Correspondence Matching Algorithm Based on Mutual Information Similarity for Multi-View Video Sequences

Ph.D. Dissertation

Lee, Soon-Young

Department of Electrical Engineering and Computer Science

Seoul National University

ABSTRACT

The multi-view video sequences are essentially used for many computer vision applications such as surveillance system. For these applications, the correspondence matching that identifies the corresponding positions of one view to another is essentially required. The correspondence matching has been fundamentally researched for a long time, however, it is still challenging for multi-view video sequences. In this dissertation, the correspondence matching algorithm and its applications for the multi-view video sequences are presented.

First, an accurate and robust similarity measure for the correspondence matching of multi-view video sequences captured by arbitrarily positioned cameras is proposed. We use an activity vector, which represents the temporal occurrence pattern of moving foreground objects at a pixel position, as an invariant feature for correspondence matching. Activity vectors are derived from a moving object detection algorithm, so it is robust to illumination changes and additive noises. Then, we devise a novel similarity measure between two activity vectors by considering the joint and individual behavior of the activity vectors. Specifically, we define random variables associated with the activity vectors and represent the similarity between them using the mutual information based similarity (MIBS) measure. Because the MIBS measure adaptively explains the behaviors between two activity vectors, it

outperforms other conventional similarity measures of binary vectors especially for a correspondence matching problem.

Then, the framework for finding correspondence matching between two multi-view surveillance sequences is proposed. In order to achieve a more accurate and robust inter-view homography, three practical techniques are utilized. The first technique is the adaptive activity area refinement which represents actual ground regions touched by foreground objects moving on the ground plane. It reduces the discrepancy between objects areas and actual ground surfaces, so that the activity vectors can effectively feature geometry surfaces in the scenes. In addition, we propose the consistent pixel positions on which the MIBS measure is reliably evaluated. At consistent pixel positions, the maximum MIBS criterion is satisfied backward and forward, therefore, we can yield more accurate correspondence matchings. Finally, the correspondence at multiple pixel positions are determined by minimizing a matching cost function associated with the MIBS measure and structure preservation terms.

The proposed correspondence matching algorithm is robust to various positions of cameras and illumination/color differences between cameras. Moreover, the proposed MIBS measure reliably represents the similarity of two binary vectors even under the additive noises. Therefore, the results of proposed algorithm demonstrate the correspondences between two different views are more accurately and reliably estimated than the conventional state-of-the-art algorithm with a relatively small computational complexity. These results indicate that the proposed algorithm is a very promising technique for various multi-view video applications for a visual surveillance such as homograpy estimation and panoramic view synthesis.

keywords: multi-view video, correspondence matching, mutual information

student number: 2007-30235

Contents

List of Figures

List of Tables

Chapter 1

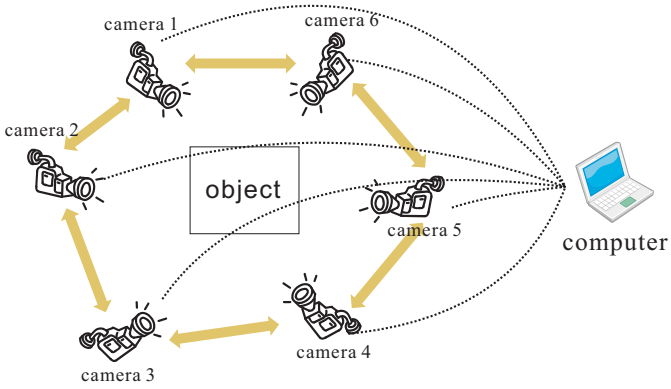
Introduction

1.1 Background and Research Issues

1.1.1 Multi-view Video Sequences

Nowadays, high definite broadcast and blu-ray disc, which are formatted by H.264 standard, has provided a great visual satisfaction. Beyond H.264, the advent of a new generation technology for the 3-D visual data enables us to experience a virtual reality, such as multi-view video and hologram. Especially, the multi-view video has been researched as a practical approach for advanced visual experiences [3–5], because it is generated by a series of conventional visual cameras with a control unit, which is easily applicable for a various circumstances as described in Fig. 1.1.

The multi-view video provides a number of fields-of-view from the spatially apart cameras to the users. Basically, the users can select one of view among the multiple views and interactively change viewing scenes. Furthermore, the discrepancies of fields-of-view between multiple cameras provide perspective and spaciousness, which play a key role to 3-D realities. Therefore, the multi-view video is exploited for many



(a)

(b)

Figure 1.1: Multi-view sequence acquisition: (a) multiple cameras. (b) Spatio-temporal frames of multi-view sequences.

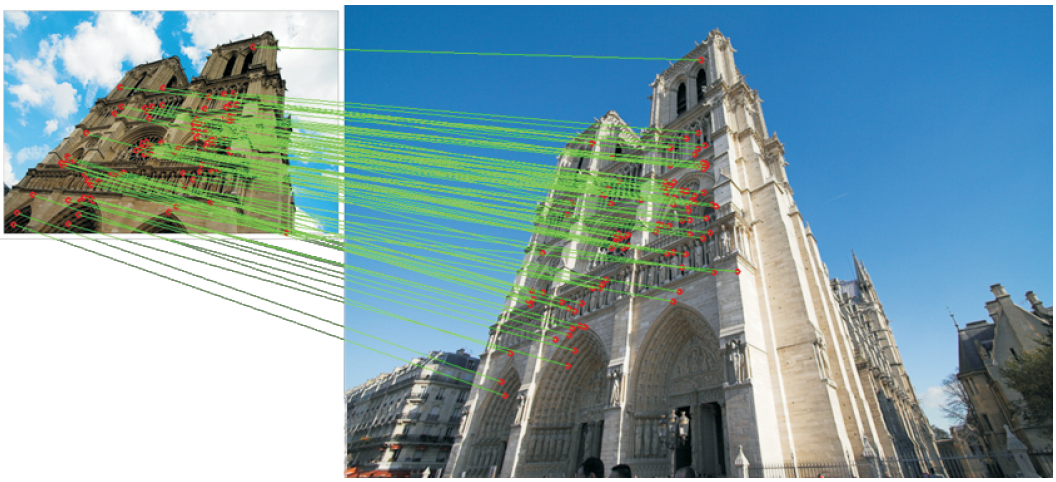


Figure 1.2: Example of the correspondence problem between two images presented in [1].

applications such as 3-D TV, 3-D reconstruction and medical imaging.

1.1.2 Correspondence Problem

The correspondence problem of the multi-view sequences is finding the corresponding position of one view to which position of another view. It facilitates many applications of visual sensor networks, such as surveillance, environmental monitoring, and panoramic view synthesis [6–8]. However, multi-view video sequences are often captured under varying illumination and lighting conditions, and multiple cameras may have different and unknown parameters, *e.g.*, positions, orientations, and zooming factors. Therefore, it is a challenging issue to determine the true correspondence matching among multiple views, and a lot of attempts have been made to develop robust correspondence matching algorithms [1, 2, 9–18].

Traditional stereo matching techniques provide a pixel-wise dense correspondence map between two images [9, 10]. In general, a window is assigned to each pixel, and

the distance between two pixels is defined as the sum of differences (SAD) of pixel intensities between the corresponding windows. Then, the matching pixel is determined that yields the smallest SAD. These stereo matching techniques are therefore sensitive to radiometric variations between images. Several advanced stereo matching algorithms have been proposed to alleviate the effect of radiometric variations, but they still suffer from the uncertainty of camera parameters [11, 12]. Another approach to the correspondence matching problem is to employ feature detection techniques, such as scale invariant feature transform (SIFT) [1] or speeded up robust features (SURF) [13]. Feature-based techniques find the correspondence only for a selected set of feature points, whereas stereo matching techniques provide dense correspondence maps over entire images. Feature-based techniques are more robust to radiometric variations and can reduce the computational complexity to find the correspondence matching. However, they also fail to work with severely different viewing positions of cameras and additionally require camera calibration techniques [19,20]. Therefore, these stereo matching or feature-based techniques are less efficient for finding the correspondences among multiple views in visual sensor network applications, where multiple cameras have quite different positions, orientations, exposure and lighting conditions.

In order to find the correspondence matching of two video sequences, Sand *et al.* independently applied an image matching method to each pair of frames [14]. This algorithm inherently suffers from the drawbacks of the correspondence matching of still images. On the other hand, several algorithms have been proposed to exploit temporal information for multi-view video matching [15–18]. The centroids of moving objects are computed over video sequences, and used as feature points to estimate the homography between two views [15–17]. However, they do not provide

sufficiently reliable matching performance, since they ignore the temporal orders of the centroids. In contrast, Caspi *et al.* employed a motion trajectory, which represents temporal locations of a moving object in order [18]. They estimated the homography by matching the trajectories of the same object observed in two views. However, these algorithms require an accurate result of object tracking, which is hard to be obtained from general video sequences containing many moving objects. As focusing on time footage of the moving objects, Ermis *et al.* proposed a correspondence matching algorithm for video sequences based on activity features of moving foreground objects [2].

1.2 Outline of the Dissertation

In this dissertation, an accurate and robust correspondence matching algorithm for multi-view video sequences captured by arbitrarily positioned cameras is proposed. In additional, we propose an inpainting technique based on the exemplar-based approach for multi-view video sequences.

Chapter 2 reviews the preliminaries about binary similarity measures and mutual information. Then, Chapter 3 proposes a correspondence matching algorithm using the multi-view video sequences. We use an activity vector, which represents the temporal occurrence pattern of moving foreground objects at a pixel position, as an invariant feature for correspondence matching. We first devise a novel similarity measure between activity vectors by considering the joint and individual behavior of the activity vectors. Specifically, we define random variables associated with the activity vectors and measure their similarity using the mutual information between the random variables. Unlike conventional Hamming distance measure, proposed

mutual information based similarity (MIBS) measure can adaptively reflect the discrepancy of different contribution derived from the counts of binary combination between two activity vectors. Therefore, the results by using the proposed MIBS measure shows accurate similarity than those by using conventional measures.

In Chapter 4, the system of finding a reliable correspondence matching between two multi-view video sequences are presented. To achieve a reliable homography transform between views, we find consistent pixel positions by employing the iterative bidirectional matching. We also refine the matching results of multiple source pixel positions by minimizing a matching cost function based on the Markov random field. Experimental results show that the proposed algorithm provides more accurate and reliable matching performance than the conventional activity-based matching algorithm, and therefore can facilitate various applications of visual sensor networks. And finally, results of panoramic view synthesis based on the proposed correspondence matching are proposed. In Chapter 4, we finally conclude this dissertation and present the limitations of our works and future works.

Chapter 2

Preliminaries

2.1 Binary Similarity Measures

Measuring the similarity or distance between two vectors are fundamental issues in many fields, such as engineering, biology, and statistics [21]. Especially, the pattern matching of unlabeled data is essentially required in most of computer vision problems. Since the binary vector which consist of 0 and 1 is widely used for pattern analysis problems such as clustering, correspondence matching and classification, many efforts have been taken to devise meaningful measure between binary vectors [22].

A binary vector $\mathbf{x}$ with T dimension is defined as

$$\mathbf{x} = (x_0, x_1, \cdots, x_{T-1}), \tag{2.1}$$

where $x_t \in \{0, 1\}$. And the similarity measure S is defined as the function that maps two binary vectors into a non-negative real number, that is

$$S : (\mathbf{x}, \mathbf{y}) \rightarrow \{0\} \cup \mathbb{R}^+, \tag{2.2}$$

where $\mathbf{x}$ and $\mathbf{y}$ are binary vectors and $\mathbb{R}^+$ denotes a set of positive real numbers.

To measure the similarity between two binary vector $\mathbf{x}$ and $\mathbf{y}$ in various approaches, the number of element-wise combinations are required. First, the occurrence count is defined as

$$\delta_t(i,j) = \begin{cases} 1 & \text{if } x_t = i \text{ and } y_t = j \\ 0 & \text{otherwise} \end{cases}, \tag{2.3}$$

where $i, j \in \{0, 1\}$. And the number of element-wise combination with respect to $\mathbf{x}$ and $\mathbf{y}$ is defined as follows.

$$K_{ij}(\mathbf{x}, \mathbf{y}) = \sum_{t=0}^{T-1} \delta_t(i,j). \tag{2.4}$$

For simple notation, we omit the argument $(\mathbf{x}, \mathbf{y})$. K_{ij} is also expressed by the data matrix of operational taxonomic units which is a 2×2 contingency table representing all combinations of K_{ij} as shown in Table 2.1. It is noteworthy that K_{11} is the number of features where the elements of $\mathbf{x}$ and $\mathbf{y}$ are both 1, K_{01} is the number of feature where the elements of $\mathbf{x}$ and $\mathbf{y}$ is $(0, 1)$, K_{10} is for $(1, 0)$ and K_{00} is for $(0, 0)$. The diagonal sum $K_{11} + K_{00}$ means the total number of matches between $\mathbf{x}$ and $\mathbf{y}$, while the other diagonal sum $K_{01} + K_{10}$ represents the total number of mismatches. Finally, the total sum $K_{11} + K_{01} + K_{10} + K_{00}$ is obviously equals to T.

Among a number of similarity measures for binary vectors, Tubbs especially have summarized various measures that are widely used in the pattern recognition area [23]. And Zhang *et al.* has intensively compared the performance of eight similarity measures in the application on a handwriting recognition [24]. Cha *et al.* proposed weighted binary measurement to improve classification performance based on the comparative study [22]. Table 2.2 lists eight binary measures widely used for binary vectors. These similarity measures are classified into two categories, one of

Table 2.1: The number of element-wise combinations of $\mathbf{x}$ and $\mathbf{y}$

$\mathbf{y}$ \ $\mathbf{x}$	1 (active)	0(inactive)
1(active)	K_{11}	K_{01}
0(inactive)	K_{10}	K_{00}

which is non-correlation based similarity measure and the other is correlation based similarity measure.

2.1.1 Non-correlation based similarity measures

A natural and intuitively appealing approach to measuring similarity between two binary vectors is to count the number of relevant matches as shown in the contingency table in 2.1. From the combination of the numbers of relevant matches, there are various similarity measures to interpret the matches and mismatches. The non-correlation based similarity measures are consist of the ratio between matches and mismatches with linear combinations, which are simply computed from K_{ij}.

For the non-correlation based similarity measure, two factors can account for all the variations [25]. The first factor is the number of 0-0 matches K_{00} which increases the similarity even when two sparse binary vectors are not related. It would be misleading to allow these 0-0 matches to contribute to the measure of association between two totally different vectors. The second factor is the weight of matches and mismatches. As mentioned above, K_{11} and K_{00} represent the number of matches, while K_{10} and K_{01} denote the number of mismatches. Depend on the weight coefficients to the matches and mismatches, the similarity measure can yields various

Table 2.2: The famous similarity measures for binary vectors

Similarity measure	Definition
Rogers-Tanimoto	$\frac{K_{11}+K_{00}}{K_{11}+K_{00}+2(K_{10}+K_{01})}$
Jaccard-Needham	$\frac{K_{11}}{K_{11}+K_{10}+K_{01}}$
Dice	$\frac{2K_{11}}{2K_{11}+K_{10}+K_{01}}$
Sokal-Michener	$\frac{K_{11}+K_{00}}{T}$
Russel-Rao	$\frac{K_{11}}{T}$
Kulczynski	$\frac{K_{11}}{K_{10}+K_{01}}$
Pearson	$\frac{K_{11}K_{00}-K_{10}K_{01}}{\{(K_{10}+K_{11})(K_{01}+K_{00})(K_{11}+K_{01})(K_{00}+K_{10})\}^{1/2}}$
Yule	$\frac{K_{11}K_{00}-K_{10}K_{01}}{K_{11}K_{00}+K_{10}K_{01}}$

results. Table 2.3 shows a summary of the famous similarity measures according to the existence of 0-0 matches and the coefficient weights. Nine combinations are described in Table 2.3 except the combinations which are apparently worthless. Note that Table 2.3 includes three unnamed combinations which provides the measures of similarity, but they have been not widely used.

Jaccard-Needham and Dice similarity measures

$$S_J = \frac{K_{11}}{K_{11} + K_{10} + K_{01}}. \tag{2.5}$$

Historically, Jaccard similarity S_J have been used in ecology fields [26]. It is clear that $S_J \to 0$ as $K_{11}/(K_{01} + K_{10}) \to 0$, and that as $(K_{01} + K_{10}) \to 0$, then $S_J \to 1$. This S_J does not consider a negative matching K_{00}, so it emphasizes the number of 1-1 matches K_{11}. As a related similarity measure, Dice similarity S_D have been devised as follows.

$$S_D = \frac{2K_{11}}{2K_{11} + K_{10} + K_{01}}. \tag{2.6}$$

It is monotonic with S_J but gives more weight to K_{11} than to mismatches. Both S_J and S_D vary from 0 to 1.

