FSCL and is supposed to be able to automatically determine the number of clusters. Lu et al[97] implemented RPCL-based method in scene classification problem, but they did not use RPCL to select the number of clusters, since as Ma and Wang [98] illustrated that it is very difficult to obtain the delearning rate in RPCL.

In this paper, I will choose FSCL based clustering method to generate the vocabulary for scene classification.

5.1 Visual Words Extraction Using SIFT

Selection of features for object recognition is crucial, since unstable features may produce unreliable recognition with changes in rotation, scaling or illumination. SIFT has been designed for extracting highly discriminative local image features that are invariant to image scaling and rotation, and partially invariant to change in illumination and viewpoint. Key-points in SIFT are derived by taking the extrema of the difference of Gaussians with multiple scales, hence, these points are supposed to be invariant to scaling. Once the key-points have been identified, the gradient directions of pixels in the vicinity of each key-point are separated into 36-bins, where each bin covers a 100 range, and each sample in the bin is weighted by the corresponding gradient magnitude. One or more orientations corresponding to the bins with the highest bin value or within 80% of the highest value are assigned to a key point, and rotation invariance is achieved by representing the key-points by descriptors (feature vectors) computed relative to this (these) orientation(s). The feature vector for each key point is formed using the 4x4 descriptor. The visual

words of image are represented by these extracted k-dimensional key points extracted by the SIFT algorithm.

5.2 Vocabulary Building Using FSCL Clustering

When the k-dimensional visual words are extracted by the SIFT method, a codebook should be built based on these visual words from the training images. Suppose the codebook C consists of N codewords C_i with k demensions, then, after applying clustering algorithm to the SIFT descriptors for all the training images, each codeword C_i can be represented as each center for corresponding cluster. Though the traditional K-means clustering method is proved to both accurate and efficient, it suffers from dead neuron problem, thus, I decide to use FSCL cluster method to obtain the codewords.

In frequency sensitive competitive learning (FSCL), suppose there are N units, let $\{m_j\} j = 1 \ldots N$ and $\{a_j\}\, j = 1 \ldots N$ represent the position and frequency of jth unit. Each unit moves among the observation samples and seeks to be located at the center of a cluster of samples. At time t, an input data vector x_t comes, the wining is selected as the one that minimize the following criterion

$$\varepsilon_t(\theta_j) = \alpha_j \|x_t - m_j\|^2, \tag{45}$$

where α_j is the update frequency defined as $a_j = c_j/t$, where c_j is the number of times that unit j has been updated up to time t. Let $P_{j,t}$ be the label to indicate if jth unit wins in the competition at time t. Then

$$p_{j,t} = \begin{cases} 1, & \text{if } \quad j = c, \\ 0, & \text{otherwise;} \end{cases} \qquad c = arg\ min_j \varepsilon_t(\theta_j). \tag{46}$$

Each m_j is updated using

$$m_j^{new} = m_j^{old} + \eta p_{j,t}(x_t - m_j^{old}). \tag{47}$$

Where η is the learning rate.

The dead units problem is solved because frequent winners are penalized so that eventually, all the units have chance to win the competition in some time.

After the codebook is generated by using FSCL, the visual words are matched with the closest codeword by calculating the distance $\mathrm{d}(\mathrm{x}, \mathrm{C_i}), \mathrm{i} = 1, \ldots \mathrm{N}$ between the visual word and each codeword. If the jthe codeword is the closest, the visual word is represented as codeword $\mathrm{C_i}$.

5.3 Experiment

The evaluation of the proposed algorithm for scene classification is based on the datasets provided by [83]. There are six categories in this dataset which are beach, building, forest, highway, industry and mountain. I use 120 images for each category, so there are total 720 images for the evaluation. These datasets are randomly divided into to two separate sets of image, half for training and half for testing. My FSCL clustering method is compared with traditional K-means clustering method in terms of the average categorization accuracies based on different number of clusters.

