AUTOMATED VIDEO PROCESSING AND SCENE UNDERSTANDING FOR INTELLIGENT VIDEO SURVEILLANCE

A Dissertation presented to the Faculty of the Graduate School

University of Missouri-Columbia

In Partial Fulfillment

Of the Requirements for the Degree

Doctor of Philosophy

by

Yu-Chia Chung

Dr. Zhihai He, Dissertation Supervisor

DEC 2010

The undersigned, appointed by the dean of the Graduate School, have examined the dissertation entitled

AUTOMATED VIDEO PROCESSING AND SCENE UNDERSTANDING FOR INTELLIGENT VIDEO SURVEILLANCE

presented by Yu-Chia Chung,

a candidate for the degree of doctor of philosophy,

and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Zhihai He

Dr. Curt H. Davis

Dr. Tony X. Han

Dr. Ye Duan

ACKNOWLEGMENTS

My deepest thanks to Dr. Zhihai He for his inspiration and mentoring. This work could not have taken place without his kind assistance. Not only for his support on my pursuit of degree but also his enthusiasm and encouragement that guide me through many difficulties. I would also like to express my gratitude to Dr. Curt H. Davis, Dr. Ye Duan and Dr. Tony H. Han who serve as members in my doctoral committee. They provide guidance by making great examples of work ethics and showing tremendous passion for perfection.

I wish to extend my appreciation to current members in Video Processing and Communication Lab, Jay Eggert, Xiwen Zhao, Wenqing Dai, Zhongna Zhao and Li Liu, for their wonderful friendship and to all the supporting friends, Xi Chen, Xin Li, Ilker Ersoy, Filiz Bunyak, Kyung-Min Han and Ting-Ning Lin who share the wonderful experience with me in MU. A special thank goes to Dr. Guilherme N. DeSouza, Dr. Kannappan Palaniappan, Betty Barfield and Shirley Holdmeier for your invaluable help.

Last, to my beloved parents, thank you simply for everything.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

ABSTRACT

Recent advances in key technologies have enabled the deployment of surveillance video cameras on various platforms, including stationary security cameras for infrastructure protection and public safety, UAV (unmanned aerial vehicles) cameras for persistent surveillance of battlefield, and cameras on mobile agents, such as vehicles, robots, and soldiers for site survey. There is an urgent need to develop advanced computational methods and tools for automated video processing and scene understanding to support various applications.

In this dissertation, we develop advanced video processing and computer vision methods for automated video processing and scene understanding. We concentrate our efforts on the following four tightly coupled tasks:

(1) *Aerial video registration and moving object detection*. We develop new similarity measures for local motion estimation and a reliability model to analyze the reliability of local motion estimation. Based on these two components, we develop a fast and reliable global camera motion estimation and video registration for aerial video surveillance.

(2) *3-D change detection from moving cameras*. We study the problem of detecting changes from multi-source videos which are captured by different moving cameras with unknown parameters at different times. We attack this problem by exploring a hierarchy of view-invariant image patch descriptors. Based on multi-scale local binary pattern (LBP) description of super-pixels and middle-level image patch labeling, we construct a hierarchy of image patch descriptors and detect changes in the video scene using multi-scale information fusion.

(3) *Cross-view building matching and retrieval from aerial surveillance videos*. Identifying and matching buildings between camera views is useful for scene understanding, battlefield surveillance, geo-location and geo-tagging of videos and photos. Our central idea is to construct a semantically rich sketch-based representation for buildings which is invariant under large scale and perspective changes.

(4) *Collaborative video compression for UAV surveillance network*. We study the problem of a network of small UAVs with limited computational and communication resources to perform collaborative video surveillance of the target environment. Based on distributed video coding, we develop a collaborative video compression scheme for a UAV surveillance network.

Our extensive experimental results demonstrate that the proposed methods and tools for automated video processing and scene understanding are efficient and promising for surveillance applications.

CHAPTER 1

INTRODUCTION

Recent advances in key technologies have enabled the deployment of surveillance video cameras on various platforms, including stationary security cameras for infrastructure protection and public safety, UAV (unmanned aerial vehicles) cameras for persistent surveillance of battlefield, and cameras on mobile agents, such as vehicles, robots, and soldiers for site survey. The amount of information generated by an integrated suite of motion imagery sensors is massive, especially in ubiquitous and persistent aerial video surveillance. Important video segments and interesting events are often buried in the massive source video and hidden in crowds of uninteresting objects. Simply bringing all source information directly to human analysts is a cognitive disaster. There is an urgent need to develop advanced computational methods and tools for automated video processing and scene understanding to support various applications.

Video surveillance usually aims at detecting and locating target for tactical intelligence as in role of maritime patrol missions [104]. Information mining in the cluttered video syntax is challenging even for human analysts. Therefore, in this ubiquitous and content-rich environment it exhibits a critical need for automated video processing. The fundamental toward video geospatial exploitation is registration [5, 10]. By definition, registration is to establish the correspondence across images. It is the enabling step toward video summarization functionalities of geo-tagging, tracking and change detection.

Registration approach usually involve direct, [11], and indirect, [6, 7], approaches to explore the spatial-temporal behavior of video frames.

In order to perform imagery data exploitation, the automated processing also hinges on large-scale multi-source information retrieval, [108]. The deployments of various surveillance cameras acquire massive video footage in the rapid pace nowadays. Bringing the raw videos direct to human analyst is a tedious cognitive challenge. Therefore, content-based video retrieval (CBVR) has become a promising direction, [102].

Automated video content retrieval is at the core of intelligence video surveillance. Despite the active research activity for content-based image retrieval, [103], video retrieval still leverage the traditional annotation in searching engine, e.g. text and audio. Recent research of video retrieval usually based on visual feature descriptor and the similarity metric defined accordingly, e.g. SIFT-like features and GLOH, [23]. Video retrieval also works in conjunction with a variety of important applications for surveillance practice such as object recognition and tracking [21, 82], scene understanding [30] and traffic flow monitoring, [105].

With the development of key technology, ground-aerial surveillance networking has become an increasingly important source of intelligence for situational awareness and decision making [3, 84]. A swarm of unmanned ground vehicle (UGV) and UAVs can coordinated a network as autonomous agents, [106]. The idea of mobile video surveillance network poses an extended dimensionality for video exploitation from multi-source videos. This new dimension is expanded on collaborative video processing of wide baseline matching [62], multi-view stereo [55, 97] and object tracking [59]. In need of efficient video archiving, there have been active research of distributive video coding (DVC) to cope with the multi-source scenario, [77, 80].

In this work, we aim to develop a set of automatic computation and communication tools to support intelligent video surveillance. As illustrated in Figure 1.1, a typical video surveillance for battlefield intelligence consists of aerial and ground video surveillance. For aerial video surveillance, a network of aerial vehicles, such as UAVs, collaborate with each other to collect important information on the ground. In this collaborative surveillance platform, the master UAV is larger in size, flying at a higher altitude, and guiding a group of slave M-UAVs that can fly at much lower altitudes. Here we assume that the master UAV has sufficient computational resources for real-time geospatial registration and is able to communicate with the control center over a wireless channel. After precise geospatial registration and global motion compensation of its camera view, the master UAV identifies and selects both static and dynamic targets for more detailed surveillance. The master UAV distributes the geospatial locations and visual target context to the slave M-UAVs, directing and guiding them to track each target. The slave M-UAV, flying a lower altitude, tracks and locks the target inside its camera view to capture detailed video information. After highly efficient video compression, the slave M-UAV forwards the compressed target video information to the master UAV. The master UAV multiplexes all the target video information from the slave M-UAVs and the geospatial location meta data of each target, and forwards these to the control center for automatic or semi-automatic (with human assistance) target recognition, information analysis, and decision making.

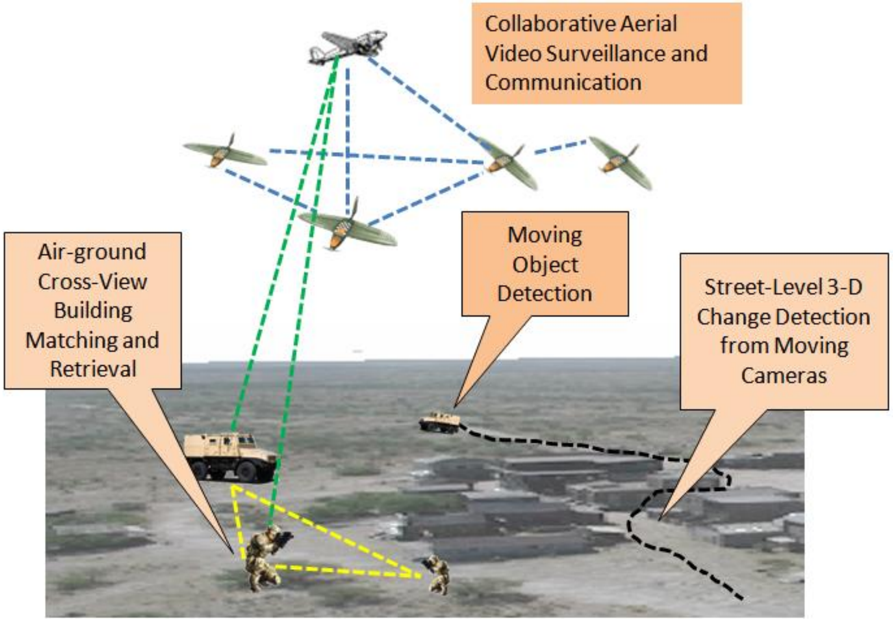

Collaborative Aerial Video Surveillance and Communication
Air-ground Cross-View Building Matching and Retrieval
Moving Object Detection
Street-Level 3-D Change Detection from Moving Cameras

Figure 1.1: Illustration of air-ground intelligent video surveillance.

On the ground, we have soldiers and ground combat vehicles. They have cameras onboard to survey the operation environment. One important task in ground-level surveillance or street-level survey missions is to identify changes, which might indicate adversary actions or potential hazards. This is the so-called *3-D change detection* problem. Furthermore, the soldiers wish to have a global awareness of the operation environment. For example, before they decide to enter an unknown building, they wish to retrieve and access the aerials surveillance video database to understand how the building looks like from different perspectives and its surrounding environment. To this end, we propose to develop air-ground building matching methods to efficiently retrieve aerial surveillance videos using ground-level images or videos. In the following, we provide a brief overview of the four major problems to be studied in this dissertation.

A) Aerial Video Registration and Moving Object Detection

Airborne surveillance has been proven an effective and importance practice for information analysis in national security. Remote sensors carried by man-pilot aircraft or unmanned aerial vehicles can monitor and secure a wide-range area for execution of specified tasks such as target searching and land cover survey. To interpret the information gathered by on-board sensors is the essential part of airborne surveillance. In the mission of man-pilot aircraft remote sensing, data analysis can be done simultaneously by human intelligence, along way the cruise. With unmanned aerial vehicles (UAVs), however, the absence of on-board human analysts would lead us to solutions like automated system or offline ground control for data processing and analysis. To achieve the real-time tactical airborne surveillance, specifically with the unmanned aerial vehicles without ground control, a visual intelligence system with computer vision tool is highly demanded. In this section we will introduce common applications of airborne surveillance and how the automated video processing plays an important role in airborne imagery.

Global motion estimation is the enabling step for many important video exploitation tasks in aerial video surveillance, including video registration, moving object detection and tracking. An efficient global motion estimation scheme for aerial video surveillance should be low-complexity, accurate, and reliable. In this work, we explore a number of methods and approaches to deal with the inherent uncertainty in motion estimation and develop a low-complexity, accurate and reliable scheme to estimate global camera motion for video registration and moving object detection. More specifically, we develop a block-based image data classification scheme to select those image regions, called *structural blocks*, which have distinctive features for reliable motion estimation. We use salient image features which are invariant to scale changes, camera zooming, and rotations and develop an adaptive compound distance metric for robust local motion estimation. We analyze the reliability of local motion estimation results and use this reliability measure as a weighting factor to determine the importance level of each local motion estimation result in global camera motion estimation. We develop a progressive scheme to detect moving objects, separate them from the background, and refine the global motion estimation. Our extensive simulation results and performance comparison with existing global motion estimation algorithms demonstrate that the proposed scheme is accurate, reliable, and has low computational complexity.

B) Air-Ground Cross-View Building Matching and Aerial Video Retrieval

In this work, we address the problem of building recognition across two camera views with large changes in scales and viewpoints. The main idea is to construct a semantically rich sketch-based representation for buildings which is invariant under large scale and perspective changes. After multi-scale maximally stable extremal regions (MSER) detection, the proposed approach finds repeated structural components of buildings, such as window, doors, and facades, and extracts semantically rich features, which are organized into a sketch-based representation of buildings. These descriptors are then clustered in association with different planes of the building and matched across video frames using spectral graph analysis. Our experiments demonstrate that the proposed approach outperforms SIFT-based matching schemes, especially for images with large viewpoint changes.

C) 3-D Change Detection from Multi-Source Videos

We study the problem of detecting changes from multi-source videos which are captured by different moving cameras with unknown parameters at different times. We attack this problem by exploring a hierarchy of view-invariant image patch descriptors. Using the five-point algorithm, SIFT and RANSAC, we track the relative camera pose within each video and obtain an approximate cross-view registration and alignment of selected video frames. Based on multi-scale local binary pattern (LBP) description of super-pixels and middle-level image patch labeling, we construct a hierarchy of image patch descriptors and detect changes in the video scene using multi-scale information fusion with Choquet integrals. We have established an image/video database for 3-D change detection and will make it available for public use. Our extensive experimental results demonstrate that the proposed method achieves a detection rate of 61% at a false alarm rate of 2% while other approaches based on conventional local photometric image descriptors fail to detect changes in the 3-D environment.

D) Distributed Coding for Collaborative Aerial Video Surveillance

This work focuses on building a power-efficient video encoder for a collaborative surveillance network. The key idea of this work is to remove the computational burden from the onboard encoder and shift the bulk of computing to its decoder. Here a channel coding scheme is introduced to build the efficient video encoder. With this idea, for a routine-fly UAV drone, it only needs to encode video with few parity-check or syndrome bits. Unseen object or changed areas are recorded by correcting the previously footage on its flight routine. This codec suits even better for fixed-camera surveillance video, since the lack of camera motion renders high correlation on video frames. Here we focus on tackling the aerial surveillance problem because mobile units require efficient codec to cope with the power consumption problem.

The rest of the chapters are organized as following: Chapter 2 presents a work of robust motion field analysis paradigm for video registration. Chapter 3 elaborates a video retrieval problem of manmade structures with a CBVR solution. A 3D change detection problem for multi-source videos is discussed in chapter 4. Chapter 5 details a collaborated video coding scheme which copes with the multi-source video encoding.

CHAPTER 2

MOTION ANALYSIS FOR VIDEO REGISTRATION

Recent advances in key technologies have enabled the development of a widening variety of platforms, including unmanned aerial vehicles (UAV), for surveillance and intelligence gathering. Aerial surveillance videos have become an increasingly important source of information for situational awareness and decision making [3, 84]. In aerial video surveillance, the UAVs fly over the areas of interest, capture videos about targets, events, and their environmental context for further information analysis and decision making. Compared to conventional videos, such as movies, news and sports videos, aerial surveillance videos have their unique characteristics: the content change in the video sequence is dominated by global camera motion. Here, the global camera motion includes camera zooming, rotation, and changes in position and perspective [2]. Although there might be local motion of objects (e.g. persons and vehicles), they only contribute to a small portion of the video scene [1, 4].

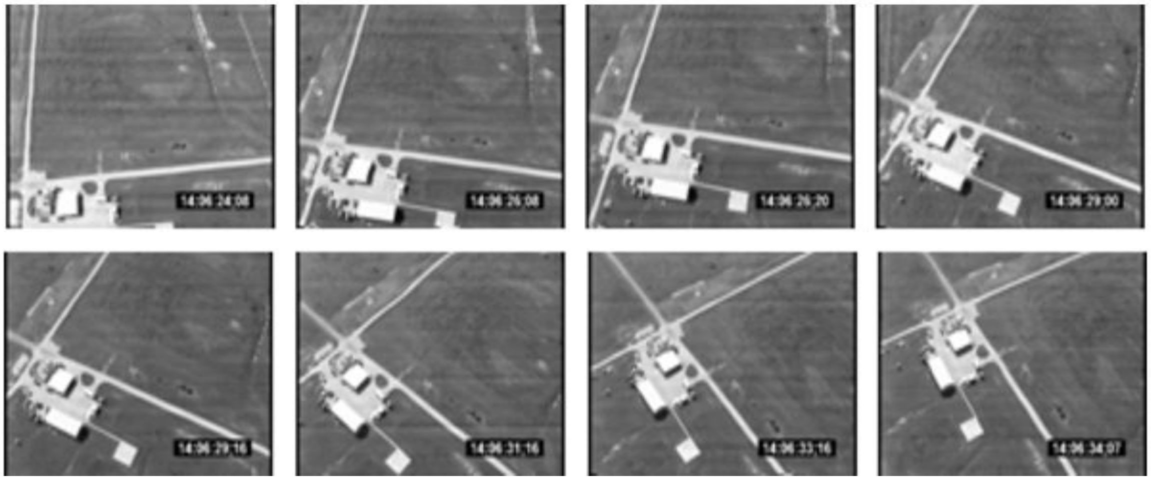


Figure 2.1: Sample video frames from an aerial surveillance video.

Fig. 2.1 shows some sample video frames from an aerial surveillance video sequence. We can see that the video frames experience translation camera motion, camera zoom in/out, rotation, as well as perspective changes. Two video frames are related by a perspective transform [2, 83]. For example, let (x, y) be the pixel position of a point object in frame I_n, where n is the frame index. After global camera motion, this point object moves to a new pixel location in frame I_{n+1}, denoted by (X, Y), as illustrated in Fig. 2.2. The relationship between (x, y) and (X, Y) is given by

$$\begin{bmatrix} X * W \\ Y * W \\ W \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.1}$$

Or, equivalently,

$$X = \frac{ax + by + c}{gx + hy + 1}$$

$$Y = \frac{dx + ey + f}{gx + hy + 1} \tag{2.2}$$

The objective of global motion estimation is to determine the following model parameters {*a, b, c, d, e, f, g, h*}. Note that camera translation, zoom in/out, rotation, affine transforms, are special cases of the transform in (2.2) [2]. For example, when *a, b, d, e, g,* and *h* are all zeros, the global camera motion in (2.2) reduces to camera translation.

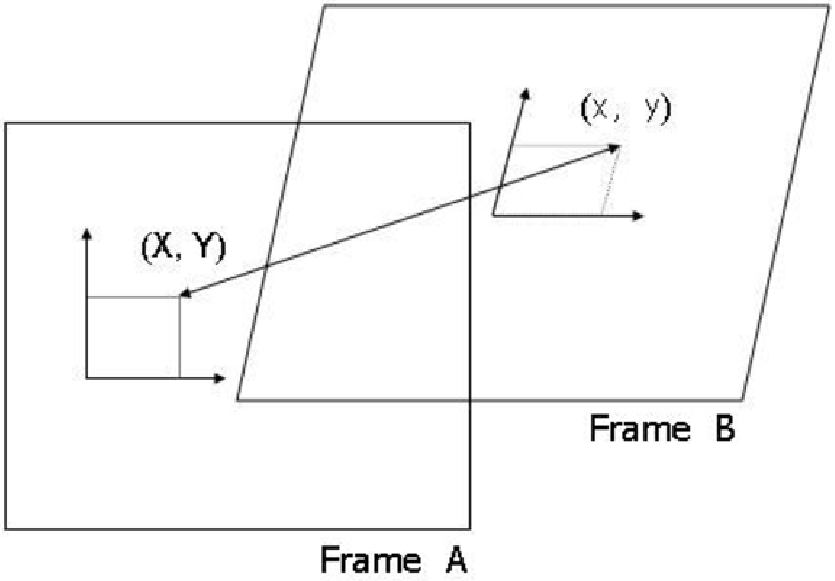


Figure 2.2: Illustration of global camera motion between two video frames.

Global motion estimation (GME) is the enabling step for many important motion imagery data exploitation tasks, including video registration which warps video frames into a common coordinate system so as to create a mosaic of the surveillance scene, moving object detection, tracking, geo-location, recognition, activity analysis, event characterization, and scene understanding [1, 4]. Global motion estimation and compensation has also been used in MPEG-4 sprite coding and Advanced Simple Profile (ASP) video coding [6, 12, 13, 5].

An efficient GME and video registration algorithm for aerial video surveillance should satisfy the following basic requirements. (1) The algorithm should have low computational complexity. This requirement becomes more critical when the algorithm operates on UAVs for online data processing, such as GME-based video compression, because the UAV often has limited computational resources. (2) The GME algorithm should be able to handle generic perspective changes. This is because the UAV, being tasked to capture detailed visual information about targets on the ground, has to fly at relatively low altitudes and often circle around the targets so as to put more image pixels on them. In this case, the camera often experiences consistent perspective changes. (3) The GME and video registration algorithm has to be resilient to noise and errors. Video capture, in practice, often has a significant amount of noise in the digital video data, including A/D (analog to digital) conversion noise, camera lens distortion, color distortion, changing light conditions, etc. The GME and registration algorithm should be sufficiently robust to survive the image noise [1].

There are two basic approaches to GME, *feature-based* and *featureless* GME algorithms. A number of feature-based GME methods are proposed in the literature [10, 11]. The work by [10] selects a subset of pixels, called dominant pixels, from the video frame and then uses a gradient method to determine the global motion parameters by minimizing the matching error of these dominant pixels between two neighboring video frames. Anchor points and invariant regions are used to establish accurate correspondence between views for global camera motion estimation. Featureless GME methods can be categorized into direct and indirect GMEs [5]. The direct GMEs determines the global motion parameters by minimizing the prediction error between corresponding pair of pixels in two frames using gradient search or other iterative methods [6, 7, 85]. This type of GME methods is able to handle generic camera model, such as the eight-parameter perspective transform and robust to image noise. However, they often suffer from high computational complexity. The indirect GME often consists of two stages; the first stage performs local motion estimation or coarse sampling in the motion field. Then the second stage follows up to refine or estimate the parametric camera models from the local motion field [5, 8, 9, 86]. The indirect GME methods often have much lower computational complexity than direct GME methods and are more suitable for real-time applications.

Although significant bodies of algorithms and methods have been developed for global motion estimation, there are a number of issues that have not been sufficiently addressed. First, the tradeoff between complexity and accuracy in global motion estimation has not been well characterized and understood. Second, motion estimation or correspondence matching between video frames often involves a significant amount of ambiguity and uncertainty. We need to develop an efficiently algorithm to manage this uncertainty to achieve reliable global motion estimation. Third, most aerial surveillance video scenes have moving objects (persons or vehicles). Since their motion is different from the background motion and is considered as noise during global motion estimation. This moving object issue needs to be carefully addressed during global motion estimation and video registration.

In this work, we propose a low-complexity reliable GME scheme for video registration and moving object detection. We propose to choose the indirect featureless global motion estimation approach because of it low computational complexity. The major contributions of this work include:

(1) Developing a block-based classification scheme to improve the accuracy of global motion estimation while reducing its computational complexity;

(2) Developing an adaptive compound distance metric for accurate motion estimation for aerial surveillance videos;

(3) Introducing a scheme to analyze the reliability of local motion estimation results and use this reliability information as a weighting factor during GME to improve its accuracy and robustness;

(4) Developing an iterative scheme to detect moving objects and progressively improve the performance of global motion estimation.

The rest of the paper is organized as follows. In Section 2.1, we give an overview of the proposed framework for global motion estimation. In Section 2.2, we present the block classification scheme and analyze its performance. Section 2.3 introduces the adaptive compound distance metric for local motion estimation. The reliability analysis is presented in Section 2.4. The proposed GME algorithm is presented in Section 2.5. Section 2.6 presents our experimental results. Section 2.7 concludes this work.

2.1 Overview of the Proposed Algorithm for Global Motion estimation

Fig. 2.3 illustrates the proposed framework for GME, video registration and moving object detection. We first classify the input aerial surveillance video into two types of blocks: *structural* and *non-structural* blocks. Structural blocks have distinctive features for reliable motion estimation. We observe that the motion estimation results of different structural blocks have different levels of accuracy and reliability. Those structural blocks with more accurate and reliable motion estimation should contribute more to the overall GME result. Therefore, we use this reliability information to weigh the importance of the corresponding structural block in the GME. Once the global camera motion parameters are obtained, we warp the video frames into a common coordinate system for video registration. After registration, we come back to estimate the local motion of non-structural blocks. Moving objects, which could be structural or non-structural blocks, are detected if their motion does not satisfy the global camera motion equation. These blocks belonging to moving objects are considered as noise for GME and removed from the list of structural blocks. The updated list of structural blocks is then used to refine the GME result. These two steps of moving object detection and global motion estimation could be repeated to detect more objects with subtle local motion and to further improve the GME accuracy. In the following sections, we will explain each component of the proposed scheme in more detail.

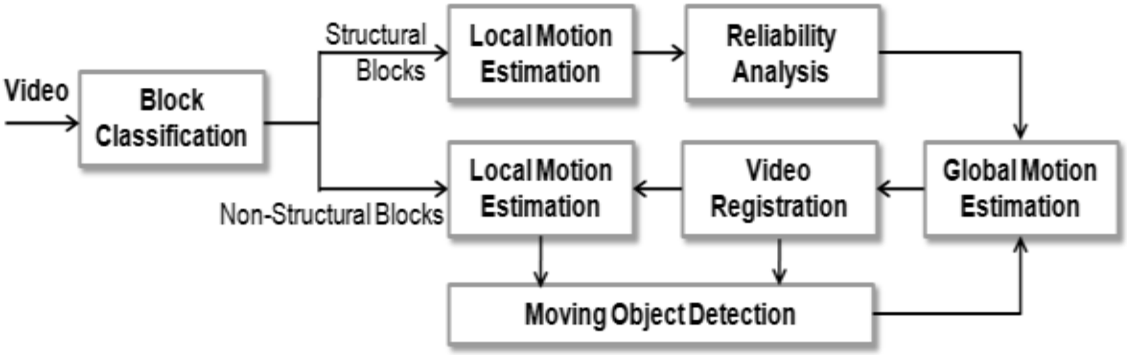


Figure 2.3: The proposed framework for global motion estimation and moving object detection.

2.2 Block Classification for Fast and Reliable Motion Analysis

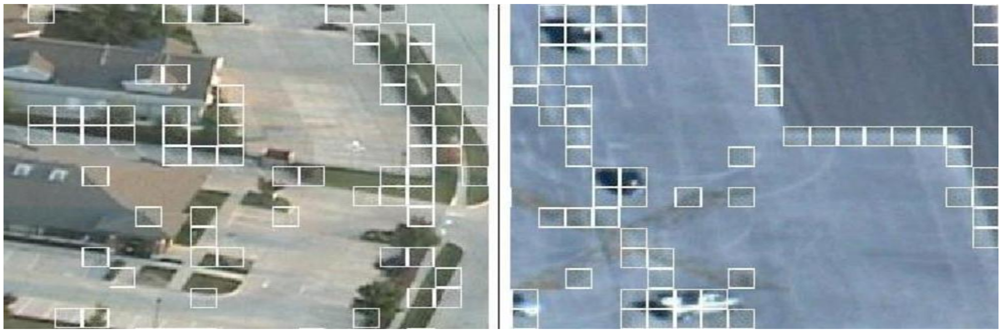
In featureless global motion estimation schemes [5, 8, 9], the video frame is often partitioned into blocks, local motion estimation is performed at every image block; and motion information of all blocks (or regions) are used to determine the global camera motion. We observe that this is not efficient. Some image blocks, for example, those in the grass area of Fig. 2.4(a) and those in the flat area of Fig. 2.4(b), have no distinctive image features and their motion estimation results are usually unreliable and noisy. This noisy motion information will degrade the overall performance of GME. Furthermore, the computational resource is also wasted on these image blocks.