Sokal-Michener and Rosers-Tanimoto similarity measures

$$S_{SM} = \frac{K_{11} + K_{00}}{T}. \tag{2.7}$$

This is one of the oldest and simplest similarity measure for the binary vectors. From the formulation, it follows that $S_{SM} \to 0$ as $(K_{00} + K11)/(K_{01} + K_{10}) \to 0$. In its complementary form, $1 - S_{SM}$, it equals to the squared Euclidean distance, that

Table 2.3: Non-correlation based similarity measures

Weighting of matches, mismatches	0-0 matches in denominator	0-0 matches in numerator	
		Included	Excluded
Equal weights	Included	Sokal-Michener $\frac{K_{11}+K_{00}}{T}$	Russel-Rao $\frac{K_{11}}{T}$
	Excluded		Jaccard-Needham $\frac{K_{11}}{K_{11}+K_{10}+K_{01}}$
Double weight for matched pairs	Included	Unnamed $\frac{2(K_{11}+K_{00})}{2(K_{11}+K_{00})+K_{10}+K_{01}}$	
	Excluded		Dice $\frac{2K_{11}}{2K_{11}+K_{10}+K_{01}}$
Double weight for unmatched pairs	Included	Rogers-Tanimoto $\frac{K_{11}+K_{00}}{K_{11}+K_{00}+2(K_{10}+K_{01})}$	
	Excluded		Unnamed $\frac{K_{11}}{K_{11}+K_{00}+2(K_{10}+K_{01})}$
Matched pairs excluded from denominator	—	Unnamed $\frac{K_{11}+K_{00}}{K_{10}+K_{01}}$	Kulczynski $\frac{K_{11}}{K_{10}+K_{01}}$

is, $\sqrt{1 - S_{SM}} = d$. Therefore, square root of the complement of S_{SM} is a metric function. In the view of probability, S_{SM} means the probability that a randomly chosen data unit achieves the same score on both variables.

As a related similarity measure, Rosers-Tanimoto similarity S_{RT} have been suggested as follows.

$$S_{RT} = \frac{K_{11} + K_{00}}{K_{11} + K_{00} + 2(K_{10} + K_{01})}. \tag{2.8}$$

which doubles the sum of mismatches $(K_{01} + K_{10})$. It is monotonic with S_{SM} and also a metric function.

Russel-Rao similarity measure

$$S_{RR} = \frac{K_{11}}{T}. \tag{2.9}$$

S_{RR} is simple similarity measure that considers only the number of 1-1 matches K_{11}. The value of S_{RR} is the probability that a randomly chosen data unit will score 1 on both variables. It excludes the number of 0-0 matches K_{00} as irrelevant in counting the number of times the two variables match but does count K_{00} in determining the number of possibilities for a match.

Kulczynski similarity measure

$$S_K = \frac{K_{11}}{K_{10} + K_{01}}. \tag{2.10}$$

The Kulczynski similarity measure is the ratio of combinations on which the elements exhibit the number of 1-1 matches to the number of mismatches. Because the number of matches are not included in the denominator, S_K varies from 0 to ∞

dramatically, therefore, it may cause unstable results when there is no mismatches between two binary vectors [27].

2.1.2 Correlation based similarity measure

Unlike the non-correlation based similarity measures, the correlation based similarity measures include product terms of K_{ij}, which denote the relative dependency. The correlation based similarity measure can explain more complicate behaviors of input data [28]. While there are many variations of correlation based similarity measures, The Pearson similarity measure and Yule similarity measure are famous for pattern recognition and classification applications [22].

Pearson similarity measure

$$S_C = \frac{K_{11}K_{00} - K_{10}K_{01}}{\{(K_{11} + K_{10})(K_{01} + K_{00})(K_{11} + K_{01})(K_{10} + K_{00})\}^{1/2}}. \tag{2.11}$$

To measuring the dependence between two general data, the Pearson product-moment correlation has been fundamentally used in statistics. S_C is directly derived from the general Pearson product-moment correlation based on binary vectors. First,

the Pearson product-moment correlation ρ is defined as follows:

$$\begin{aligned}
\rho &= \frac{\mathrm{cov}(\mathbf{x},\mathbf{y})}{\{\mathrm{var}(\mathbf{x})\mathrm{var}(\mathbf{y})\}^{1/2}} \\
&= \frac{\sum_{t=0}^{T-1}(x_i-\overline{x})(y_i-\overline{y})}{\left\{\left[\sum_{t=0}^{T-1}(x_i-\overline{x})^2\right]\left[\sum_{t=0}^{T-1}(y_i-\overline{y})^2\right]\right\}^{1/2}} \\
&= \frac{\sum_{t=0}^{T-1}x_iy_i - \frac{1}{T}\left(\sum_{t=0}^{T-1}x_i\right)\left(\sum_{t=0}^{T-1}y_i\right)}{\left\{\left[\sum_{t=0}^{T-1}x_i^2 - \frac{1}{T}\left(\sum_{t=0}^{T-1}x_i\right)^2\right]\left[\sum_{t=0}^{T-1}y_i^2 - \frac{1}{T}\left(\sum_{t=0}^{T-1}y_i\right)^2\right]\right\}^{1/2}}. \qquad (2.12)
\end{aligned}$$

For binary vectors $\mathbf{x}$ and $\mathbf{y}$, the sums of elements of $\mathbf{x}$ and $\mathbf{y}$ can be expressed by using K_{ij} as follows.

$$\begin{aligned}
\sum_{t=0}^{T-1}x_iy_i &= K_{11}, \\
\sum_{t=0}^{T-1}x_i &= K_{11}+K_{10}, \\
\sum_{t=0}^{T-1}y_i &= K_{11}+K_{01}, \\
\sum_{t=0}^{T-1}x_i^2 &= K_{11}+K_{10}, \\
\sum_{t=0}^{T-1}y_i^2 &= K_{11}+K_{01}.
\end{aligned}$$

Substituting theses expression into Eq. 2.12 gives

$$\begin{aligned}
\rho &= \frac{K_{11}-(K_{11}+K_{10})(K_{11}+K_{01})/T}{\{[K_{11}+K_{10}-(K_{11}+K_{10})^2/T][K_{11}+K_{01}-(K_{11}+K_{01})^2/T]\}^{1/2}} \\
&= \frac{K_{11}T-(K_{11}+K_{10})(K_{11}+K_{01})}{\{(K_{11}+K_{10})[T-(K_{11}+K_{10})](K_{11}+K_{01})[T-(K_{11}+K_{01})]\}^{1/2}} \\
&= \frac{K_{11}K_{00}-K_{10}K_{01}}{\{(K_{11}+K_{10})(K_{01}+K_{00})(K_{11}+K_{01})(K_{10}+K_{00})\}^{1/2}}. \qquad (2.13)
\end{aligned}$$

The correlation similarity measure take into account the statistical factors such as means and variances of input data. Therefore, it is invariant to linear transformations of $\mathbf{x}$ and $\mathbf{y}$. The correlation similarity measure is not a metric function. When it is converted to complementary form to correspond to distance, it cannot satisfy the triangle inequality and moreover, it shows a perfect correlation between non-identical elements, such as two column vectors, one of which is the other multiplied by a scalar.

Yule similarity measure

$$S_Y = \frac{K_{11}K_{00} - K_{10}K_{01}}{K_{11}K_{00} + K_{10}K_{01}}. \tag{2.14}$$

The Yule similarity measure is closely related with the data matrix in Table 2.1, which the numerator of S_Y is the determinant of the data matrix. And the upper bound of S_Y is 1 when the match is perfect and the lower bound is -1 when there are no matches at all. S_Y balances the number of matches against that of mismatches effectively, so it is useful for the applications such as classification and taxonomy.

In summary, we have reviewed eight similarity measures which are widely researched in pattern recognition and classification areas. Conventional works have evaluated the performance of various binary similarity measures for binary template matching and handwriting identification. The performance of similarity measure highly depends on the behaviors of the input binary vectors. For instance, the Yule similarity measure shows a stable performance for the template matching [23], whereas the Dice and Pearson similarity measures relatively outperform for the handwriting identification [24]. Therefore, a user should select the similarity measure which best suited the needs of the particular matching problem.

2.2 Mutual Information

Since the concept of the mutual information has been originated by Shannon, mutual information is widely used for many engineering fields such as communication, visual sensing and complexity analysis. Especially in computer vision area, the mutual information provides a robust measure of the similarity between two data that are handled as probability densities. Since Viola *et al.* first has suggested the mutual information can be employed to the medical image alignment [29], many researches apply the mutual information into the registration of multi-modal CT and MR images [30–32] and stereo matching algorithm [33, 34]. In this section, we review a fundamental information theory about entropy and mutual information.

We first revisit the entropy which is defined for a discrete random variable X and its probability mass function $p(x) = P(X = x)$ [35]. The entropy $H(X)$ of a random variable X is defined as

$$H(X) = -\sum_{x} p(x) \log p(x). \tag{2.15}$$

Obviously, $H(X)$ is always semi positive regardless of X. The entropy means an expected value of information, which measures an uncertainty of a random variable X. In a practical usage, we need to extend the entropy to a pair of random variables X and Y. The joint entropy $H(X, Y)$ of a pair of discrete random variables (X, Y) with a joint probability mass function $p(x, y)$ is defined as

$$H(X, Y) = -\sum_{x} \sum_{y} p(x, y) \log p(x, y). \tag{2.16}$$

Furthermore, we introduce mutual information, which is a measure of the amount of information that is commonly contained in X and Y. The mutual information $I(X; Y)$ is the relative distance between the joint distribution $p(x, y)$ and the product

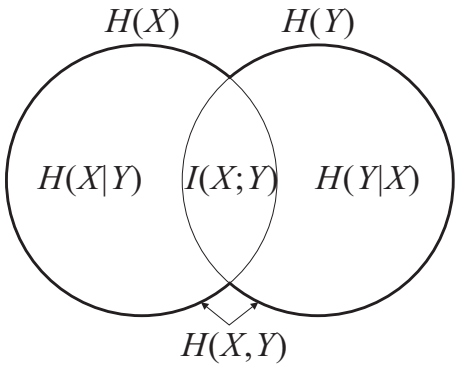


Figure 2.1: Venn diagram for the relationship between mutual information and entropies of (X, Y).

of distribution $p(x)p(y)$ as following:

$$I(X;Y) = -\sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)}. \tag{2.17}$$

The mutual information can be interpreted as a linear combination of entropies from its definition and finally draw following three relationships.

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y), & (2.18)\\ &= H(Y) - H(Y|X), & (2.19)\\ &= H(X) + H(Y) - H(X,Y). & (2.20) \end{aligned}$$

Fig. 2.1 shows the Venn diagram for a relationship between $H(X)$, $H(Y)$, $H(X|Y)$, $H(Y|X)$, and $I(X;Y)$. The areas bounded by circle lines represent the information or the uncertainties of the random variables. Note that, the intersection of information in X and Y is corresponding to the mutual information $I(X;Y)$.

Because the entropy is always larger than or equals to zero, Eq. (2.18) and (2.19) represent minimum and maximum boundaries of a mutual information as following:

$$0 \leq I(X;Y) \leq \min(H(X), H(Y)). \tag{2.21}$$

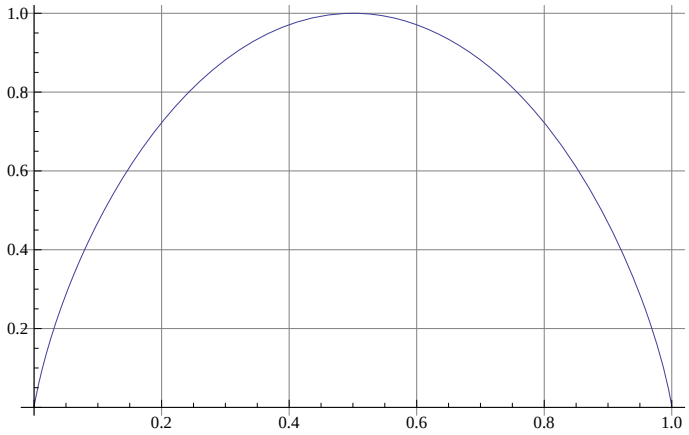


Figure 2.2: The graph of $H(X)$ with respect to p.

Notice that, the minimum value of $I(X;Y)$ occurs when X and Y are independent, and the maximum value occurs when X and Y completely dependent each other.

It is noteworthy to analyze the entropy of the binary random variable, *i.e* the discrete random variable X with the alphabet $\{0, 1\}$. When the probability of success is p, then $H(X)$ is represented as

$$H(X) = -p \log p - (1-p) \log(1-p). \tag{2.22}$$

The graph of $H(p)$ with respect to p is represented in Fig. 2.2. The figure illustrates that the entropy is zero when $p = 0$ or 1 which means X has no randomness and maximized when $p = 0.5$, which means the uncertainty is maximum.

Chapter 3

Mutual Information based Similarity Measure for Binary Activity Vectors

3.1 Introduction

Recently, video technologies have been converging with networking systems and facilitating many applications based on visual sensor networks, such as visual surveillance, environmental monitoring and panoramic view synthesis [6–8]. A visual sensor network usually employs multi-view videos captured by multiple cameras. Correspondence matching among multiple views is one of the most important and challenging issues to analyze the multiple videos and provide useful visual information. Practically, we find the corresponding pixels which are the projections of a same scene point in 3-D space onto the different image planes. A lot of efforts have been made to efficiently find the correspondence matching [1, 2, 9–18].

However, in real multiple view video scenarios such as wide area surveillance applications, the conventional approaches have difficulties to find the reliable correspondence between multiple cameras by the stereo matching algorithms [9–12] or the feature based matching algorithms [1, 13] due to the large photometric changes and severely different camera parameters. Compared to the correspondence matching algorithms for the still images, the consecutive frames of the video sequences enable to utilize the activity information from the moving objects, which provide robust matching performances to the photometric variations and view differences. Nevertheless, many algorithms based on the activity information require accurate object tracking results which are very challenging in general dynamic video sequences [15–18] or suffer from erroneous matches due to the unsuitable similarity measure [2].

The correspondence matching algorithm for the video sequences presented in this work exploits the activity vectors that the occurrence patterns of the foreground objects. The primary contribution of this work is to propose a new similarity measure based on the mutual information of the activity vectors so that it can reliably explain various behaviors of the activity vectors between the matching position pairs. For a single position matching, the proposed similarity measure shows more robust and reliable performance than the Hamming distance measure that the conventional work used in [2]. As utilizing the proposed similarity measure, our algorithm handles the correspondence problem for multiple pixel positions in order to estimate the reliable homography between two views. We carefully select the multiple pixel position at which the similarity measures of the corresponding position pairs are reliably evaluated and determine the corresponding pixel positions by using MRF framework associated both the similarity measures of activity vectors and local structure

preservation. We demonstrate that the proposed matching algorithm is robust to the photometric changes and arbitrary camera allocations and also does not need any prior knowledge such as the camera parameters and the poses of the objects.

In this chapter, we propose a accurate and robust measure to describe the similarity between two features for multi-view video sequences. We regard activity vectors as outputs of binary random variables, and measure their similarity based on the mutual information between two random variables. While the conventional Hamming distance only counts the elements with different values between activity vectors [2], the proposed mutual information based similarity (MIBS) measure considers all possible combinations of 0's and 1's between activity vectors and yields more reliable matching performance. Experimental results demonstrate that the proposed MIBS measure provides accurate and reliable matching results in various camera configurations, and yields better performance than other similarity measures including the work in [2].

The rest of this chapter is organized as follows. In Section 3.2, we present the mutual information based similarity measure for activity, which is our main idea. In Section 3.3, we give the qualitative and quantitative experiments results for various sample video sequences. In Section 3.4, we finally make a conclusion.