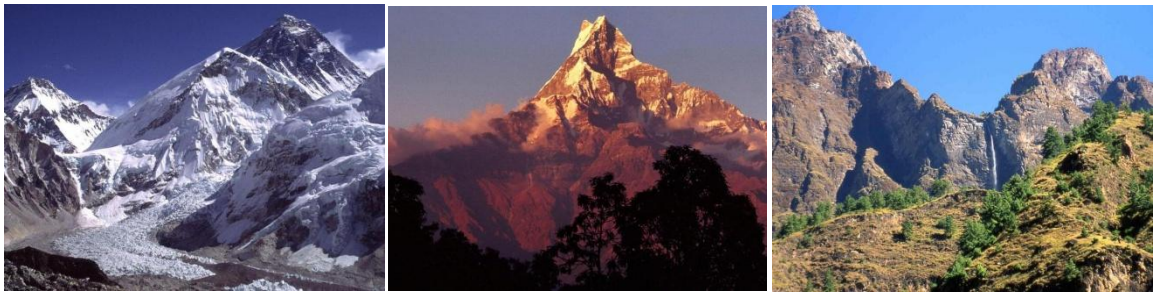
(a) Beach

(b) Building

(c) Forest

(d) Highway

(e) Industry



(f) Mountain

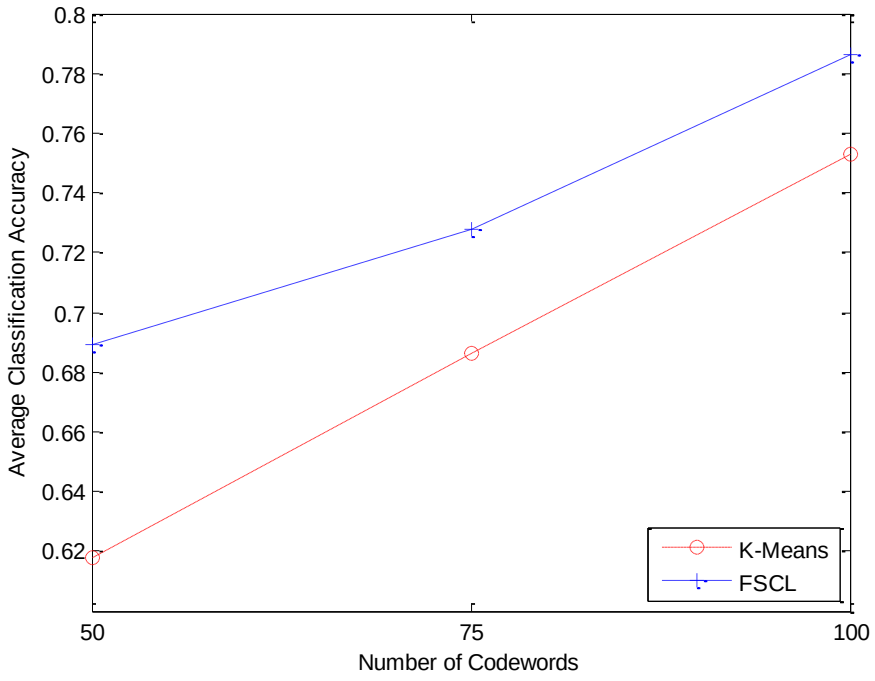
Figure 5.1 Sample images for the experiment

Canny edge detector is applied to every image, then, interesting points are sampled based on these detected edges. Visual words are formed by applying SIFT method around these interesting points in each image. In my experiment, the extracted SIFT descriptors are 128 dimensional. The vocabulary is obtained by using the FSCL clustering method among all the SIFT descriptors for the training images. In FSCL, the training time n is set as 100 and the learning rate η is set as $0.1 * e^{-0.05*(n-1)}$. Once the codebook is generated, the visual words are quantized to the closest codeword, the pLSA is applied to learn from the training images using the above mentioned EM algorithm. For a new testing image d_{new}, using EM algorithm, $P(z_k|d_{new})$ maximizes the likelihood of d_{new} with the probability $P(w_i|z_k)$ which was learned in the training step .Finally, SVM classifier is used to classify the class label for the testing images.

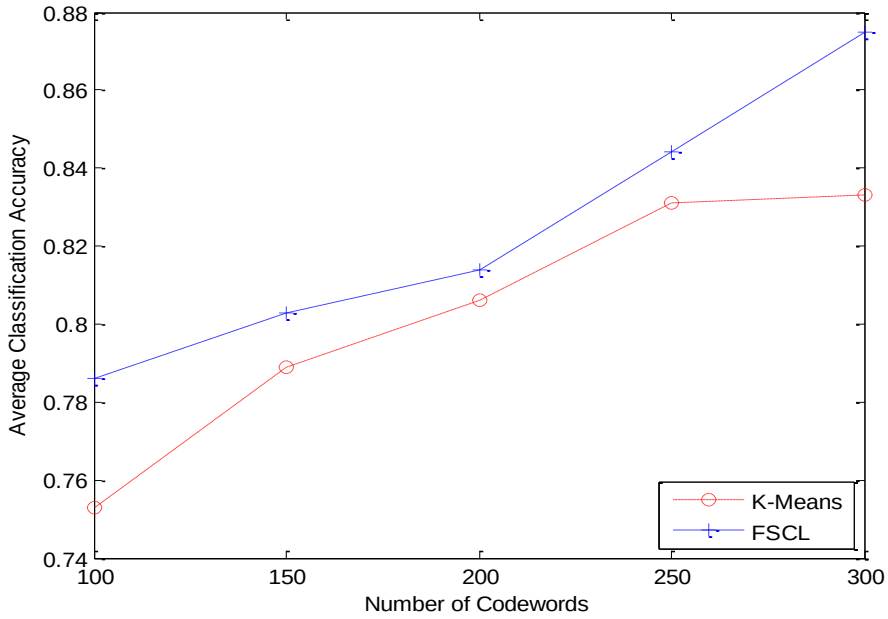
In order to test the FSCL method, I compared it with the traditional K-means method. Since the scene classification performance is relied on the number of codewords which is also the number of clusters during the codebook generation process, I compared both methods based on different number of codewords. The results for the comparison between FSCL and K-means methods are shown in Figure 5.2 with the size of codewords ranging from 50 to 500. I can find that FSCL based method outperforms the K-means based method if the number of codeword is from 50 to 500.