Motivated by this observation, we propose to classify the image blocks into two categories: structural blocks and non-structural blocks. Structural blocks, such as buildings edges, corners, road lines, contours and patterns, have distinctive features for accurate and reliable motion estimation. Methods for detecting corner, line and curvilinear features have been developed in the literature and used for motion estimation, object tracking, and image registration [1]. These methods often operate on assumptions about the image content and are not able to handle generic videos in aerial video surveillance. For example, image registration and object tracking methods based on corner detection and tracking work well on videos of urban scenes with a lot of buildings and may fail on videos of rural or natural scenes. In addition, they often have high computational complexity.

In this work, we propose a simple yet efficient method for image content classification. Our basic idea is that, if an image block has strong low-to-medium frequency components, it often contains a significant amount of structural information, such as edges, corners, contours, or patterns. We partition the image into blocks, for example, 8×8 or 16×16 blocks. We then apply discrete cosine transform (DCT) to each block. Let $\{x_i \mid 0 \leq {}_i \leq S-1\}$ be the DCT coefficients with x_0 being the DC coefficient. For each block, we define

$$R = \frac{\sum_{i=1}^{\gamma(S-1)} (x^i)^2}{\sum_{i=1}^{S-1} (x^i)^2} \tag{2.3}$$

Here, γ is a parameter between 0 and 0.5 and R represents the ratio between the energy of low-to-medium frequency components and the overall energy. In this work, we set γ to be 0.2. We select the fraction of blocks, for example the top 15%, which have the highest structural energy ratios, as the structural blocks, and with the rest being classified as non-structural ones. Fig. 2.4 shows two examples of classification results. The structural blocks are highlighted with white boxes. In Section 2.6, we will present experimental results to demonstrate the advantage of this structural block classification scheme in GME.



(a) (b)

Figure 2.4: Classification into structural (highlighted with white boxes) and non-structural blocks.

2.3 Motion Search Using Invariant Distance Metrics

The next step of our GME scheme is to determine the motion of structural blocks. Basically, for each structural block, we need to find a block in the previous frame which has the minimum distance to it. In aerial video surveillance, the camera often experiences consistent camera motion and parameter changes, such as rotation, zooming, and changes in camera position and perspective. A desired distance metric should be invariant to camera motion, local object motion, and robust to image noise [1, 14]. In this work, we propose to explore three distance metrics: (1) sum of absolute difference, (2) intensity profile, and (3) histogram of gradient. Since each of them has both advantages and disadvantages in GME on aerial surveillance videos, we propose to form an adaptive compound distance metric from them for accurate and reliable motion estimation.

2.3.1 Sum of Absolute Difference

The sum of absolute difference (SAD) has been extensively used for motion estimation in video coding [6]. Let **A** and **B** be two image blocks and $\{a_{ij}\}$ and $\{b_{ij}\}$ be their pixels. The SAD between blocks **A** and **B** is given by

$$d_0(A,B) = \sum_{ij} \|a_{ij} - b_{ij}\| \tag{2.4}$$

One of the major advantages of this SAD metric is its low computational complexity. However, it is invariant only under translational motion and is not efficient for estimating other types of camera motion, such as rotation, zooming, and perspective changes.

2.3.2 Intensity Profile

To handle other types of camera motions, such as rotation, zoom in/out, and perspective changes, we introduce the second distance metric called *intensity profile*. Fig. 2.5 shows two example video frames with camera rotation and zooming. If a distance metric is invariant under camera rotation and zooming, those two white dots in Figs. 2.5(A) and (B) should have the minimum (or even zero) distance.

The intensity profile aims to characterize the intensity distribution in an image region around a point location. Let $O_A = (x_A, y_A)$ be the center position (pixel) of block A. Let $C(O_A, r)$ be a circle centered at O_A with a radius r, as illustrated in Fig. 2.5. The average intensity on this circle is given by

$$m(O_A, \qquad r) = \frac{1}{|C(O_A, r)|} \oint_{C(O_A, r)} I_t(x, y) d_x d_y \tag{2.5}$$

where R is the maximum radius to search. For example, we can set R to be the block width. The function $m(O_A, r)$ is called the intensity profile for pixel O_A or block A. Similarly, we can define the intensity profile for the center pixel of block B:

$$m(O_B, \qquad r) = \frac{1}{|C(O_B, r)|} \oint_{C(O_B, r)} I_{t-1}(x, y) d_x d_y \tag{2.6}$$

We can see that if pixel O_A in frame I_{t-1} moves to O_B in frame I_t, the intensity profiles $m(O_B, r)$ and $m(O_A, r)$ will be the same even with camera rotation. However, with camera zoom, $m(O_B, r)$ and $m(O_A, r)$ will be different. For example, if the camera zooms out, $m(O_A, r)$ will match the first segment of $m(O_B, r)$ after being scaled horizontally (either compressed or stretched), as illustrated in Fig. 2.5. Based on this observation, we can define a new distance measure, called *intensity profile distance*, as follows:

$$d_1(A,B) = \min_{(1-\delta)<\lambda<(1+\delta)} \max_{0<r<(R/\lambda)} |m(O_A, \lambda \cdot R) - m(O_B, r)| \tag{2.7}$$

where λ is the scaling factor, and $[1 - \delta,\ 1 + \delta]$ is the search range for λ. It can be seen that the distance (or similarity) metric $d_1(A,B)$ is invariant under camera rotation and zoom.

To compute the intensity profile distance, we first need to compute the average pixel value $m(O_A, r)$. Here, r can be integers. For example, we can choose $r = 1, 2, 3, \cdots, 16$. This computation involves table look-up to find the pixels on the circles and additions to compute their average. The min-max operation in (2.7) is performed on a 1-D array of a small size. Therefore, the overall computational complexity is acceptable. Certainly, its complexity is higher than that of SAD. In Section 2.5, we will characterize the computational complexity and performance improvement of this new distance metric in GME.

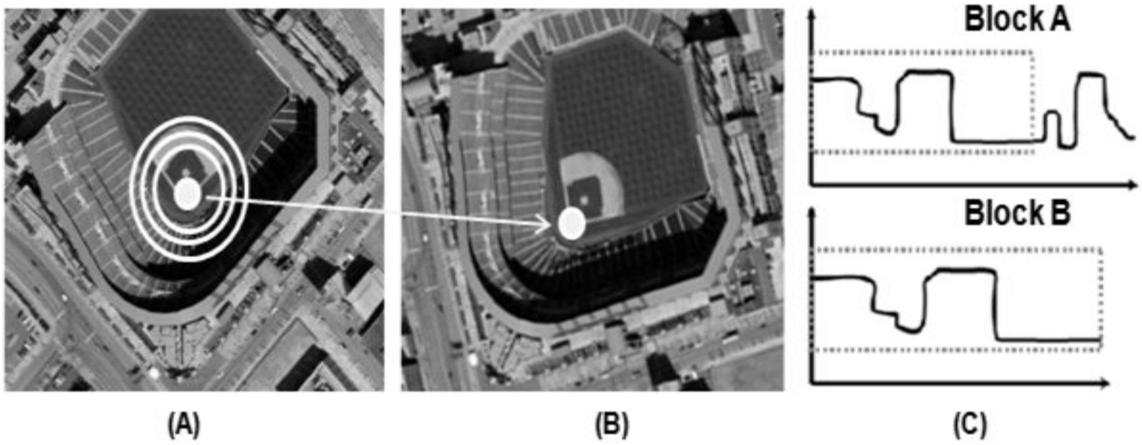


Figure 2.5: Definition of intensity profile for blocks.

2.3.3 Histogram of Gradient

The SAD captures the average difference between two blocks. The intensity profile captures the intensity distribution around a point location. We observe that, for accurate and reliable motion estimation, it is also important to consider the structural information within the neighborhood. Histogram of gradient (HOG) has been used as a descriptor in SIFT (scale-invariant feature transform) to describe key points for object recognition and tracking [14]. In this work, we propose to modify HOG to form a distance metric for accurate and reliable motion estimation. To computer HOG, we partition the block into basic units. For example, as illustrated in Fig. 2.6, we partition a 16 × 16 block into M units. For example, we can choose M to be 8 or 16. We compute the gradient for each unit. The direction of the gradient can be obtained by correlating the unit with a set of units with directional patterns and finding the direction with the maximum correlation [14]. For the convenience of computation and implementation, we can uniformly quantize this direction into K discrete values. Let $\mathbf{H}_A = \{\mathbf{H}_A[m] \mid 1 \leq m \leq \mathrm{M}\}$ be the HOG of all sub-blocks in block **A**. Similarly, we can compute the HOG **H**$_B$ for block **B**. If block **B** is rotated from block **A**, then **H**$_B$ will be a circular shifted version of **H**$_A$, i.e.,

$$H_B[m] = H_A[m \oplus \Delta m] \tag{2.8}$$

for some integer Δm. Here, $\oplus$ represents addition with modulo M and Δm is related to the amount of camera rotation. Let

$$H_A^{\Delta m} = \{H_A(m \oplus \Delta\, m \,|1 < m < M)\} \tag{2.9}$$

A new distance metric between blocks **A** and **B** can be then defined as

$$d_2(A, B) = \min_{\Delta m} \|H_A^{\Delta m} - H_B\|p \tag{2.10}$$

Here, $\|\cdot\|p$ represents the L_p-norm of the vector. In this work, we set p = 1. It can be seen that distance metric is invariant under camera rotation. The HOG is able to capture some structural information about the block and may help us improve the accuracy and reliability of motion estimation. Certainly, this distance metric has higher computational complexity than the SAD. In Section 2.5, we will analyze the computational complexity and performance improvement by this new distance metric in GME.

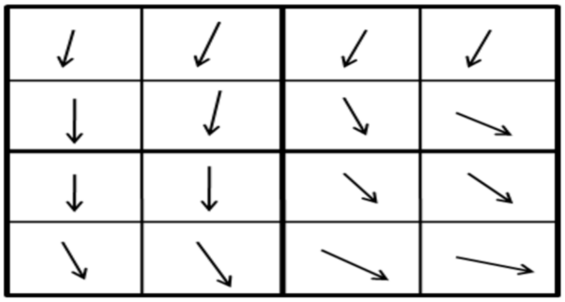


Figure 2.6: Illustration of histogram of gradient.

2.3.4 A Compound Feature for Accurate and Reliable Motion Estimation

The SAD is efficient only for estimating translational motion while the other two are efficient in handling more complicated camera motion, such as rotation and zooming. However, they have higher computational complexity than SAD. We also observe that, in aerial video surveillance, the camera motion changes over time. During most of the time, the camera just has translational motion when the aerial vehicle is flying forward. Occasionally, it has zooming or rotation when the camera is adjusted by a human operator or perspective change when the vehicle is making turns. Therefore, the selection of the distance metric should be adaptive and dynamic. Motivated by this observation, we propose to use an adaptive compound distance metric for motion search:

$$d(A,B) = w_0 \cdot d_0(A,B) + w_1 \cdot d_1(A,B) + w_2 \cdot d_2(A,B) \tag{2.11}$$

where $w_i \in \{0, 1\}$. $\boldsymbol{w} = [w_0, w_1, w_2]$ will be adaptively chosen according to the current camera motion pattern. For example, if we know from the previous frames that the camera translation motion is dominant, we set $\boldsymbol{w} = [1, 0, 0]$ and only compute the SAD which has low computational complexity. If the camera zoom is dominant, we set $\boldsymbol{w} = [0, 1, 0]$ and only compute the intensity profile. We may also compute SAD and set $\boldsymbol{w} = [\delta, 1, 0]$ where δ is a small value between 0 and 0.5. In this way, we can fuse the information from two distance metrics and hopefully increase the robustness of motion search. When the camera rotation is dominant, we may set $\boldsymbol{w} = [\delta, 1, 0]$ or $\boldsymbol{w} = [\delta, 0, 1]$. In this work, we just use the above heuristic approach to determine $\boldsymbol{w}$. In our feature work, we shall develop a more systematic way to determine $\boldsymbol{w}$.

Once the distance metric is established, for each structural block **B** in frame I_t, we can find its best match **A*** in frame I_{t-1} which has the minimum distance to B:

$$A^* = \arg\min_{A \in N(B)} d(A,B) \tag{2.12}$$

Here, N(B) represents the neighborhood (or search window) around the center position of block **B** in frame I_{t-1}.

2.4 Reliability Analysis for Global Motion Estimation

In this work, the camera motion parameters are determined from local motion estimation results of structural blocks using a least mean squared error (LSME) approach, which will be explained in Section 2.5.1. We observe that the estimation results of some structural blocks are much more accurate and reliable than those of other blocks because they have more distinctive image features. We would like those blocks with more reliable local motion estimation to play a more important role in GME. On the other hand, we would like to de-emphasize those blocks with noisy and unreliable local motion estimation when determining the global camera motion parameters. To this end, we propose to analyze the reliability of local motion estimation and use this reliability measure as a weighting factor during LMSE for camera parameter estimation.

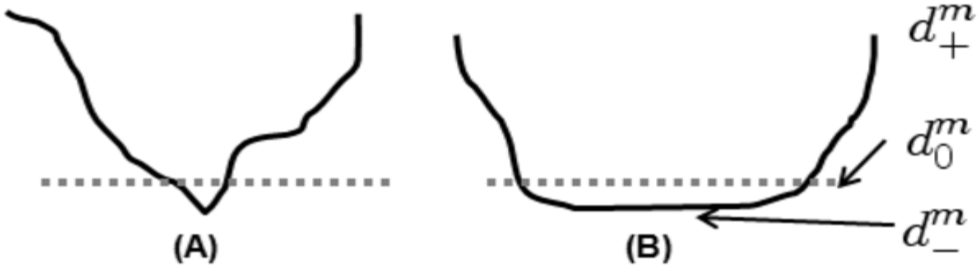


Figure 2.7: Illustration of reliability definition.

Fig. 2.7 shows two cases of local motion estimation within a neighborhood where a minimum distance is found. In case (A), the distance at the minimum location is distinctively smaller than those in its neighborhood. While in case (B), the minimum distance is not distinctively smaller or there might be multiple minimum locations. This implies that this minimum location or estimated motion vector is not reliable because there are many other locations or motion vectors with similar distance values. Certainly, the larger the variance of these motion vectors is, the more unreliable the local motion estimation is. Based on this observation, we define the reliability for local motion estimation.

Let $\{\mathbf{B}^m | 1 \leq m \leq \mathrm{M}\}$ be the set of structural blocks in frame I_t that we have selected from block classification. For each structural block B^m, we find the top L best matches for B^m in the previous frame I_{t-1} and the estimation results are denoted by $\Lambda = \{(\mathrm{V}_\mathrm{j}^\mathrm{m}, \mathrm{d}_\mathrm{j}^\mathrm{m}) | 1 < j < L\}$ where $V_j^m = (\dot{x}_j^m, \dot{y}_j^m)$ represents the motion vector and d_j^m is the corresponding distance. Let V^m be the motion vector determined by the local motion estimation and d_-^m be the corresponding distance. Certainly,

$$d_-^m = \min_j d_j^m$$

(2.13)

We also define

$$d_+^m = \max_j d_j^m$$

(2.14)

Let

$$d_0^m = d_-^m + \alpha \cdot (d_-^m - d_+^m)$$

(2.15)

where α is a threshold value between 0 and 1. By default, we set $\alpha = 0.1$. Here, α can be considered as the level of image noise. We choose a subset of those motion vectors in Λ whose distance measurements are very close to the minimum d_-^m, and denote this subset by

$$\Lambda\_ = \{(V_k^m, d_k^m) | d_k^m < d_-^m\}$$

(2.16)

Here, we re-label the elements in the set Λ_ by index k, $1 \leq k \leq K_m < L$. Certainly, $V^m \in$ Λ_. We define the reliability measure as

$$\gamma^m = \frac{1}{1 + \eta \cdot \sum_{k=1}^{K_m} \left\| V_k^m - V^m \right\|_2}$$

(2.17)

Here, η is a positive penalty rate and || · ||2 represents the L2-norm. We can see that $0 < \gamma^m < 1$. If the value of K_m is small (close to 1) or the motion vectors V_k^m are very close to V^m, the corresponding reliability measure γ^m will be very close to 1. Otherwise, γ^m will be small and approaching 0. Fig. 2.8(a) shows one example video frame. We use all blocks as structural blocks and estimate their local motion and compute its reliability.

Fig. 2.8(b) shows the reliability of each block. We can see that the blocks in the top-right region are almost flat and their reliability values are very small. The blocks in the center-left region have very high reliability because they do have distinctive image features. It is interesting to see that those blocks in the parking lot (top-center) area have medium reliability because they do look similar to each other and cause some uncertainty during local motion estimation.

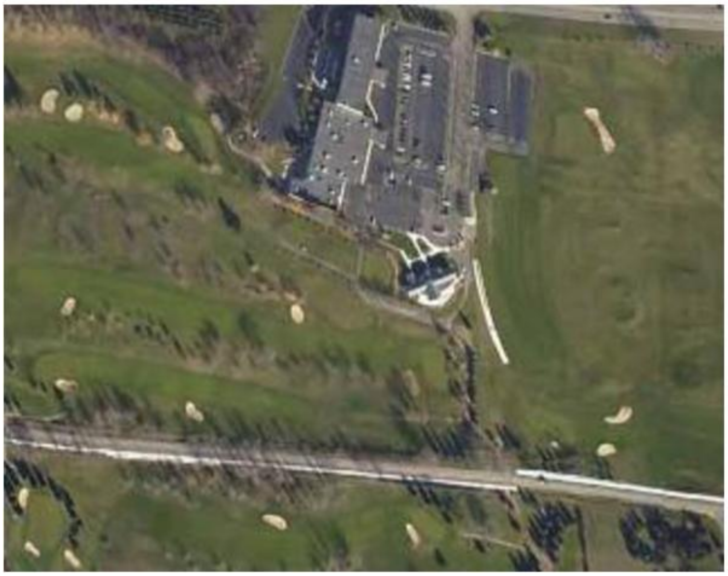


Figure 2.8: (a) A sample video frame from aerial surveillance videos.

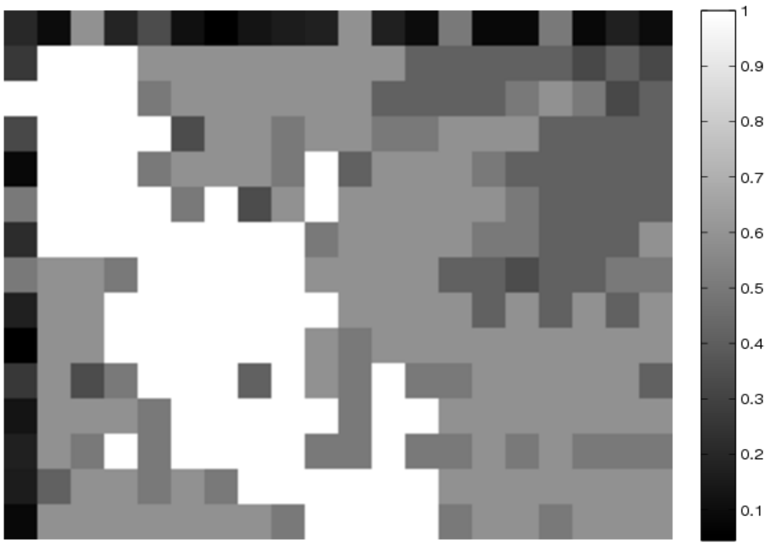


Figure 2.8: (b) Reliability map of Fig. 2.8: (a).

In the following, we use one example to demonstrate the importance of reliability analysis in global motion estimation. Fig. 2.9 shows two consecutive video frames in which a rectangular object moves towards bottom-right in the camera view. In local motion estimation, we partition the image into blocks and select a fraction, e.g. 15%, of them as structural blocks. Most likely, those blocks along the edges will be selected. We then find the local motion for each structural block. We can see that each block on the edge will match any block along the edge with the same distance value. By default, the motion estimation will choose the first best match which yields a motion vector pointing downwards, as illustrated in Fig. 2.9. However, the four blocks at corners will find their correct motion vectors, which point to bottom-right. Since we have a lot more edge blocks than corner blocks, if we treat them equally important, the GME will determine that the object moves downwards, which is incorrect. If we perform the above reliability analysis, those edge blocks will have a very small reliability since they have many similar best matches, while those corner blocks will have a very high (close to 1.0) reliability value. In this case, the global motion estimation will automatically rely more on those corner blocks than those edge blocks and generate the correct object motion. Certainly, this is just one simple example to illustrate the importance of reliability analysis. In actual GME and registration of aerial surveillance videos, we observe that the reliability analysis enables us to discover the most salient image features in a statistical manner without sophisticated and computationally intensive computer vision processing.

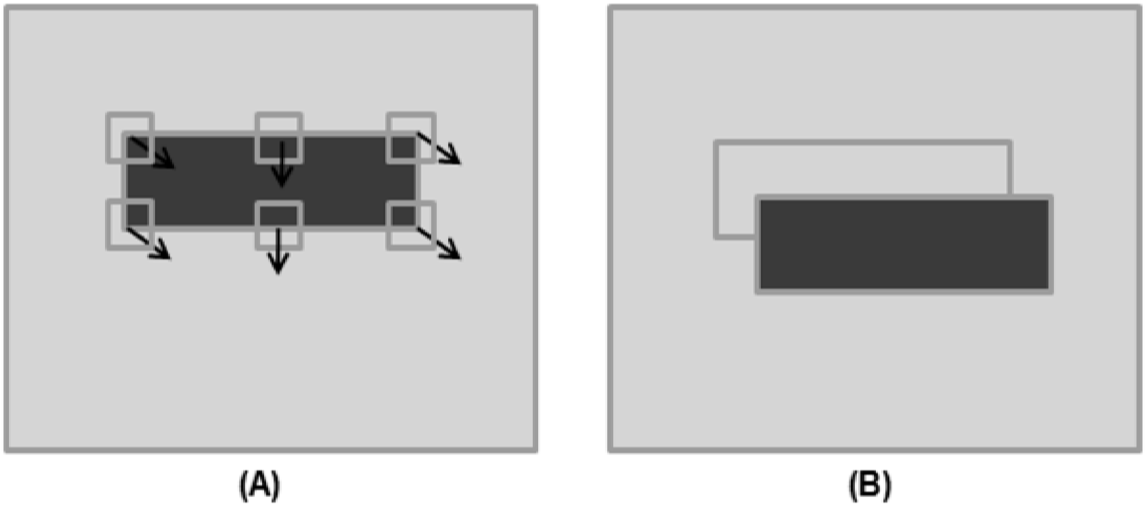


Figure 2.9: A rectangular object moving in a video scene.

2.5 Global Motion Estimation

As discussed in Section 2.1, if a video sequence experiences global camera motion, two consecutive video frames are then associated by a perspective transform defined in (2.2). (2.2) is also called the global motion equation. It has 8 parameters. Theoretically, if we know the local motion vectors of 8 points, more specifically, if we know the pixel position (x, y) of each point in frame I_{t-1} is mapped into a new position (X, Y) in frame *It,* we can then solve this global motion equation and obtain the 8 global motion parameters [a, b, c, d, e, f, g, h]. However, in practice, local motion estimation often involves a significant amount of uncertainty and ambiguity and it is often hard to accurately find the true motion vector for point. Therefore, we employ a large number of correspondence pairs and determine the 8 global motion parameters by a weighted least mean squared error (LMSE) procedure.

Let $\{\mathbf{B}^m | 1 \leq m \leq \mathrm{M}\}$ be the set of structural blocks in frame I_t. Let (x_m, y_m) be the center position of block B^m. We estimate its local motion and determine that (x_m, y_m) is mapped to (X_m, Y_m) in frame I_{t+1}. According to the global motion equation in (2.2), we have

$$X_m = \frac{a \cdot x_m + b \cdot y_m + c}{g \cdot x_m + h \cdot y_m + 1}$$

$$Y_m = \frac{d \cdot x_m + e \cdot y_m + f}{g \cdot x_m + h \cdot y_m + 1}$$

(2.18)

It can be rewritten into the following linear form:

$$P_m \cdot G = Q_m$$

(2.19)

where

$$P_m = \begin{bmatrix} x_m & y_m & 1 & 0 & 0 & 0 & -x_m \cdot X_m & -y_m \cdot X_m \\ 0 & 0 & 0 & x_m & y_m & 1 & -x_m \cdot Y_m & -y_m \cdot Y_m \end{bmatrix}$$

(2.20)

$$G = [a\ b\ c\ d\ e\ f\ g\ h]^T$$

(2.21)

and

$$Q_m = [X_m\ Y_m]^T$$

(2.22)

In this work, we use a least mean square error (LMSE) procedure to determine the global motion parameters **G** which aims to minimize the following the square error:

$$E = \sum_{m=1}^{M} [P_m \cdot G - Q_m]^T \cdot [P_m \cdot G - Q_m] \tag{2.23}$$

Note that the reliability of block B^m is γ^m. If we use γ^m as a weight, the square error becomes

$$E_W = \sum_{m=1}^{M} \gamma^m \cdot [P_m \cdot G - Q_m]^T \cdot [P_m \cdot G - Q_m] \tag{2.24}$$

Write

$$P \quad = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix} and\ Q = \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_m \end{bmatrix} \tag{2.25}$$

Define

$$W = diag\{\gamma^1, \gamma^1, \gamma^2, \gamma^2, \dots, \gamma^M \gamma^M\} \tag{2.26}$$

The solution to the LMSE problem in (2.24) is given by

$$G = (P^T W P)^{-1} P^T W Q \tag{2.27}$$

Once the camera parameters are determined, we can warp the video frame into a common coordinate system so as to create a mosaic for the video sequence.

2.5.1 Moving Object Detection and Global Motion Refinement

Typical aerial surveillance videos have moving objects (persons or vehicles) in the scene. It should be noted that those structural blocks that are used for GME could have blocks from moving objects. The motion estimation of these structural blocks could also have high reliability levels. Note that the motion of moving objects is different from the global camera motion. In this case, the motion information of the moving objects will act as noise and degrade the overall GME performance. Therefore, it is very important to detect moving objects; separate them out from the background, and refine the GME result.

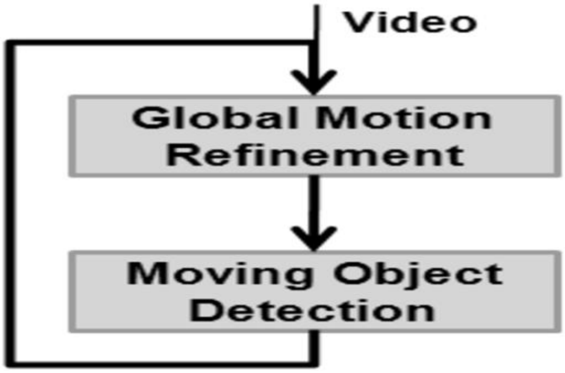


Figure 2.10: Moving object detection and global motion refinement.

We consider moving object detection as a hypothesis testing problem. An object is considered as moving if its motion does not satisfy the global motion equation. We observe that the moving object detection and global motion refinement is a mutually dependent and recursive process, as illustrated in Fig. 2.10. Initially, when the moving objects are included in the GME process, the estimated global camera motion won't be very accurate. Using this global motion equation, we can only detect those fast moving objects whose motion is significantly different from the global motion. Once these objects are detected and excluded, the global motion estimation will be more accurate. With this more accurate global motion equation, we can detect objects with slower motion. This process can be repeated to further improve the GME accuracy and detect more moving objects. In Figs. 2.11 and 2.12, we plot the values of the eight global motion parameters (a, b, c, d, e, f, g, h) at each iteration for test Video (5) in Fig. 2.15. We can see that, with moving object detection and global motion refinement, these global motion parameters quickly converges to their true values. To measure the accuracy of GME, we warp the current frame I_{t+1} into the coordinate system of frame I_t and measure the average pixel-level registration error. Fig. 2.13 shows the average registration error (in unit of pixels) as a function of iteration number. We can see that the registration error is reduced significantly with the iterative motion object detection and global motion refinement.