3.2 Similarity Measure for Correspondence Matching

The conventional measures including the Hamming distance are devises to measure the dissimilarity between two activity vectors. However, it cannot fully reflect the behavior of activity vectors, possibly leading to incorrect matching results. We propose a more accurate and robust similarity measure, called MIBS, which describes

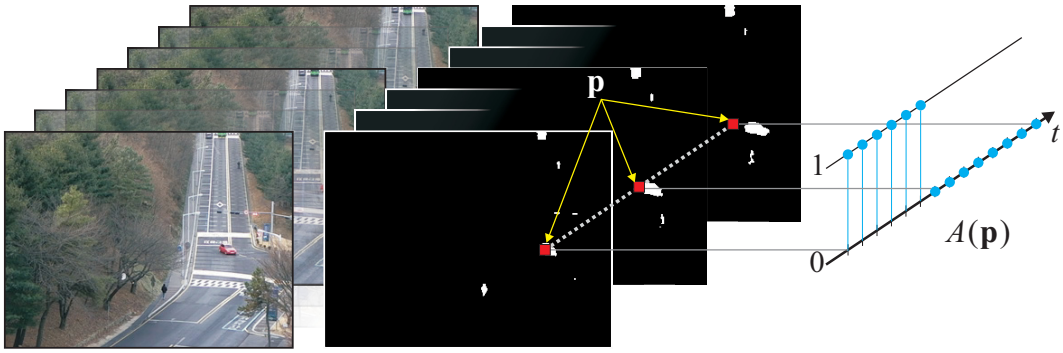


Figure 3.1: An example of activity vector. The t-th element in an activity vector $A(\mathbf{p})$ has binary value 1 or 0, respectively, when $\mathbf{p}$ belongs to a foreground object or the background at time t.

the joint behavior of two activity vectors more faithfully using the mutual information.

3.2.1 Activity Based Correspondence Matching

Let $\mathcal{I} = \{I_t : t = 0, 1, \ldots, T-1\}$ be a video sequence composed of T frames, in which I_t denotes an image frame at time t. For each frame I_t, we detect moving foreground objects using the background subtraction method in [36]. Then, we generate the binary map I_t^B, in which the pixels belonging to the foreground objects and the static background are assigned binary values of 1 and 0, respectively. We then define the binary time series $A(\mathbf{p})$ at each pixel position $\mathbf{p}$ by

$$A(\mathbf{p}) = (I_0^B(\mathbf{p}), I_1^B(\mathbf{p}), ..., I_{T-1}^B(\mathbf{p})) \tag{3.1}$$

where $I_t^B(\mathbf{p})$ is the binary value of $\mathbf{p}$ in I_t^B. We refer to $A(\mathbf{p})$ as the activity vector, since it represents the temporal occurrence pattern of moving foreground objects at $\mathbf{p}$ through a video sequence [2]. We also say that pixel $\mathbf{p}$ is active at time t when

$I_t^B(\mathbf{p}) = 1$, or inactive otherwise. Fig. 3.1 illustrates an activity vector. The left images represent original video frames, and the middle ones are the binary maps in which white and black pixels depict foreground objects and the static background, respectively. The time series $A(\mathbf{p})$ is determined by the binary values at the red points, which have the same pixel position $\mathbf{p}$ through the video sequence.

We assume that cameras are static while capturing multi-view video sequences. Therefore, each pixel position $\mathbf{p}$ corresponds to a unique scene point in the real world. Also, the activity vector $A(\mathbf{p})$ represents the temporal occurrence pattern of foreground objects at $\mathbf{p}$. Therefore, when two cameras capture the same scene point, the corresponding pixels in the two views should yield the same activity vector in the ideal case. For instance, in Fig. 3.2(a), two cameras, which are networked and time-synchronized to each other, capture a moving object and generate two video sequences $\mathcal{I}$ and $\mathcal{J}$. In Fig. 3.2(b), $\mathbf{p}$ and $\mathbf{q}$ are the projected pixels of the same scene point $\mathbf{x}$ onto $\mathcal{I}$ and $\mathcal{J}$, respectively. Since $\mathbf{p}$ and $\mathbf{q}$ correspond to each other, the activity vectors $A(\mathbf{p})$ and $A(\mathbf{q})$ are identical as shown in Fig. 3.2(c), even though the cameras have different parameters. This indicates that the activity vector is an efficient invariant feature to find the correspondences in multi-view video sequences [2]. More specifically, for a source pixel position $\mathbf{p}$ in $\mathcal{I}$, we find the activity vector $A(\mathbf{p})$. Then, we measure the similarity $s(\mathbf{p}, \mathbf{q})$ between $A(\mathbf{p})$ and $A(\mathbf{q})$ for each candidate pixel position $\mathbf{q}$ in $\mathcal{J}$. Finally, we decide the candidate pixel position $\mathbf{q}^*$, which maximizes the similarity $s(\mathbf{p}, \mathbf{q}^*)$, as the best matching position. In other words,

$$\mathbf{q}^* = \arg\max_{\mathbf{q} \in \mathcal{J}} s(\mathbf{p}, \mathbf{q}) \tag{3.2}$$

for each $\mathbf{p} \in \mathcal{I}$.

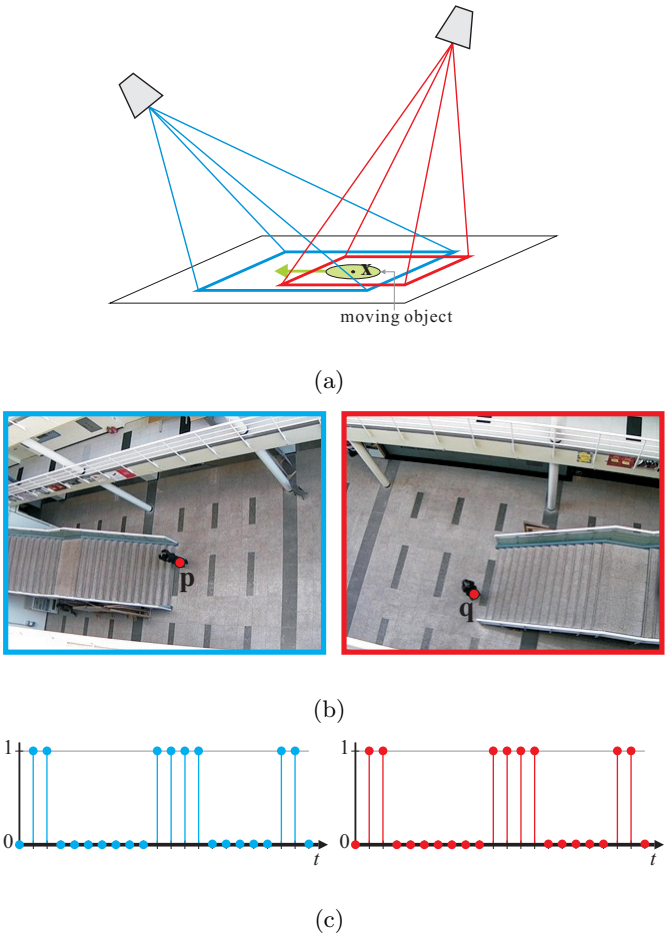


Figure 3.2: Activity vectors at corresponding pixels. (a) A scene point **x** is captured by two different cameras. (b) **p** and **q** are the projected pixels of **x** onto the two views. (c) The activity vectors at **p** and **q** are identical, even though the two cameras have different parameters.

3.2.2 Generalized Similarity Measure for Activity

In order to obtain reliable matching results via (3.2), an efficient similarity measure $s(\mathbf{p}, \mathbf{q})$ should be defined. In the Ermis *et al.*'s algorithm [2], the Hamming distance is used to measure the dissimilarity between two activity vectors, which is the number of element positions in which the two vectors have different values. In this work, we propose the MIBS measure by considering the joint behavior of activity vectors more thoroughly.

Let us analyze a given pair of activity vectors, $A(\mathbf{p})$ and $A(\mathbf{q})$, by considering all the possible combinations of the binary values of the corresponding elements. Let $K_{mn}(\mathbf{p}, \mathbf{q})$ represent the number of element positions, at which $A(\mathbf{p})$ has value m and $A(\mathbf{q})$ has value n,

$$K_{mn}(\mathbf{p}, \mathbf{q}) = |\{t : A_t(\mathbf{p}) = m \text{ and } A_t(\mathbf{q}) = n\}|, \quad m, n \in \{0, 1\}, \tag{3.3}$$

where $|\cdot|$ denotes the cardinality of a set, and $A_t(\cdot)$ is the t-th element of an activity vector $A(\cdot)$. From now on, we omit the arguments $\mathbf{p}$ and $\mathbf{q}$ from $K_{mn}(\mathbf{p}, \mathbf{q})$ for simpler notations. We have four numbers K_{00}, K_{01}, K_{10}, and K_{11}, since the activity vectors are composed of binary elements. Notice that the number of total elements in each activity vector is T, which equals to the length of the video sequence. Therefore, K_{mn}'s have the following properties:

$$K_{00} + K_{01} + K_{10} + K_{11} = T, \tag{3.4}$$

$$0 \leq K_{mn} \leq T, \quad m, n \in \{0, 1\}. \tag{3.5}$$

Note that the subscripts m and n of K_{mn} are associated with $A(\mathbf{p})$ and $A(\mathbf{q})$, respectively, and thus $K_{01} \neq K_{10}$ in general.

In the ideal case, when $\mathbf{p}$ and $\mathbf{q}$ correspond to each other, the activity vectors $A(\mathbf{p})$ and $A(\mathbf{q})$ should be exactly the same, resulting in $K_{00} + K_{11} = T$ and $K_{01} + K_{10} = 0$. However, in practical situations, the activity vectors are generally different, although similar, due to various factors such as acquisition noise, occlusion, and background subtraction errors. As $A(\mathbf{p})$ and $A(\mathbf{q})$ become more similar, the sum $K_{00} + K_{11}$ increases and the sum $K_{01} + K_{10}$ decreases at the same time, according to the property in (3.4). It is worth to note that $K_{01} + K_{10}$ is the Hamming distance in [2], which is a reasonable measure of the dissimilarity between the activity vectors.

However, in a typical video sequence, it is observed that foreground moving objects occupy relatively smaller areas than the static background. Furthermore, an activity vector derived at a pixel position is generally very sparse such that the number of active elements is very small in comparison with the total number of video frames. Consequently, the information of moving objects has a bigger contribution in characterizing the behavior of activity vectors than that of the static background. More specifically, K_{00}, K_{01}, K_{10}, and K_{11} have different importance in the correspondence matching procedure. In this context, the Hamming distance, which counts only K_{01} and K_{10} with equal weight regardless of the distributions of binary elements, cannot fully represent the activity information. Therefore, we introduce a generalized similarity measure by employing all K_{mn}'s, which is given by

$$s(\mathbf{p}, \mathbf{q}) = \sum_{m,n \in \{0,1\}} \alpha_{mn}(\mathbf{p}, \mathbf{q}) K_{mn}(\mathbf{p}, \mathbf{q}) \tag{3.6}$$

where $\alpha_{mn}(\mathbf{p}, \mathbf{q})$ is a weighting parameter reflecting the relative importance of $K_{mn}(\mathbf{p}, \mathbf{q})$ in the similarity measure. For example, when $\alpha_{00}(\mathbf{p}, \mathbf{q}) = \alpha_{11}(\mathbf{p}, \mathbf{q}) = 1$ and $\alpha_{01}(\mathbf{p}, \mathbf{q}) = \alpha_{10}(\mathbf{p}, \mathbf{q}) = 0$, the generalized similarity measure in (3.6) is reduced

to

$$s(\mathbf{p}, \mathbf{q}) = K_{00} + K_{11} = T - (K_{01} + K_{10}). \tag{3.7}$$

In this case, the maximization of the similarity measure $s(\mathbf{p}, \mathbf{q})$ yields the same result as the minimization of the Hamming distance $K_{01} + K_{10}$.

3.2.3 Mutual Information Based Similarity Measure

The relative importance of K_{mn}'s in the similarity measure relies on the joint probability distribution of the binary values (m, n)'s in activity vectors. Thus, we regard activity vectors as the realization of joint random variables, and measure their similarity using the mutual information of the random variables. Theoretically, the mutual information represents the amount of information commonly contained in two random variables [35]. A large amount of mutual information means that the random variables are highly correlated and have similar probability distributions. The mutual information has been employed in many applications. For instance, it has been used to understand the joint behavior as well as the individual behavior of two sparse vectors in [29, 32]. Moreover, the mutual information does not require prior knowledge even for multi-modal sequences acquired by different sensors [31]. Because of these desirable characteristics, we also use the mutual information to measure the similarity between two activity vectors, which are captured by different cameras with different parameters and separately preprocessed using the background subtraction method.

Let us first define Bernoulli random variables X and Y at pixel positions $\mathbf{p}$ and $\mathbf{q}$, which randomly take binary values $A_t(\mathbf{p})$ and $A_t(\mathbf{q})$ at time t, respectively. Then the probability distributions of X and Y are empirically estimated from K_{mn}'s in (3.3), since the length of a video sequence is long enough to represent the statistical

properties faithfully. Specifically, the joint probability $p(m,n)$ of the event $\{X = m, Y = n\}$ is estimated by

$$p(m,n) = \frac{K_{mn}}{T}, \quad m,n \in \{0,1\}. \tag{3.8}$$

The marginal probabilities of the events $\{X = m\}$ and $\{Y = n\}$ are then given by

$$\begin{aligned} p(m) &= \Pr\{X = m\} = \frac{K_{m*}}{T} = \frac{1}{T}(K_{m0} + K_{m1}), \quad m \in \{0,1\}, \\ p(n) &= \Pr\{Y = n\} = \frac{K_{*n}}{T} = \frac{1}{T}(K_{0n} + K_{1n}), \quad n \in \{0,1\}, \end{aligned} \tag{3.9}$$

where K_{m*} denotes the number of elements in $A(\mathbf{p})$ with binary value m, and K_{*n} denotes the number of elements in $A(\mathbf{q})$ with value n. Then, the mutual information $I(X;Y)$ between X and Y can be written as [35]

$$\begin{aligned} I(X;Y) &= \sum_{m,n\in\{0,1\}} p(m,n) \log_2 \frac{p(m,n)}{p(m)p(n)} \\ &= \sum_{m,n\in\{0,1\}} \frac{K_{mn}}{T} \log_2 \frac{TK_{mn}}{K_{m*}K_{*n}}. \end{aligned} \tag{3.10}$$

We employ this mutual information as the activity similarity measure

$$s(\mathbf{p},\mathbf{q}) = I(X;Y), \tag{3.11}$$

and call it as the MIBS measure. Note that this is equivalent to setting the weight parameters in the generalized measure in (3.6) to

$$\alpha_{mn}(\mathbf{p},\mathbf{q}) = \frac{1}{T} \log_2 \frac{TK_{mn}}{K_{m*}K_{*n}}, \quad m,n \in \{0,1\}. \tag{3.12}$$

The mutual information $I(X;Y)$ is the intersection of the information in X with the information in Y, and the following inequalities hold [35]:

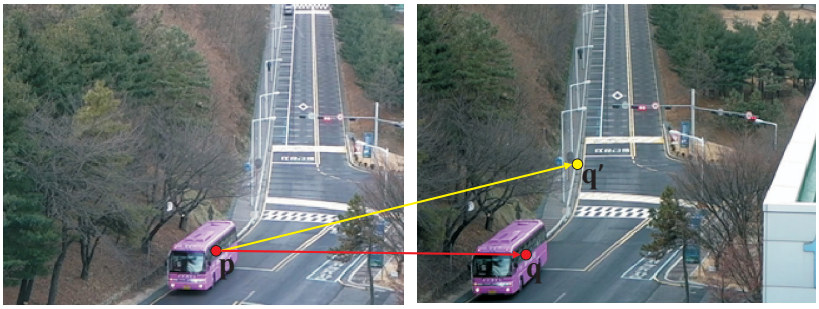
$$0 \le I(X;Y) \le \min\{H(X), H(Y)\} \tag{3.13}$$

where $H(X)$ and $H(Y)$ are the entropies of X and Y, given by

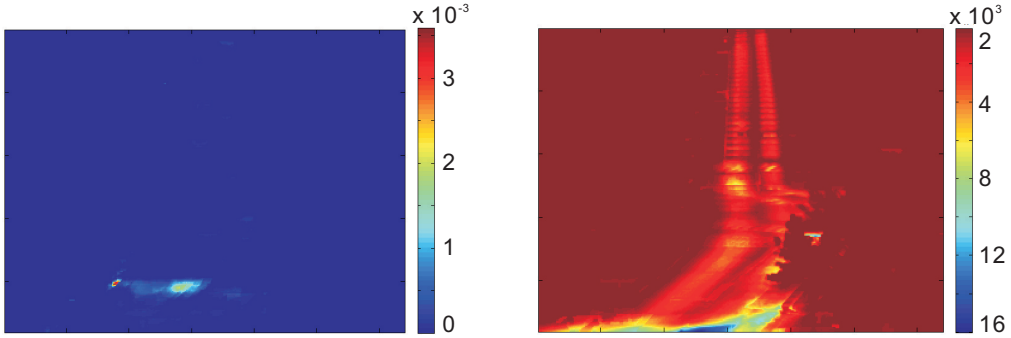
$$\begin{aligned} H(X) &= -\sum_{m} \frac{K_{m*}}{T} \log_2 \frac{K_{m*}}{T}, \\ H(Y) &= -\sum_{n} \frac{K_{*n}}{T} \log_2 \frac{K_{*n}}{T}. \end{aligned}$$

In typical video sequences for surveillance applications, the activity vector $A(\mathbf{p})$ contains much less active elements than inactive ones, *i.e.*, $K_{1*} \ll K_{0*}$. Thus, the entropy $H(X)$ gets larger as K_{1*} increases, and reaches the maximum value of 1 when $K_{1*} = K_{0*}$. In other words, $H(X)$ represents the self information of the activity vector [35]. Let us consider the extreme case when $\mathbf{p}$ always belongs to the static background at which a foreground object never occurs. Then, we have $K_{1*} = 0$, $K_{0*} = T$, and $H(X) = I(X;Y) = 0$. In such a case, the mutual information cannot convey meaningful information for the correspondence matching. On the contrary, a large value of $H(X)$ provides more confidence on the correspondence matching result of $\mathbf{p}$. Similarly, a large value of $H(Y)$ increases the reliability of the correspondence matching of $\mathbf{q}$.