In order for the detailed examination, I also compute the confusion matrix for FSCL and K-means method when the codeword size is 400. In Table 5.1 and Table 5.2, I can observe that classification accuracy of FSCL based method is higher than K-

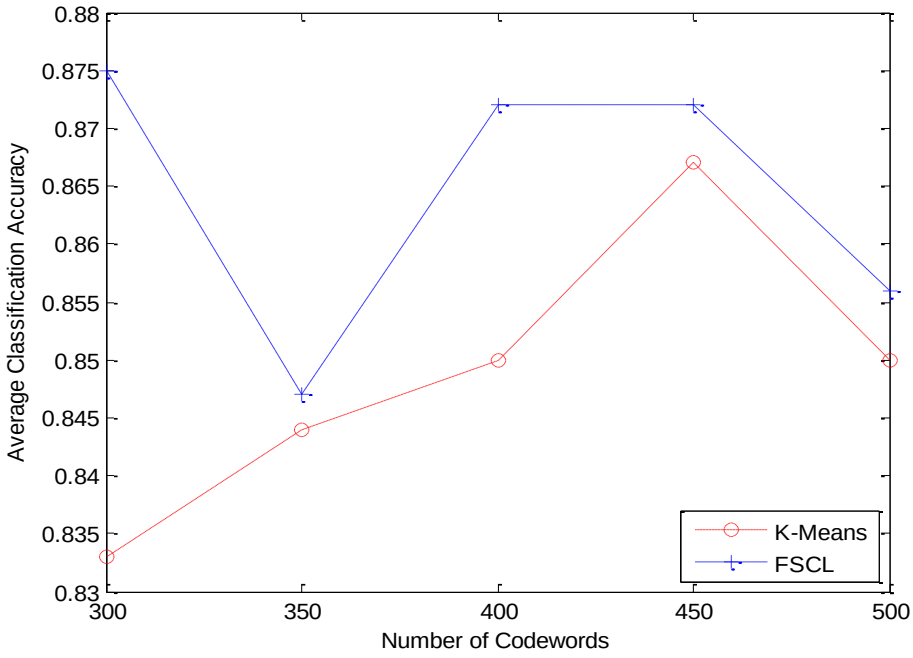
means based method in five categories, namely Category 1 (Beach), Category 2 (Building), Category 3 (Forest), Category 4 (Highway) and Category 6 (Mountain), while in Category 5 (Industry), the misclassification rate is higher for FSCL than K-means.



(a) Classification accuracy with codeword size from 50 to 100



(b) Classification accuracy with codeword size from 100 to 300



(c) Classification accuracy with codeword size from 100 to 300

Figure 5.2 Comparison results between FSCL and K-means methods

Table 5.1 Confusion Matrix of Scene Classification based on K-means Method with 400 Codewords

	Category1	Category2	Category3	Category4	Category5	Category6	Accuracy
Category1	58	0	0	0	2	0	96.67
Category2	13	45	2	0	0	0	75.00
Category3	0	5	54	1	0	0	90.00
Category4	0	0	4	56	0	0	93.33
Category5	1	0	8	1	49	1	81.67
Category6	1	0	0	0	14	45	75.00

Table 5.2 Confusion Matrix of Scene Classification based on FSCL Method with 400 Codewords

	Category1	Category2	Category3	Category4	Category5	Category6	Accuracy
Category1	60	0	0	0	0	0	100.00
Category2	12	46	1	0	1	0	76.67
Category3	0	4	54	1	1	0	90.00
Category4	0	0	2	58	0	0	96.67
Category5	1	0	10	1	47	1	78.33
Category6	0	0	0	0	11	49	81.67

CHAPTER 6

CONCLUSION

In this thesis, I presented novel method of scale invariant feature extraction and probabilistic model construction.