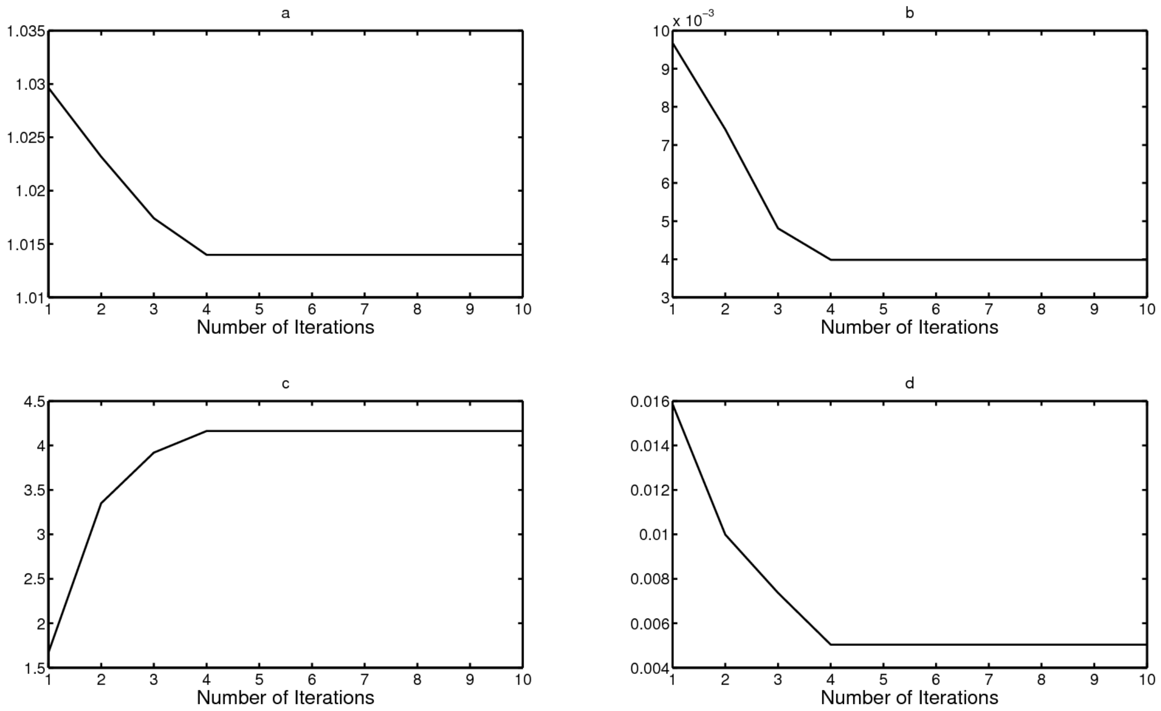


Figure 2.11: Convergence of transform estimation.

It should be noted that when we refine the GME, we only need to re-compute the global motion parameters **G** using (2.27). Since moving objects could also be classified as non-structural blocks, we need to estimate the local motion for all non-structural blocks. Before the local motion estimation, we remove the global camera motion by warping the two neighboring video frames into a common coordinate system. We observe that, once the global motion is removed, the object motion is usually translational. Therefore, for low computational complexity, we can use the SAD distance metric for motion estimation of non-structural blocks.

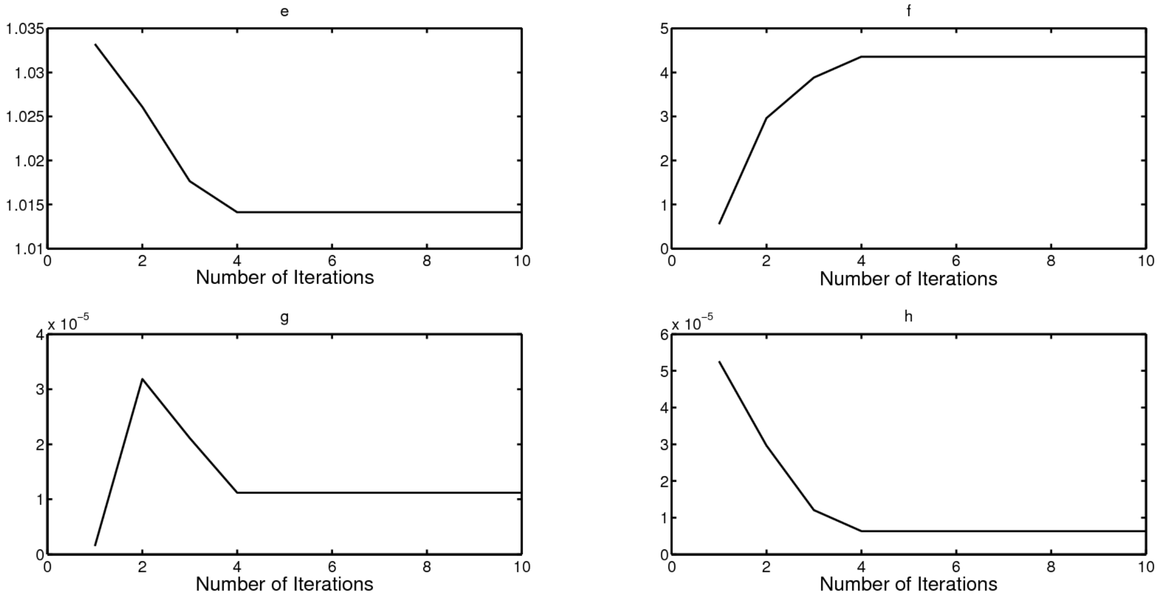


Figure 2.12: Convergence of transform estimation.

2.5.2 Algorithm Description

In this section, we summarize the proposed algorithm for GME and moving object detection.

Step 1: *block classification*. Partition the video frame into blocks. Apply DCT to each block. Compute the energy ratio using (2.3). Classify the blocks into structural and non-structural blocks based on their energy ratios.

Step 2: *local motion estimation of structural blocks*. Using the compound distance metric in (2.11), estimate the motion vector of each structural block. As discussed in Section 2.3.4, the distance metric selection should be adaptive based on the camera motion of previous frames.

Step 3: *reliability analysis*. Compute the reliability m of each structural block using (2.17).

Step 4: *compute global motion parameters*. Based on the motion vectors of structural blocks and their reliability values, compute the global motion parameters **G** using (2.27).

Step 5: *motion estimation for non-structural blocks*. Based on the global motion parameters, warp the video frame into the coordinate system of the previous frame. Estimate the motion vectors of non-structural blocks.

Step 6: *moving object detection*. Detect moving objects. An object is considered as moving if its motion does not satisfy the global motion equation.

Step 7: *iteration*. Repeat Steps 4 and 6 to refine global motion estimation and detect more moving objects. The iteration can be stopped when the changes of global motion parameters become very small.

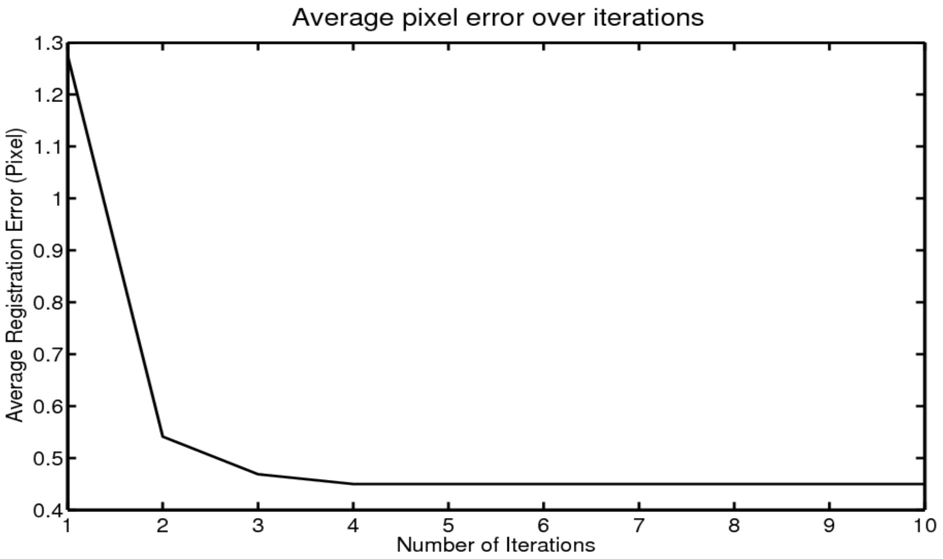


Figure 2.13: Average registration error to iterations.

It can be seen that the computational complexity of the proposed GME algorithm lies in three major components: block classification, local motion search, and global camera parameters estimation. Table 1 shows the average computational complexity of these three components (in percentage of overall complexity). It can be seen that the major complexity lies in local motion search. As in standard video compression systems, such as MPEG-4 or H.263 [6], block-based local motion search can be implemented at low complexity using fast search algorithms, such as diamond search and three step search [17, 18]. Therefore, the proposed global motion estimation algorithm has a similar level of computational complexity as a simple-profile MPEG-4 or H.263 video encoder. In Section 2.6.4, we will conduct complexity comparison with existing GME algorithms.

Table 2.1: Computational complexity of major components.

Component	Complexity (%)
Block Classification	13.22%
Local Motion Search	75.15%
Global Motion Estimation	6.63%
Others	5%

2.6 Experimental Results

In this section, we evaluate the proposed algorithm for GME. The experimental evaluation consists of two parts. First, we provide experimental results to justify the proposed ideas and methods, such as block classification, adaptive selection of distance metrics, and reliability analysis. Second, we compare the proposed algorithm with other GME algorithms in terms of both computational complexity and estimation accuracy. Although a number of GME algorithms have been proposed in the literature, due to the lack of sufficient technical detail, it is often difficult to implement them and achieve the same level of performance. In this work, we choose to compare the proposed algorithm against the GME algorithm provided by the Microsoft MPEG-4 Visual Reference Software [19], one of the state-of-the-art algorithms for GME. We refer to this algorithm as *MS-GME*.

2.6.1 Experimental Setup

To evaluate the algorithm performance, especially, the estimation accuracy of global camera parameters, we need the ground truth of these camera parameters. However, many aerial surveillance videos do not come along with camera meta data. Even in some cases where the meta data is available, it is often not sufficiently accurate because of sensor measurement noise. For example, the GPS sensor for camera location and gyroscope for camera orientation may have measurement noise. In this work, we propose two approaches to addressing this issue. First, we develop a MATLAB tool to simulate aerial video surveillance on a computer. More specifically, we set one large satellite picture on the ground plane, move the camera along a trajectory, vary the camera parameters, and capture a sequence of video frames based on a virtual camera model specified by these parameters. In this case, we know and have full control of the ground-truth camera parameters. From the video sequence, we use the GME algorithms to estimate the global camera parameters. For every pixel in the current frame, we use both the ground-truth global camera parameters and those obtained from GME to warp the pixel into the coordinate system of the previous frame. The location difference between these two warped pixels is called *pixel registration error*. We use the mean and variance of pixel registration errors of all pixels for performance evaluation.

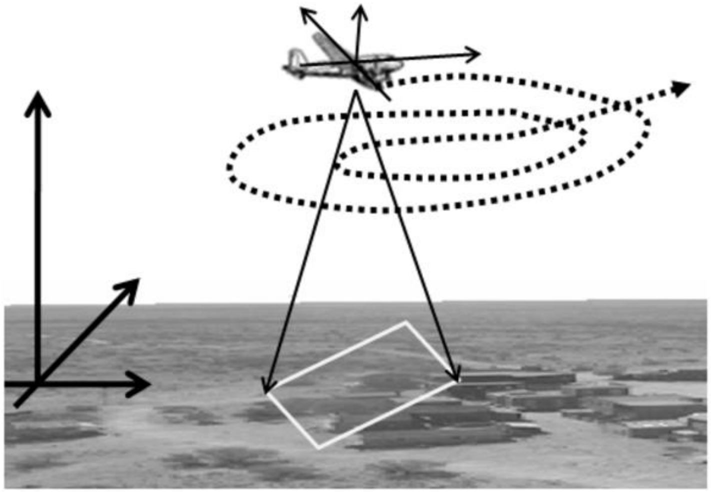
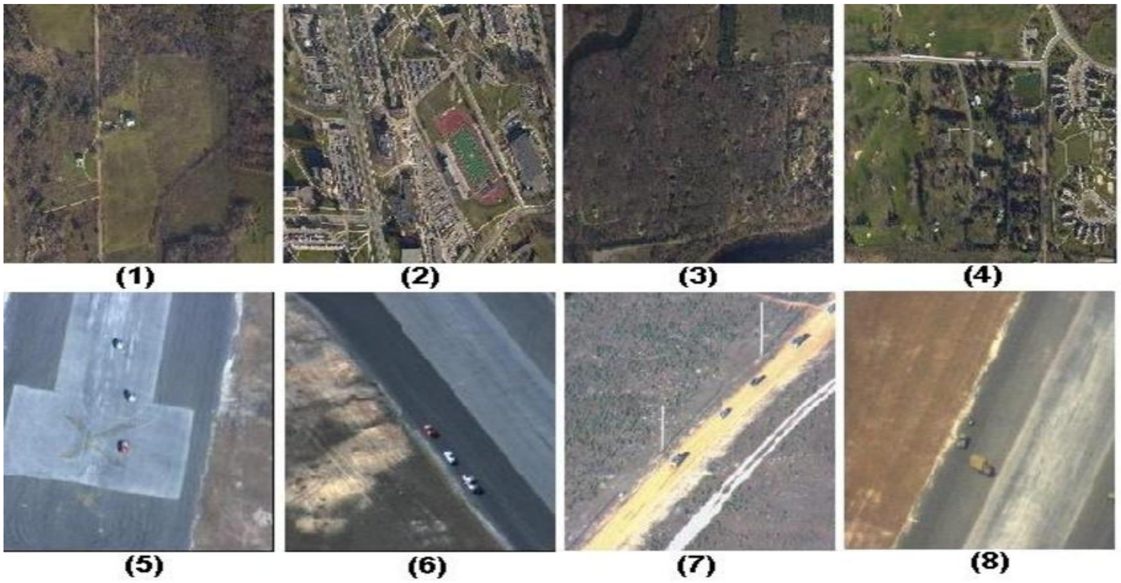


Figure 2.14: Illustration of computer simulation of aerial video surveillance.

Fig. 2.15 shows the test video sequences that we use for performance evaluation. The top row, videos (1) to (4) are obtained from computer simulation. Besides these simulated aerial surveillance videos, we also use aerial surveillance videos collected from field deployment of UAVs (unmanned aerial vehicles), as shown in the bottom row of Fig. 2.15. All test videos have a resolution of 640×480 at 30 frames per second. Unfortunately, some field aerial surveillance videos do not have camera parameter meta data. Although some videos do have the meta data, it is not sufficiently accurate to be used as ground-truth for performance comparison. In this case, we propose to introduce a new measure, called *pixel intensity difference*, to measure the performance. More specifically, after global motion estimation for two neighboring frames I_{t-1} and I_t, we use the global camera parameters to warp frame In into the coordinate system of I_{t-1} using (2). Let $\Omega(n - 1, n)$ be the overlapped image area of I_{t-1} and I_t (after warping). If the estimation is accurate, at every location z in Ω, two pixels from I_{t-1} and I_t, denoted by $I_{t-1}[z]$ and $I_t[z]$, should have the same intensity value. Motivated by this observation, we use the average pixel intensity difference:

$$PD(t-1,t) = \frac{1}{|\Omega(t-1,t)|} \sum_{z \in \Omega(t-1,t)} |I_{t-1}[z] - I_t[z]| \tag{2.28}$$

to measure the accuracy of GME. Here, $|\,\Omega\,(t - 1,\, t)|$ represents the total number pixels in the overlapped area.



(1) (2) (3) (4)

(5) (6) (7) (8)

Figure 2.15: Test video clips.

2.6.2 Experimental Justification of the Proposed Algorithm

In this section, we present experimental results to justify the proposed ideas and methods, including block classification, adaptive distance metric, and reliability analysis, and characterize their impact on the overall performance of GME. As discussed in Section 2.3, the block classification selects blocks with distinctive features for reliable motion estimation and therefore improves the performance of GME. Fig. 2.16 shows the mean and variance of pixel registration errors of GME on Video (1) with different percentages of structural blocks. The result for Video (2) is shown in Fig. 2.17. It can be seen that as we choose more and more structural blocks, the registration error reduces. However, if we choose all blocks as structural blocks, which implies no block classification, the registration error is significantly increased. This demonstrates that block classification improves the performance of GME. Simulation results over other test videos yield similar results. In this work, based on our experience, we choose the top 30-40% of blocks as structural blocks.

As discussed in Section 2.4, the reliability analysis emphasizes the importance of structural blocks which have reliable and accurate location motion estimation. Fig. 2.18 shows the mean of pixel registration errors of GME on Video (4) with and without reliability analysis. It can be seen that, with reliability analysis, the average registration error is significantly reduced.

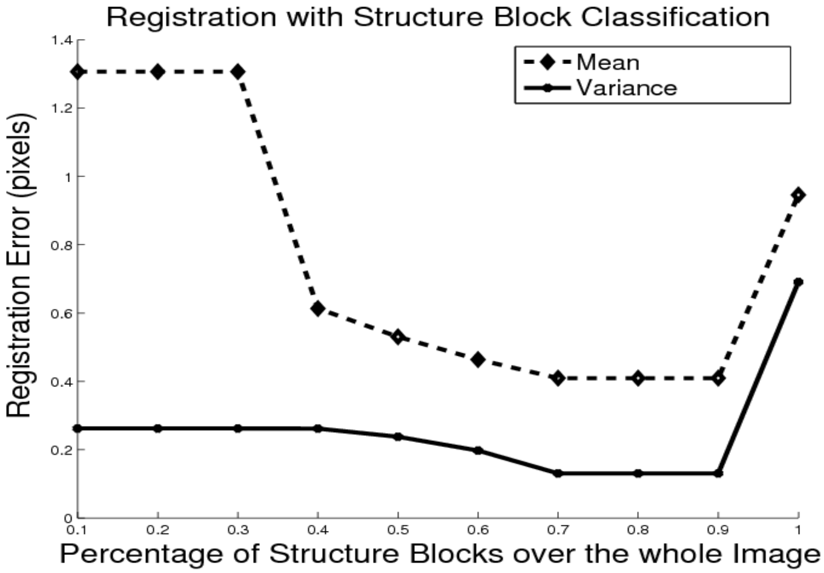


Figure 2.16: Mean and variance of pixel registration errors of GME on Video (1) with different percentages of structural blocks.

In Section 2.3.4, we propose an adaptive compound distance metric for local motion estimation based on three distance metrics, SAD, intensity profile, and histogram of gradients. We observe that these three have both advantages and disadvantages in terms of computational complexity and motion estimation performance. For video sequences with different global camera motion characteristics, we need to select different combinations of them in an adaptive manner, as illustrated in (2.11). In the following experiment, we choose different weight vectors in (2.11), obtain different compound distance metrics, and evaluate their performance in GME. Table 2 lists the average pixel registration errors for test videos (1) to (4). The minimum values are indicated in bold. We can see that a combination of SAD and intensity profile has the minimum average pixel registration error on Videos (1) and (4); a combination of SAD and histogram of gradients has the minimum on Video (2); and intensity profile has the minimum on Video (3). We can also see that SAD itself ([1 0 0]) is not efficient for estimating global camera motion and it is not helpful to use all of these three distance metrics simultaneously ([1, 1, 1]), either. These experimental results justify the need of the adaptive compound distance metric.

Table 2.2: Average pixel registration error of GME with different compound distance metrics.

Weight	[1 0 0]	[0 0 1]	[0 1 0]	[1 0 1]	[0 1 1]	[1 1 0]	[1 1 1]
Video (1)	0.3081	0.3080	0.3898	0.3081	0.2950	**0.2300**	0.2821
Video (2)	0.4642	0.5065	0.6751	**0.3652**	0.4272	0.4680	0.3766
Video (3)	5.6243	4.2064	**0.5059**	5.6203	2.7651	2.8858	3.7973
Video (4)	0.4182	0.3683	0.3028	0.5023	0.3170	**0.3011**	0.4880

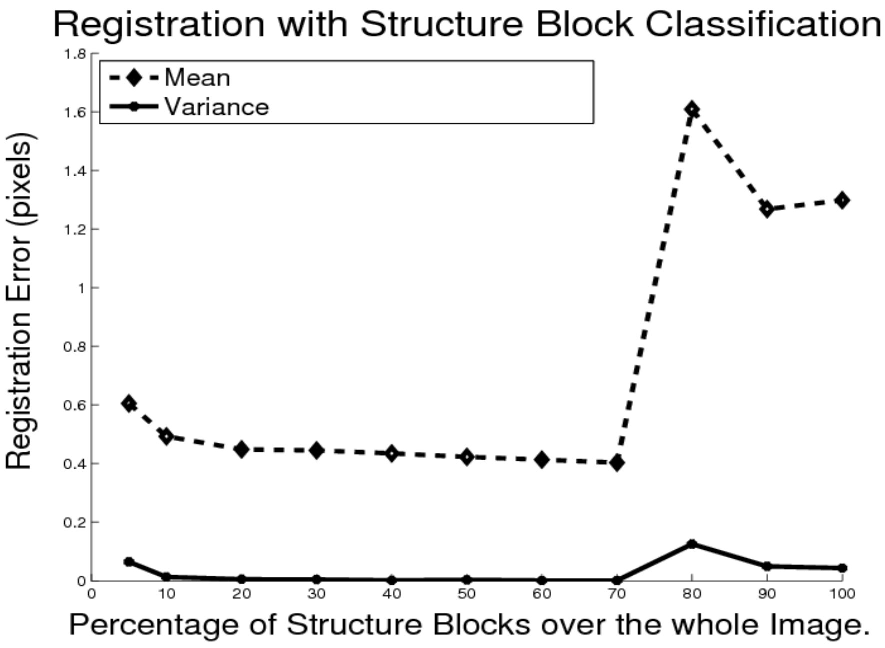


Figure 2.17: Mean and variance of pixel registration errors of GME on Video (2) with different percentages of structural blocks.

2.6.3 Performance Comparison with MS-GME

In the following, we compare the proposed GME algorithm with the MS-GME algorithm implemented in the Microsoft MPEG-4 Visual Reference Software [19]. We evaluate their computational complexity and GME performance. We use the field aerial surveillance videos for test and use the pixel intensity difference for performance measurement. Fig. 2.19 shows the running time (in seconds per frame) comparison between MS-GME and the proposed algorithm on Video (5). Fig. 2.20 shows its GME accuracy comparison results. Simulation results on Video (7) are shown in Fig. 2.21 and Fig. 2.22, respectively. Comparison results for the average running time (in seconds per frame) and the average pixel intensity difference on Videos (5), (7), and (8) are listed in Table 2.3. It can be seen that the proposed GME algorithm achieves much higher estimation accuracy than the MS-GME algorithm while its computational complexity is significantly (about 15-20 times) lower than that of MS-GME. In addition, the computational complexity of the proposed algorithm does not vary much with different input videos. This is highly desirable in practical software/hardware implementation of GME.

2.6.4 Video Registration and Moving Object Detection

In this section, we demonstrate the application of the proposed GME algorithm in video registration and moving object detection. Once the global camera motion parameters are estimated, we can use them to warp the video frames into a common coordinate system with (2.2) so as to create a mosaic for the video scene. Once the background camera motion is removed, we can detect moving objects using simple background subtraction and silhouette extraction algorithms that have been developed for stationary cameras

[15, 16]. Figs. 2.23 and 2.24 show the video registration results for Videos (2) and (5), respectively. It can be seen that image features, such as road edges, are well aligned. Fig. 2.25 shows the moving objects detection results for three test videos (7), (8), and (5). The first two videos (7) and (8) contain a single moving object, as indicated by a bounding box in the first two rows of Fig. 2.25. The third video (5) has multiple vehicles. Fig. 2.25 (the third row) shows the moving trajectory of each vehicle.

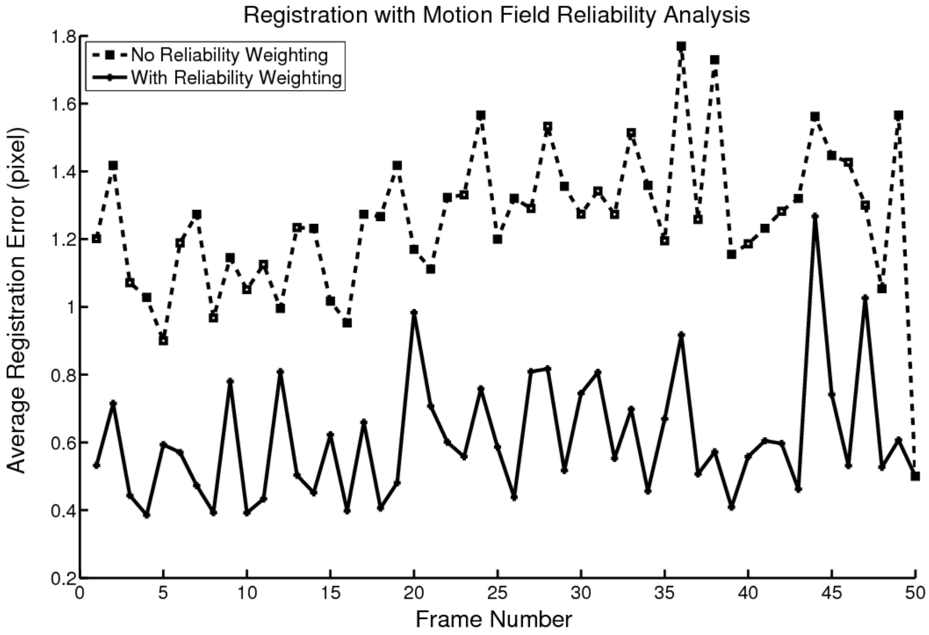


Figure 2.18: Average registration error of GME on Video (4) with and without reliability analysis.

Table 2.3: Performance comparison with MS-GME.

Test	Running Time (s)		Average Pixel Intensity Difference	
Video	This Work	MS-GME	This Work	MS-GME

Video(5)	0.046	0.634	1.615	3.386
Video(6)	0.042	0.616	1.017	3.918
Video(7)	0.046	1.233	2.529	4.296

2.7 Discussion

GME is the enabling step for video registration, moving object detection and tracking in aerial video surveillance. In this work, we have explored various methods to deal with the inherent uncertainty and image noise in motion analysis, and developed a low-complexity, accurate and reliable scheme to estimate the global camera motion for video registration and moving object detection. More specifically, we have introduced a block-based image data classification scheme to select those image regions, called *structural blocks*, with distinctive features for reliable motion estimation. We have introduced an adaptive compound distance metric for motion estimation which is able to efficiently handle camera rotation and zoom. We have develop a scheme to analyze the reliability of local motion estimation and use this reliability measure to as a weighting factor to influence the importance level of each structural block in global camera motion estimation. A progressive iterative scheme is proposed to detect moving objects, separate them from the background, and refine the GME. Our extensive simulation results demonstrate that the proposed GME scheme is accurate, reliable, and has low complexity. In our future work, we shall develop a systematic approach to determine the weighting vector for the compound distance metric. We also need to explore more sophisticated image features and distance metrics for accurate and reliable motion estimation.

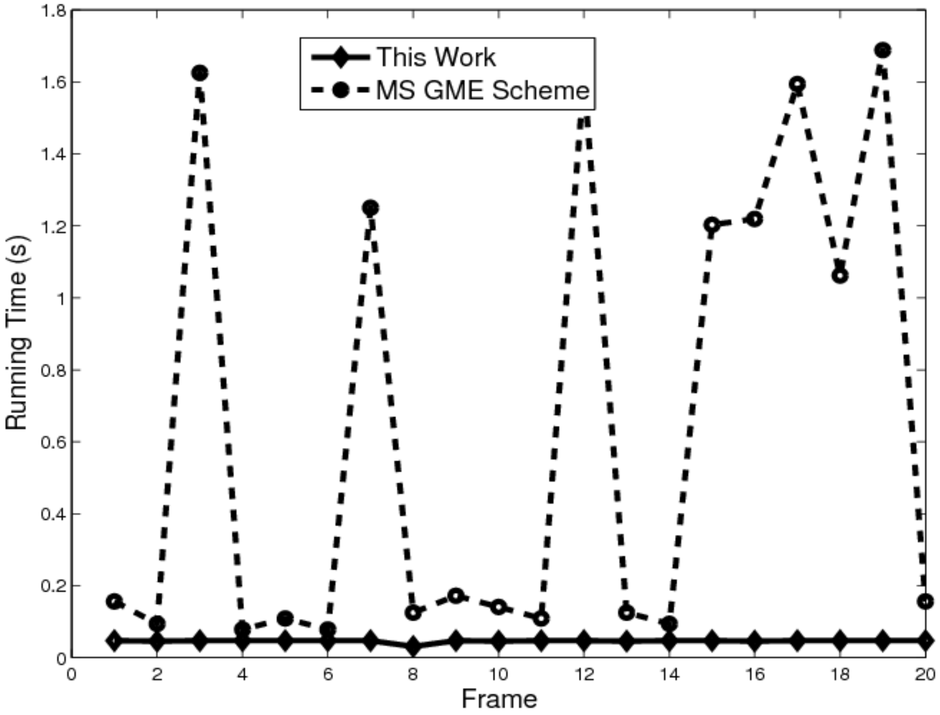


Figure 2.19: Running time comparison with the MS-GME scheme on Video (5).