Fig. 3.3 shows the correspondence matching result of the proposed MIBS measure. In this example, we select a pixel position $\mathbf{p}$ in the left view in Fig. 3.3(a) and find its matching position $\mathbf{q}$ in the right view. Fig. 3.3(b) shows the similarities of all candidate matching positions, which are computed from (3.11), where red and blue regions represent high and low similarities, respectively. In comparison, Fig. 3.3(c) shows the similarity map of the Hamming distance measure, where red regions represent small Hamming distances. We see that the Hamming distance measure does not generate a compactly distinctive region of small distances and returns an incorrect matching position $\mathbf{q}'$ in the right view in Fig. 3.3(a). In contrast, the proposed MIBS measure clearly identifies the region of high similarities and yields an accurate



(a)



(b) (c)

Figure 3.3: Comparison of the activity-based matching results of the proposed MIBS measure and the conventional Hamming distance measure [2]. (a) $\mathbf{q}$ and $\mathbf{q}'$ depict the best matching points to $\mathbf{p}$, which are obtained by the proposed algorithm and the conventional algorithm, respectively. (b) The color map of the proposed algorithm, where red regions depict high similarity values. (c) The color map of the conventional algorithm, where red regions depict small Hamming distances.

Table 3.1: Specifications of the sample sequences.

Name	Number of frame	Camera configuration	Activities
ArtCollege	100,000	zoom, rotation	low
Crossroad	66,000	perspective	high
Desk	150,000	zoom	high
Hall	100,000	zoom, rotation	medium
ParkingLot	200,000	rotation	low
Road	172,000	translation	medium
Jahayeon	100,000	zoom, rotation	high
Stair	126,000	rotation	medium
Library	150,000	zoom, translation	medium
Soccer	100,000	perspective	high

matching position $\mathbf{q}$.

3.3 Experimental Results

3.3.1 Test Sample Sequences

We evaluate the performance of the proposed algorithm using 10 pairs of test video sequences for indoor and outdoor scenes. The image resolution is 320×240 and the frame lengths are varying from 100,000 to 200,000. The test video sequences are captured by two synchronized cameras with various configurations, such as translations for the 'Library' and 'Road' sequences, different zoom factors for the 'Hall,' 'Desk,' and 'ArtCollege' sequences, rotational transformations for the 'Jahayeon'

(a)

(b)

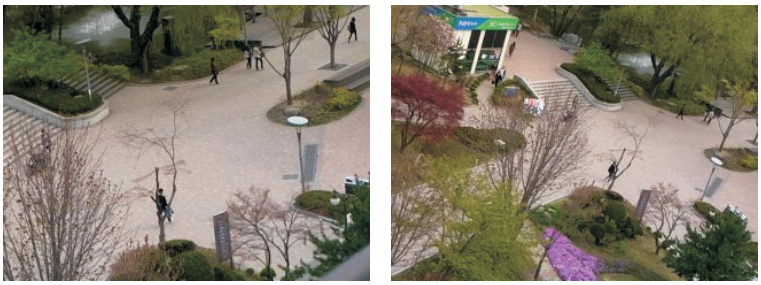
(c)

(d)

Figure 3.4: Experimental sequences. Left image is $\mathcal{I}$ and right image is $\mathcal{J}$. (a) Hall. (b) Desk. (c) Road. (d) Stair.

(a)

(b)



(c)

Figure 3.5: Experimental sequences. Left image is $\mathcal{I}$ and right image is $\mathcal{J}$. (a) Library. (b) ArtCollege. (c) Jahayeon.

(a)

(b)

(c)

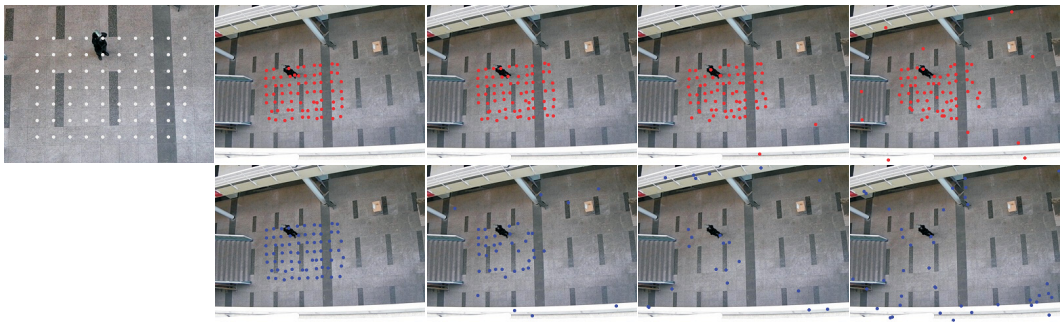
Figure 3.6: Experimental sequences. Left image is $\mathcal{I}$ and right image is $\mathcal{J}$. (a) ParkingLot. (b) Soccer. (c) Crossroad.

and 'ParkingLot' sequences, and different viewing angles for the 'Stair,' 'Soccer,' and 'Crossroad' sequences.

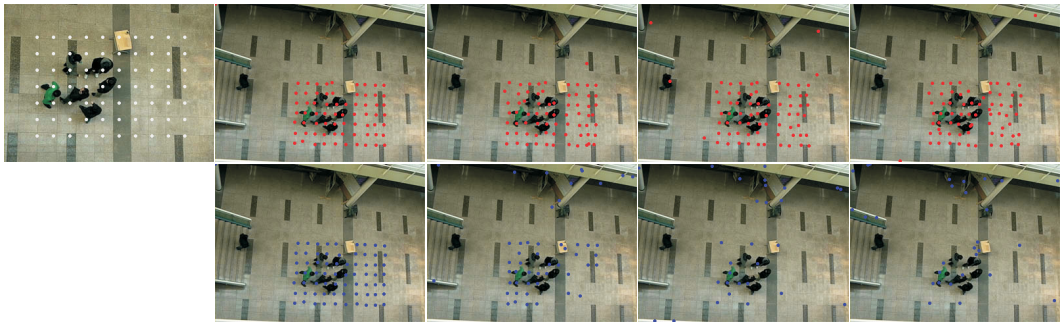
3.3.2 Performance of MIBS Measure

Next, we compare the matching performance of the proposed MIBS measure with that of the conventional Hamming distance measure [2]. We sample pixel positions on a regular grid in one view, which contain activities, and then find the corresponding pixel positions in the other view via (3.2). For a fair comparison, we use the same adaptive activity areas for generating activity vectors.

In each sub-figure in Fig. 3.7∼3.10, the first column shows one view with regularly sampled pixel positions. The upper and lower frames in the second column show the corresponding pixel positions in the other view, which are obtained by maximizing the MIBS measure and minimizing the Hamming distance measure, respectively. It is observed that the proposed MIBS measure yields accurate matching results and preserves regular grid structures in the target frames of most test sequences. However, the conventional measure causes incorrect matching pixels, for example, near the boundaries of the active regions in the 'Library' and 'ArtCollege' sequences, as shown in Fig. 3.8(c) and Fig. 3.9(a). Moreover, when using the conventional measure, the 'Soccer' sequence has a lot of incorrect matching pixels in Fig. 3.10(a). It is because the 'Soccer' sequence contains many large and fast moving objects, which cannot be clearly detected by the background subtraction method, and generates erroneous activity vectors. On the contrary, the proposed similarity measure is more robust to the noise in activity vectors, and therefore provides a significantly better matching result. Also, in the 'Crossroad' sequence in Fig. 3.10(b), note that the conventional measure fails to find the correspondences of

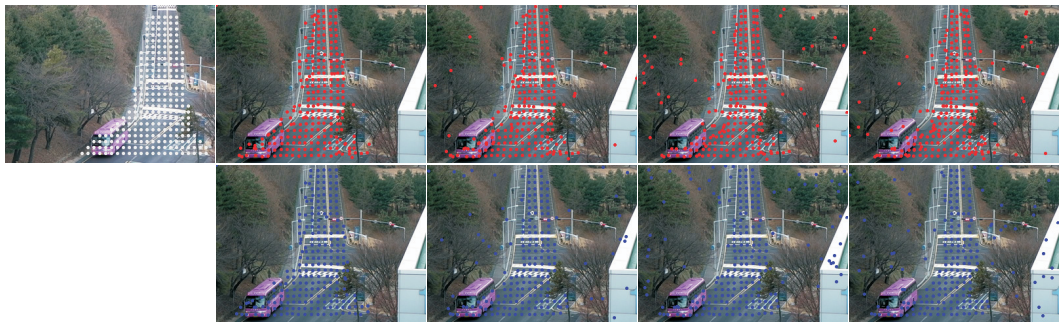


(a)

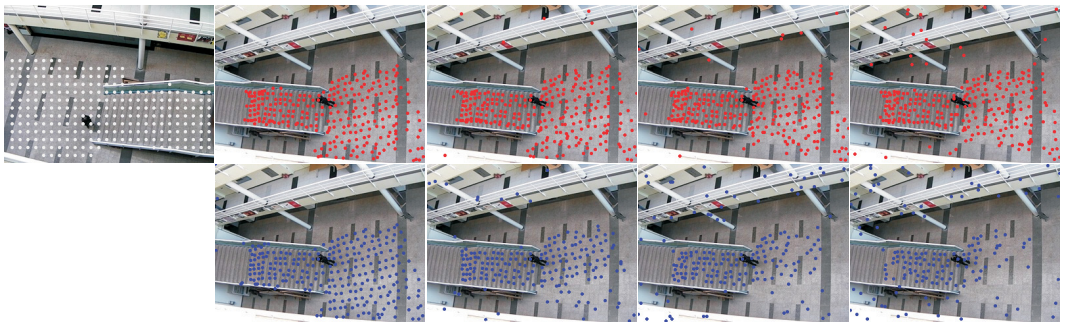


(b)

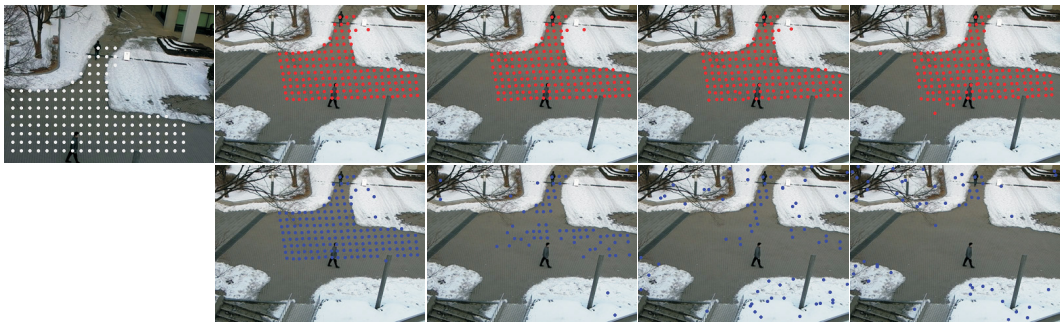
Figure 3.7: Correspondence matching results of the proposed MIBS measure and the conventional Hamming distance measure [2]. (a) Hall. (b) Desk. The first column shows the regularly sampled source pixels in one view, and the upper and lower figures in the second column show their corresponding pixels found by the proposed measure and the conventional measure, respectively. The third, fourth, and fifth columns represent the correspondence matching results when the binary activity vectors include the noise with probability of 0.02, 0.05, and 0.07, respectively.



(a)

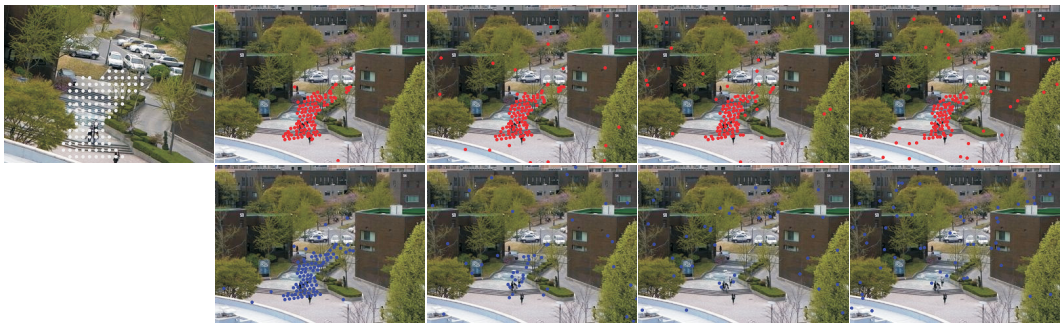


(b)



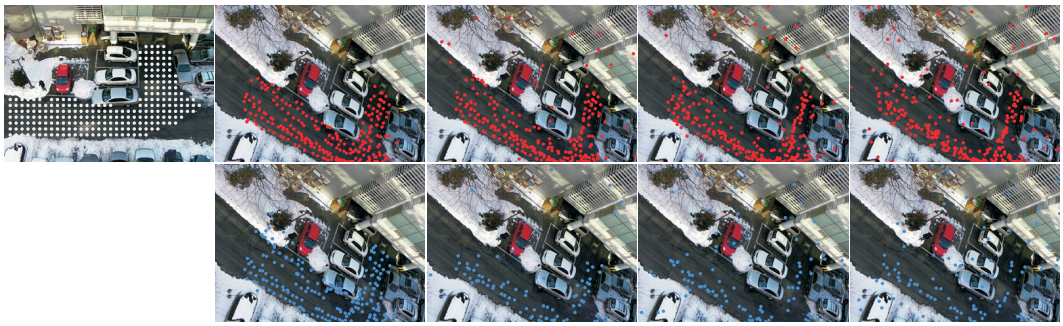
(c)

Figure 3.8: Correspondence matching results of the proposed NIBS measure and the conventional Hamming distance measure [2]. (a) Road. (b) Stair. (c) Library.



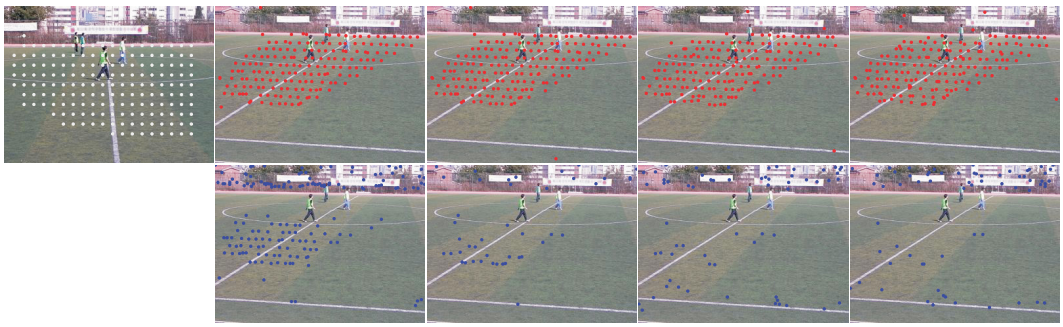
(a)

(b)



(c)

Figure 3.9: Correspondence matching results of the proposed MIBS measure and the conventional Hamming distance measure [2]. (a) ArtCollege. (b) Jahayeon. (c) ParkingLot.