For the object tracking problem, new multiple feature fusion algorithm is developed in particle filter frame work. Results of experimentation indicate that my algorithm with occlusion tolerance as high as 80% outperforms some representative real-time tracking algorithms that may be considered as viable competitors for the current approach. The second experiment with human data demonstrates that my UPF algorithm outperforms the traditional Mean Shift algorithm in tracking accuracy with results very close to the ground truth. Thus it may be concluded that the algorithm does not sacrifice any tracking accuracy to achieve robustness against occlusions.

For the video stabilization problem, PCA-SIFT features and adaptive particle filter methods are designed to compute the motion of neighboring frames. Particle filter can refine the results of the initial guess of the global motion. Adaptively selecting the particle size and the noise of the particle filter reduces the complexity of the computation. A novel SIFT-BMSE cost function is also in the particle filter framework proposed here. Experiments have confirmed its effectiveness and accuracy.

For the WCE video segmentation problem, I uses SIFT algorithm for vocabulary building and video frame classification. The experiment confirms that my pLSA model performs better than the traditional SVM classifiers based on the SIFT

feature. However, the difficulty in constructing constant sized SIFT feature vectors for SVM training and testing suggests that the SIFT features may not be the right features for use in an imaging application with an SVM or any other classifier with a supervised learning scheme. In my pLSA approach, the SIFT features derived in color spaces performed as expected much better than the grey SIFT features. However, the surprise result here is that the SIFT features in the RGB color space yield higher classification accuracy than those in the HSV color space, unlike in some prior works in supervised classification approaches for imaging applications. Comparison of my results with the state of the art SVM classifiers for endoscopy video segmentation indicates that pLSA, despite its unsupervised mode of operation, yields competitive performance. These results could be improved using better feature sets and properly chosen codebook sizes.

For the scene classification method, I proposed a FSCL based method to provide a good vocabulary generation method in scene classification. I further compared this method with the widely used K-means clustering method. The experiment results show that my proposed method outperforms the K-means based method even if the codewords numbers in the vocabulary is changing.

REFERENCE

[1] K. Rangachar and R. C. Jain. Computer Vision; Principles. IEEE Computer Society Press, 199.

[2] K. Ikeuchi, T. Shakunaga, M. Wheeler, and T. Yamazaki.Invariant histograms and deformable template matching for target recognition. In Proceedings of IEEE CVPR 96, pages 100–105, 1996.

[3] I. Haritaoglu, L. S. Davis, and D. Harwood. w4 who? when?where? what? a real time system for detecing and tracking people. In FGR98 (submitted), 1998.

[4] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):780–785, 1997.

[5] D. Koller, K. Danilidis, and H.-H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. International Jmynal of Computer Vision, 10(3):257–281, 1993.

[6] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In Proceedings of IEEE CVPR 97, pages 495–501,1997.

[7]Thomaz Zielke, Michael Brauckmann, and Irner von Seelen. Intensity and edge based symmetry detection with an application to car following. CVGIP: Image Understanding. 58(2):177-190, September 1993.

[8] UI Regensburger and Volker Graefe. Visual Recognition of obstacles on roads.In Proceedings of the IEEE/RSJ/GI international Conference on Intelligent Robots and Systems, pages 982-987, Munich, Germany, September 1994.

[9]Stefan Bohrer, Thomaz Zielke, and Volker Freiburg. An integrated obstacle detection framework for intelligent cruise control on motorways. In Proceedings of the Intelligent Vehicle Symposium, pages 276-281, Detroit, Mi, September 1995.

[10] S. M. Smith. ASSET 2:Real-time motion segmentation and shape tracking. In Proceedings of the International Conference on Computer Vision. Pages 237-244. Cambridge, MA, 1995.

[11] Walter J. Gillner. Motion based vehicle detection on motorways. In Proceedings of the Intelligent Vehicles Symposium, pages 483-487, Detroit, MI, September 1995.

[12] W. Kruger, W. Enkelmann, and S. Rossle. Real time estimation and tracking of optical flow vectors for obstacle detection. In Proceedings of the Intelligent Vehicles Symposium, Pages 304-309. Detroit, Mi, September 1995.

[13] A. G. H. J. Yang, Y. Jiang and C. Ngo. Evaluating bag-of-visual-words representations in scene classification. In Proc. of International Workshop on Multimedia Information Retrieval, pages 197-206,2007.

[14] C. J. B. J. C. Gemert, J. M. Geusebroek and A. W. M. Smeulders. Kernel codebooks for scene categorization. In Proc. of the 10th European Conference on Computer Vision, pages 696-709, 2008.