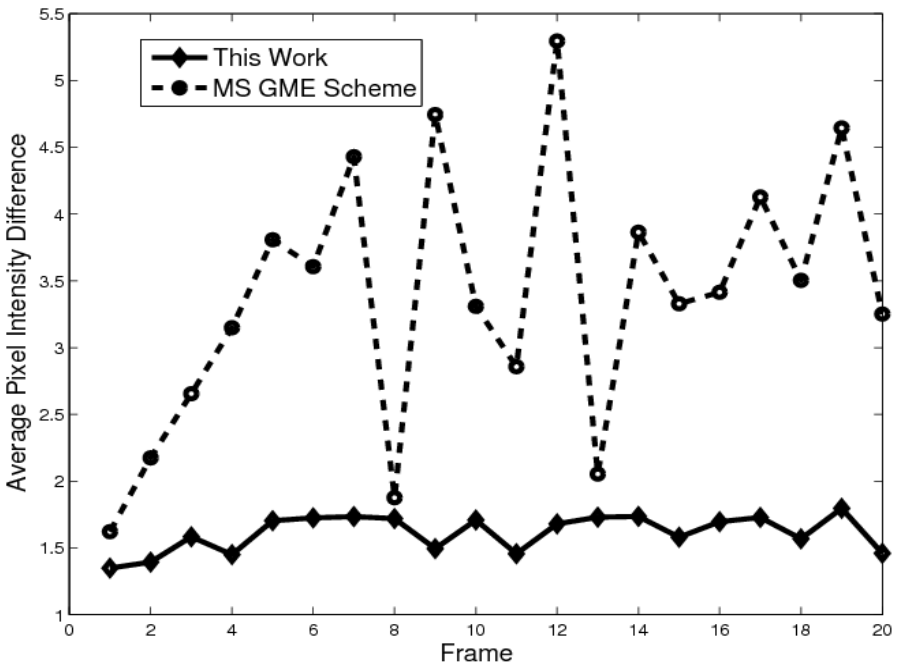


Figure 2.20: Estimation accuracy comparison with the MS-GME scheme on Video (5).

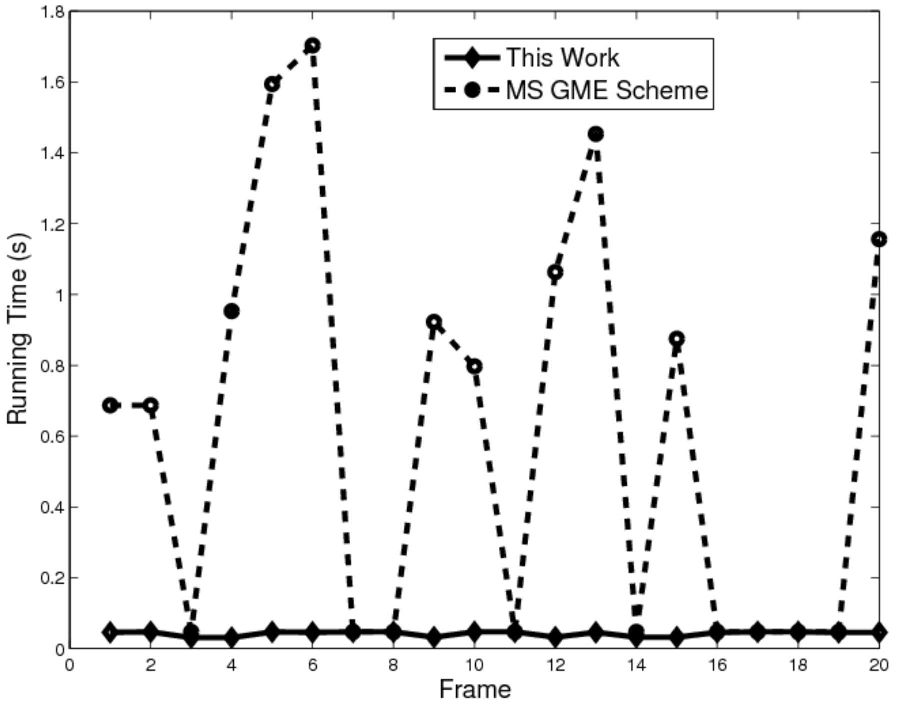


Figure 2.21: Running time comparison with the MS-GME scheme on Video (7).

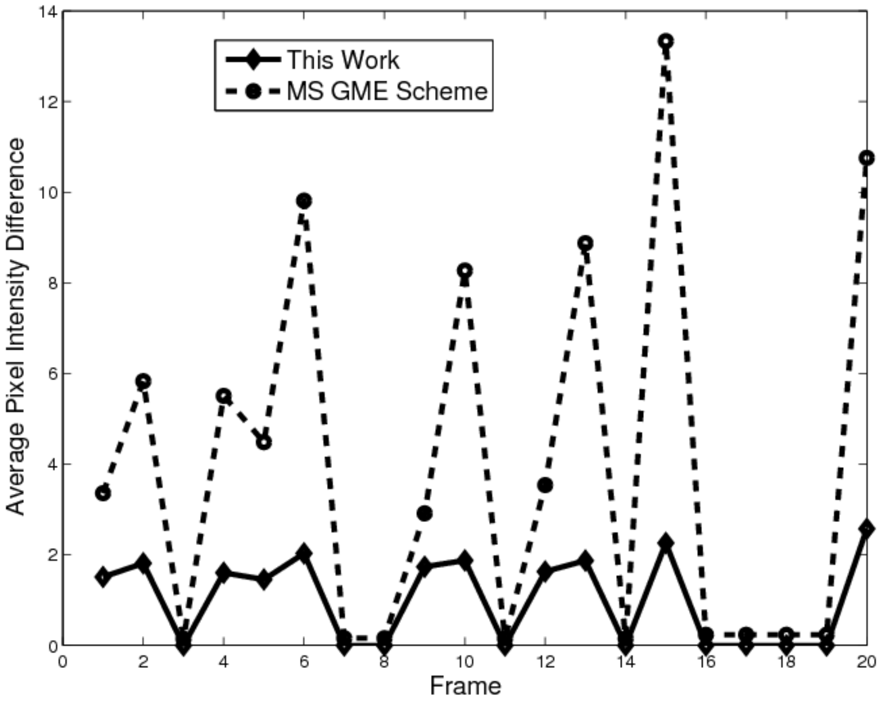


Figure 2.22: Estimation accuracy comparison with the MS-GME scheme on Video (7).

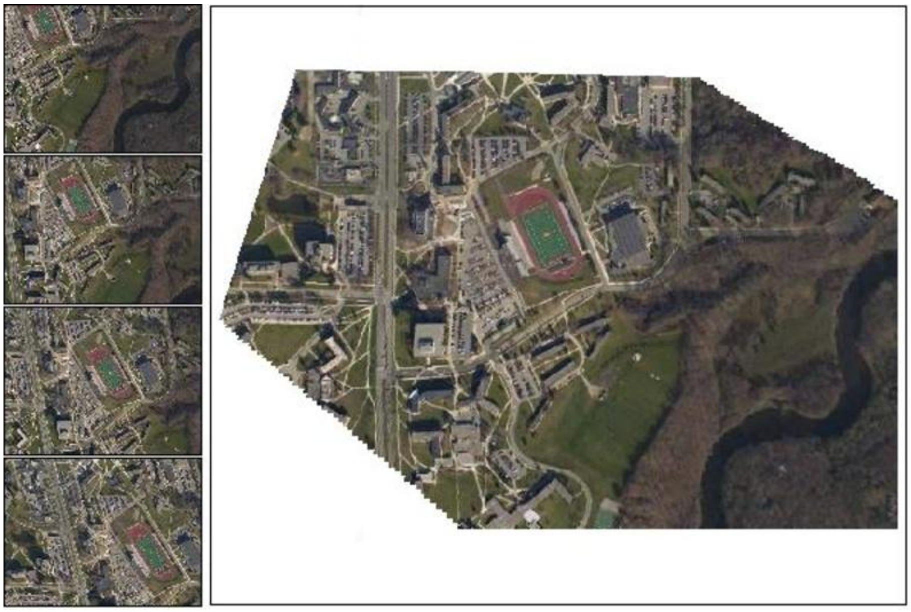


Figure 2.23: Video registration results for Video (2).

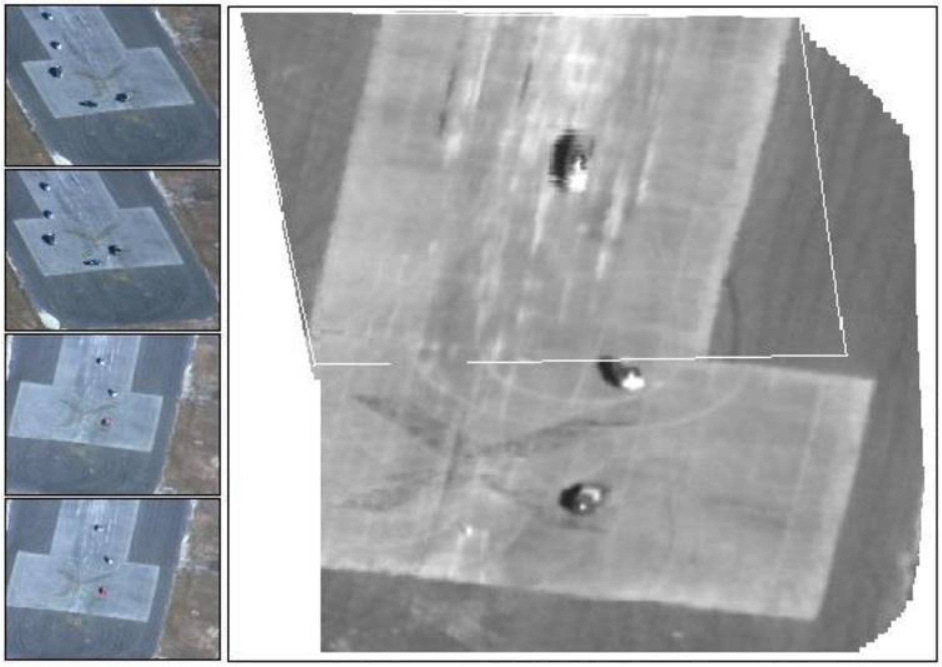


Figure 2.24: Video registration results for Video (5).

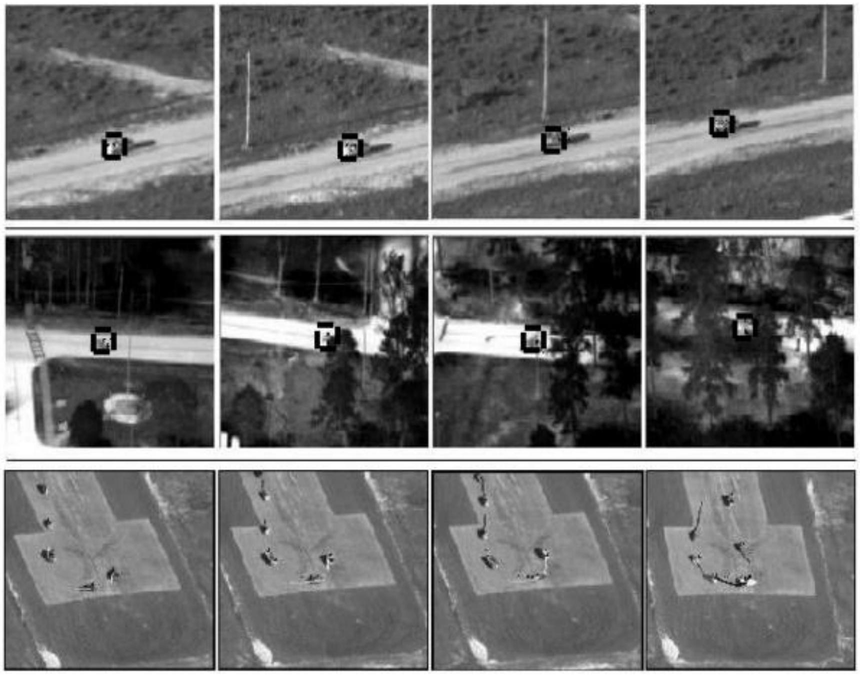


Figure 2.25: Moving object detection results for Videos (6), (7), and (8).

CHAPTER 3

CONTENT-BASED BUILDING RETRIEVAL FROM AERIAL SURVEILLANCE VIDEOS

Identifying and matching buildings between camera views is useful for scene understanding, robot navigation, battlefield surveillance, geo-location and geo-tagging of videos and photos [20]. For example, within the context of battlefield surveillance, a solider can take a photo using a hand-held camera of a building, retrieve and review the video clips from the aerial surveillance video database which have continuous surrounding views of the same building to understand the environment for situational awareness and informed action planning. In this work, we propose to develop a method for building recognition using sketch-based representation and spectral graph matching.

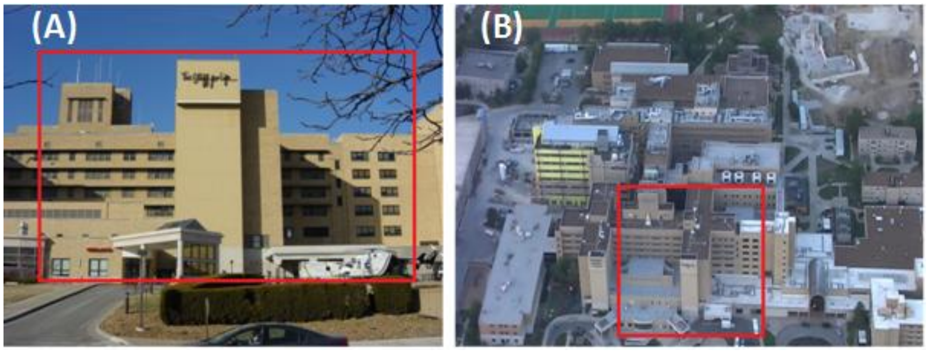


Figure 3.1: (A) and (B): Building identification between two videos with large changes in scales and perspectives.

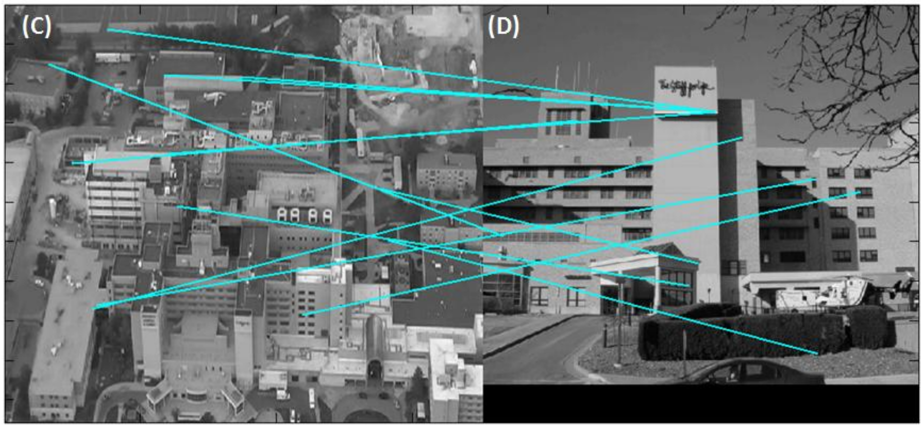


Figure 3.1: (C) and (D): SIFT matching results.

Recent years have seen great advances in developing local appearance descriptors for objects, such as SIFT [21], PCA-SIFT [22], GLOH (gradient location and orientation histogram) [23], shape context [24], steerable filters [25], etc. These local image descriptors have found many important applications in computer vision, such as wide-baseline matching [26], object recognition and tracking [21, 82], texture matching [27, 90], image retrieval [28], robot navigation [29], and scene understanding [30]. The central goal of developing local image descriptors is to make them invariant under image transforms and camera motion, such as image rotations, camera zoom, changes in scale and perspectives, and image noise, while maintaining high repeatability and discriminative power. According to the comprehensive performance evaluation conducted by Mikolajczyk and Schmid [23], SIFT and SIFT-like GLOH features exhibit the highest matching accuracies and recall rates, especially for scales changes in the range 2-2.5 and image rotations in the range 30 to 45 degrees. Performance for all local descriptors degraded with image blurring which affects the accuracy and reliability of edge, shape, and gradient information used in these local descriptors. In the practice of building recognition, images are often taken by different persons at different times from different platforms (e.g. ground-level or airborne surveillance cameras). These types of images typically have large changes in scales, camera perspectives, illuminations, and strong image blurring due to camera motion, where existing local feature matching methods are not able to provide satisfying performance. Figure 3.1 (C) and (D) show one example of SIFT matching. SIFT has found 4547 and 2179 key points in images (C) and (D). However, only very few matches are found between them. Here, we set the distance ratio to be 0.65.

Man-made structure detection has been studied in the literature [20, 89]. For aerial images, especially those collected at high altitudes, roof-top and shadow detection is the key step to building detection [33]. To this end, low-level image primitives, e.g. edges, lines and corners, are extracted and then grouped together using either geometric models [31] or statistical models, e.g. Markov Random Field (MRF) [32]. Within the context of ground-level images, approaches to building detection are much different. Kumar and Hebert [20] show how to classify man-made structures from landscape using causal multi-scale random field. Color and texture features have been used in [36, 37] to segment buildings from scenes. Vailaya *et al.* [35] use the edge coherence histograms for scene classification and building detection. Sarkar and Soundararajan [34] develop a perceptual organization framework to group low-level edge features that belong to a building using spectral graph partitioning.

While a number of approaches have been developed for man-made structure detection, few methods have been developed in the literature for building retrieval or matching between camera views. In their recent work, Rajashekhar *et al.* proposed a method for building retrieval using cross-ratio which is invariant under perspective changes. This method relies on accurate and reliable line detection and is mainly used for buildings with crossing linear features. Zhang and Košecká [47] developed a promising approach for building recognition based on SIFT and voting and a probabilistic model. On the ZuBuD database, they reported an average of 90.4% correction recognition. It should be noted that most building views in the ZuBuD have relatively small viewpoint changes.

To develop an efficient scheme for building recognition with large viewpoint changes, we propose to explore a sketch-based representation which is semantically rich and largely invariant under large scale and perspective changes. Human visual system has a remarkable capability in recognizing and matching objects under drastically different viewing conditions. As we perceive and recognize a building, we often attempt to remember its outline, major structural components (e.g. windows and doors), their appearance characteristics and spatial configurations. In this work, we propose to capture this type of representation by a sketch. We focus our efforts on office buildings, which typically have repeated structural components, such as windows, doors, poles, exterior decorations, etc. After multi-scale maximally stable extremal regions (MSER) detection, the proposed approach attempts to find major structural components of buildings, and extract patterned semantic features, which are organized into a sketch-based representation of buildings. These descriptors are then clustered in association with different planes of the building and matched across video frames using spectral graph analysis.

This work has the following three major contributions: (1) detection of patterned structural components of buildings; (2) construction of a sketch-based representation of buildings; and (3) extension of spectral graph analysis for global image matching under large changes in scale and perspectives.

The rest of the chapter is organized as follows. The multi-resolution MSER detection is presented in Section 3.2. Section 3.3 explains the detailed procedure to construct a sketch-based representation of buildings. The extension of spectral graph analysis for global image matching under large changes in scale and perspectives is presented in Section 3.4. Section 3.5 presents our experimental results. Section 3.6 concludes the paper with further discussions and future work.

3.1 Multi-Scale MSER Detection

Our first step to capturing the major appearance characteristics of buildings is based on MSER detection. The original concept of maximally stable extremal regions

(MSERs) was proposed by Matas *et al.*[38]. MSER detection finds a set of distinguished image regions where each inner-pixel intensity value is less (greater) than a certain threshold, and all intensities around the boundary are greater (less) than the same threshold. An extremal region is maximally stable when the area (or the boundary length) of the segment changes the least with respect to the threshold. The set of MSERs is closed under continuous geometric transformations and is invariant to affine intensity changes, providing a highly efficient region detector for local image matching [23, 39].

We note that structural components of buildings, such as windows and doors, have different physical sizes and their sizes in the image depends on the image resolution. To address this issue, we follow Forssén and Lowe's approach in [39], extending the MSER detector to multiple scales. We construct a Gaussian pyramid and apply the MSER detection separately at each resolution. Fig. 3.2 (B) shows the MSER detection results for the image in Fig. 3.2(A). A closer look of the building area is shown in Fig. 3.2(C). We can see that all window areas have been successfully detected. Fig. 3.2(D) shows the extracted MSERs. Figs. 3.2(E) and (F) show the extracted MSERs at scales 2 and 3.

If the camera parameters and physical sampling distance of the images are known, e.g. in airborne video surveillance, we can carefully choose a subset of scales which reflects the typical sizes of buildings components.

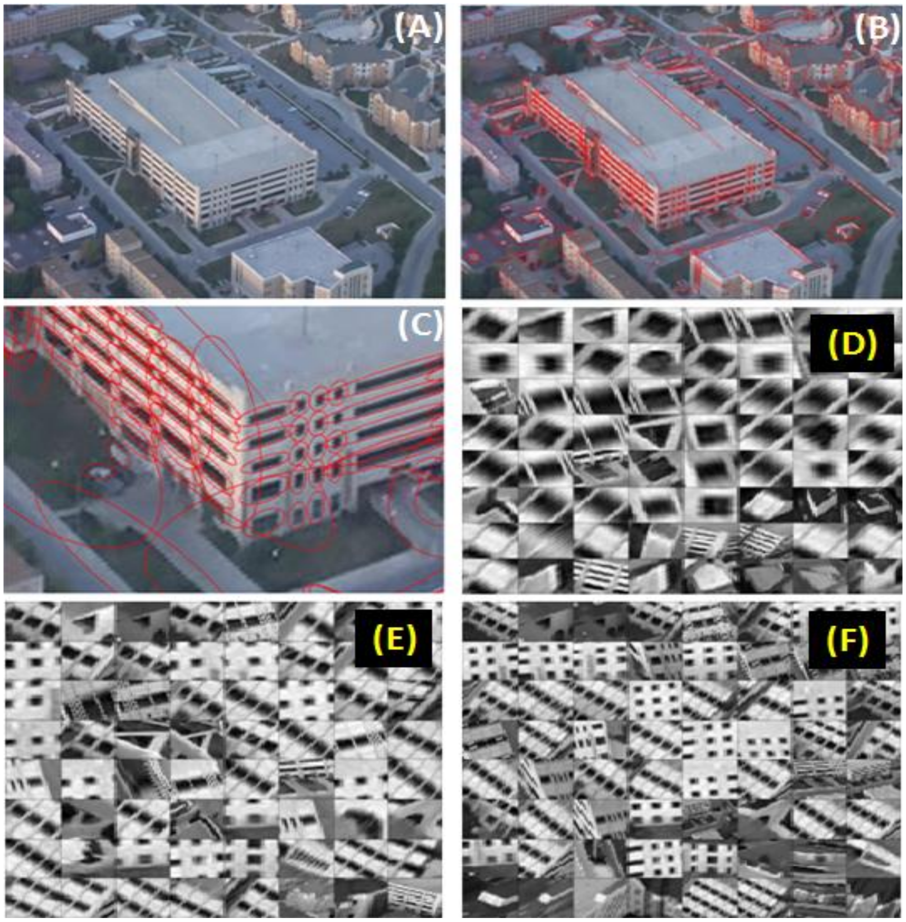


Figure 3.2: MSER detection results. (A) the original image; (B) MSER detection results indicated with red ellipses; (C) a close look at the parking garage area; (D) detected MSERs; (E) MSER detection results at scale 2; and (F) MSER detection results at scale 3.

3.2 A Sketch-Based Representation

After MSER detection, typically, the next step in image matching is to extract local image features [23, 39]. As discussed in Section 3.1, in the practice of building recognition, the image pairs often have large changes in scales, camera perspectives, illuminations, and strong image blurring due to camera motion, where existing low-level local feature matching methods are not able to provide satisfying performance. To address this problem, we propose to construct a semantically rich sketch-based representation of buildings so as to extract higher-level features which are largely invariant under severe changes in scales and perspectives.

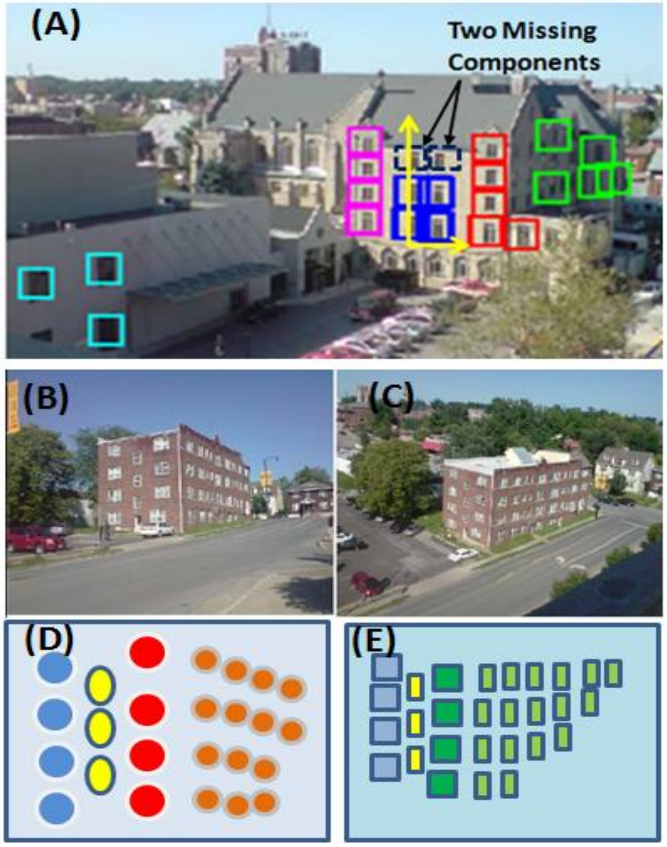


Figure 3.3: a sketch-based representation of a building in two different views.

Our approach is mainly motivated by the following observation: office buildings typically have repeated structural components, such as windows, doors, and exterior decorations, being configured in certain spatial patterns in various styles, as shown in Figs. 3.1 and 3.6. These types of regularity and patterns provide reliable features for building recognition. To detect repeated structural components from MSERs, we take the following major steps:

1. *Normalization*. To improve its scale and affine invariant capability, a normalization operation is applied to the elliptic regions detected by MSER: all regions are mapped to circular patches of constant radius [23]. More specifically, we compute the singular value decomposition (SVD) each patch's covariance matrix $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$ and rectify each patch using the following transform:

$$\boldsymbol{x} = s\boldsymbol{A}\widehat{\boldsymbol{x}} + \boldsymbol{m}, \qquad \boldsymbol{A} = 2\boldsymbol{U}\boldsymbol{D}^{1/2} \tag{3.1}$$

   Here, s is a scaling factor to control the normalized patch size [38]. In this work, we set the size to be 41 pixels.

2. *Extracting local appearance features.* Let $\{\boldsymbol{B}_k | 1 \leq k \leq K\}$ be the set of normalized patches. For each path, we use HOG (Histogram of Oriented Gradients) [44] as the local patch descriptor. To differentiate image patches on different sides / planes of the 3-D building, we also include the principle direction (θ_x, θ_y) of each patch into the feature vector. Let $\{\boldsymbol{f}_k | 1 \leq k \leq K\}$ be the set of features for all normalized patches.