(a)

(b)

Figure 3.10: Correspondence matching results of the proposed MIBS measure and the conventional Hamming distance measure [2]. (a) Soccer. (b) Crossroad.

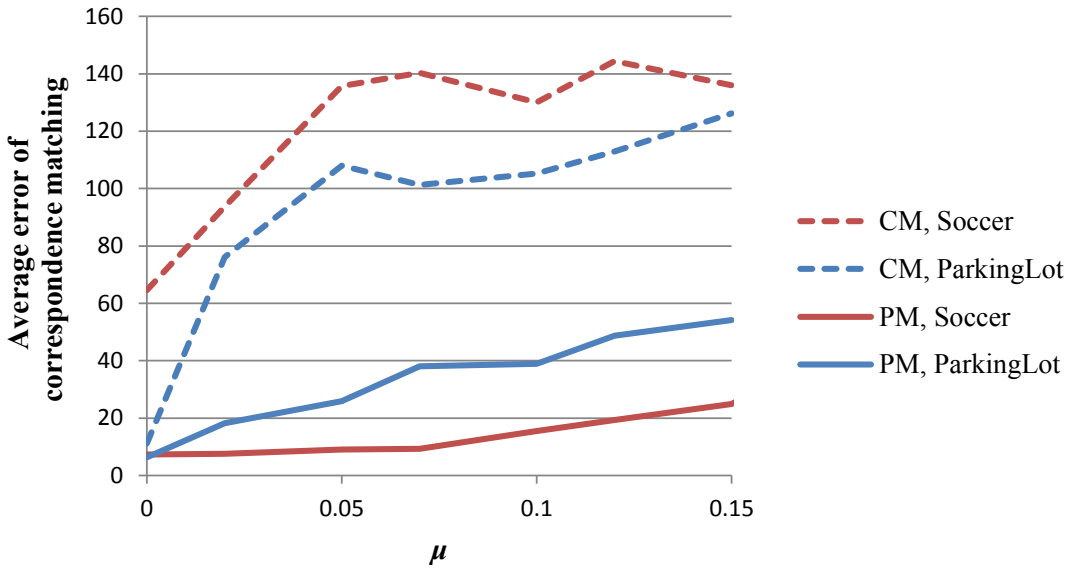


Figure 3.11: Comparison of the correspondence matching errors between the proposed MIBS measure and the conventional Hamming distance measure [2]. The 'Soccer' and 'ParkingLot' sequences are used in this test. The average Euclidean distance between computed matching positions and the ground truth ones is measured according to the noise probability μ. The solid and dashed curves are the results of the proposed measure (PM) and the conventional measure (CM), respectively.

most pixels. The 'Crossroad' sequence contains many foreground objects close to the cameras, and the detected objects occupy large areas and/or are not fully included in one of the two views. Therefore, the true object areas on the ground plane cannot be determined correctly and the activity vectors become unreliable. The proposed measure also yields a relatively worse matching result on the 'Crossroad' sequence than on the other sequences. However, the proposed measure still provides better performance than the conventional measure.

We also test the robustness of the similarity measures against the noise in activity vectors. We generate a binary random noise image U_μ, whose pixel values are 1 with probability μ. Then we add the noise image U_μ to a binary map I_t^B, obtained by the background subtraction method, to generate a noisy map

$$\tilde{I}_t^B = I_t^B \oplus U_\mu, \tag{3.14}$$

where $\oplus$ denotes the pixel-wise exclusive OR operation. We derive the activity vectors from the noisy binary maps and use them to find the correspondences. In Fig. 3.7∼3.10, the third, fourth, and fifth columns compare the matching results with the noise probability $\mu = 0.02$, 0.05, and 0.07, respectively. For all test sequences, the conventional measure provides severely degraded matching results even with a small noise probability $\mu = 0.02$, and provides almost randomly scattered distributions of matching pixels at $\mu = 0.07$. In contrast, the proposed measure maintains the robust matching performance even at $\mu = 0.07$. This is because the conventional Hamming distance measure only counts the numbers K_{01} and K_{10}, which are very sensitive to the noise. On the other hand, the proposed MIBS measure adaptively reflects all combinations of binary values, and exploits the numbers K_{00} and K_{11} as well to find the corresponding pixels more reliably. For a quantitative comparison, we also measure the average Euclidean distances between computed matching positions and the ground truth ones. Fig. 3.11 plots the average matching errors on the 'Soccer' and 'ParkingLot' sequences according to the noise probability μ. The errors with the conventional measure rapidly increase as μ gets higher, whereas those with the proposed measure increase only slightly until $\mu = 0.1$.

3.3.3 Comparison to Other Similarity Measures

In this subsection, we compare the matching performance of the proposed MIBS measure with that of various binary similarity measures introduced in Section 2.1. The pixel positions in $\mathcal{I}$ are selected on a regular grid as shown in Fig. 3.12∼3.14(a). We find the corresponding positions with changing the similarity measures but preserve other conditions for a fair comparison.

For both quantitative and qualitative comparison, we uses three sample sequences 'Soccer,' 'ParkingLot,' and 'Jahayeon' that provides the ground truth. And we exploit 9 different similarity measures 'Hamming(Socak-Michener), 'Ermis,' 'Jaccard-Needham,' 'Dice,' 'Correlation,' 'Yule,' 'Russel-Rao,' 'Rogers-Tanmoto,' and 'Kulzinsky' including the proposed MIBS measure. In Fig. 3.12, the results of the MIBS, 'Jaccard-Needham,' 'Dice,' and 'Kulzinsky' show relatively more reliable matching than other measures. The results of 'ParkingLot' and 'Jahayeon' Sequences also demonstrate similar performance trends. For a quantitative comparison, we also measure the average Euclidean distances between computed matching positions and the ground truth ones as described in Fig. 3.15. The graphs also demonstrate the MIBS provides less average matching errors than other similarity measures.

3.4 Conclusion

In this chapter, we proposed a correspondence matching algorithm for multiple video sequences. We employed the activity vector for correspondence matching, which is a temporal occurrence pattern of moving foreground objects at a specific pixel position. In order to efficiently compare two activity vectors, we considered the activity

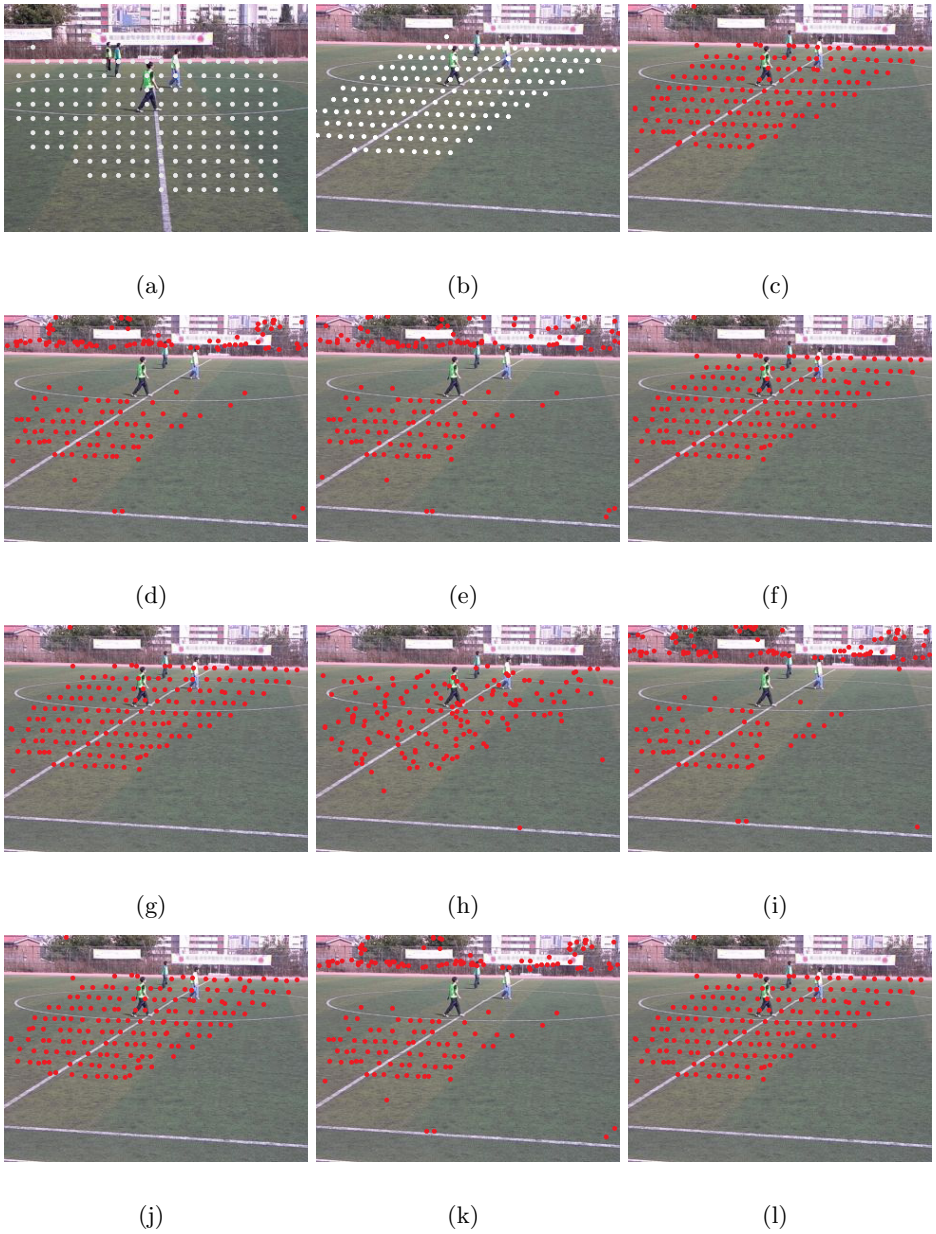
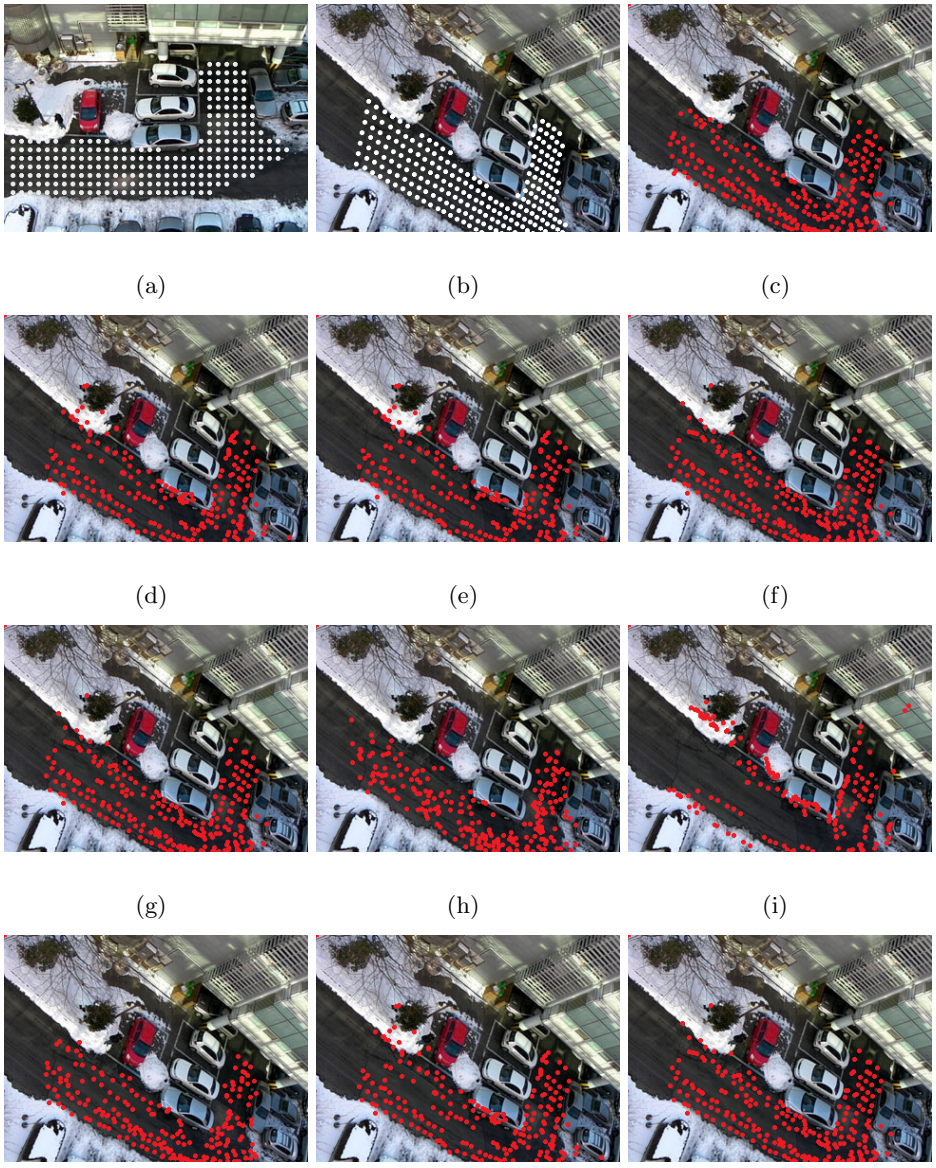


Figure 3.12: Comparative results to various similarity measures for the 'Soccer' sequence. (a) Initial grid positions. (b) Ground truth. (c) The results of MIBS measure, (d) Hamming, (e) Ermis, (f) Jaccard-Needham, (g) Dice, (h) correlation, (i) Yule, (j) Russel-Rao, (k) Rosergs-Tanmoto, and (l) Klzinsky.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 3.13: Comparative results to various similarity measures for the 'ParkingLot' sequence. (a) Initial grid positions. (b) Ground truth. (c) The results of MIBS measure, (d) Hamming, (e) Ermis, (f) Jaccard-Needham, (g) Dice, (h) correlation, (i) Yule, (j) Russel-Rao, (k) Rosergs-Tanmoto, and (l) Klzinsky.

(a) (b) (c)

(d) (e) (f)

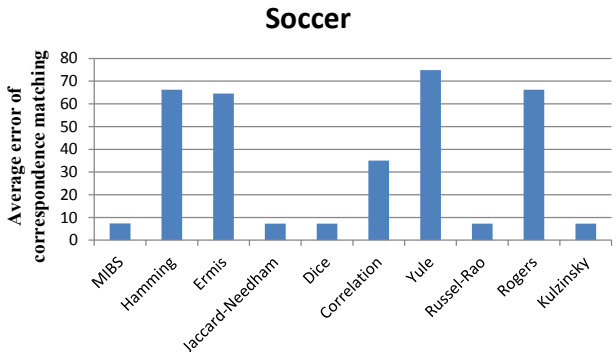
(g) (h) (i)

(j) (k) (l)

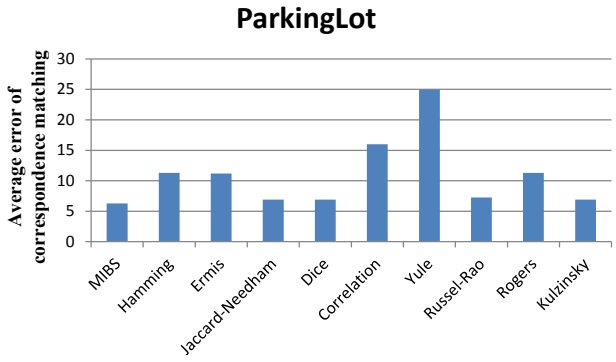
Figure 3.14: Comparative results to various similarity measures for the 'Jahayeon' sequence. (a) Initial grid positions. (b) Ground truth. (c) The results of MIBS measure, (d) Hamming, (e) Ermis, (f) Jaccard-Needham, (g) Dice, (h) correlation, (i) Yule, (j) Russel-Rao, (k) Rosergs-Tanmoto, and (l) Klzinsky.