[15] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. Machine Learning, 42(1):177-196, 2001.

[16] J. M. O. F. Monay, P. Quelhas and D. Gatica. Contextual classification of image patches with latent aspect models. Jmynal on Image and Video Processing, pages 1-20, 2008.

[17] A. Z. A. Bosch and X. Munoz. Scene classification via plsa. In Proc. of European Conference on Computer Vision, pages 517-530, 2006.

[18] A. Z. A. Bosch and X. Munoz. Scene classification using a hybrid generative/discriminative approach.IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(4):712-727, 2008.

[19] F. Monay and D. Gatica. Modeling semantic aspects for cross-media image indexing. IEEE Transaction on Pattern Analysis and Machine Intelligence, 29(10):1802-1817, 2007.

[20] F. Monay and D. G. Perez. Plsa-based image autoannotation: constraining the latent space. In Proc. of the 12th annual ACM International Conference on Multimedia, pages 348-351, 2004.

[21] Arulampalam, S., Maskell, S., Gordon, and T. N.,Clapp. A tutorial on particle filters for on-line non-linear/non-Gaussian bayesian tracking. IEEE Trans. Signal Process, pages 174–188, 2002.

[22] Kobayashi, Y., Sugimura, D., Sato, Y.: 3d head tracking using the particle filter with cascaded classifiers. In: BMVC (2006)

[23] Nummiaro, K., Koller-Meier, E., Svoboda, T., Roth, D., van Gool, L.: Color-based

object tracking in multi-camera environment. In: 25th Pattern Recognition Symposium,

DAGM (2003)

[24] T. Chang and S. Gong. “Tracking multiple people with a multicamera system ”. In IEEEWorkshop onMulti-Object Tracking, 2001.

[25] Hu, W.-M., Hu, M., Zhou, X., Tan, T.-N., Lou, J., Maybank, S.J.: Principal axis based correspondence between multiple cameras for people tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(4), 663–671 (2006)

[26] Y. Boykov and D. Huttenlocher. Adaptive Bayesian recognition in tracking rigid objects. In Comp. Vis. and Pattern Rec, pages 697–704, 2000.

[27] J. Black, T. Ellis, and P. Rosin. Multi view image surveillance and tracking. IEEE Workshop on Motion and Video Computing,, pages 169–174, 2002.

[28] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. Int. J. of Comp. Vis, pages 5–28, 1998.

[29] Bhattacharyya, A.: On a measure of divergence between two statistical populations de_ned by probability distributions Bull. Calcutta Math. Soc.35 (1943)99-109

[30] S. Khan, O. Javed, Z. Rasheed, and M. Shah. “Human tracking in multiple cameras”. In Proceedings of ICCV, 2001.

[31] T. Zhao, M. Aggarwal, R. Kumar, and H. Sawhney. Real-time wide area multicamera stereo tracking. IEEE Conference on Computer Vision and Pattern Recognition, 1:976–983, 2005.

[32]O. Javed, Z. Rasheed, O. Alatas, and M. Shah. “KnightM: A real time surveillance system for multiple overlapping and non-overlapping cameras”. In Proceedings of ICME, 2003.

[33] K. Kim and L. Davis, “Multi-Camera Tracking and Segmentation of Occluded People on Ground Plane Using Search-Guided Particle Filtering,” Proc. Ninth European Conf. Computer Vision,2006.

[34] L. Lee, R. Romano, and G. Stein. “Monitoring activities from multiple video streams: Establishing a common coordinate frame”. IEEE Trans. on PAMI, 22(8):758–768, Aug 2000.

[35] R. E. Kalman. A new approach to linear filtering and prediction problems. Trans. ASME-J. Basic Eng., 82:35–45, 1960.

[36] R. Rosales and S. Sclaroff. 3D trajectory recovery for tracking multiple objects and trajectory guided recognition of actions. In Comp. Vis. and Pattern Rec, pages 117–123, 1999.

[37] Arulampalam, S., Maskell, S., Gordon, and T. N., Clapp. A tutorial on particle filters for on-line non-linear/non-Gaussian bayesian tracking. IEEE Trans. Signal Process, pages 174–188, 2002.

[38] N. K., K.-M. E., and V. G. L. A color-based particle filter. In Workshop on Generative-Model-Based Vision, June 2002.

[39] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking,” IEEE Trans. Signal Process., vol. 50, no. 2, pp. 174–188, Feb. 2002.

[40] S. Erturk, “Digital image stabilization with sub-image phase correlation based global motion estimation,” IEEE Trans. on Consumer Electronics, Nov. 2003.