3. *Clustering for regularity detection.* We then apply the k-means algorithm to cluster patches $\{\boldsymbol{B}_k\}$. Fig. 3.3(A) shows an example of clustering results. We can see that the

image patches of windows are separated into clusters with different shapes on different sides of the building, indicated by different colors.

4. *Regularity-driven spatial extension*. We observe that this initial step of regularity detection is not able to detect all repeated structural components due to the limited discriminative power of local features, image noise and illumination variations, shadows, and the inherent ambiguity in clustering and classification. To improve the detection results, we assume that the repeated structural components have a lattice spatial configuration. We compute the principle directions of the lattice using voting. We then incrementally search the MSER near the lattice to identify additional structural component in each cluster. Fig. 3.3(A) shows an example of two missing windows being detected using this procedure.

We also extract HOG features describe the façade of the building, i.e. surrounding areas of these structural components. These clusters of structural components, their features (described in Section 3.2), and descriptors of their surveillance areas form a sketch representation of the building, as shown in Figs. 3.3(D) and (E).

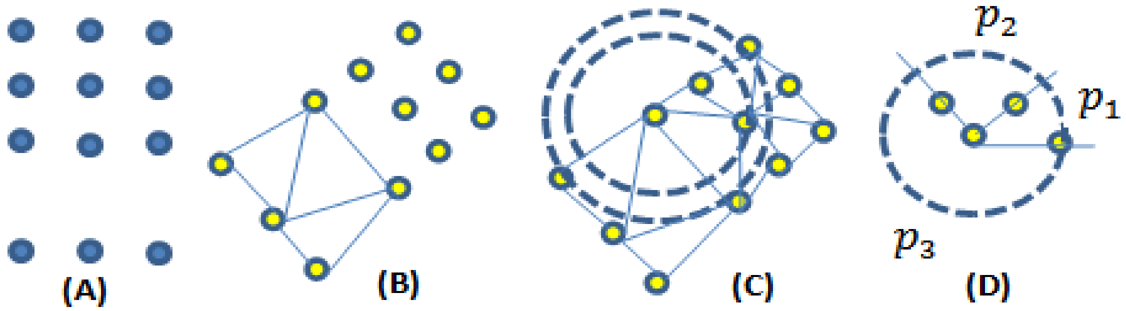


Figure 3.4: geometrical invariants of point patterns.

3.3 Spectral Graph Matching

Spectral graph theory, [91], attempts to characterize the global structure of graphs using the eigen-values and eigenvectors of the node adjacency matrix. Recently, researchers have applies spectral graph theory for object recognition, image matching, and segmentation. In his earlier work [40], Umeyama developed an eigen-decomposition approach for matching graphs of the same size. Scott and Longuet-Higgins were among the first to apply spectral graph analysis for matching 2-D point features between two arbitrary shapes or point patterns [41]. They use a proximity matrix to describe the affinity of all possible pair-wise matches, and the eigenvectors of this matrix are used to determine the point correspondences. One of its major drawbacks is that it cannot cope with relatively large rotations in the image plane. Shapiro and Brady proposed an improved method in [42] by comparing the eigenvectors of the point proximity matrix for more accurate and robust point matching. Sclaroff and Pentland [43] proposed an algorithm based on the eigen-modes of a shape matrix for point correspondence and shape recognition. Carcassoni and Hancock [90] embedded spectral graph analysis into the framework of EM algorithm and significantly improve the method's robustness to noise and error in point features.

During our experiments, we observe that Shapiro and Brady's algorithm is simple, reliable, and is able to deal with relatively large changes in scales and perspective changes. We recognize that local appearance features, such as SIFT and HOG features, are insufficient for our task of building recognition. This is because a building is represented by a collection of repeated structural components (nodes). Within each cluster, nodes share similar appearance. To address this issue, we propose to incorporate the following geometrical invariants to characterize the spatial configuration of nodes.

(1) *Area ratio of triangle pairs*. As illustrated in Fig. 3.4(B), for each cluster of nodes in the building sketch, we perform Delaunay triangulation and compute its area ratio of neighboring triangle pairs. Van Gool et al. [45] demonstrate that this geometric feature is invariant under perspective transform. In this work, we will use the average, minimum, and maximum area ratio of neighboring triangle pairs as our first set of geometric features.

(2) *Configuration* e*ntropy of k-hop neighbors*. We note that the area ratio feature has a strong discriminative power only when the cluster has a non-uniform pattern. To provide complimentary information for spatial configuration, we introduce the second features: the spatial distribution of k-hop neighbors. As illustrated in Figs. 3.4(C), on the Delaunay triangulation, we find the set of k-hop neighbors for each node, denoted by $\mathcal{N}_i = \{o_{i1}, o_{i2}, \cdots o_{iL}\}$, which will partition the circular area around the node into L sections, as shown in Fig. 3.4(D). Let p_l be the area of section o_l normalized by the total area of the circle. We define the following configuration entropy:

$$\boldsymbol{H_k}(\mathcal{N}_i) = \sum_{l=1}^{L} p_{il} \log_2 \frac{1}{p_{il}} \tag{3.2}$$

During our experiments, we find that $\boldsymbol{H}(\mathcal{N}_o)$ is very effective in differentiating boundary, corner, and inside nodes. In this work, we use the entropy of up to 3-hop neighbors as our second set of geometric features:

$$\boldsymbol{H}(\mathcal{N}_i) = [\,\boldsymbol{H_1}(\mathcal{N}_i), \boldsymbol{H_2}(\mathcal{N}_i), \boldsymbol{H_3}(\mathcal{N}_i)] \tag{3.3}$$

We note that different clusters of structural components may reside on different planes / sides of the 3-D building. To avoid the 3-D matching problem, we propose to perform spectral graph matching on each cluster independently and a joint decision is made on matching results of clusters for building recognition. An image region is determined to be match of the building in the query image if this region contains matches to all or most clusters in the query image. Let $\{\boldsymbol{F}_i = [\, \boldsymbol{f}_i \,|\, \boldsymbol{\alpha}_i, \boldsymbol{H}(\mathcal{N}_i) \,] \mid 1 \leq i \leq N\}$ be the set of augmented features for node i in the query image. Here, $\boldsymbol{f}_i$is the HOG appearance feature, α_i represents the area ratio statistics of neighboring triangle pairs, and $\boldsymbol{H}(\mathcal{N}_i)$ is the entropy of k-hop neighbors. Let $\{\boldsymbol{g_k} | 1 \leq k \leq M\}$ be the corresponding feature set for the reference image. For the query image, we construct a Gaussian-weighted proximity matrix:

$$\boldsymbol{F} = [\boldsymbol{F}_{ij}]_{N\times N}, \qquad \boldsymbol{F}_{ij} = e^{-\frac{d(\boldsymbol{f}_i,\boldsymbol{f_j})}{2\sigma^2}}. \tag{3.4}$$

Here, $d(\boldsymbol{f}_i, \boldsymbol{f_j})$ represents the distance between two features. Similarly, we can construct the proximity matrix

$$\boldsymbol{G} = [\boldsymbol{G}_{kl}]_{M\times M}, \qquad \boldsymbol{G}_{ij} = e^{-\frac{d(g_k, \boldsymbol{g_l})}{2\sigma^2}} \tag{3.5}$$

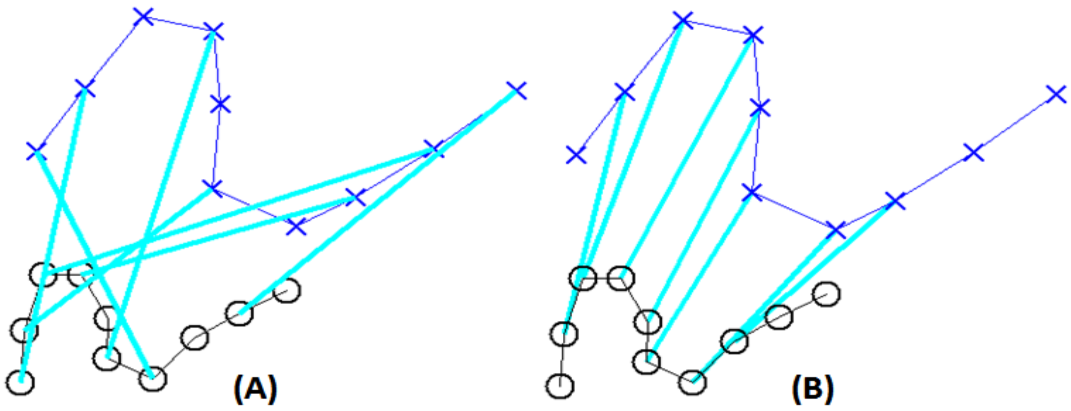


Figure 3.5: spectral graph matching: (A) with local appearance feature only and (B) using both appearance and geometric invariants of nodes.

for the reference image. Note that both are symmetric square matrices. Let

$$\boldsymbol{F} = \boldsymbol{V_1}\boldsymbol{D}_1\boldsymbol{V}_1^T, \qquad \boldsymbol{G} = \boldsymbol{V_2}\boldsymbol{D}_2\boldsymbol{V}_2^T, \tag{3.6}$$

be their singular value decomposition. Note that these two clusters in the query and references may have different number of nodes, i.e., $N \neq M$. Let $K = \min(N, M)$. Let $\{\xi_i^1\}_{1\leq i\leq K}$ and $\{\xi_i^2\}_{1\leq i\leq K}$ be the first K row vectors of matrices $\boldsymbol{V_1}$ and $\boldsymbol{V_2}$, respectively. We construct the following association matrix

$$\boldsymbol{Z} = [Z_{ij}]_{K\times K}, \qquad Z_{ij} = ||\xi_i^1 - \xi_i^2||^2 \tag{3.7}$$

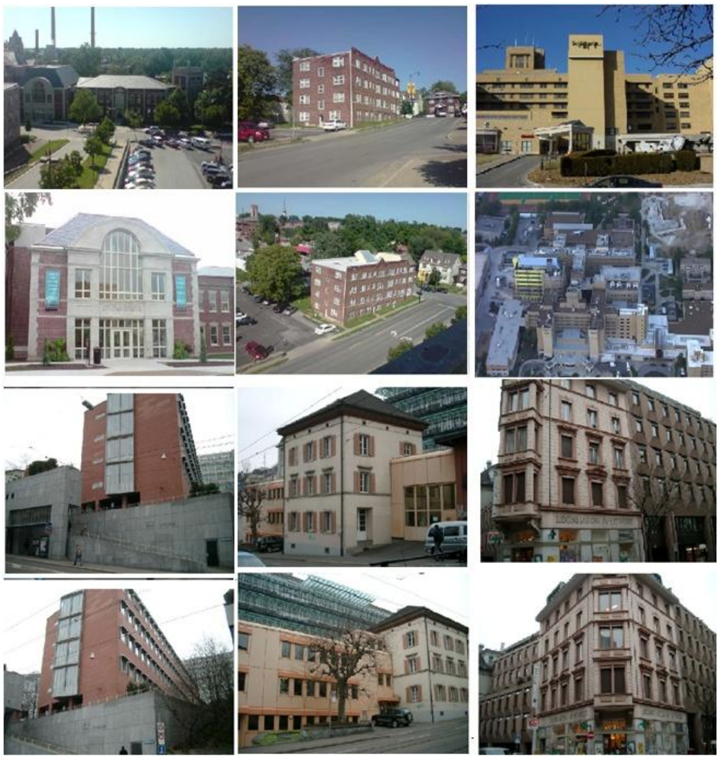


Figure 3.6: sample images from the ZuBuD and our Ground-Aerial datasets.

using the Euclidean distance. Following Shapiro and Brady's approach [42], we determine the node-to-node correspondence as follows: if Z_{ij} is the minimum of row i of matrix $\boldsymbol{Z}$, then node i in the query image is matched to node j in the reference image. Fig. 3.5 shows one example of matching result on point patterns. We assign the features extracted from the query image to points labeled with circles and those extracted from the reference image to points labeled with crosses. Fig. 3.5(A) shows the spectral graph matching results with local appearance feature only while Fig. 3.5(B) shows the results with both appearance and geometric features. We observe that the appearance features are useful in inter-cluster matching while the geometric features are useful in intra-cluster matching. In Section 3.4, we will conduct detailed evaluation on the matching performance.

During our building recognition, we first detect the MSERs in both query and reference images and identify clusters of structural components, as discussed in Section 3.1. Then, for each cluster in the query image, we perform the above spectral graph matching procedure to find the matching (a cluster) in the reference image. We also extract HOG features describe the façade of the building, i.e. surrounding areas of these structural components. A region in the reference image is determined to the match if it has the best overall matching performance on all clusters and the surrounding area.

3.4 Experimental Results

Experiments were conducted on two datasets. The first one is the ZuBuD database which has 201 buildings with 5 images per building. All buildings are randomly selected in Zurich, Switzerland. A detailed description of this database is provided in [46]. The second dataset, called Ground-Aerial, simulates the ground-aerial battlefield surveillance scenario discussed in Section 1. The dataset was about 500-1000 feet above the ground and collected continuous aerial videos of 23 buildings (mainly office buildings) on the ground. The dataset also has images about these buildings taken by a ground-level camera. Some example images from these two datasets are shown in Fig. 3.6. The top three pairs are from the Ground-Aerial and the bottom three pairs are from ZuBuD.

Our performance evaluation consists of two parts. First, we evaluate the local matching performance of the proposed method at the image patch level. Second, we evaluate its performance in building retrieval.

(1) ***Local Matching Performance Evaluation***

For each test image pair, we select those MSERs for major structural components of buildings, such as windows, doors, and exterior decorations. We then use the SIFT algorithm and our method to find the match for each MSER. We visually examine the correctness of matching and calculate the recall statistics:

$$recall = \frac{\#\, correct\ matches}{\#\, correspondences} \times 100\%. \tag{3.8}$$

Table I summarizes the recall statistics of both methods on both datasets. We can see that, by using semantically rich pattern-level features, the proposed method outperforms SIFT in matching the major structural components of buildings. It should be noted that, although the patch-level recall is low, especially on the challenging Ground-Aerial dataset, the overall building recognition performance will be much higher after aggregating the decision on a large number of image patches.

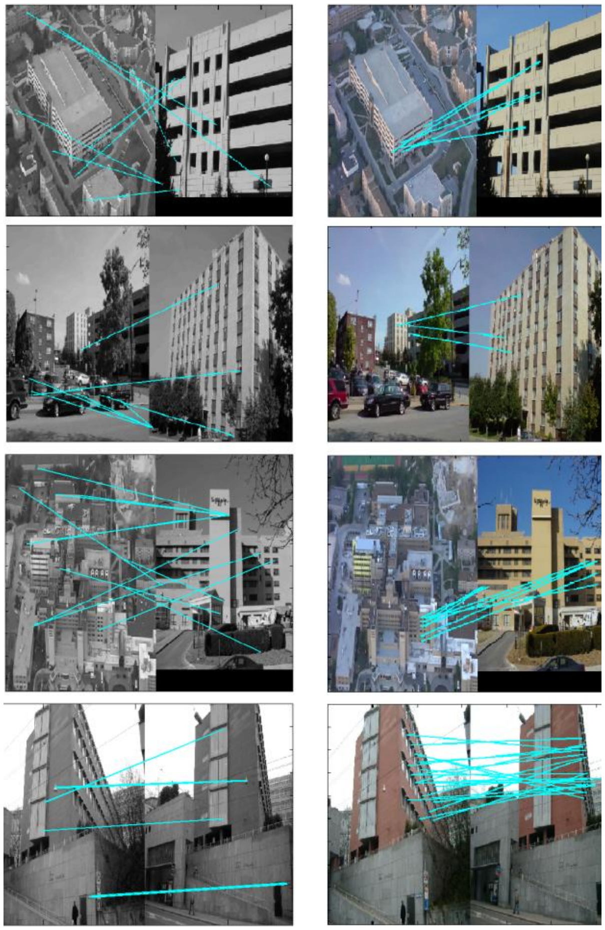


Figure 3.7: building matching performance comparison. Left: results by SIFT-based matching; right: results by this work.

Table 3.1: Image patch matching performance (recall in %)

Method	**Ground-Aerial**	**ZuBuD**
SIFT	21.0%	53.1%
This Work	50.3%	66.8%

*(2) **Building matching performance***

In this experiment, we use a query image, i.e. one camera view of a building, to find the building from a database which has images of other buildings and other views of the same building. We compare our method with the Zhang and Košecká's method [47] which is based on SIFT and voting and a probabilistic model. On the ZuBuD database, they reported an average of 90.4% correction recognition. During our experiments, we find that, for image with relatively small viewpoint changes, SIFT-based matching is very efficient and our algorithm has no clear advantage.

As discussed in Section 3.1, the focus of our effort is to address the challenge in building recognition with large changes in viewpoints. We select those 25 building image pairs with the largest changes in scales and viewpoints from the ZuBuD and Ground-Aerial datasets. Our algorithm achieves 81% correct recognition while the SIFT-based matching algorithm has 48% correct recognition. Fig. 3.7 shows 4 examples of building matching results with SIFT-based matching [47] and the proposed algorithm. It can be seen that the proposed algorithm is able to match building between images with large viewpoint changes.

3.5 Discussion

In this work, we have considered the problem of building recognition between images with large changes in scales and viewpoints. Based on multi-scale MSER detection, we detect repeated structural components of buildings and construct a sketch representation of building. Based on spectral graph analysis, we develop building recognition scheme.

Our experimental results demonstrated that the proposed method outperforms SIFT-based recognition schemes, especially for images with large viewpoint changes. In our future work, we shall extend this method to more generic building types by considering additional modalities of features, such as contour.

CHAPTER 4

3-D Change Detection from Multi-Source Videos

We study the problem of detecting changes from multi-source videos which are captured by different moving cameras with unknown parameters at different times. The objective of 3-D change detection is to identify keypoints, image patches or features that cannot find matches across camera views. As the key challenge, we need to make sure that these unmatched image patches only belong to new objects in the scene and nowhere else. We attack this problem by exploring a hierarchy of view-invariant image patch descriptors. Using the five-point algorithm, SIFT and RANSAC, we track the relative camera pose within each video and obtain an approximate cross-view registration and alignment of selected video frames. Based on multi-scale local binary pattern (LBP) description of super-pixels and middle-level image patch labeling, we construct a hierarchy of image patch descriptors and detect changes in the video scene using multi-scale information fusion with Choquet integrals. We have established an image/video database for 3-D change detection and will make it available for public use. Our extensive experimental results demonstrate that the proposed method achieves a detection rate of 61% at a false alarm rate of 2% while other approaches based on conventional local photometric image descriptors fail to detect changes in the 3-D environment.

4.1 Introduction

3-D change detection from multi-source videos has great potentials in battlefield intelligence, infrastructure security, robot navigation, etc. For example, a video cam-era can be deployed on a soldier, a convoy, or an un-manned ground vehicle (UGV) to automatically detect changes in its surrounding environment when it passes through a combat zone which has been surveyed before. Changes might indicate adversary actions and potential hazards. 3-D change detection, if successfully developed, can be also used in many homeland security scenarios to detect new modifications or damages to infrastructures, such as buildings, bridges, and driveways, as well as left objects from patrol vehicles in areas which cannot be covered by stationary surveillance cameras. This task might be hard or even impossible for human beings due to our limited memory and data processing capabilities.

In 3-D change detection from multi-source videos, we need to compare two video sequences captured by different cameras with unknown parameters at different times. Because of the unrestricted motion of persons or vehicles which carries the camera, there are often large changes in scales and perspectives between these two camera views. This poses significant challenges in 3-D change detection from multi-source videos. In conventional problems of wide baseline matching [62], multi-view stereo [55, 97], and object tracking [59], the objective is to find matches between camera views. However, in 3-D change detection, the problem becomes more challenging. First, we need to make sure image patches belonging to the existing scene accurately match across camera views. Second, we need to identify keypoints, image patches or features that cannot find matches across camera views and make sure that they only belong to new objects and nowhere else.

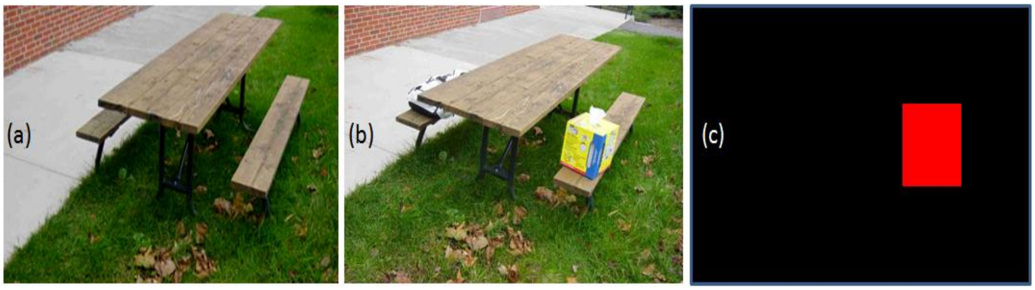


Figure 4.1: 3-D change detection; (a) and (b): two videos taken by different cameras at different times; (c) detection result. The red box represents changes that indicate new object(s).

4.1.1 Related Work

In the past decade, we have seen significant advances in the development of local photometric descriptors for image matching, such as SIFT [59], PCA-SIFT [63], HoG (histogram of oriented gradients) [60], shape context [64], etc. The central goal of local image description is to make them invariant under image transforms and camera motion, such as image rotations, camera zoom, changes in scale and perspectives, and image noise, while maintaining high repeatability and discriminative power [59, 60, 96]. There are two major issues in direct use of these local image descriptors for 3-D change detection. First, due to large changes in camera scales and perspectives, there are often a large number of unmatched keypoints, which are not necessarily new objects. In our experiments, we observe that SIFT keypoints with the largest feature distance are not necessary on the new objects. For example, in Figure 4.2, we show the top 5%, 10%, and 25% of keypoints in the input image. We can see that most keypoints with the largest distance are not on the new object, the yellow box. Second, keypoints may not be generated or selected on the new objects. In this case, the new objects will be missed during change detection.

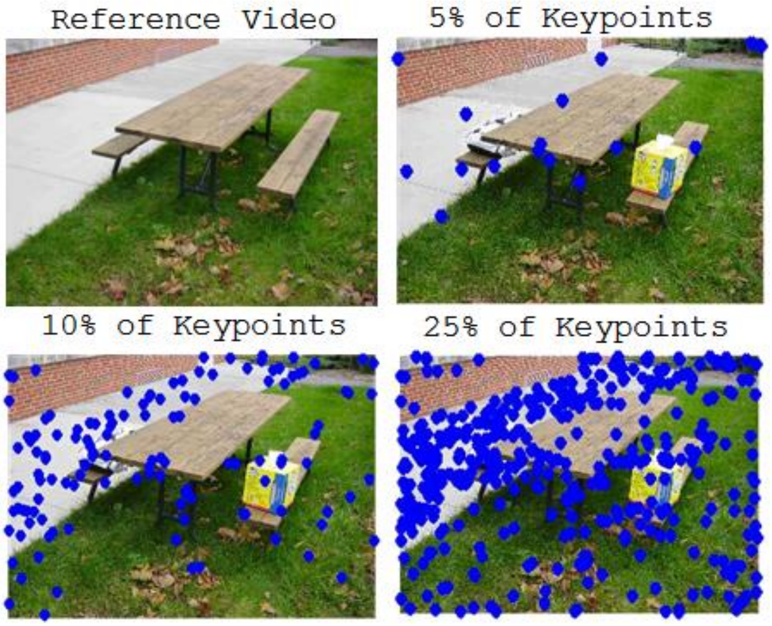


Figure 4.2: SIFT matching for change detection by assuming that keypoints with largest matching distance are from the new objects. Here, we show the top 5%, 10%, and 25% of keypoints with the largest distance.

For stationary cameras, change detection is often formulated as background modeling, subtraction, and moving object detection problem [49, 94]. A thorough survey of such algorithms is given in [50]. Change detection has also been well-studied in the area of aerial surveillance for detecting moving vehicles, new constructions of buildings and roads [65]. When the aircraft is flying high and the camera viewing distance is large and the ground object structure is relatively small, this problem reduces to geospatial video registration and pixel or image patch classification, distinguish significant changes such as new buildings from insignificant changes in vegetation, weather, and lighting conditions [65, 70, 71]. When the 3-D structure of ground objects, such as buildings, becomes more significant, occlusion and motion parallax are two major issues in change detection [72, 93]. Pollard and Mundy developed a 3-D voxel-based method for change detection in an urban environment from aerial surveillance images, where probability distributions for surface occupancy and image appearance are stored in each voxel [48]. This method is able to efficiently handle occlusion and motion parallax. Image registration and change detection have also been studied for medical image analysis [57, 58].

The ideal situation in 3-D chance detection is to create a 3-D model of the scene from both videos and compare their models in the 3-D domain. Differences between models then indicate new objects. A number of promising algorithms have been developed in the literature for 3-D scene reconstruction using structure from motion [51, 53] or stereovision [54] methods. Saxena et al developed a learning-based method to infer 3-D structures from a single image using Markov Random Field (MRF) analysis [52, 92]. It has successfully created qualitatively correct 3-d models for 64.9% of 588 images downloaded from the internet. Many algorithms are able to create photo-realistic 3-d models which are both qualitatively accurate and visually pleasing. However, because of the intrinsic ambiguity between local image features and the 3-d location of the point, it remains a very challenging task to construct a quantitatively accurate 3-D model from images [54, 55, 56]. If the 3-D model is not quantitatively accurate and the local alignment is precise, it will generate a large number of false positives during change detection. Furthermore, 3-D modeling and scene reconstruction often suffer from computational complexity.

4.2 Camera Pose Tracking and Frame Registration

To detect changes, we first need to make sure that two video frames under comparison have sufficient overlap of camera views. To this end, we need to track the relative pose of the camera in each video to determine the subset of video frames from both videos that have sufficient overlap of camera views. This consists of two major operations, intra-view camera pose tracking and inter-view frame registration. For the intra-view camera pose tracking, we follow Nistér's five-point algorithm [61]. More specifically, we find SIFT matching between adjacent frames within each video sequence. With the random sample consensus scheme (RANSAC), we estimate the fundamental matrix and camera parameters up to a scaling factor [61]. In the mean time, we also perform cross-view SIFT matching in a one-to-multiple manner. More specifically, let $F_A(n)$ and $F_B(m)$ be the n-th and m-th frames of videos A and B, respectively. We apply SIFT matching between these two frames and let $d_S[F_A(n),\ F_B(m)]$ be the average distance of matched keypoints. Let $\rho_S[F_A(n),\ F_B(m)]$ be fraction of keypoints that are matched between these two frames. We define the following distance metric

$$D[F_A(n),\ F_B(m)] = \beta \cdot d_S[F_A(n),\ F_B(m)] + \quad \rho_S[F_A(n), \qquad F_B(m)] \tag{4.1}$$

where β is a normalization parameter. This distance metric is used to measure the content similarity between two video frames. It consists two major indicators: the fraction of matched keypoints $\rho_S[F_A(n),\ F_B(m)]$ and the average matching distance $d_S[F_A(n),\ F_B(m)]$. We observed that video frames from these two views may often lose their synchronization even after their initial synchronization. Therefore, we need to find its best match within a neighborhood. More specifically, for frame $F_A(n)$ in Video A, we find its best match by minimizing the distance metric in (4.1) as follows:

$$m^* = \arg \min_{m \in [n-\Delta, n+\Delta]} D[F_A(n), F_B(m)]. \tag{4.2}$$

Here, Δ is the size of the search window in units of frames. Once the best matching has been found, we can determine the homography between these two frames. In this work, we use the following 8-parameter perspective transform:

$$\begin{bmatrix} X \cdot W \\ Y \cdot W \\ W \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{4.3}$$

where $[x, y]$ and $[X, Y]$ are the keypoint coordinates in frames $F_A(n)$ and $F_B(m^*)$, respectively. With bundle adjustment, we can find the optimal set of parameters $\{a, b, c, d, e, f, g, h,\}$ to best fit the matched SIFT keypoints. Figure 4.4 shows one example of view registration results. It should be noted, even after this registration, the frame-to-frame alignment error could still be large due to the 3-D structure of the video scene.