Table 3.2: Average errors of correspondence matching according to various similarity measures

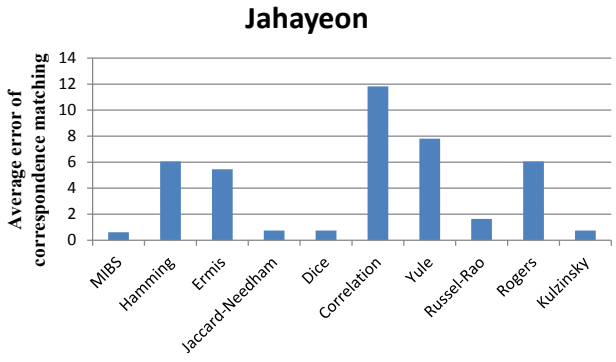
Similarity measure	Soccer	ParkingLot	Jahayeon
MIBS	7.365405	6.298014	0.613001
Hamming	66.217085	11.305671	6.058103
Ermis	64.555733	11.198556	5.454871
Jaccard-Needham	7.242479	6.899561	0.744769
Correlation	35.08293	15.982141	11.824747
Yule	74.904488	24.963444	7.806933
Russel-Rao	7.306032	7.273160	1.641155
Rogers-Tanmoto	66.217085	11.305671	6.058103
Kulzinsky	7.242479	6.899561	0.744769



(a)



(b)



(c)

Figure 3.15: Average errors of correspondence matching to the ground truth. (a) 'Soccer,' (b) 'ParkingLot,' and (c) 'Jahayeon' sequences.

vectors as the random variables and measured their similarity by using the mutual information of the related random variables. Since the mutual information describes all the behaviors of the activity vectors, it provides more accurate and reliable matching results than that of the conventional Hamming distance measure. Furthermore, the proposed MIBS measure yields very robust results when the input sequences are collapsed by the additive noises. Simulation results demonstrated that the matching pairs from the proposed MIBS measure yielded promising results in both quantitative and qualitative comparison to Hamming distance measure even under the additive noises. And the comparison to other similarity measures, the MIBS measure shows good matching performances. Therefore, the proposed MIBS measure is a very promising for finding correspondence matching in multi-view video sequences, furthermore, it is applicable to variable applications such as visual surveillance and panoramic view generation.

Chapter 4

Correspondence Matching for Multi-view Surveillance Video Sequences using MIBS measure

4.1 Introduction

Recently, video technologies have been converging with networking systems and facilitating many applications based on visual sensor networks, such as visual surveillance, environmental monitoring and panoramic view synthesis [6–8]. A visual sensor network usually employs multi-view videos captured by multiple cameras. Correspondence matching among multiple views is one of the most important and challenging issues to analyze the multiple videos and provide useful visual information. Practically, we find the corresponding pixels which are the projections of a same scene point in 3-D space onto the different image planes. A lot of efforts have been made to efficiently find the correspondence matching [1, 2, 9–18].

However, in real multiple view video scenarios such as wide area surveillance applications, the conventional approaches have difficulties to find the reliable correspondence between multiple cameras by the stereo matching algorithms [9–12] or the feature based matching algorithms [1, 13] due to the large photometric changes and severely different camera parameters. Compared to the correspondence matching algorithms for the still images, the consecutive frames of the video sequences enable to utilize the activity information from the moving objects, which provide robust matching performances to the photometric variations and view differences. Nevertheless, many algorithms based on the activity information require accurate object tracking results which are very challenging in general dynamic video sequences [15–18] or suffer from erroneous matches due to the unsuitable similarity measure [2].

We have found the pixel-wise correspondences between two view frames by only using the MIBS measure in the previous chapter. However, it is insufficient to discover the homography between two views from the multiple surveillance cameras. Therefore, as utilizing the proposed similarity measure, our algorithm handles the correspondence problem for multiple pixel positions in order to estimate the reliable homography between two views. We carefully select the multiple pixel position at which the similarity measures of the corresponding position pairs are reliably evaluated and determine the corresponding pixel positions by using MRF framework associated both the similarity measures of activity vectors and local structure preservation. We demonstrate that the proposed matching algorithm is robust to the photometric changes and arbitrary camera allocations and also does not need any prior knowledge such as the camera parameters and the poses of the objects.

In this chapter, we propose a more accurate and robust correspondence matching

algorithm for multi-view video sequences. We regard activity vectors as outputs of binary random variables, and measure their similarity based on the mutual information between two random variables. While the conventional Hamming distance only counts the elements with different values between activity vectors [2], the proposed mutual information based similarity (MIBS) measure considers all possible combinations of 0's and 1's between activity vectors and yields more reliable matching performance. Moreover, we find the correspondence matching for multiple pixel positions to estimate the homography transform between two views. Specifically, we select reliable pixel positions by iteratively applying the bidirectional matching, and refine the matching positions by optimizing a cost function based on the Markov random field (MRF). Experimental results demonstrate that the proposed MIBS algorithm provides accurate and reliable matching results in various camera configurations, and yields better performance than the conventional algorithm [2].

The rest of this chapter is organized as follows. Section 4.2 reviews the previous researches on the correspondence matching for still images and video sequences. In Section 4.3, we present the framework of the correspondence matching for the multi-view surveillance video sequences, which is our main idea. Next, we explain the correspondence matching for the multiple pixel positions by using MRF framework in order to establish a reliable homography between two different views in Section 4.3. In Section 4.4, we give the qualitative and quantitative experiments results for various sample video sequences and applications to the panoramic view generation. In Section 4.5, we finally make a conclusion.

4.2 Related Works

4.2.1 Correspondence Matching of Images

Traditional stereo matching algorithms obtain a pixel-wise dense correspondence map between two rectified images [9, 10]. Specifically, a window is assigned to each pixel and the corresponding pixels between two images are determined such that their windows yield the smallest difference of pixel intensities. Hence the stereo matching is sensitive to the variations of epipolar constraint, illumination, and camera parameters. Advanced stereo matching algorithms are robust to the severe photometric variations, however, still suffer from the uncertainty of camera parameters [11, 12]. Therefore, the stereo matching techniques are not proper to general surveillance applications in which the parameters of multiple cameras are very different.

Another approach to find the correspondence of images is based on the feature detection methods such as scale invariant feature transform (SIFT) [1] or speeded up robust features (SURF) [13]. While the stereo matching techniques provide a dense correspondence map over a whole image, the feature based methods find the correspondence only for the selected several feature points. Therefore, they are robust to the photometric variations including the changes of illumination and/or lighting conditions, and furthermore, can alleviate the computational complexity to find the correspondences among multiple images. However, they still fail to work under severely different camera geometries, *e.g.*, viewing orientation, and therefore require camera calibration techniques [19, 20].

4.2.2 Correspondence Matching of Videos

In order to find the correspondence matching of two video sequences, Sand *et al.* considered a video sequence as a set of image frames and independently applied the image matching scheme to each pair of image frames [14]. For a given source pixel in one image, the corresponding target pixel in the other image is selected which yields the most similar color value to that of the source pixel. Since this algorithm basically uses the image matching method frame by frame, the correspondence matching result is also vulnerable to the variations of color and/or lighting conditions.

Several algorithms exploit the temporal information of video sequences for view matching [15–18]. The methods in [15–17] compute the centroids of the moving objects over the video sequences, and estimate the homography between two views by considering the centroids as feature points for matching. However, the temporal orders of centroids are not considered for matching which yields unreliable matching performances. On the other hand, Caspi *et al.* employed a motion trajectory which represents a sequence of temporal locations of a moving object over the video sequence [18]. They estimated a homography by matching the two trajectories between two views associated with a same object. However, these algorithms require an accurate result of object tracking between multiple video sequences, which is generally hard to be obtained from the video sequences containing many moving objects.

Ermis *et al.* proposed a correspondence matching algorithm for multiple video sequences based on an activity feature of foreground objects [2]. By performing segmentation to each image, a binary image is obtained which represents the foreground objects and the background region. Then, the activity feature is defined

at each pixel position in the binary image, as the temporal series of binary values through the video sequence. Since the corresponding pixels between two different views should yield a same activity feature, for a given pixel in one view, its corresponding pixel in the other view is selected we determine the corresponding pixels between two views which yield the most similar activity features each other. The activity based correspondence matching algorithm only requires a simple moving object detection, and is more robust to the illumination changes or noises compared to the object tracking (trajectory) based matching approaches. Moreover, it is also suitable for the surveillance applications which employ the multiple cameras with significantly different camera parameters such as viewing angles and positions.

However, the conventional algorithm employs the Hamming distance to measure the similarity between the binary feature vectors, which is not always optimal to find the true correspondence [2]. Especially, it may yield unreliable matching results when the source video includes fast moving objects. Therefore, it is required to devise a novel similarity measure for activity which is more accurate and robust to the errors in moving object detection. Furthermore, in order to facilitate various multi-view applications such as a panoramic view generation, an accurate correspondence matching algorithm for multiple views is also needed.

4.3 Proposed Algorithm

4.3.1 Adaptive Activity Area

For typical video sequences in surveillance applications, a ground plane is included in the static background and foreground objects move on the ground plane. Thus, we practically find the correspondences between the two ground planes of different views

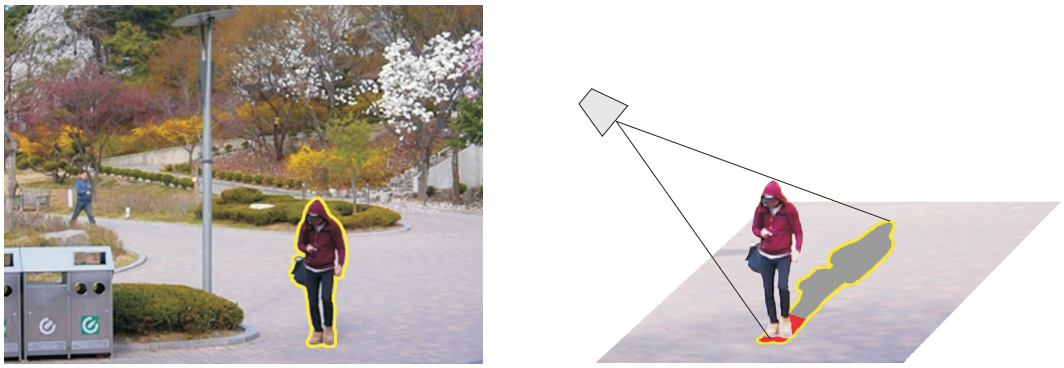
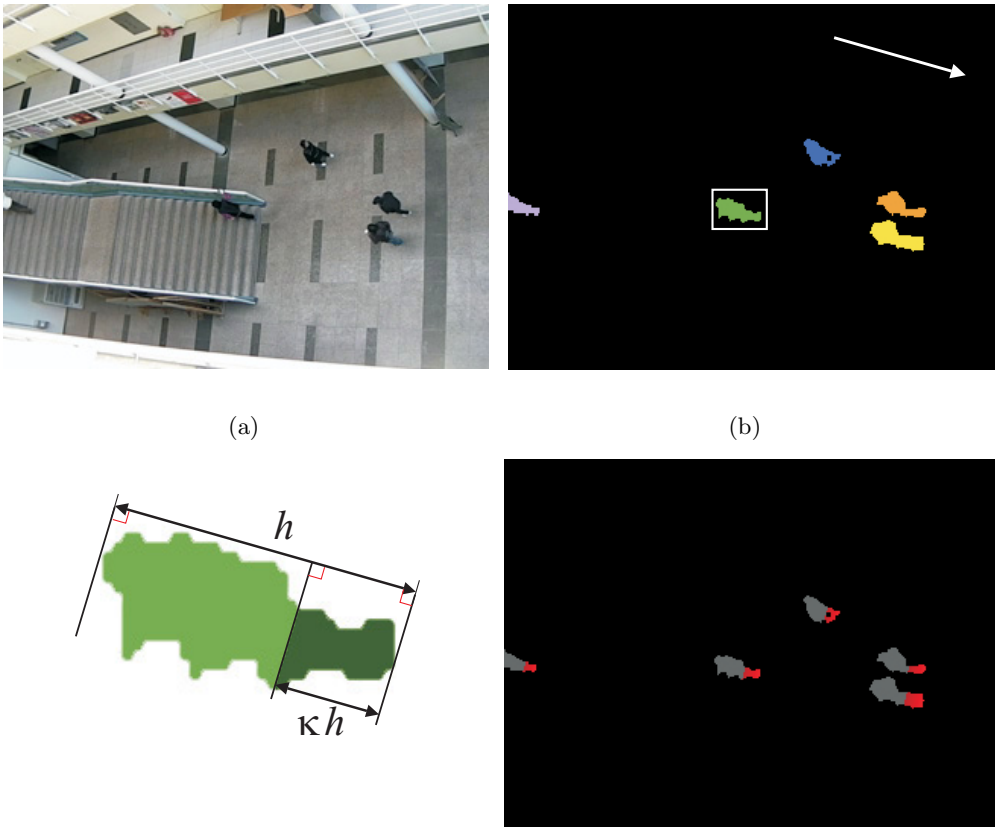


Figure 4.1: Discrepancy between the detected foreground object and the true object area on the ground plane.

using activity vectors. However, foreground objects, detected by the background subtraction method, do not coincide with their touching areas on the ground plane, since a camera is usually configured at a skew angle from the ground plane. For example, when the camera is directed toward the ground at a skew angle as shown in Fig. 4.1, the grey area is detected as an object region, whereas the red area is the true active area on the ground. Therefore, if we form activity vectors from detected object regions directly, then we may obtain incorrect matching results.

The Ermis *et al.*'s algorithm [2] simply uses rectangular bottom areas of bounding boxes of detected objects to compute activity vectors. In this work, we estimate valid activity areas adaptively according to the ground direction of a captured scene. We assume that, as shown in Fig. 4.2(a), a video sequence contains a number of pedestrians standing on the ground plane and the camera does not capture objects upside down. This assumption is acceptable in the surveillance scenarios which we focus on. Then, a moving pedestrian forms an elongated shape along its height direction in Fig. 4.2(b).



(a) (b)

(c) (d)

Figure 4.2: Adaptive activity areas. (a) The 'Stair' sequence. (b) The detected foreground objects are marked by different colors. The white arrow represents the ground direction. (c) h denotes the height of an object along the ground direction, and κ is the ratio for the valid activity area. (d) The adaptive activity areas are denoted by the red color.

Then our algorithm estimates the ground direction which indicates the direction of the ground plane of the objects without any camera parameters. In detail, we identify the moving objects which is denoted by O as illustrated in Fig. 4.2 (b), and define the mass covariance matrix $B(O)$ as

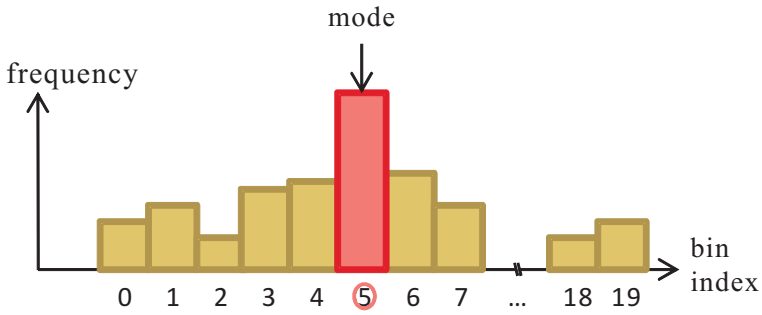
$$B(O) = \begin{bmatrix} b_{xx} & b_{xy} \\ b_{xy} & b_{yy} \end{bmatrix}, \tag{4.1}$$

$$\begin{aligned} b_{xx} &= \sum (o_x - c_x)^2, \\ b_{xy} &= \sum (o_x - c_x)(o_y - o_y), \\ b_{yy} &= \sum (o_y - c_y)^2, \end{aligned}$$

where (o_x, o_y) denote the pixel position in O, and (c_x, c_y) is the center of mass of O. $B(O)$ is positive semidefinite, there are two positive eigen values and their corresponding eigen vectors that represent the dominant directions of the object shapes [37]. Therefore, we select the eigen vector corresponding to the large eigen value of $B(O)$, which denotes the long side direction of the object shape. Fig. 4.2 (b) shows the ground direction $\mathbf{n}$ as depicted by a white arrow. Note that we use a single ground direction which is determined by averaging ground directions, because the grounds are approximately planar. The ground direction is estimated from all long side directions of whole objects. We first make 20 angle bins that cover from 0 to 180 degrees and count the number of objects which are dropped into corresponding angle bins as describe in Fig 4.3.