[41] L. Xual, and X. Lin, “Digital Image Stabilization Based on Circular Block Matching,” IEEE Trans. on Consumer Electronics, Vol. 52, No.2, May 2006.

[42] A. Censi, A. Fusiello, and V. Roberto. “Image stabilization by features tracking,” International Conference on Image Analysis and Processing, 1999.

[43] D. LoI, “Distinctive image features from scale-invariant keypoints,” International Jmynal of Computer Vision, Vol. 60, No.2, pp.91-110, 2004.

[44] C. Chung, and H. Chen, “Feature-based full-frame image Stabiliztion,” IEEE International Symposium on Multimedia 2007

[45] R. Hu, R. Shi, I Shen, and W.Chen, “Video Stabilization using scale-invariant features” Proceedings of the 11th International Conference Information Visualization ,pp**.** 871-877 , 2007.

[46] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato, “SIFT features tracking for video stabilization” Proceedings of the 14th International Conference on Image Analysis and Processing, pp. 825-830, 2007.

[47] Y. Ke, and R. Sukthankar, “PCA-SIFT: A More Distinctive Representation for Local Image Descriptors” Computer Vision and Pattern Recognition, 2004.

[48] Z. Pan, and C. Ngo, “Selective object stabilization for home video consumers,” IEEE Tran .on Consumer Electronics, Vol, 51, No.4, Nov. 2005.

[49] A. A. Yeni, and S. Erturk, “Sast digital image stabilization using one bit transform based sub-image motion estimation,” IEEE Trans. on Consumer Electronics,Vol.51, No.3, Aug.2005.

[50] A. Litvin, J. Konrad, and W.C. Karl, “Probabilistic video stabilization using kalman filtering and mosaicking,” in IS&T/SPIE Symposium on Electronic Imaging , 2003, pp.663-674.

[51] S. Zhou, R. Chellappa, and B. Moghaddam, “Visual tracking and recognition using appearance-adaptive models in particle filters”. IEEE Transactions on Image Processing (TIP), Vol. 11, pp. 1434-1456, November 2004.

[52] J. Yang, D. Schonfeld, and M. Mohamed, “Robust video stabilization based on particle filtering tracking of projected camera motion,” IEEE Transactions on Circuits and Systems for Video Technology, accepted for future publication.

[53] S. Krishnan, P. Wang, C. Kugean, M. Tjoa,, ”Classification of endoscopic images based on texture and neural network” Proc. 23rd Annual IEEE Int. Conf. in Engineering in Medicine and Biology, (4), pp. 3691-3695, 2001.

[54] A. Maieron et al,”Multicenter retrospective evaluation of capsule endoscopy in clinical routine,” Endoscopy, vol. 36, pp. 864-868, 2004.

[55] G. Gay,M. Delvaux, and J. Key, ”The role of video capsule endoscopy in the diagnosis of digestive diseases: A review of current possibilities,” Endoscopy, vol. 36, pp. 913-920, 2004.

[56] P. Swain, ”Wireless capsule endoscopy and Crohn’s disease,” , Gut, vol. 54, pp. 323-326, 2005.

[57] A. Culliford, J. Daly, B. Diamond, M. Rubin, and P. H. R. Green, ”The value of wireless capsule endoscopy in patients with complicated celiac disease,” Gastrointestinal Endoscopy, vol. 62, no. 1, pp. 55-61, 2005.

[58] M.T. Coimbra, J.P.S. Cunha, ”MPEG-7 Visual Descriptors-Contributions for Automated Feature Extraction in Capsule Endoscopy”, IEEE transactions on circuits and systems for video technology, Vol. 16, No. 5, pp. 628-637, 2006.

[59] M. Boulougmya, E. Wadge, V. S. Kodogiannis, and H. S. Chowdrey, ”Intelligent systems for computer-assisted clinical endoscopic image analyses,” in Proc. 2nd Int. Conf. Biomed. Eng., , Innsbruck, Austria, 2005, pp. 405-8.

[60] M. Mackiewicz, J. Berens and M. Fisher, ”Wireless Capsule Endoscopy Color Video Segmentation,” IEEE Transaction on Medical Imaging,, vol. 27, no. 12, December,2008

[61] P. Howarth, A. Yavlinsky, D. Heesch, and S. Ruger, ”Medical image retrieval using texture, locality and color,” in Proc. Cross Language Evaluation Forum, 2005, pp. 740-749.

[62] P. Wang, S. Krishman, C. Kugean, and M. P. Tjoa, ”Classification of endoscopic images based on texture and neural network,” in Proc. 23rd Annu. Int. Conf. IEEE Eng. Med. Biol. Sci., , 2001, pp. 3691-5.