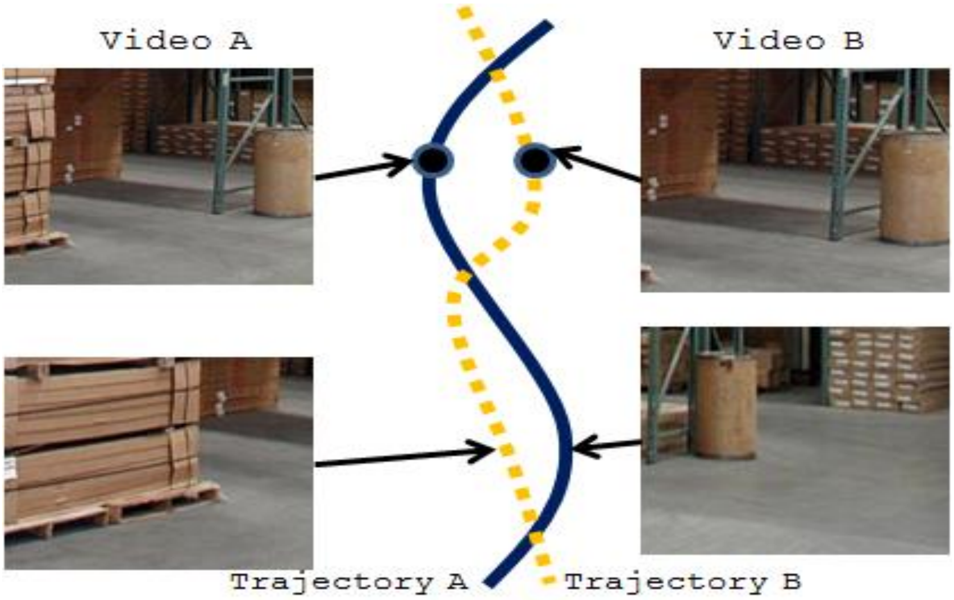


Figure 4.3: relative camera pose tracking and approximate view registration.

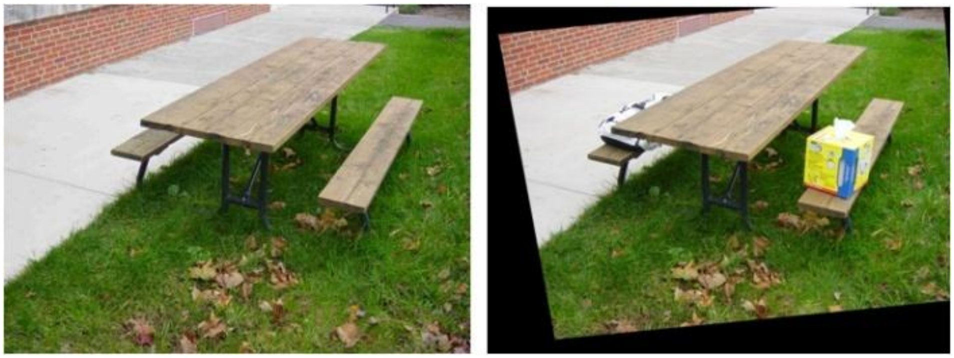


Figure 4.4: camera pose tracking and inter-view frame registration result.

Note that the above inter-view registration operation is computationally intensive. We can couple the intra-view camera pose tracking with this inter-view synchronization. Specifically, from the camera-pose tracking of both videos, we can determine the video segments where their camera views are similar. We then perform the inter-view synchronization and registration only within these segments for change detection.

4.3 Hierarchical View-Invariant Image Patch Description

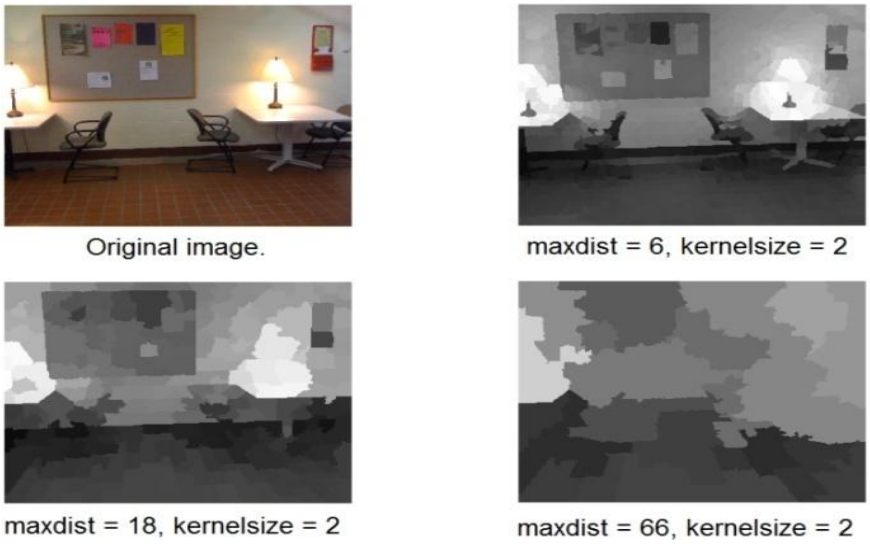
We observe that existing local photometric descriptors for image matching such as SIFT [59, 98] and HoG (histogram of oriented gradients) [60] cannot be directly used for our purpose of 3-D change detection due to the following two reasons. First, these low-level image features are still sensitive to large changes in camera scale and perspective, which will generate a large number of mismatches. Second, selected keypoints might not be on the new objects, which will result in missed detection. For accurate and reliable 3-D change detection, we propose to use a multi-scale fusion-based approach for change detection based on super-pixels and view-invariant image patch descriptors.

4.3.1 Multi-Scale Super-Pixel and LBP Features

First, we find small homogeneous regions in the image, called s*uper-pixels* and use them as our basic unit of image matching and change detection [52, 66, and 67]. We use Vedaldi and Soatto's quick shift and kernel method to construct super-pixels from the original video frame [67, 95]. We observe that high-frequency image components, such as textures and local gradients, are sensitive to changes in camera scales and perspectives. To control the amount of high-frequency details being incorporated in the local descriptor, we use multi-scale super-pixels. Figure 4.5 shows one example of multi-scale super-pixels produced with different kernel sizes.

For the super-pixel image at each scale, we use the local binary pattern (LBP) descriptions and their histograms. At different scales, the LBP operator is extended to different neighborhood sizes. The video frame under comparison is partitioned into blocks. For each block in frame $F_A(n)$, we search all blocks with its neighborhood in frame

$F_B(m^*)$ and find the best match block with the minimum LBP feature distance. This distance will be used later for computing the confidence map for change detection.



Original image.

maxdist = 6, kernelsize = 2

maxdist = 18, kernelsize = 2

maxdist = 66, kernelsize = 2

Figure 4.5: Super-pixels generation with different kernel sizes using Vedaldi and Soatto's quick shift and kernel method [67].

4.3.2 Middle-level Image Patch Description

We observe that the above multi-scale LBP description of super-pixels is not sufficient for accurate and reliable change detection. At large scales, it is only able to detect image patches with significant changes and rough location of the new objects. When the scale reduces, more high-frequency components are incorporated into the local patch description, making the matching and change detection more sensitive and less stable with scale and perspective changes. To address this issue, we introduce a middle-level description of image patches which is view-invariant and is able to further refine the change detection results.

Our basic idea is to label each local image patch based on its internal texture pattern and use the histogram of its neighboring patch labels as a middle-level patch descriptor.

More specifically, using a learning-based approach, we label each image patch into the following 6 categories: smooth, with a single extreme, with smooth transition, with a single edge, with multiple edges, and with random textures, as illustrated in Figure 4.6. To this end, we first use the HoG features [60]. To more efficiently capture the internal edge pattern of the image patch, we propose to use the finite ridgelet transform which has found successful applications in image denoising, restoration, and compression [68].

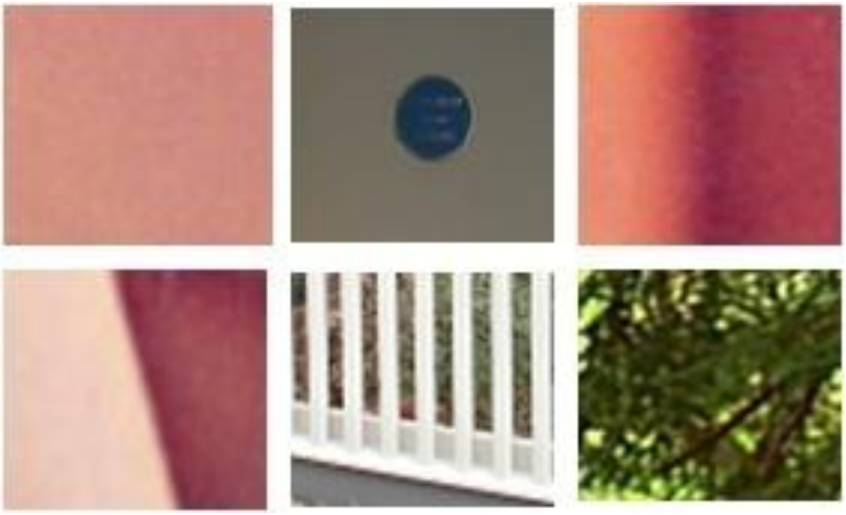


Figure 4.6: automatic labeling of image patches into 6 categories: smooth, with a single extreme, with smooth transition, with a single edge, with multiple edges, and with random textures.

Given an integrable 2-D function $f(\boldsymbol{x})$, for example, an image, its continuous ridgelet transform (CRT) is defined as

$$\Re_f(a,b,\theta) = \int \phi[a,b,\theta](\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}, \tag{4.4}$$

where the ridgelets $\phi[a,b,\theta](\boldsymbol{x})$ in 2-D are defined by a wavelet-type function:

$$\phi[a,b,\theta](\boldsymbol{x}) = a^{-\frac{1}{2}}\,\phi\left(\frac{x_1 cos\theta + x_2 sin\theta - b}{a}\right) \tag{4.5}$$

In practice, an image is a discrete set of pixels. Its discrete ridgelet transform, or finite ridgelet transform (FRT), [099], can be implemented by summations of image pixels along a certain set of lines. These lines are defined in a finite geometry. Let $Z_p = \{0, 1, \cdots, p-1\}$ be a finite field with modulo p operations, where p is a prime number [17]. The FRT of an image patch of size $p \times p$ is defined as

$$f_R[k,l] = \frac{1}{\sqrt{p}} \sum_{(i,j)\in L_{kl}} F(n)[i,j]. \tag{4.6}$$

Here, L_{kl} denotes a line of pixels or a scanning pattern of the image patch:

$$L_{kl} = \{[i,j] : j = ki + l \ \ (mod\, p), i \in Z_p\} \tag{4.7}$$

and as a special case

$$L_{pl} = \{[l,j] : j \in Z_p\}. \tag{4.8}$$

Figure 4.7 shows one example of these scanning line patterns for an image patch of size 7×7. After FRT, we use the histogram of the $f_R[k,l]$ as our second set of features. We hand label a large set of training set of image patches and use SVM (support vector machine) to learn the classifier. Once each image patch is labeled into one of those 6 categories, we use the histogram of neighboring patch labels as a feature to find the best match in its neighborhood and obtain its minimum distance. The minimum distance will be used later to compute the confidence for change detection.

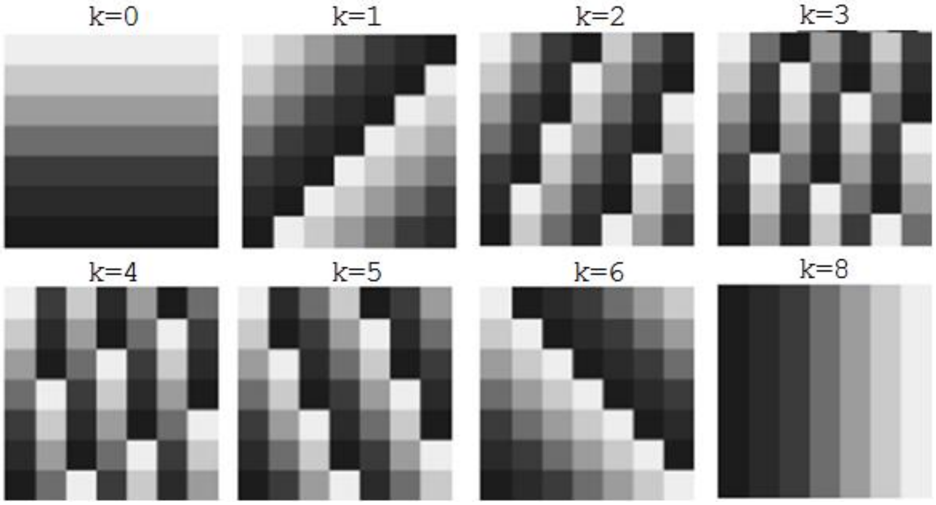


Figure 4.7: lines used finite ridgetlet transform of a 7×7 image patch. Each sub-figure is indexed by k, which represents the slope of the lines. Pixels of the same intensity with the same value of l are averaged together to produce the FRT output $f_R[k,l]$.

4.4 Multi-Scale Fusion for Change Detection

Our approach to change detection hinges on the ability to automatically fuse various pieces of partial evidence at multi-scales into a global assessment. At each scale, using the LBP description of super-pixels, the middle-level patch description, correlation-based matching with the neighborhood, we can obtain distance map for each patch in the video frame. Patches of large distance values will more likely be new objects. To form a joint decision of change detection, we propose to aggregate this type of uncertainty information (described in the form of probability or membership function) at multiple scales using Choquet integral [69]. The Choquet integral considers a non-additive measure and confidence outputs over a set X of information sources, such as features, samples classification, clustering analysis results, etc. A non-additive measure is a set function $g: 2^X \to [0, 1]$, such that

(a) $g(\emptyset) = 0; g(X) = 1;$

(b) $g(B) \geq g(A)$ if $B \supset A$;

(c) $\lim g(A_i) = g(\cup A_i)$ if $\cdots \supset A_i \supset \cdots \supset A_1$.

Let $h: X \to [0, 1]$ be a confidence function. The continuous version of Choquet integral is given by

$$C = \int_X h(x) \circ g = \int_0^1 g(A_\alpha) d\alpha, \tag{4.9}$$

where

$$A_\alpha = \{x \,| h(x) \geq \alpha\}. \tag{4.10}$$

Its discrete version can be computed with the following summation [69]:

$$C = \sum_{i=1}^{n} h(x_i)[g(A_i) - g(A_{i+1})] = \sum_{i=1}^{n} [h(x_i) - h(x_{i+1}) g(A_i) \tag{4.11}$$

In this work, the non-additive measure $g(A)$ represents the probability for an image patch A to be a change, which is computed based on its matching distance in comparison with the decision threshold. In our experiments, we assume that the distances of image patches follow a zero-mean half-normal distribution with variance σ^2 and set

$$g(A) = \frac{2\sigma}{\pi} e^{-d(A)^2\sigma^2/\pi},$$

(4.12)

where $d(A)$ is the best matched feature distance for patch A. The Choquet integral will produce an fused confidence map indicating the probability for a patch to be a change. We select the region with the maximum likelihood, for example, top 5%, as the change detection output.

4.5 Experimental Results

In this section, we describe our database for performance evaluation and present the experimental results.

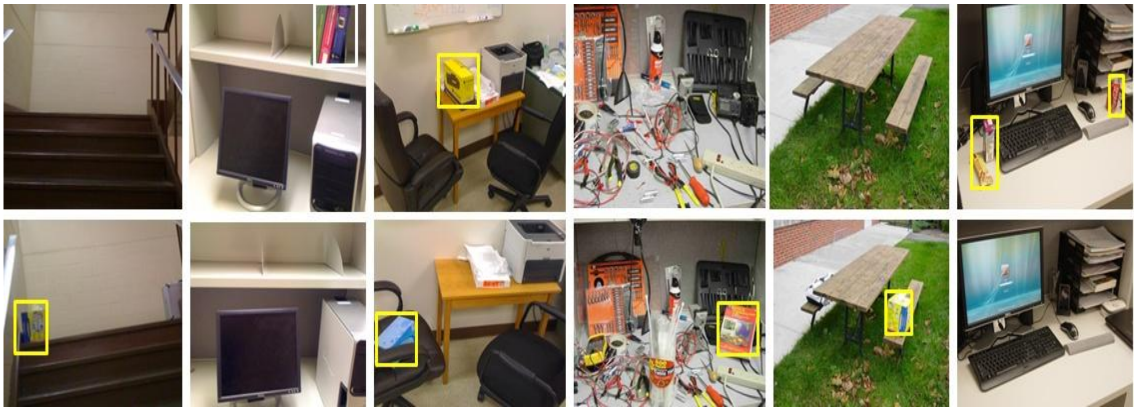


Figure 4.8: sample video frames from our database. Here, we show 6 pairs of video frames with new objects being hand labeled with a bounding box.

4.5.1 Database for 3-D Change Detection

We have established a database for 3-D change detection. We have collected image/video samples from 12 indoor and outdoor environments. We then introduce 1-2 new objects in the scene, such as boxes or bags, walk over the same scene from different perspective and record a set of samples. The set size ranges from 20 to 30. From this set, we can choose any pair and use them as test data for 3-D change detection. That means, we have more than $\binom{2}{20} \times 12 = 4560$ possible test cases. For each sample with new objects, we hand label the location of new objects with one or more bounding boxes, as shown in Figure 4.8. This dataset will be made available for public use.

4.5.2 3-D Change Detection with SIFT

To our best knowledge, our work is among the first to detect changes in a 3-D environment from videos captured by moving cameras with large view variations. We could not identify similar work in the literature for effective performance comparison. Instead, we adapt the SIFT matching for 3-D change detection and use it for performance comparison. Specifically, we find SIFT points on two video frames F_A (n) and F_B (m^*) under comparison. For each keypoint in F_A (n), we find its the minimum distance to those keypoints in F_B (m^*). We choose the top μ% of keypoints with the largest distance as indicators of new objects. We refer to these keypoints as new keypoints. We then compare it with the hand labeled ground-truth and compute the fraction of new keypoints, denoted by β, that are within the bounding box of the new object. In Figure 4.9, we show the percentage of new keypoints that fall onto the new objects using the original images as the input. The x-axis is the percentage of keypoints being chosen as new keypoints that have the largest distance. It also shows the results if we use the warped images after view registration. We can see that very few, less than 7% of new keypoints, are falling onto the new objects. Furthermore, the view registration does not help the SIFT matching. Instead, it even degrades the performance since the local gradient information has been distorted by the registration process. Figure 4.10 shows one example with 5%, 10%, and 25% of keypoints. This shows that 3-D change detection using conventional local photometric descriptors do not work.

4.5.3 3-D Change Detection Using the Proposed Algorithm

We implement the proposed 3-D change detection method and test its performance using our image database de-scribed in Section 4.1. Figure 4.11 shows four examples of 3-D change detection results. Our criterion for successful detection is that, if the fraction of detected image patches is larger than 50%, we claim a successful detection. Figure 4.12 shows the ROC curve obtained. It can be seen that the proposed method achieves a 61% of object detection rate at a false alarm rate of 2%. This level of performance is very promising, given that 3-D change detection remains an open and challenging problem. In Singh et al's recent work [73], they have achieved near 70% of detection rate at a false alarm rate of 2% on the PETS 2006 database. It should be noted that these are surveillance videos captured by stationary cameras. To our best knowledge, our work is among the first attempting to detect changes in a 3-D environment from multi-source videos captured by different moving cameras. The problem is much more challenging. This work is the first step of our efforts on this challenging problem. In our future work, we shall further refine the algorithm and improve the detection performance.

4.6 Conclusion

In this work, we studied the problem of change detection in a 3-D environment from multi-source videos captured by different cameras at different times. The key challenge here is to make sure image patches in the existing video scene are able to accurately find matches between camera views with relatively large changes in scales and perspectives. In the meantime, we need to make sure that t image patches belonging to the new object cannot find good matches and those unmatched patches are only within the new object. To address these issues, we proposed a hierarchy of view-invariant image patch descriptors based on multi-scale super-pixels representation, LBP features, and middle-level patch labeling. We developed a multi-scale fusion scheme for change detection based Choquet integrals. Our experimental results demonstrated that the proposed method achieves very promising performance.

In our future work, we shall further improve the detection accuracy and reliability of the method. One interesting method we would like to explore is the learning-based approach, learning the characteristics of image patches from past video frames and the reference videos for change detection. We will also couple object tracking with 3-D change detection and extend the fusion process to multiple video frames. We expect that this will significantly improve the overall performance and reduce the computational complexity.

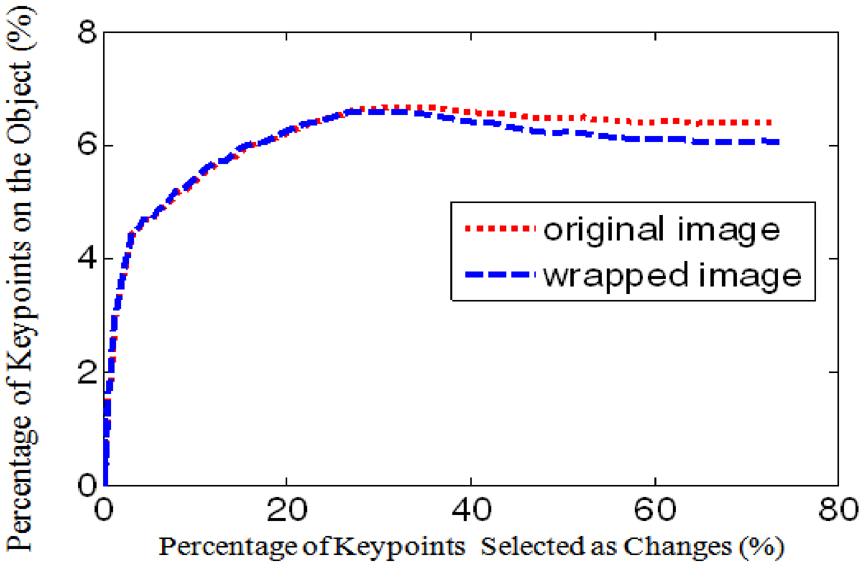


Figure 4.9: the percentage of SIFT keypoints with the largest distance that fall onto the new objects with and without view registration.

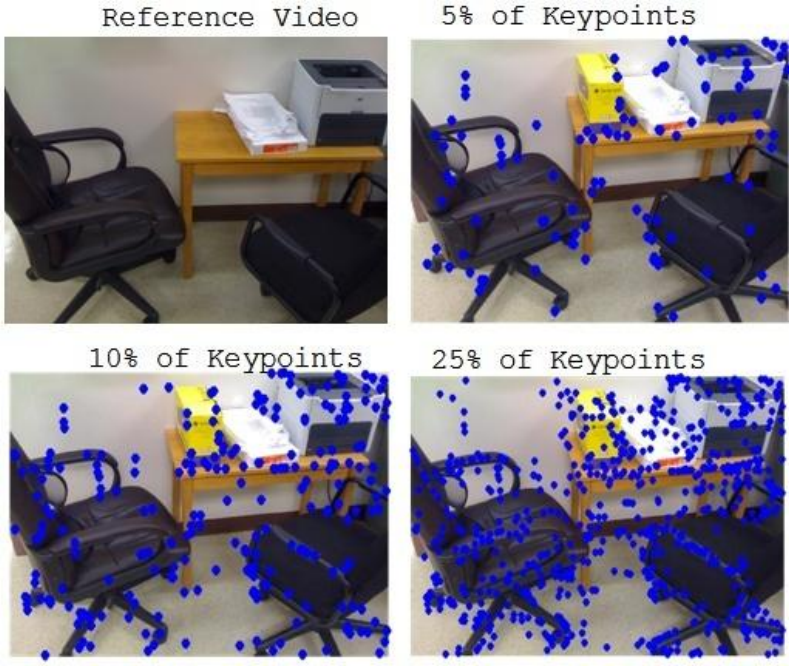


Figure 4.10: SIFT matching for change detection by assuming that keypoints with largest matching distance are from the new objects. Here, we show the top 5%, 10%, and 25% of keypoints with the largest distance.

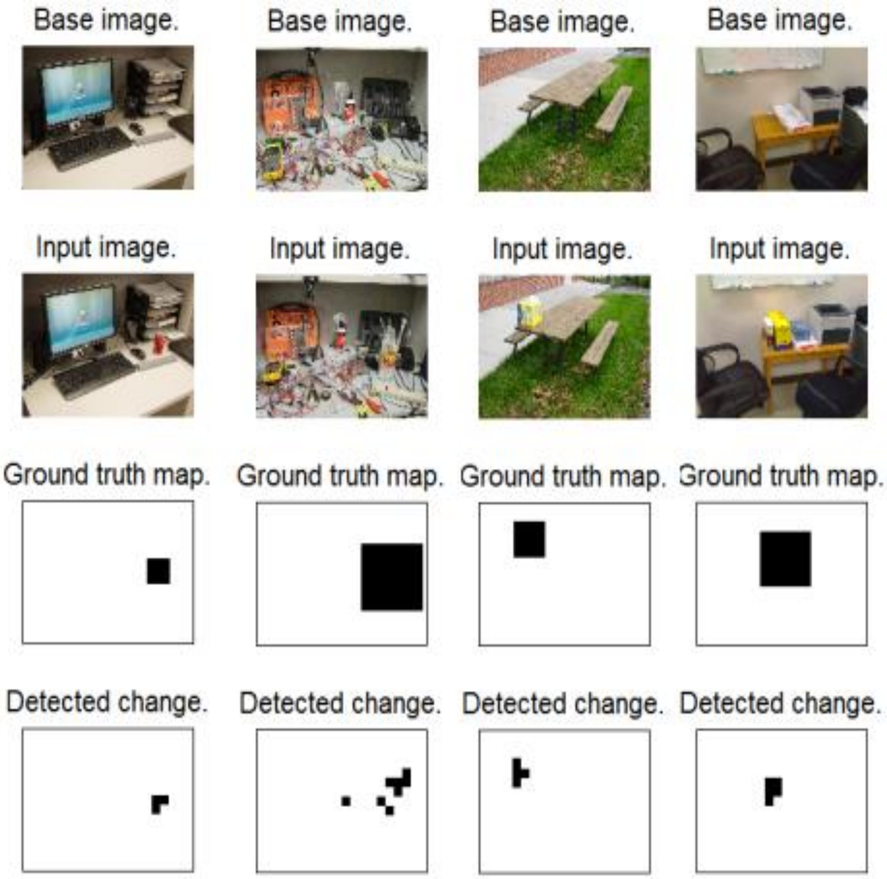


Figure 4.11: examples of change detection results. The first row shows video frames from the reference video; the second row shows frames from the input video; the third row shows the hand labeled ground-truth; and the fourth row shows the detection results.

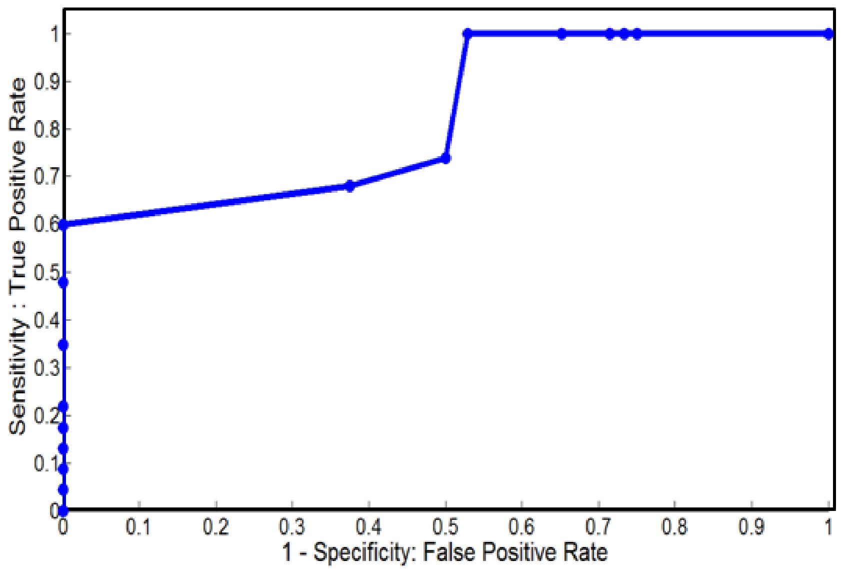


Figure 4.12: ROC curve of the proposed 3-D change detection.