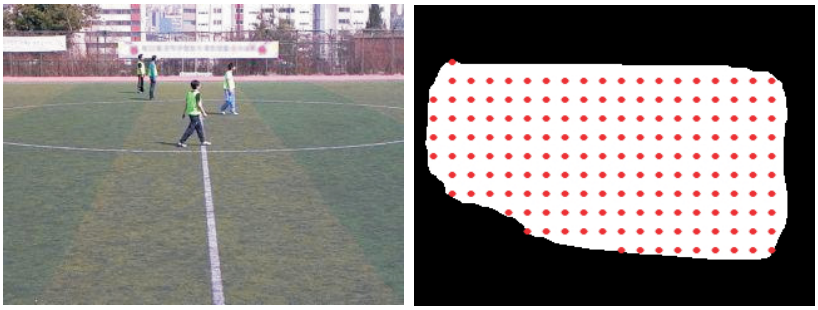
Finally, we adaptively set the valid activity area as a lower part of each object along the ground direction, as shown in Fig. 4.2(c), where h is the object height along the ground direction and the ratio κ is empirically set to 0.25. In Fig. 4.2(d), the

(a) (b)



(c)

Figure 4.3: Ground direction is decided from the angle frequency histogram. (a) The dominant direction of each object is denoted as red lines. (b) The ground direction of the 'Soccer' sequences is determined as denoted by a red line. (c) The histogram of the angles of the dominant directions.



(a) (b)

Figure 4.4: Selection of reliable pixel positions. (a) The 'Soccer' sequence. (b) Reliable pixel positions are selected on the regular grid in the white region, in which there are activities.

resultant adaptive activity areas are depicted by red regions, whereas the initially detected foreground objects consist of grey as well as red regions.

4.3.2 Selection of Consistent Pixel Positions

In order to select pixel positions for correspondence matching, we first sample initial positions on the regular grid in $\mathcal{I}$. However, we may not be able to find reliable matching for some of the pixel positions due to the lack of activity information. For instance, when $\mathbf{p}$ is located within the static background at which no foreground object appears through the entire video sequence, the activity vector $A(\mathbf{p})$ is a trivial zero sequence. In such a case, even if the identical vector $A(\mathbf{q})$ is found at $\mathbf{q}$, it is also a zero sequence. Thus, $A(\mathbf{p})$ and $A(\mathbf{q})$ yield $K_{00} = T$ and $K_{01} = K_{10} = K_{11} = 0$, and the MIBS measure becomes $s(\mathbf{p}, \mathbf{q}) = I(X;Y) = 0$ according to (3.10). Moreover, there might be multiple $\mathbf{q}$'s in the static background region, which have the zero sequence as the activity vectors, causing ambiguity in the correspondence matching. Therefore, we choose only the pixel positions with some amount of activities. Fig. 4.4

shows selected pixel positions for the 'Soccer' sequence. The white region represents the set of pixels at which the numbers of active elements in $A(\mathbf{p})$'s are larger than 1% of the total frame length T, and the red dots show regularly sampled pixel positions within the white region.

Then we refine the initial pixel positions by performing the iterative bidirectional matching to obtain consistent pixel positions. We say that a pair of pixel positions $\mathbf{p} \in \mathcal{I}$ and $\mathbf{q} \in \mathcal{J}$ are consistent, when the corresponding pixel of $\mathbf{p}$ is $\mathbf{q}$ and the corresponding pixel of $\mathbf{q}$ is $\mathbf{p}$. Note that a pair of consistent pixel positions are highly probable to be the true corresponding points. Let us first define the forward matching function from pixel $\mathbf{p}$ in $\mathcal{I}$ to $\mathcal{J}$ as

$$f_{\mathcal{I}\to\mathcal{J}}(\mathbf{p}) = \arg\max_{\mathbf{q}\in\mathcal{J}} s(\mathbf{p},\mathbf{q}). \tag{4.2}$$

Similarly, let us define the backward matching function from pixel $\mathbf{q}$ in $\mathcal{J}$ to $\mathcal{I}$ as

$$f_{\mathcal{J}\to\mathcal{I}}(\mathbf{q}) = \arg\max_{\mathbf{p}\in\mathcal{I}} s(\mathbf{q},\mathbf{p}). \tag{4.3}$$

Given an initial pixel $\mathbf{p}^{(0)}$, the bidirectional matching process performs the forward matching and the backward matching iteratively. Specifically, for a given pixel $\mathbf{p}^{(k)}$ in $\mathcal{I}$, we first find its corresponding pixel $\mathbf{q}^{(k)} = f_{\mathcal{I}\to\mathcal{J}}(\mathbf{p}^{(k)})$ in $\mathcal{J}$ by the forward matching. Then, by the backward matching of $\mathbf{q}^{(k)}$, we find $\mathbf{p}^{(k+1)} = f_{\mathcal{J}\to\mathcal{I}}(\mathbf{q}^{(k)})$ in $\mathcal{I}$. If $\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)}$, we terminate the bidirectional matching and declare the pair $(\mathbf{p}^{(k)}, \mathbf{q}^{(k)})$ as consistent. On the contrary, when $\mathbf{p}^{(k+1)} \neq \mathbf{p}^{(k)}$, we perform the bidirectional matching again with $\mathbf{p}^{(k+1)}$, and check whether $\mathbf{p}^{(k+2)} = \mathbf{p}^{(k+1)}$. This is iteratively repeated until $\mathbf{p}^{(k)}$ converges to a consistent pixel position. In addition, to obtain a balanced distribution of consistent pixel positions, we exclude $\mathbf{p}^{(k)}$ from the set of consistent pixel positions when the distance between $\mathbf{p}^{(k)}$ and $\mathbf{p}^{(0)}$ becomes

Figure 4.5: The bidirectional matching for selecting consistent pixel positions. $\mathbf{p}^{(0)}$ in the left frame is an initial pixel position, which is matched to $\mathbf{q}^{(0)}$ in the right frame by the forward matching. Then, the backward mapping matches $\mathbf{q}^{(0)}$ to $\mathbf{p}^{(1)}$. Finally, $\mathbf{p}^{(1)}$ and $\mathbf{q}^{(0)}$ are consistent.

larger than the half of the grid interval. Moreover, to prevent the exceptional cases of infinite loop, we also exclude the pixel positions that do not converge within 10 iterations.

Fig. 4.5 illustrates how the bidirectional matching finds consistent pixel positions in the 'ArtCollege' sequence. An initial pixel position $\mathbf{p}^{(0)}$ is depicted by the red dot in the left frame, and its corresponding pixel position $\mathbf{q}^{(0)}$ is found in the right frame. However, we see that $\mathbf{p}^{(0)}$ and $\mathbf{q}^{(0)}$ are not the projections of the same scene point in the real world. Thus, $\mathbf{p}^{(1)}$ is found again by the backward matching, which is in turn matched to $\mathbf{q}^{(0)}$ by the forward matching. Therefore, $\mathbf{p}^{(1)}$ and $\mathbf{q}^{(0)}$ are declared as a pair of consistent pixel positions.

4.3.3 MRF-Based Optimization

We optimize the correspondence matching results for multiple consistent pixel positions based on the MRF [38]. By considering the relationship between neighboring pixel positions, the MRF optimization refines the correspondence matching result of each pixel position and improves the matching accuracy. In Fig. 4.6, red dots show the consistent pixel positions in one view of the 'Soccer' sequence and yellow lines represent the neighboring pairs of consistent pixel positions in the MRF structure. Let $\mathcal{C}$ denote the set of consistent pixel positions in $\mathcal{I}$, and $\mathcal{N}$ denote the set of pairs of neighboring positions in $\mathcal{C}$. We refine the matching results by minimizing the cost function

$$E(\mathbf{Q}) = \sum_{\mathbf{p}_i \in \mathcal{C}} D_{\mathbf{p}_i}(\mathbf{q}_i) + \sum_{(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{N}} V_{\mathbf{p}_i, \mathbf{p}_j}(\mathbf{q}_i, \mathbf{q}_j), \tag{4.4}$$

where $\mathbf{q}_i$ denotes the matching pixel in $\mathcal{J}$, which corresponds to a consistent pixel $\mathbf{p}_i$ in $\mathcal{C}$, and $\mathbf{Q}$ is the set of all $\mathbf{q}_i$'s. Also, $D_{\mathbf{p}_i}(\mathbf{q}_i)$ is the data cost, which measures the dissimilarity between $\mathbf{p}_i$ and $\mathbf{q}_i$, and $V_{\mathbf{p}_i, \mathbf{p}_j}(\mathbf{q}_i, \mathbf{q}_j)$ is the smoothness cost, which measures the discontinuity between the matching results of the neighboring pixels.

For accurate matching, the design of the data cost is the most fundamental part in the MRF framework [12]. We define the data cost $D_{\mathbf{p}_i}(\mathbf{q}_i)$ using the MIBS measure:

$$D_{\mathbf{p}_i}(\mathbf{q}_i) \quad = \quad \frac{s_{\max}(\mathbf{p}_i) - s(\mathbf{p}_i, \mathbf{q}_i)}{s_{\max}(\mathbf{p}_i) - s_{\min}(\mathbf{p}_i)}, \tag{4.5}$$

where $s_{\max}(\mathbf{p}_i)$ and $s_{\min}(\mathbf{p}_i)$ are the maximum and the minimum values of $s(\mathbf{p}_i, \mathbf{q}_i)$ over all candidate matching positions in $\mathcal{J}$; $s_{\max}(\mathbf{p}_i) = \max_{\mathbf{q} \in \mathcal{J}} s(\mathbf{p}_i, \mathbf{q})$ and $s_{\min}(\mathbf{p}_i) = \min_{\mathbf{q} \in \mathcal{J}} s(\mathbf{p}_i, \mathbf{q})$. Hence $D_{\mathbf{p}_i}(\mathbf{q}_i)$ is normalized into the range of $[0, 1]$. Also, $D_{\mathbf{p}_i}(\mathbf{q}_i)$ is inverse proportional to $s(\mathbf{p}_i, \mathbf{q}_i)$, and thus yields a small cost when $\mathbf{p}_i$ corresponds to $\mathbf{q}_i$.

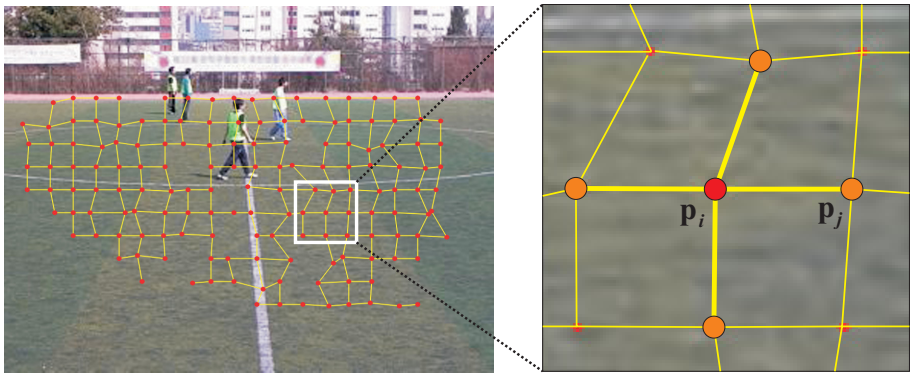


Figure 4.6: Red dots depict the consistent pixel positions, and yellow lines represent the pairs of neighboring positions in the MRF optimization.

The smoothness cost $V_{\mathbf{p}_i,\mathbf{p}_j}(\mathbf{q}_i, \mathbf{q}_j)$ assumes that the pixels $\mathbf{q}_i$ and $\mathbf{q}_j$ in $\mathcal{J}$, corresponding to the neighboring pixels $\mathbf{p}_i$ and $\mathbf{p}_j$ in $\mathcal{I}$, should be also spatially adjacent. We may employ the difference between the displacement vectors, $(\mathbf{q}_i - \mathbf{p}_i)$ and $(\mathbf{q}_j - \mathbf{p}_j)$, to define the smoothness cost. However, when the two cameras have very different configurations, the displacement vectors may not be similar. In order to compensate for this discrepancy in the camera configurations, we first estimate a coarse homography H from $\mathcal{I}$ to $\mathcal{J}$ by using the pairs of consistent pixels, based on the simple outlier elimination technique, RANSAC [39]. Then we compute the displacement vectors $\mathbf{v}_{\mathbf{p}_i}$ and $\mathbf{v}_{\mathbf{p}_j}$, by taking $H\mathbf{p}_i$ and $H\mathbf{p}_j$ instead of $\mathbf{p}_i$ and $\mathbf{p}_j$, which are given by

$$\begin{aligned} \mathbf{v}_{\mathbf{p}_i} &= \mathbf{q}_i - H\mathbf{p}_i, \\ \mathbf{v}_{\mathbf{p}_j} &= \mathbf{q}_j - H\mathbf{p}_j. \end{aligned} \tag{4.6}$$

The smoothness cost is then defined as the difference between $\mathbf{v}_{\mathbf{p}_i}$ and $\mathbf{v}_{\mathbf{p}_j}$. More specifically, we employ the truncated quadratic cost function, which is widely used

in many energy minimization tasks [38], to define the cost

$$V_{\mathbf{p}_i,\mathbf{p}_j}(\mathbf{q}_i,\mathbf{q}_j) = \beta \min(\|\mathbf{v}_{\mathbf{p}_i} - \mathbf{v}_{\mathbf{p}_j}\|^2, V_{\max}), \tag{4.7}$$

where $V_{\max}$ is a truncation factor and β is a weighting coefficient, which are empirically set to 25 and 0.7, respectively, in all experiments. We obtain the solution of the energy minimization problem in (4.4) using the graphcut algorithm [38,40], since it provides a near-optimal solution efficiently [41]. We use the α-β-swap algorithm rather than the α-expansion algorithm, since the smoothness cost $V_{\mathbf{p}_i,\mathbf{p}_j}(\mathbf{q}_i,\mathbf{q}_j)$ is semi-metric. And it can be optimized by various other discrete MRF solvers, such as high order max flow [42], belief propagation [43] or tree-reweighting message passing(TRW) [44, 45].

4.3.4 Additional Color Information

While the MIBS measure for activity in the previous chapter efficiently finds the correspondence matching for most of the video sequences, it suffers from the ambiguity to find the true correspondence when one image contains multiple foreground objects as shown in Fig. 4.7. For a given active pixel $\mathbf{p}$ in I_t^B, its false correspondences and the true one in J_t^B are denoted by the blue and red arrows, respectively. In general, this ambiguity may be alleviated by comparing the activity vectors through all the frames, however, it still yields a drawback when the temporal activity characteristics of multiple moving objects are also similar. Thus this ambiguity causes an error to count K_{ij} and degrades the performance of the activity based correspondence matching.

Therefore, we employ a color similarity measure together with the activity similarity measure, assuming that a same object yields almost a same color in the images

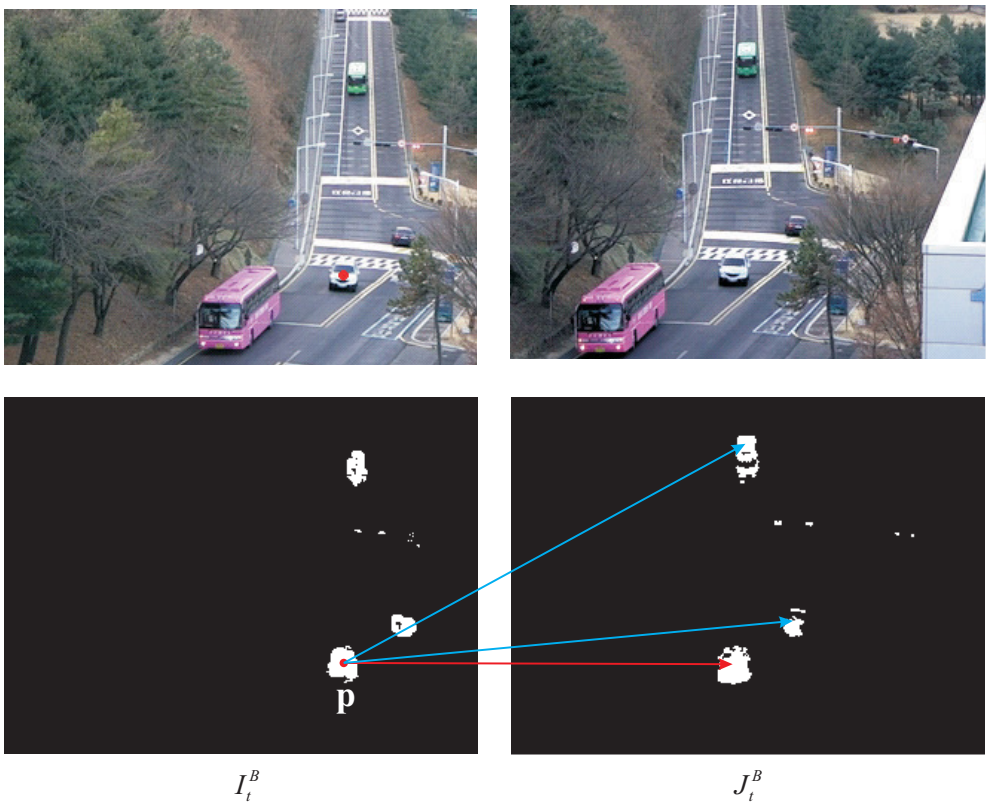


Figure 4.7: The uncorrelative activities: the blue arrows represent the objects that definitely not include the position corresponding to the **p**. The red arrow indicates the correlative object that **p** is located on.

of different views. Let $I_t(\mathbf{p})$ and $J_t(\mathbf{q})$ denote the 3-tuple vectors of red, green and blue color values at the pixels $\mathbf{p} \in I_t$ and $\mathbf{q} \in J_t$, respectively. If the number of frames, at which $I_t(\mathbf{p})$ and $J_t(\mathbf{q})$ are similar each other, is larger, then **p** and **q** are more probable to be corresponding each other. Hence we first count the number of frames yielding the similar color values between two active pixels **p** and **q** as

$$C(\mathbf{p},\mathbf{q}) = \left|\left\{t \mid \|I_t(\mathbf{p}) - J_t(\mathbf{q})\| < \sigma \text{ and } I_t^B(\mathbf{p}) = J_t^B(\mathbf{q}) = 1\right\}\right|, \tag{4.8}$$

where. $\|\cdot\|$ denotes the Euclidean distance between the color vectors, and σ is the

threshold of the color similarity which is fixed to 30 in our experiments. Since $C(\mathbf{p}, \mathbf{q})$ is meaningful only when both $\mathbf{p}$ and $\mathbf{q}$ are active simultaneously, we normalize $C(\mathbf{p}, \mathbf{q})$ by K_{11} and define a color similarity measure as follows.

$$s_{\text{color}}(\mathbf{p}, \mathbf{q}) = \begin{cases} 0, & N_{11}(\mathbf{p}, \mathbf{q}) = 0, \\ C(\mathbf{p}, \mathbf{q})/K_{11}(\mathbf{p}, \mathbf{q}), & \text{otherwise.} \end{cases} \tag{4.9}$$

Note that $\mathbf{p}$ and $\mathbf{q}$ are highly probable to be corresponding each other when $s_{color}(\mathbf{p}, \mathbf{q})$ is larger, and $0 \leq s_{\text{color}}(\mathbf{p}, \mathbf{q}) \leq 1$.