[63] P. Spyridonos, F. Vilarino, J. Vitria’, F. Azpiroz, and P. Radeva, ”Anisotropic feature extraction from endoluminal images for detection of intestinal contractions,” in Proc. MICCAI, 2006, vol. 2, pp. 161-168.

[64] J. Cunha, M.Coimbra, P. Campos, and J.M. Soares, ”Automated topographic segmentation and transit time estimation in endoscopic capsule exams,” IEEE Trans. Med. Imag., vol. 27, no. 1, pp. 19-27, Jul. 2008.

[65] F. Jurie, B. Triggs, ”Creating efficient codebooks for visual recognition”, Proceedings of International Conference on Computer Vision,2005.

[66] A. Bosch, A. Zisserman, and X. Muoz, ”Scene classification using a hybrid generative/discriminative approach,” IEEE Transactions on Pattern Analysis and Machine Intelligence,vol. 30, no. 4, pp. 712-727, 2008.

[67] Y.Linde, A.Buzo, and R.M.Gray, ”An algorithm for vector quantizer design,” IEEE Transactions on Communications, vol. 28(1), pp. 84-95,1980.

[68] Lloyd, S. P. ”Least square quantization in PCM”. IEEE Transactions on Information Theory , 28 (2): 129-137.1982.

[69] L. Fei-Fei and P. Perona. ”A Bayesian hierarchical model for learning natural scene categories,” in IEEE Conference on Computer Vision and Pattern Recognition,volume 2, pages 524-531, 2005.

[70] A. Oliva and A. Torralba. ”Modeling the shape of the scene: A holistic representation of the spatial envelope” International Jmynal of Computer Vision, 42(3):145-175, 2001

[71] D. G. LoI.”Distinctive image features from scale-invariant keypoints.” International Jmynal of Computer Vision,60(2):91-110, 2004.

[72] A. Z. A. Bosch and X. Munoz.”Scene classification via plsa.” In Proc. of European Conference on Computer Vision, pages 517-530, 2006.

[73] A. G. H. J. Yang, Y. Jiang and C. Ngo. ”Evaluating bag-of-visual-words presentations in scene classification.”In Proc. of International Workshop on Multimedia Information Retrieval,pages 197-206,2007.

[74] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, T. Tuytelaars,and L. Van Gool. ”Modeling scenes with local descriptors and latent aspects.” ICCV, vol. 1:883-900, 2005.

[75] T. Hofmann. ”Unsupervised learning by probabilistic latent semantic analysis.” Machine Learning, 42(1):177-196, 2001.

[76] D. Blei , A.N., M. Jordan. Jmynal of Machine Learning Research 3: pp. 993-1022.

[77] J. Niebles, H. Wang and F. Li. ”Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words.” International Jmynal of Computer Vision, 2008.

[78] Z. Lu, Y. Peng, H. Ip,”Image categorization via robust pLSA, ” Pattern Recognition Letters, Vol. 31 , Issue 1,pages: 36-43, 2010.

[79] Y.Linde, A.Buzo, and R.M.Gray, “An algorithm for vector quantizer design,” IEEE Transactions on Communications, vol. 28(1), pp. 84-95,1980

[80] Lloyd, S. P. "Least square quantization in PCM". IEEE Transactions on Information Theory 28 (2): 129–137.1982

[81] S. C. Ahalt, A. K. Krishnamurthy, P. Chen, and D. E. Melton. Competitive learning algorithms for vector quantization. Neural Networks, 3(3):277–290, 1990.

[85] J. H. Friedman. An overview of predictive learning and function approximation. From Statistics to Neural Networks, Proc.NATO/ASI workshop, pages 1–61, 1994.

[86] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages 524-531, 2005.

[87] J. Luo and M. Boutell. Natural scene classification using overcomplete ica. Pattern Recognition, 38(10):1507-1519, 2005.

[88] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. International Jmynal of Computer Vision, 42(3):145-175, 2001

[89] D. G. LoI. Distinctive image features from scale-invariant keypoints. International Jmynal of Computer Vision, 60(2):91-110, 2004.

[90] M. Szummer and R. W. Picard. Indoor-outdoor image classification. In Proc. of IEEE International Workshop on Content-based Access of Image and Video Databases, pages 42-51, 1998.

[91] A. Z. A. Bosch and X. Munoz. Scene classification via plsa. In Proc. of European Conference on Computer Vision, pages 517-530, 2006.

[92] A. G. H. J. Yang, Y. Jiang and C. Ngo. Evaluating bag-of-visual-words representations in scene

classification. In Proc. of International Workshop on Multimedia Information Retrieval, pages 197-206,2007.