CHAPTER 5

DISTRIBUTED CODING FOR SURVEILLANCE NETWORK

Chapters 2 to 4 are devoted to an elaboration upon computer vision tools that tackle the tedious task of video summarization for intelligence. In this chapter, video processing will go beyond content retrieval to explore the possibility of building an efficient video coding paradigm wherein surveillance network. This is to consider a power-limited surveillance network of cameras mounted on mobile units which differ from those tethered in a ground-wired system. In application as airborne surveillance, UAVs require low-complexity acquisition-and-storage devices. Therefore the video coder and decoder (codec) need to be power-efficient and it shall increase the capability of mobile surveillance network via prolonged coverage with proper video coding choice.

5.1 Introduction

Many advanced video archiving codec as the ITU-T H.26x or MPEG utilizing motion compensation and entropy coding. As shown in table 2.1 from chapter 2 the motion search would consume the most computational complexity in motion compensation. The mobile units, however, cannot acquire unlimited power supply while operating and must have efficient video encoding. Hence, it is a straight forward solution to adopt a low-complexity encoder which shifts the bulk of computation to the decoder. Without losing generality, this work narrows the problem to study a network of small UAVs with limited computational and communication channel resources yet to perform collaborative video surveillance. Based on Wyner and Ziv's theoretic results from 1970s and channel codes, we develop a distributed video compression scheme for a UAV network. The idea is for small UAVs to perform channel coding instead of source coding to limit the cost on video acquisition. The computational cost is then shifted to the decoder end which could be Eagle Eye or ground command base. In this way, all the intensive computation is done by the decoding and leave encoder with power-efficient task to encode videos. Nevertheless, its rate-distortion behavior has to be carefully studied for the control of video quality. The simulated results in section 5.4 will demonstrate the promising performance of this work and its potentials.

Starting with section 5.2, an overview is given to brief the integration of distributed video coding (DVC), channel code and the UAV surveillance platform. Section 5.3 expends the history of DVC and its potential on this work. A simulated video rate-distortion analysis is given in section 5.4 for a comparison of existing works and our edge. Section 5.5 sums up the challenge and potentials of this work.

5.2 Collaborative Video Compression Using Distribution Video Coding

In a surveillance network, autonomous vehicle mounted with video camera has high mobility and safe for patrolling on the battlefield and hazardous area in industry. Airborne surveillance usually employs unmanned vehicles to patrol or monitor the sky. An unmanned airborne vehicle (UAV) is piloted by onboard computer with global position system or navigated with ground base control. One or multiple video cameras combined can provide a wide coverage over a broad surveillance area. In practice videos are registered offline to perform content analysis such as change detection. Video sequences are stitched together to create a bigger picture of view so called panorama, as shown in Fig. 2.23. Hence, archived airborne videos that panned over a wide area or circled around have *mutual information* that shares with each other and may overlap with the new incoming video footage. This makes the *side information* required by distributed video coding (DVC) available from image registration and enable the low-complexity encoder using DVC.

In video compression, the basic thinking is to remove the redundant information which only needed to be encoded once. Encoder verifies the essential but repeated elements and only to code new or the changed portions. The post processing for surveillance video, such as background extraction or registration introduced in chapter 2, works as information fusion and sum up the essential but redundant portion of video footages. The key idea of this work is to utilize the coded footage as side information and remove the computational burden from the onboard encoder. Here the *channel coding* scheme is introduced to build the efficient video encoder. With this idea, for a routine-fly UAV drone, it only needs to encode video with few parity-check or syndrome bits. Unseen object or changed areas are recorded by correcting the previously footage on its flight routine. This codec suits even better for fixed-camera surveillance video, since the lack of camera motion renders high correlation on video frames. Here we focus on tackling the aerial surveillance problem because mobile units require efficient codec to cope with the power consumption problem.

Another question may be raised is, how does channel or distributed coding compared to the conventional source coding in terms of performance. In the scenario of airborne video surveillance, an on-board video camera must be power efficient and memory-saving for storage. For the state-of-art video encoding, high compression rate or low data bit rate cost high computational power at the encoder end. On the other hand, if introducing intra-coding, simple video encoder scheme would save on computational power which prolongs the battery life but shortens the playback time in videos. These lead to one tradeoff problem between power saving for video encoder scheme and limited memory storage quota. The state of art video transcoder can encode a video sequence with less than a bit rate of 50 kbps to perform an over 40 dB PSNR visual quality. Draw back from the advanced video encoding, however, is that the computational cost is high in terms of power consumption and compression time.

In this chapter a channel codec, Low-Density Parity-Check Accumulate (LDPCA) code [75], without feedback channel is introduced and shows its potential application on a power-efficient encoding scheme for video surveillance. Figure 5.1 gives an overview for the proposed system. At the encoder side onboard an UAV, distributed coder with transform-domain bit-plane channel coding is used to compress the video. The acquired sequence can be archived for post processing or transmitted via an error-prone channel in real-time. At the decoder end, the side information for the acquired sequence is registered from another pass of the video archiving which can be footage from high-altitude air patrol or satellite remote sensing. The channel decoder, LDPCA, then joint the side information and parity bit to decode the video. In case few bit planes fail in decoding, the decoder can replace the bit plane with the coarser version from the side information. This gives us a better error resistant in rate-distortion behavior while operating with lower bandwidth or severe noise.

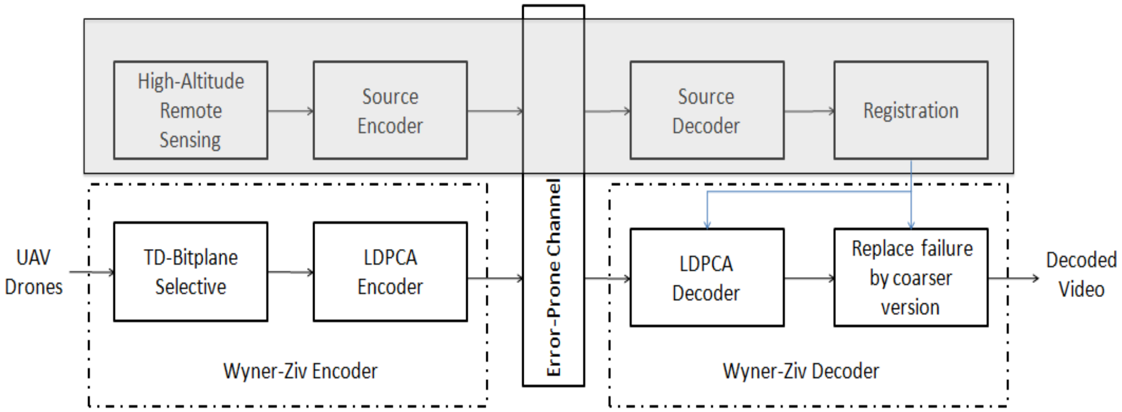


Figure 5.1: The layout of distributed coding for mobile surveillance network.

The next section will walk through the background of distributed video coding and details the design and implementation of proposed coding scheme.

5.3 Distributed Video Coding

A majority of lossless video coding belongs to source coding which intends to remove the redundant information from source signal sequence and achieve high compression rate. In the state of art codec such as H.264 standard, two coding approaches are adopted to explore the redundant information. First, *intra-frame* coding exploits the pixel correlation lie within single video frame and removes the redundant information. The *inter-frame* coding, on the other hand, exploits the spatial-temporal pixel correlation and removes redundancy across video frames inside a group of picture (GOP). Theoretically, intra-coding is with low complexity comparing to inter-coding yet introduce more redundancy on compressed video sequence. These two combined also result in an asymmetry that encoding has typically 5 to 10 times more complexity than decoding task in current video coders.

In 1976, Wyner-Ziv video coding [74] revealed a possibility to achieve inter-frame coding efficiency with an hybrid intra-frame encoder and inter-frame decoder video codec. It was started by Slepian and Wolf's work in 1973, [100], that proving two dependent sources, it can be jointly decoded and approaching the channel capacity according to Shannon's limit. The work has shortly been followed by Wyner and Ziv to provide a counterpart of distributed coding that does lossy compression.

Distributed coding refers to encoding sources A and B (or more) independently but to perform joined decoding for these multiple sources. It also introduced the idea of *side information* which is an erroneous version of original encoded message. For instance, we refer a coarse version of A as the side information of B since these two sources are dependent and can be jointly decoded. Based on side information provided for B, the decoder can simply pull parity bits of B to correct the broken message and recovers the original sequence. It is of our interest to study the case that the side information is not available at the encoder, and how to predict the minimum (parity check) bit rate to recover the original message at the decoder end.

Among the most popular implementations of Slepian-Wolf and Wyner-Ziv codec, a feedback channel is required for decoder to retrieve an increment of bits from encoders.

Upon requiring each increment bits, decoders can improve the bit-error rate and achieve the desired rate-distortion performance.

The state of art distributed video transcoder has also made the most from conventional inta-frame motion compensation because of its high compression rate. Most of distributed codec which required side information introduce a hybrid coding scheme. For instance [77] and [78] encode half of its frames as key frames with conventional intra-coded image. The side information for Wyner-zive frames are then obtained by interpolation or motion compensation between key frames in a GOP. Nevertheless, the complexity and computational cost for intra coding remains uncut and needed feedback channel to adapt its bit rate. It is not feasible for airborne video capture to bear on the computational cost introduced by motion-compensated coding nor is the feedback channel available. Hence, this work focus on utilizing computer vision tools to register the side information at the decoder end. It not only shifts the computational complexity to the decoder but also proposes a solution to rate allocation in the circumstance that there is no real-time feedback channel for the UAV surveillance. The next section will expand on retrieving and preparing side information for distributed coding that achieves good visual quality and robust to noise.

5.3.1 Side Information

In conventional DVC, the side information for decoder came from independent channel coders. People become interested in WZ-DVC because of its error resistance behavior in the lower bit rate. Therefore it has been a great effort made toward an adapted version of hybrid codec combining source coding and channel coding, [75]. The preceding works in the literature, however, only coded even frames (or sub-frame rate) with DVC. This is to acquire the side information from motion-compensated frames or through interpolation. Indeed it is one way to adopt DVC coding but the rate-distortion performance only counted for a portion of original video sequence. Unlike broadcasting or video archiving application, fortunately, video surveillance records similar and repetitive content in it natural practice. One obvious example is in the warehouse security cam which has almost the same footage (background). That highly dependent video footage is a perfect fit for distributed coding. While the idea seems promising for us to adopt DVC for surveillance video, there is still one trick lying behind until we can practice it for airborne surveillance - the camera motion. To acquire the side information and enable the joint decoding for airborne video, this work proposes using geo-tagging information for decoders to retrieve dependent video footage. Combining onboard GPS and gyroscope, all surveillance video can be geo-tagged and the extra data rate is trivial comparing to video itself. Figure 5.2 shows a set of simulated side information (the second row) prepared in the decoder end, it serves as a coarse version of original sequence (top row) to be recovered. One most obvious corruption perceived on side information is the blocky effect due to down sampling.

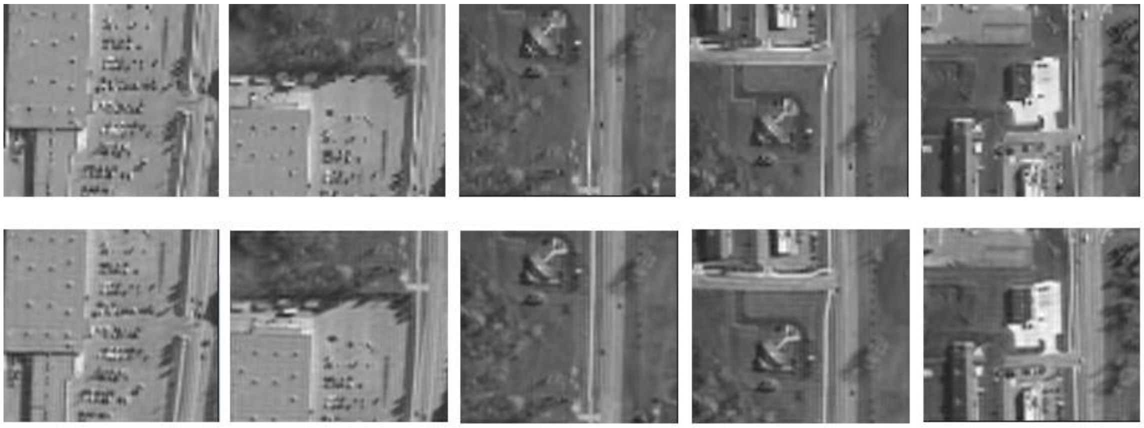


Figure 5.2: Simulated side information (the second row) prepared in the decoder end for the original video (the first row).

In this way, encoder is able to encode each frame without tucking intra-coded frames required for side information. Theoretically intra encoding is low-complexity but with higher data rate. Distributed coding allow statistically dependent sources, in our case video, to be intra-encoded with low data rate and inter-decoded without lose of reconstruction quality. This is all made possible with archived *side information* registered from archived video footages. In video surveillance, recorded sequence could be thousands of hours and covering every inch in the secured area. Video data are all statistically dependent in terms of spatial overlapping of scenes. The main reason we can exploit the statistical dependency from airborne videos is that the entire camera array is calibrated and geo-located. Onboard airborne cameras are all calibrated for offline video processing not only to stitch multiple camera views but also make it possible to store the geo-location of video frames. This camera calibration allows us to exploit the redundancy across video data to perform distributed video coding.

In the following, the side information is referred to the archived imagery and video footages that is statistically dependent with new video inputs. Videos from different mission may carry the same scenes despite image resolution or perspective change. With multiple unmanned airborne vehicles or other remote sensing imagery e.g. satellite image, the decoder can use side information collectively from other source to reconstruct video sequence. Hence, the redundancy can be efficiently removed in the encoder end.

5.3.2 Distributed Video Coding

The UAV onboard independent encoder is designed based on a LDPCA syndrome code. LDPCA code is a hard competition to Turbo code not only because the comparable performance but also its less complex encoder scheme. Similar to Turbo code, its decoder uses iterative belief propagation and log likelihood ratio, however, punctured turbo codec need to be selected carefully or it will easily lose the error resistance. An elaboration of the proposed compression paradigm will be given in the following sections.

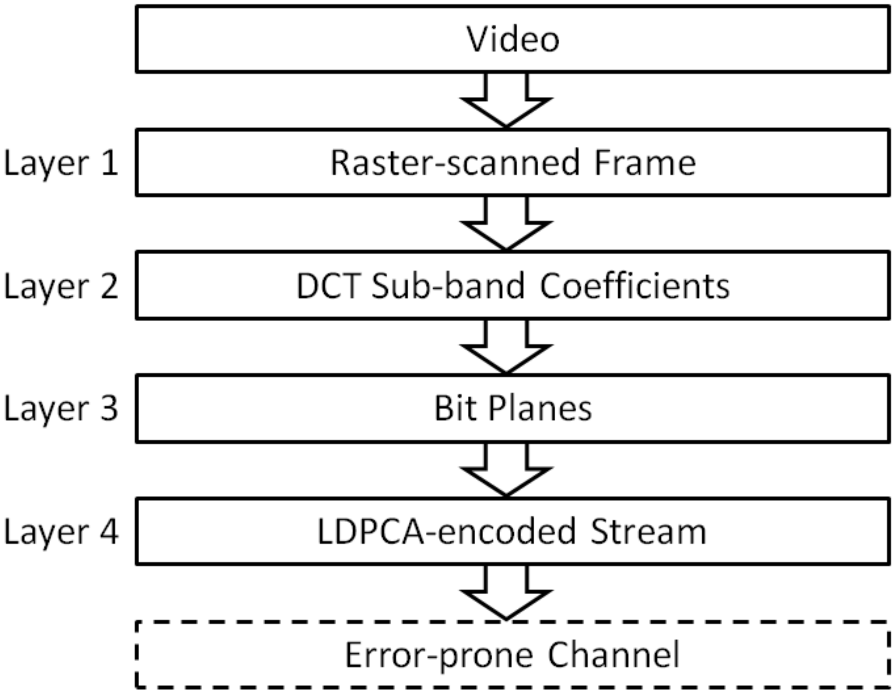


Figure 5.3: Encoding layers for distributed video coding.

A Transform-Domain Encoding

For each frame, DCT is applied to generate sub-band channels across all macro blocks of image, as layer 2 in Figure 5.3. The same sub-channel response is then grouped together to form a sub-band stream. For instance, applying a 4-by-4 DCT will end up with 16 sub-band streams for a single image. After interleaving of DCT channels, each sub-band is further broken in to bit planes and apply LDPCA code, layer 3 and 4 in Figure 5.3. Note that for a fixed DCT size it renders a fixed length of sub-channel sequence. For instance, applying a 4-by-4 DCT on a 20-by-20 image gives you a sub-channel stream of length 25. In selection of LDPCA code, it needs to be made sure that the generation matrix can cover the total length of sub-channel bit planes. It is obvious that the compression is low-complexity and no intensive motion field analysis required for this video transcoder.

B Feedback-Channel-Free Decoding

At the decoder, side-information is acquired via registration with the tagged geo-information in videos. It can be seen as original source image traveled through a noise-prone channel. Distributed video decoding requires dependency model describing this noise channel for massage-passing iterative decoding. Since only side information is available, the parametric model is built upon exploiting its partial statistics. To model the correlation between original message and the corrupted version (side information frame), most people use Laplace virtual noise channel model which is an error probability density function of a sequence. With the estimated error probability density function, the LDPCA are able to predict the log likelihood ratio for each bit in a bit plane layer of each DCT sub-band.

Usually the virtual noise channel model is a statistical dependency model based on the variance of each sub-band of side information. This work adopts a model proposed by Chien and Karam [76] while it requires conventional intra-coding to provide motion-compensated key frames. To be free from transmitting reference key frames, a geo-registered low resolution images based on videos' META data is provided as side information in this work. If we see the side information and the original sequence are two dependent sources independently coded, which is true because they are identical image with vary resolutions (and bias from registration), the dependency model can be built from the side information and project the real dependency between the original message and side information at the decoder. This gives us the advantage to leave the feedback channel behind and still efficiently decode the source with required parity/syndrome bits.

In order to describe the difference between the original sequence and its coarse version, i.e. side information, a quantized Laplace distribution of PDF is usually used. This work adopts the model used in BLAST-DVC [76],

$$P(D=d)=\begin{cases}\displaystyle\int_{254.5}^{\infty} 0.5\cdot\sigma\cdot exp(-\sigma|x|)dx, & if\ d\ =\ 255\\ \displaystyle\int_{d-0.5}^{d+0.5} 0.5\cdot\sigma\cdot exp(-\sigma|x|)dx, & if\ -255<d<255\\ \displaystyle\int_{-\infty}^{-254.5} 0.5\cdot\sigma\cdot exp(-\sigma|x|)dx, & if\ d\ =\ -255\end{cases}$$

(5.1),

or in another form given in [101],

$$P(D=d)=\begin{cases}1-e^{-\frac{1}{\sqrt{2}\sigma}}, & if\ d\ =\ 0\\ sinh(\dfrac{1}{\sqrt{2}\sigma})\cdot e^{-\frac{\sqrt{2}|d|}{\sigma}}, & if\ 1\leq|d|\leq 254\\ \dfrac{1}{2}\cdot e^{-254.5\cdot\frac{\sqrt{2}}{\sigma}}, & if\ |d|\ =255\end{cases}$$

(5.2)

, here σ is the variance of DCT sub-band from acquired side information at the decoder. In this work one modification is made on Chien and Karam's work to better predict the noise channel. The modification is to flatten the DC's PDF and make it uniform distributed because DC has huge variance according to experimental observation.

The work by Westerlaken et al. [79] gives a close estimation of bit plane conditional probability to compute log likelihood ratio based on (5.2):

$$pr(0) = P\left(Q^{(b)} = 0 \middle| Y = y, Q^{(b+1)}, \dots, Q^{(L-1)}\right) = \sum_{i=0}^{2^b-1} P_N(q(x_p, 2^{b+1}) \cdot 2^{b+1} + i - y) \tag{5.3}$$

$$pr(1) = P\left(Q^{(b)} = 1 \middle| Y = y, Q^{(b+1)}, \dots, Q^{(L-1)}\right) = \sum_{i=2^b}^{2^{b+1}-1} P_N(q(x_p, 2^{b+1}) \cdot 2^{b+1} + i - y) \tag{5.4}$$

with $Q^{(b)}$equals the b^{th} biplane value, Y the symbol value, Laplacian PDF $P_N(n)$ from (5.2), $q(x,y)$ equals the $log_2(y)^{\text{th}}$ bit plane value of x and $x_p = \sum_{i=b+1}^{L-1} Q^{(i)} \cdot 2^i$.

The LDPCA then utilizes $pr(1)/pr(0)$ as intrinsic log likelihood ratio to decode video sequence through belief propagation.

5.3.3 Rate Control

One major contribution of this work is a proposed DVC paradigm without feedback channel. A majority of hybrid video compression paradigm introduce feedback from decoder to request an increment on bit rate. Indeed, without feedback channel, LDPC may not be achieving the theoretical channel capacity limit. It is because of the nature of parity check bit which is introduced as redundancy in channel coding to correct error instead of carry information. Hence it is simple for decoder to report failure and require extra parity bits. This is not feasible for most video surveillance since decoding is not required in sync with encoding, especially for airborne surveillance.

To achieve sub-optimum performance, this work builds a rate-control function based on channel error probability model and the entropy of estimated channel-coded bits. The proposed compression scheme is also able to evaluate the higher bit rates performance by means of the quantization matrix in [80] for the transform domain Wyner-Ziv coding.

As shown in figure 5.3, to allocate proper parity-check/syndrome bits for each frame, each sub-band coefficients are decomposed into bit plane layers. Based on bit-plane layer decomposition, for each bit plane the conditional entropy is computed according to (5.3) and (5.4):

$$R_b^q = \rho \cdot \left(\frac{H_{x_q y_q}^b}{B_b \cdot L_N} \right) \tag{5.5}$$

where R_b^q is the number of bits assigned for the q^{th} bit plane of sub-band b, B_b is the number of bit plane assigned for the DCT band according to [80], L_N is the full syndrome length of the chosen LDPCA and $H^{b}_{x_q y_q}$ is the q^{th} plane-wise predicted conditional entropy of sub-band *b*. A rate-allowance factor ρ is set to 1.1 reflecting the strength of camera motion field. In practice, the rate control also make exception for the MSB and bit planes to have the full code length.

Note that there is a difference between this rate control function and source-entropy coding. We cannot allocate bits based on the randomness of bit stream directly because we have no knowledge of the side information on the decoder end. In other words, direct entropy coding cannot reflect the channel noise effect on side information and might not recover the original message due to lack of parity bits. On the other hand, (5.5) utilizes the prediction of channel error and works the bit allocation accordingly. Thus the significance of each bit plane with consideration of its channel response can be measured by its conditional entropy. Similar to entropy coder idea we share with source coding, the bit budget is then allocated to each bit plane proportionally.

Another contribution of this work is, instead of simulating the distributed video coding only with sub-frame rate disregarding the key frame, [77], video sequences are coded with DVC thoroughly. In our experimental results, videos are reconstructed with a rate-distortion behavior close to the state of art hybrid video coding.

5.4 Experimental Results

The encoding efficiency of distributed compression hinges on the quality of retrieved side information image from video footages. Side information prepared in this work is generated by the virtual motion video simulator used in chapter 2 to render the ground truth for camera motion-field analysis. To prepare experimental videos, first the original UAV video is generated with known flight trajectory and perspective angles. The coarse version of original video, i.e. side information, is registered from the same video coverage given a large-scale remote sensing image and known camera trajectory. For side information, however, video is down sampled with a vertical scaling factor of 2.5 and applied 2D smoothing afterward. Figure 5.4 give an example of the maximum absolute difference between one of experimental UAV video and its side information.

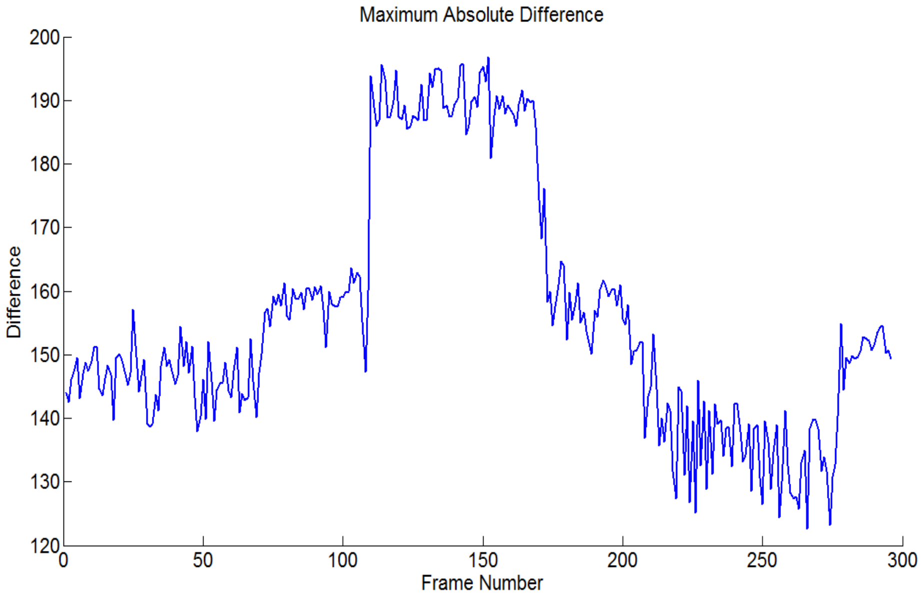


Figure 5.4: Pixel-level difference between simulated side information and original frames.

Figure 5.5 and Figure 5.6 demonstrate the estimation of DCT-domain bit-plane error PDF based on eq. (5.2). The data are centered at 1023 in the buffer because in our implementation it means to deal with the pixel depth up to 512 for fixed-point computation. It shows that DC band caries larger variance than AC bands according to the model and it was proven true that more bit planes should be dedicated to DC across the image.

In both figure 5.7 and figure 5.8, the top notches are made by distributed coding with LDPCA and feedback channels. Decoder with feedback channel make request of incremental parity bits until it achieve the optimum PSNR constrained by quantization. The bottom rate-distortion behavior demonstrates the DVC without feedback channel. Rate control in this case is also lack of intelligence that allowing each bit plane to have uniform quota budget which perform the worst. The reconstruction quality of videos with proposed DVC in this work is about 1dB off the optimum DVC with feedback channel. It is encouraging that through exploiting the partial statistics of side information we can compete with DCV with feedback channel enabled.

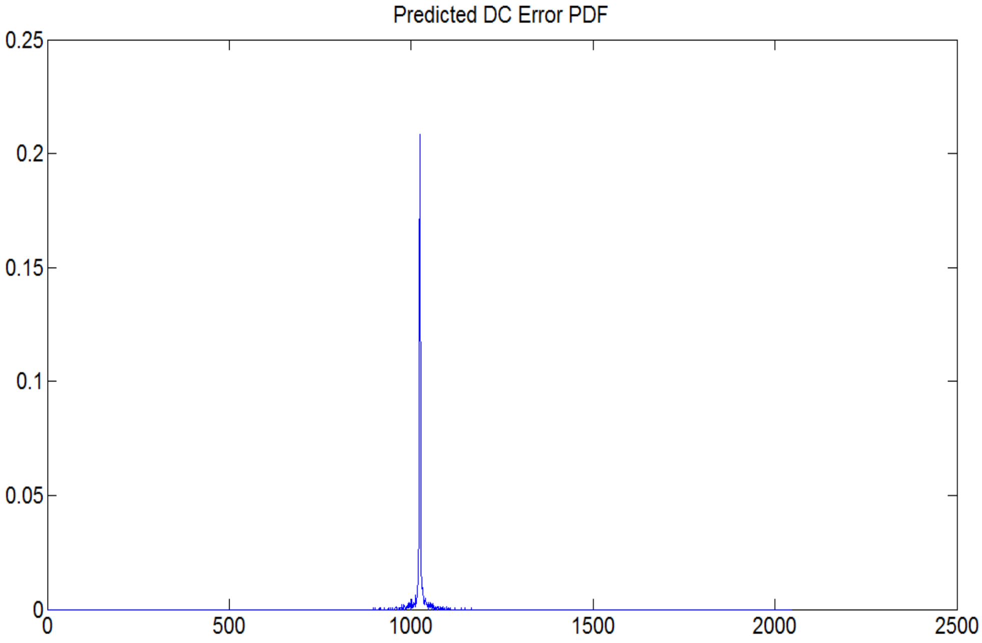


Figure 5.5: Practical DC-channel error PDF for fixed-point computation.

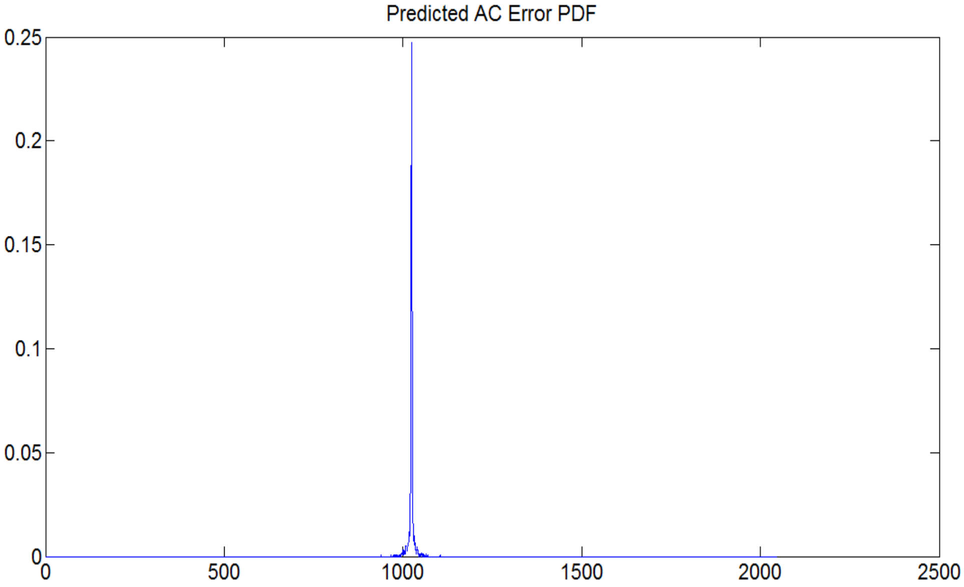


Figure 5.6: Practical AC-channel error PDF for fixed-point computation.

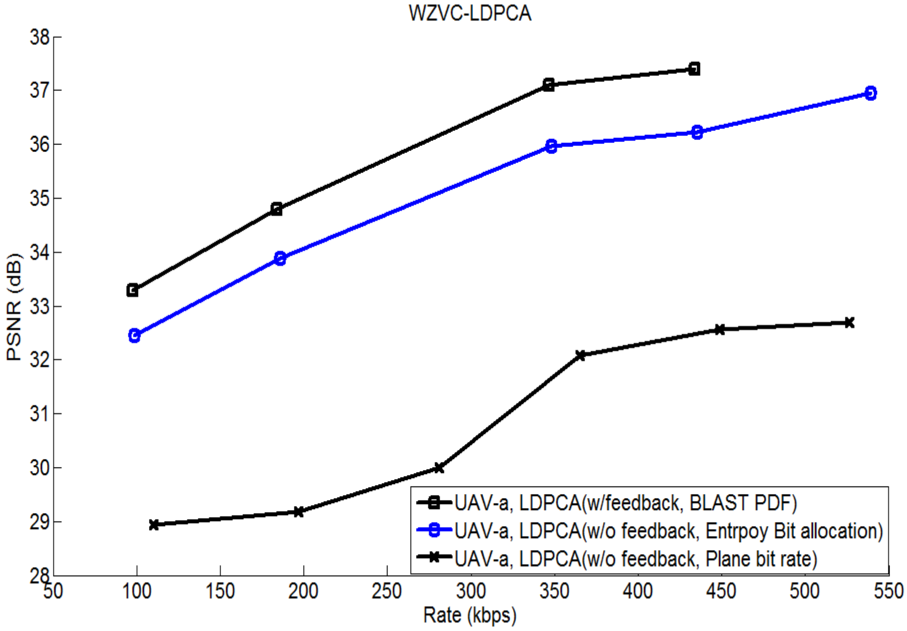


Figure 5.7: LDPCA rate-distortion performance of proposed distributed video coding on airborne video A.

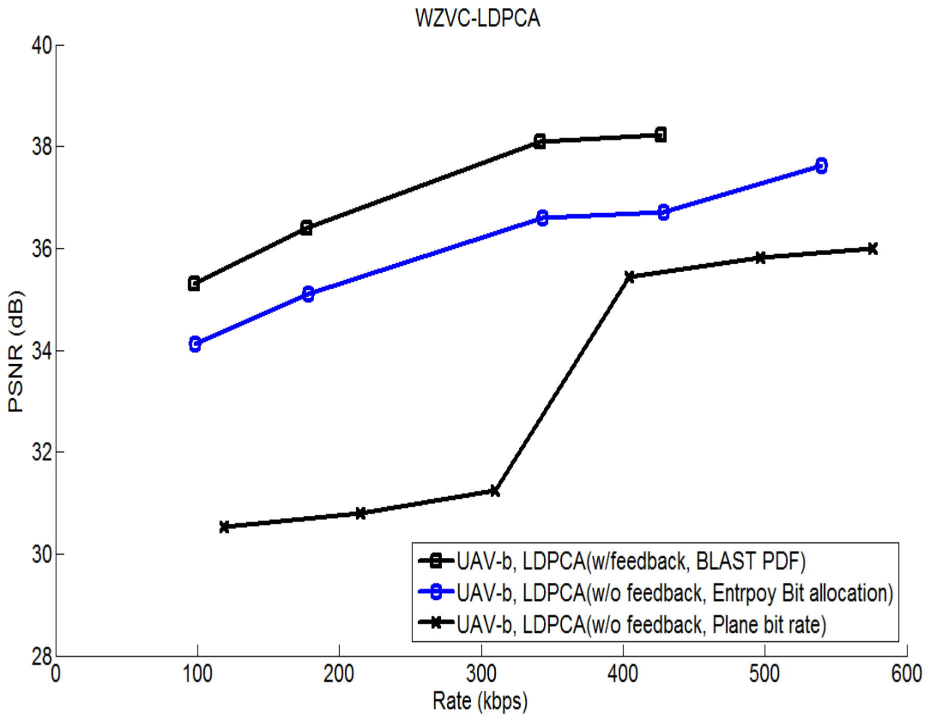


Figure 5.8: LDPCA rate-distortion performance of proposed distributed video coding on airborne video B.

5.5 Conclusion

In this work a distributed channel coding paradigm is proposed for mobile video surveillance network. It manages to have a low-complexity encoder and shift the bulk of computation to the decoder end. The main idea is form image geo-registration that makes side information available for decoder to share mutual information with encoder. The experimental results also show its potential to compete with the state-of-art hybrid DCV transcoder only this work requires no feedback channel. Another contribution from this collaborative coding is to investigate the possibility of full DVC scheme for video surveillance networks. Hence the total frame rate is objectively referred to Wyner-ziv coder without any bias.

Because of its better error-resistance in lower transmission rate to protect the visual quality, DVC can be used to cope with noise-prone channel in some extreme practice such as airborne surveillance. For challenges posted by video surveillance such as thermal noise and motion blur, distributed code can also minimized the abruptness of video quality. One foreseen improvement is to close the performance gap between two-way channel decoder and the feed-forward channel only decoder.

CHAPTER 6

CONCLUDING REMARKS AND FUTURE DIRECTIONS

In this dissertation, we have developed advanced video processing and computer vision methods for automated video processing and scene understanding in the following four areas: (1) Aerial video registration and moving object detection; (2) 3-D change detection from moving cameras; (3) Cross-view building matching and retrieval from aerial surveillance videos; (4) Collaborative video compression for UAV surveillance network. Computational intelligence is a promising tool for information analysis. Simply bringing all source information directly to human analysts is a cognitive disaster. There is an urgent need to develop advanced computational methods and tools for automated video processing and scene understanding. It is important to make decisions with refined, digested and unbiased information because the less but systematic access to information always makes more sense.

In our future work, to further extend our existing capabilities in aerial video surveillance and intelligent video processing, we shall conduct further research in the following areas:

1. Automated ground-aerial coordination for a swarm of ground unmanned vehicles and plane-style drones. The network shall allow autonomous agents to communicate with each other and adapt with environmental obstacles to retrieve more comprehensive information.

2. Real-time situational awareness with embedded target classifier for tactical intelligence. This application need to leverage with the advance of computational intelligence to cope with power-limited mobile agent system.

BIBLIOGRAPHY

[1] L.G. Brown, “A survey of image registration techniques,” *ACM Computing Surveys*, 24 (1992) 326376.

[2] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision,* Cambridge University Press, 2000.

[3] Robert S. Zitz, “Precision engagement relies on geospatial intelligence,” *Pathfinder*, pp. 7-10, October, 2003.

[4] R.P. Wildes, D.J. Hirvonen, S.C. Hsu, R. Kumar, W.B. Lehman, B. Matei, W.Y. Zhao, “Video Georegistration: Algorithm and Quantitative Evaluation,” in *Proc. ICCV 01*, vol. II, pp. 343–350, July 2001.

[5] Y. Su, M. -T. Sun, and V. Hsu, “Global motion estimation from coarsely sampled motion vector field and the applications,” *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 15, No. 2, pp. 232 - 242, Feb. 2005.

[6] *MPEG-4 Video Verification Model version 18.0*, ISO/IEC JTC1/SC29/WG11, 2001.

[7] F. Dufaux and J. Konrad, “Efficient, robust, and fast global motion estimation for video coding,” *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 497C501, Mar. 2000.

[8] Y. T. Tse and R. L. Baker, “Global zoom/pan estimation and compensation for video compression,” *Proceedings of ICASSP91,* Toronto, ON, Canada, May 1991, pp. 2725C2728.

[9] H. Jozawa, K. Kamikura, A. Sagata, H. Kotera, and H.Watanabe, “Two-stage motion compensation using adaptive global MC and local affine MC, *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 7, no. 2, pp. 75C85, Feb. 1997

[10] Y. Keller and A. Averbuch, “Fast gradient methods based on global motion estimation for video compression,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, No. 4, pp. 300 - 309, April 2003.

[11] T Tuytelaars and L Van Gool, “MatchingWidely Separated Views Based on Affine Invariant Regions,” *International Journal of Computer Vision*, 2003.

[12] M. Irani, P. Anandan, and S. Hsu, “Mosaic based representations of video sequences and their application,” *in Proc. Int. Conf. Computer Vision*, Boston, MA, June 1995, pp. 605-611.

[13] N. Grammalidis, D. Beletsiotis, and M. Strintzis, “Sprite generation and coding in multiview image sequences,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 302C311, Mar. 2000.

[14] Lowe, D. G., “Distinctive Image Features from Scale-Invariant Keypoints”, *International Journal of Computer Vision*, 60, 2, pp. 91-110, 2004. 29

[15] X. Chen, Z. He, J. Keller, D. Anderson, and M. Skubic, “Adaptive Silhouette Extraction in Dynamic Environments Using Fuzzy Logic”, *International conference on Fuzzy System*, Atlanta GA, Oct 2006.

[16] A. Elgammal, D. Harwood, and L.S. Davis, “Non-Parametric Model for Background Subtraction,” *Proc. IEEE Intl Conf. Computer Vision 99*, FRAME-RATE Workshop, 1999.

[17] R. Li, B. Zeng, andM. L. Liou, “A new three-step search algorithm for block motion estimation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 438442, Aug. 1994.

[18] S. Zhu and K.-K. Ma, “A new diamond search algorithm for fast block matching motion estimation,” in *Proc. Int. Conf. Inform., Commun., Signal Process.*, Singapore, Sept. 912, 1997, pp. 292296.

[19] Microsoft MPEG-4 Visual Reference Software, Version: Microsoft-FDAM-1-2.3-001213, Dec. 2000.

[20] S. Kumar and M. Hebert. “ Man-Made Structure Detection in Natural Images using a Causal Multiscale Random Field.”, *IEEE International Conference on Computer Vision and Pattern Recognition* (CVPR), 2003. [Ku03]

[21] D. Lowe. “ Distinctive image features from scale-invariant keypoints. ”, *IJCV*, 60(2):91–110, 2004.

[22] Y. Ke and R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. In *CVPR*, 2004.

[23] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE T. PAMI*, 27(10):1615–1630, 2005.

[24] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[25] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[26] T. Tuytelaars and L. Van Gool. Matching widely separated views based on afne invariant regions. *International Journal of Computer Vision*, 1(59):61–85, 2004.

[27] S. Lazebnik, C. Schmid, and J. Ponce. Sparse texture representation using afne-invariant neighborhoods. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Madison, Wisconsin, USA, pages 319–324, 2003.

[28] C. Schmid and R. Mohr. Local gray-value invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 19(5):530–534, 1997.

[29] Zhe Lin and In So Kweon, "Robust invariant features for object recognition, pose estimation and topological navigation," *International Journal on Human Friendly Welfare Robotic Systems (IJHWRS*), Volume 6, No. 1, Mar. 2005.

[30] Jiangjian Xiao, Hui Cheng, Feng Han, Sawhney, H. Geo-spatial aerial video processing for scene understanding and object tracking, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, June 2008.

[31] C. Lin and R. Nevatia. Building detection and description from a single intensity image. *Computer Vision and Image Understanding*, 72:101–121, 1998.

[32] S. Krishnamachari and R. Chellappa. Delineating buildings by grouping lines with mrfs. *IEEE Trans. on Pat. Anal. Mach. Intell.*, 5(1):164–168, 1996.

[33] H. Mayer. Automatic object extraction from aerial imagerya survey focusing on buildings. Computer Vision and Image Understanding, 74(2):138–149, 1999.

[34] S. Sarkar and P. Soundararajan. Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata. IEEE Trans. on Pat. Anal. Mach. Intell., 22(5):504–525, 2000.

[35] A. Vailaya, A. K. Jain, and H. J. Zhang. On image classification: City images vs. landscapes. Pattern Recognition, 31:1921–1936, 1998.

[36] X. Feng, C. K. I. Williams, and S. N. Felderhof. Combining belief networks and neural networks for scene segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence, 24(4):467–483, 2002.

[37] S. Konishi and A. L. Yuille. Statistical cues for domain specific image segmentation with performance analysis. In Proc. IEEE Int. Conf. CVPR, pages 125–132, 2000.

[38] J. Matas, O. Chum, U. M., T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In BMVC, 2002.

[39] Per-Erik Forssén and David G. Lowe, "Shape descriptors for maximally stable extremal regions," International Conference on Computer Vision (ICCV), Rio de Janeiro, Brazil, October 2007.

[40] S. Umeyama, “An eigen-decomposition approach to weighted graph matching problems”, IEEE PAMI, 10, pp. 695–703, 1988.

[41] G.L. Scott and H.C. Longuet-Higgins, “An algorithm for associating the features of 2 images”, Proceedings of the Royal Society of London Series B (Biol.), 244, pp. 21–26, 1991.

[42] L.S. Shapiro and J.M. Brady, “Feature-based correspondence- an eigenvector approach”, Image and Vision Computing, 10, pp. 283–288, 1992.

[43] S. Sclaroff and P. Pentland, Modal matching for correspondence and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 17(6), 1995, 545–561.

[44] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, 2005.

[45] Van Gool, L. , Proesmans, M. and Zisserman, A. Planar homologies as a basis for grouping and recognition Image and Vision Computing (1998).

[46] H. Shao, T. Svoboda, and L. Van Gool. ZUBUD-Zurich buildings database for image based recognition. Technical report No. 260, Swiss Federal Institute of Technology, 2003.

[47] Wei Zhang and Jana Kosecka. Localization Based on Building Recognition.IEEE Workshop on Computer Vision Applications for the Visually Impaired, in conjunction with CVPR, 2005.

[48] T. Pollard and J. L. Mundy, Change Detection in a 3-d World, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-6, June 2007.

[49] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, pages 246–252, 1999.

[50] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *IEEE Trans on Image Processing*, Vol. 14, No. 3, 2005.

[51] Ashutosh Saxena, Min Sun, Andrew Y. Ng., 3-D Reconstruction from Sparse Views using Monocular Vision, In ICCV workshop on Virtual Representations and Modeling of Large-scale environments (VRML), 2007.

[52] Learning 3-D Scene Structure from a Single Still Image, Ashutosh Saxena, Min Sun, Andrew Y. Ng. In ICCV workshop on 3D Representation for Recognition (3dRR-07), 2007.

[53] Make3D: Learning 3D Scene Structure from a Single Still Image, Ashutosh Saxena, Min Sun, Andrew Y. Ng. IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI), vol. 30, no. 5, pp 824-840, 2009.

[54] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV, 47, 2002.

[55] D. A. Forsyth and J. Ponce. Computer Vision : A Modern Approach. Prentice Hall, 2003.

[56] R. Zhang, P. Tsai, J. Cryer, and M. Shah. Shape from shading: A survey. IEEE PAMI, 21:690–706, 1999.

[57] Myronenko A., Song X.(2009): "Image Registration by Minimization of Residual Complexity.", Computer Vision and Pattern Recognition, (CVPR'09), pp. 49-56.

[58] D. L. G. Hill, P. G. Batchelor, M. Holden, and D. J. Hawkes. Medical image registration. Physics in medicine and biology, 46(3):R1–R45, Mar. 2001.

[59] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. IJCV, 60(2), 2004.

[60] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In CVPR, 2005.

[61] D. Nist´er. An efficient solution to the five-point relative pose problem. IEEE Trans. Pattern Anal. Mach. Intell., 26(6):756–770, 2004.

[62] T. Tuytelaars and L. Van Gool. Matching widely separated views based on afne invariant regions. International Journal of Computer Vision, 1(59):61–85, 2004.

[63] Y. Ke and R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. In CVPR, 2004.

[64] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(4):509–522, 2002.

[65] R. P. Wiles, D. J. Hirvonen, S. C.Hsu, R. Kumar, W. B. Lehman, B. Matei, and W.-Y. Zhao, Video georegistration: algorithm and quantitative evaluation, IEEE International Conference on Computer Vision (ICCV), vol. 2, pp. 343-350, July 2001.

[66] D. Comaniciu, V. Ramesh, and P. Meer. The variable bandwidth mean shift and data-driven scale selection. In Proc. ICCV, 2001.

[67] Quick Shift and Kernel Methods for Mode Seeking A. Vedaldi and S. Soatto. in *Proceedings of the European Conference on Computer Vision*, 2008.

[68] M. N. Do and M. Vetterli, The finite ridgelet transform for image representation, *IEEE Transactions on Image Processing*, vol. 12, pp. 16-28, Jan. 2003.

[69] H. Tahani, J. M. Keller. “Information Fusion in Computer Vision Using the Fuzzy Integral,” *IEEE Trans. On Systems, Man and Cybernetics*, Vol. 20, No. 3, May-June 1990, pp. 733 – 741.

[70] Radke, R.J.; Andra, S.; Al-Kofahi, O.; Roysam, B.; Image change detection algorithms: a systematic survey *IEEE Transactions on Image Processing*, Volume 14, Issue 3, March 2005 Page(s):294 - 307.

[71] Bovolo, F.; Bruzzone, L.; Marconcini, M.; A Novel Approach to Unsupervised Change Detection Based on a Semisupervised SVM and a Similarity Measure, *IEEE Transactions on Geoscience and Remote Sensing*, Volume 46, Issue 7, July 2008 Page(s):2070 - 2082.

[72] Chang Yuan; Medioni, G.; Jinman Kang; Cohen, I.; Detecting Motion Regions in the Presence of a Strong Parallax from a Moving Camera by Multiview Geometric Constraints, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 29, Issue 9, Sept. 2007 Page(s):1627 – 1641.

[73] Singh, M., Parameswaran, V., and Ramesh, V., Order consistent change detection via fast statistical significance testing, IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008. June 2008.

[74] A. Wyner and J.Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory,* vol. IT-22, no. 1, Jan. 1976.

[75] Varodayan, D., Aaron, A., Girod, B.: “Rate-Apdaptive Distributed Source Coding Using Low-Density Parity-Check codes”, In: Proc. Asilomar Conference on Signals and Systems, Pacific Grove, CA (2005)

[76] CHIEN W.-J., KARAM L.J.: “Bitplane selective distributed video coding”. Proc. IEEE Asilomar Conf. on Signals, Systems, and Computers, October 2008

[77] A. Aaron, S. Rane, R. Zhang, B. Girod, “Wyner–Ziv coding for video: applications to compression and error resilience”, in: Proc. of IEEE Data Compression Conf. (DCC), Snowbird, UT, March 2003, pp. 93–102.

[78] E. Peixoto and R. L. Queiroz and D. Mukherjee,, “A Wyner-Ziv Video Transcoder”, *to appear, IEEE Trans. Circuits and Systems for Video Technology*, 2009.

[79] R. P.Westerlaken, S. Borchert, R. K. Gunnewiek, and R. L. Lagendijk. Analyzing symbol and bit plane-based LDPC in distributed video coding. In *Proc. Int’l Conf. Image Processing (ICIP)*, 2007.

[80] C. Brites, J. Ascenso, and F. Pereira, "Improving Transform Domain Wyner-Ziv Video Coding Performance", IEEE International Conf on Acoustics, Speech and Signal Processing, Toulouse, France, May 2006.

[81] R Szeliski, ”Computer Vision: Algorithms and Applications”, Research.Microsoft.Com.

[82] Murphy-Chutorian, E. Trivedi, M.M., “Pattern Analysis and Machine Intelligence, IEEE Transactions on”, Publication Date: April 2009 Volume: 31, Issue: 4 On page(s): 607-626.

[83] Zhigang Zhu, Edward M. Riseman, Allen R. Hanson and Howard Schultz, “An efficient method for geo-referenced video mosaicing for environmental monitoring,”*Machine Vision and Applications*, Volume 16, Number 4 / September, 2005. Page(s) 203-216

[84] Shastry, A.C. and Schowengerdt, R.A, “Airborne video registration and traffic-flow parameter estimation,” *Intelligent Transportation Systems, IEEE Transactions on*, vol 6, issue 4, Page(s):391 - 405, Dec. 2005.

[85] Zoltn Szlvik, Tams Szirnyi and Lszl Havasi, “Video camera registration using accumulated co-motion maps,” *ISPRS Journal of Photogrammetry and Remote Sensing*, Volume 61, Issue 5, January 2007, Pages 298-306.

[86] A. Criminisi and A. Zisserman, “A Plane Measuring Device” *Image and Vision Computing*, , 1999a On page(s): 625 - 634 vol. 17(8).

[87] M. J. Rantz, R. T. Porter, D. Cheshier, D. Otto, C. H. Survey, III, R. A. Johnson, M. Skubic, H. Tyrer, Z. He, G. Demiris, J. Lee, G. L. Alexander, and G. Taylor, “TigerPlace, a state-academic-private project to revolutionize traditional long term care,” *J. Housing for the Elderly*, Oct. 2007.

[88] Zhongna Zhou Xi Chen Yu-Chia Chung Zhihai He Han, T.X. Keller, J.M. , “Activity Analysis, Summarization, and Visualization for Indoor Human Activity Monitoring”, *Circuits and Systems for Video Technology, IEEE Transactions on*, Volume: 18, Issue: 11, page(s): 1489-1498, Nov. 2008.

[89] A. Rajashekhar, Subhasis Chaudhuria, and Vinay P. Namboodiri. “Retrieval of images of man-made structures based on projective invariance.”, *Pattern Recognition*, Vol. 40, Pages 296-308, Issue 1, January 2007.

[90] M. Carcassoni and E. R. Hancock. “Point pattern matching with robust spectral correspondence”. IEEE Conference on Computer Vision and Pattern Recognition, Volume: 1, On page(s): 649-655, June 2000.

[91] P. Felzenszwalb and D. Huttenlocher. “Efficient graph-based image segmentation.”, IJCV, 59, 2004.

[92] G. Zheng and X. Zhang. “A unifying MAP-MRF framework for deriving new point similarity measures for intensity-based 2D-3D registration.” In ICPR, volume 2, pages 1181–1185, 2006.

[93] Christoph H. Lampert, Hannes Nickisch, Stefan Harmeling: "Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer", IEEE Computer Vision and Pattern Recognition (CVPR), Miami, FL, 2009.

[94] R. D. Eastman, J. Le Moigne, and N. S. Netanyahu, “Research issues in image registration for remote sensing”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007, June 2007.

[95] D. Comaniciu and P. Meer. “Mean shift: A robust approach toward feature space analysis.” IEEE Trans. Pattern Anal. Mach. Intell, 24(5), 2002.

[96] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys. “3D Model Matching with View-point Invariant Patches (VIPs) .” In Proc. CVPR, 2008.

[97] M. Pollefeys et al. “Detailed real-time urban 3d reconstruction from video.” IJCV, 78(2-3), 2008.

[98] K. Mikolajczyk and C. Schmid. “A performance evaluation of local descriptors.” IEEE T. PAMI, 27(10):1615–1630, 2005.

[099] R. Lidl and H. Niederreiter, “Introduction to Finite Fields and Their Applications”, *Cambridge University Press*, revised edition, 1994.

[100] D. Slepian and J.K. Wolf: “Noiseless coding of correlated information sources,” *IEEE Trans. on Inform. Theory*, vol. IT-19, pp. 471–480, July 1973.

[101] Morbee M, Prades-Nebot J, Pizurica A, Philips W : "Rate allocation algorithm for pixel-domain distributed video coding without feedback Channel", in Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, vol. 1, pp: I-521–I-524, April 2007.

[102] M. R. Lyu, E. Yau, and K. S. Sze, “iview: An intelligent video over internet and wireless access system,” in *Proc. 11th Int. World Wide Web Conf. (WWW2002), Practice and Experience Track*, Honolulu, HI, 2002.

[103] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain,“Content-based image retrieval at the end of the early years,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 12, pp. 1349–1380, 2000.

[104] Giompapa, S.; Gini, F.; Farina, A.; Graziano, A.; Croci, R.; Distefano, R., “Maritime border control multisensor system”, *Aerospace and Electronic Systems Magazine*, IEEE Vo-

lume: 24 , Issue: 8 Digital Object Identifier: 10.1109/MAES.2009.5256382 Publication Year: 2009 , Page(s): 9 - 15

[105] Shastry, A.C.; Schowengerdt, R.A, “Airborne video registration and traffic-flow parameter estimation”, *Intelligent Transportation Systems, IEEE Transactions on*, Volume: 6 , Issue: 4 Digital Object Identifier: 10.1109/TITS.2005.858621 Publication Year: 2005 , Page(s): 391 – 405.

[106] Barnes, L.; Garcia, R.; Fields, M.; Valavanis, K., “Swarm formation control utilizing ground and aerial unmanned systems”, *Intelligent Robots and Systems, IROS 2008. IEEE/RSJ International Conference on*, Issue Date : 22-26 Sept. 2008 On page(s): 4205 – 4205.

[108] Hoi, S.C.H.; Lyu, M.R. “A Multimodal and Multilevel Ranking Scheme for Large-Scale Video Retrieval”,; *Multimedia, IEEE Transactions on* Volume: 10 , Issue: 4 Digital Object Identifier: 10.1109/TMM.2008.921735 Publication Year: 2008 , Page(s): 607 - 619.

VITA

Yu-Chia Chung was born in Taipei, Taiwan in 1979. He received his B.S. degree in electrical engineering from Yuan Ze University, Taiwan in 2000. From 2001 to 2003 he was in fulfillment of military obligation and was admitted into the program of electrical and computer engineering for graduate study at the University of Missouri in Spring 2004.

Mr. Chung worked with the Video Processing and Communication Lab and the Center for Geospatial Intelligence in the University of Missouri from 2004 and become a Ph.D candidate in year 2009. In 2010, he also did an internship with DCM Research Resources LLC where he practiced computer vision tool for video processing.

His research interests include computer vision, communication system, pattern recognition, video processing and compression.