[93] J. Liu and M. Shah. Scene modeling using co-clustering. ICCV,2007.

[94] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, T. Tuytelaars,and L. Van Gool. Modeling scenes with local descriptors and latent aspects. ICCV, Vol. 1:883-900, 2005.

[95] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. Machine Learning, 42(1):177-196, 2001.

[96] D. Blei , A.N., M. Jordan. Jmynal of Machine Learning Research 3: pp. 993–1022.

[97] J. Niebles, H. Wang and F. Li. Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. International Jmynal of Computer Vision, 2008

[98] L. Xu, A. Krzyzak, E. Oja, Rival penalized competitive learning for clustering analysis, RBF net, and curve detection. IEEE Trans. Neural Networks 4 (7), 636-648, 1993

[99] Z. Lu, Y. Peng, H. Ip, Image categorization via robust pLSA, Pattern Recognition Letters. Vol. 31 , Issue 1,pages: 36-43, 2010

[100] J. Ma , T. Wang, A cost-function approach to rival penalized competitive learning (RPCL). IEEE Trans. Systems Man Cybernet., Part B 36 (4), 722–737, 2006.

[101] Nummiaro, K., Koller-Meier, E., Svoboda, T., Roth, D., van Gool, L.: Color-based object tracking in multi-camera environment. In: 25th Pattern Recognition Symposium, DAGM (2003)

[102] T. Chang and S. Gong. “Tracking multiple people with a multicamera system ”. In IEEEWorkshop onMulti-Object Tracking, 2001.

[103] Hu, W.-M., Hu, M., Zhou, X., Tan, T.-N., Lou, J., Maybank, S.J.: Principal axis based correspondence between multiple cameras for people tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(4), 663–671 (2006)

[104] Y. Boykov and D. Huttenlocher. Adaptive Bayesian recognition in tracking rigid objects. In Comp. Vis. and Pattern Rec, pages 697–704, 2000.

[105] J. Black, T. Ellis, and P. Rosin. Multi view image surveillance and tracking. IEEE Workshop on Motion and Video Computing,, pages 169–174, 2002.

[106] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. Int. J. of Comp. Vis, pages 5–28, 1998.

[107] Bhattacharyya, A.: On a measure of divergence between two statistical populations de_ned by probability distributions Bull. Calcutta Math. Soc.35 (1943)99-109

[108] S. Khan, O. Javed, Z. Rasheed, and M. Shah. “Human tracking in multiple cameras”. In Proceedings of ICCV, 2001.

[109] T. Zhao, M. Aggarwal, R. Kumar, and H. Sawhney. Real-time wide area multicamera stereo tracking. IEEE Conference on Computer Vision and Pattern Recognition, 1:976–983, 2005.

[110]O. Javed, Z. Rasheed, O. Alatas, andM. Shah. “KnightM: A real time surveillance system for multiple overlapping and non-overlapping cameras”. In Proceedings of ICME, 2003.

[111] K. Kim and L. Davis, “Multi-Camera Tracking and Segmentation of Occluded People on Ground Plane Using Search-Guided Particle Filtering,” Proc. Ninth European Conf. Computer Vision,2006.

[112] L. Lee, R. Romano, and G. Stein. “Monitoring activities from multiple video streams: Establishing a common coordinate frame”. IEEE Trans. on PAMI, 22(8):758–768, Aug 2000.

[113] R. E. Kalman. A new approach to linear filtering and prediction problems. Trans. ASME-J. Basic Eng., 82:35–45, 1960.

[114] R. Rosales and S. Sclaroff. 3D trajectory recovery for tracking multiple objects and trajectory guided recognition of actions. In Comp. Vis. and Pattern Rec, pages 117–123, 1999.

[115] Y. Shen, P. Guturu, B. P. Buckles, D. Thyagaraju, and K. Namuduri “Video Stabilization Using Principal Component Analysis and Scale Invariant Feature

Transform in Particle Filter Framework" IEEE Trans. on Consumer Electronics, Aug.2009

[116] Y. Shen, P. Guturu, T. Damarla, B. P. Buckles "Particle Filter Based Object Tracking with Discriminative Feature Extraction and Fusion" ISVC (2) 2008: 246-256

[117] D. Ang, Y. Shen, P. Duraisamy "Video Analytics for Multi-camera Traffic Surveillance" IWCTS '09

[118] P. Duraisamy, Y. Shen, X. Yuan, "Image registration error analysis using pattern recognition algorithms" SPIE, 14 September 2010, Vol. 7799

[119] P.Duraisamy, Y. Shen, K. Namuduri "Error analysis and performance estimation of two different mathematical methods for image registration" SPIE,14 September 2010, Vol. 7799