In the sequel, we consider the similarity measure for activity in (3.11) and the similarity measure for color in (4.9) together, to define a total similarity measure s_{total}.

$$s_{\text{total}}(\mathbf{p}, \mathbf{q}) = (1 - \alpha) s_{\text{MIBS}}(\mathbf{p}, \mathbf{q}) + \alpha s_{\text{color}}(\mathbf{p}, \mathbf{q}), \tag{4.10}$$

where s_{MIBS} is the MIBS measure and α is a weighting parameter and empirically set to a constant belonging to the range $[0, 1]$. The weighting factor α is constant for any $\mathbf{p}$ and $\mathbf{q}$, but it depends on the input video sequences $\mathcal{I}$ and $\mathcal{J}$ in order to take account of the variances of the video sequence containing the colorful objects.

In order to test the effect of the color similarity, we find the corresponding position $\mathbf{q}$ of the given reliable site $\mathbf{p}$ by the following criterion

$$\mathbf{q}^* = \arg \max_{\mathbf{q} \in \mathcal{J}} s_{\text{total}}(\mathbf{p}, \mathbf{q}). \tag{4.11}$$

Note that, total similarity s_{total} provides both activity information and color information for finding the correct correspondence.

Fig. 4.8 demonstrates the comparison results on the 'Parkinglot' sequence whose the ground truth correspondence is available. For the reliable sites in Fig. 4.8 (a), Fig. 4.8 (b) shows the corresponding positions by considering the color similarity

(a)

(b)

(c)

Figure 4.8: (a) Consistent pixel positions in $\mathcal{I}$. (b) Matching result without the color similarity measure. (c) Matching results with the color similarity.

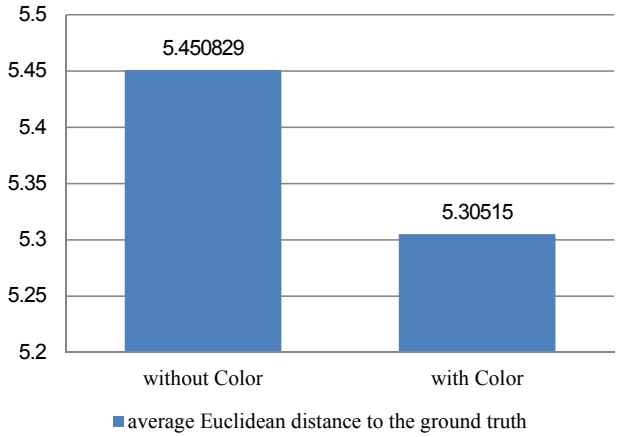


Figure 4.9: Average errors of correspondence matching according to the color similarity measure.

cost, which α is set to 0.42 in Eq. (4.11). For comparison, Fig. 4.8 (c) illustrates the corresponding positions of the identical reliable sites using the criterion without the color similarity cost by setting α to zero in Eq. (4.11). The average Euclidean distance to the ground truth diminishes when the color similarity cost is considered as shown in Fig. 4.8 (d).

4.4 Experimental results

4.4.1 Performance Evaluation of Adaptive Activity Area

In Fig. 4.10, we compare the correspondence matching errors, which are obtained with adaptive activity areas, rectangular bottom areas [2], and entire object areas. Each matching error is the average Euclidean distance between computed matching pixels and the ground truth ones, which are manually obtained. Both the adaptive activity areas and the rectangular bottom areas reduce the matching errors signifi-

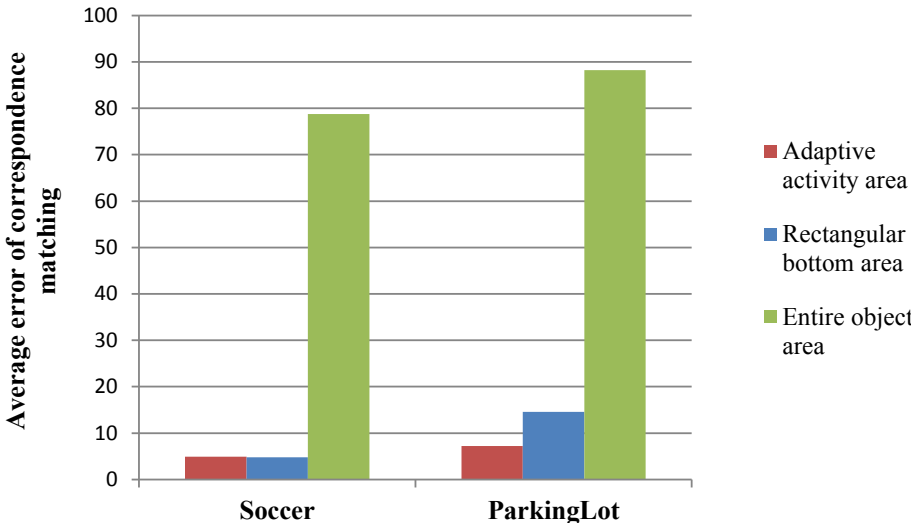


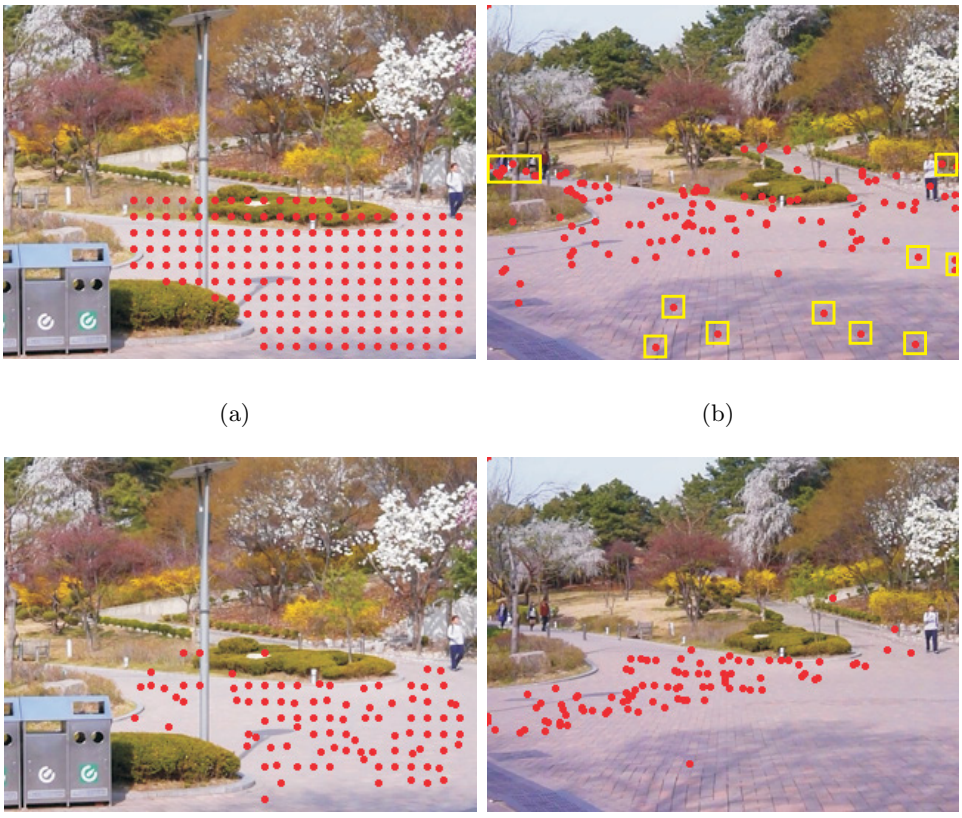
Figure 4.10: Comparison of the correspondence matching errors, which are obtained with the proposed adaptive activity areas, the rectangular bottom areas [2], and the entire object areas. The average Euclidean distance is measured between computed matching pixels and the manually obtained ground truth ones.

cantly as compared with the entire object areas. The adaptive activity areas yield a comparable result to the rectangular bottom areas on the 'Soccer' sequence, which has the vertical ground direction. However the proposed adaptive areas provide a smaller matching error than the rectangular bottom areas on the 'ParkingLot' sequence, where the ground direction forms an oblique angle with the horizontal line.

4.4.2 MRF Optimization with Consistent Pixel Positions

The next experiment demonstrates how the iterative bidirectional matching in the previous section improves the matching performance. In other words, we compare the matching performance of initially sampled pixel positions (IP's) and consistent pixel positions (CP's). Figs. 4.11(a) and (b) show IP's in $\mathcal{I}$ and their matching positions in $\mathcal{J}$ on the 'Crossroad' sequence. Outlier pixels with incorrect matching results are marked by yellow boxes. Fig. 4.11(c) shows the CP's, which are refined from the IP's by employing the bidirectional matching, and Fig. 4.11(d) shows their matching positions, which yield more reliable and consistent results. In Fig. 4.13, we also quantitatively compare the matching performance by measuring the average transform errors, $\|\mathbf{q} - H\mathbf{p}\|$, where H is the homography transform between the two views obtained with IP's or CP's, respectively. An inlier ratio is defined as the number of matching pairs in RANSAC [39] to the total number of pairs. We see that CP's always provide smaller transform errors than IP's, regardless of inlier ratios.

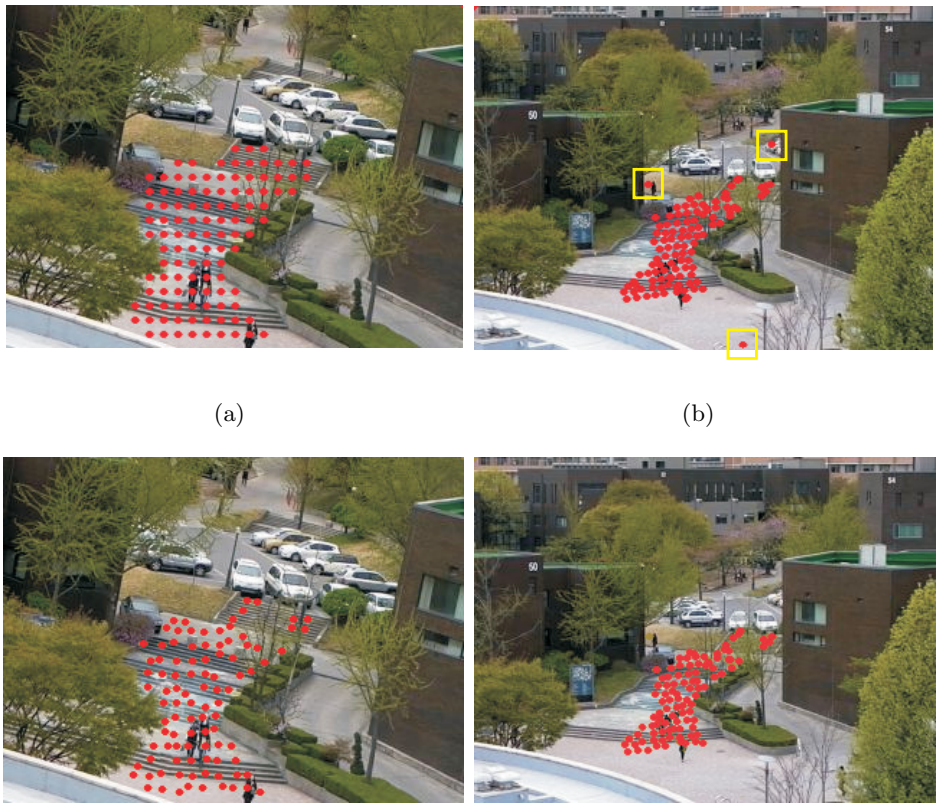
Fig. 4.14∼4.16 evaluates the performance of the MRF optimization. The left-most column shows consistent pixel positions in one view. The second and the third columns show the matching results in the other view, which are obtained by the proposed correspondence matching algorithm with and without the MRF optimization,



(a) (b)

(c) (d)

Figure 4.11: Matching results with initial pixel positions (IP's) and consistent pixel positions (CP's) on the 'Crossroad' sequence. (a) IP's and (b) their matching positions, where outlier pixels are marked by yellow boxes. (c) CP's and (d) their matching positions.



(a) (b)

(c) (d)

Figure 4.12: Matching results with initial pixel positions (IP's) and consistent pixel positions (CP's) on the 'ArtCollege' sequence. (a) IP's and (b) their matching positions, where outlier pixels are marked by yellow boxes. (c) CP's and (d) their matching positions.

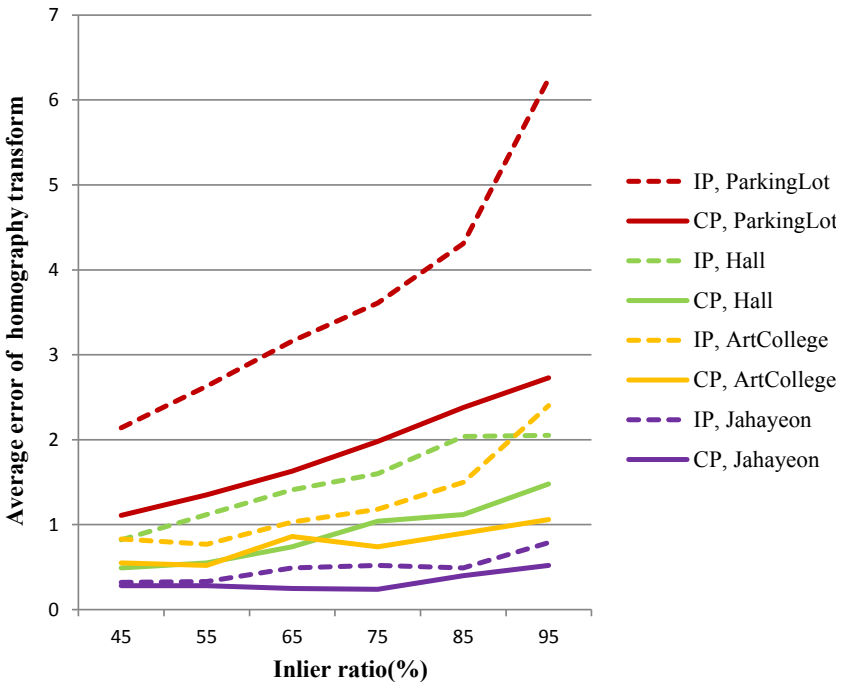


Figure 4.13: Average errors of the homography transforms computed with IP's and CP's, respectively, in terms of the inlier ratios.

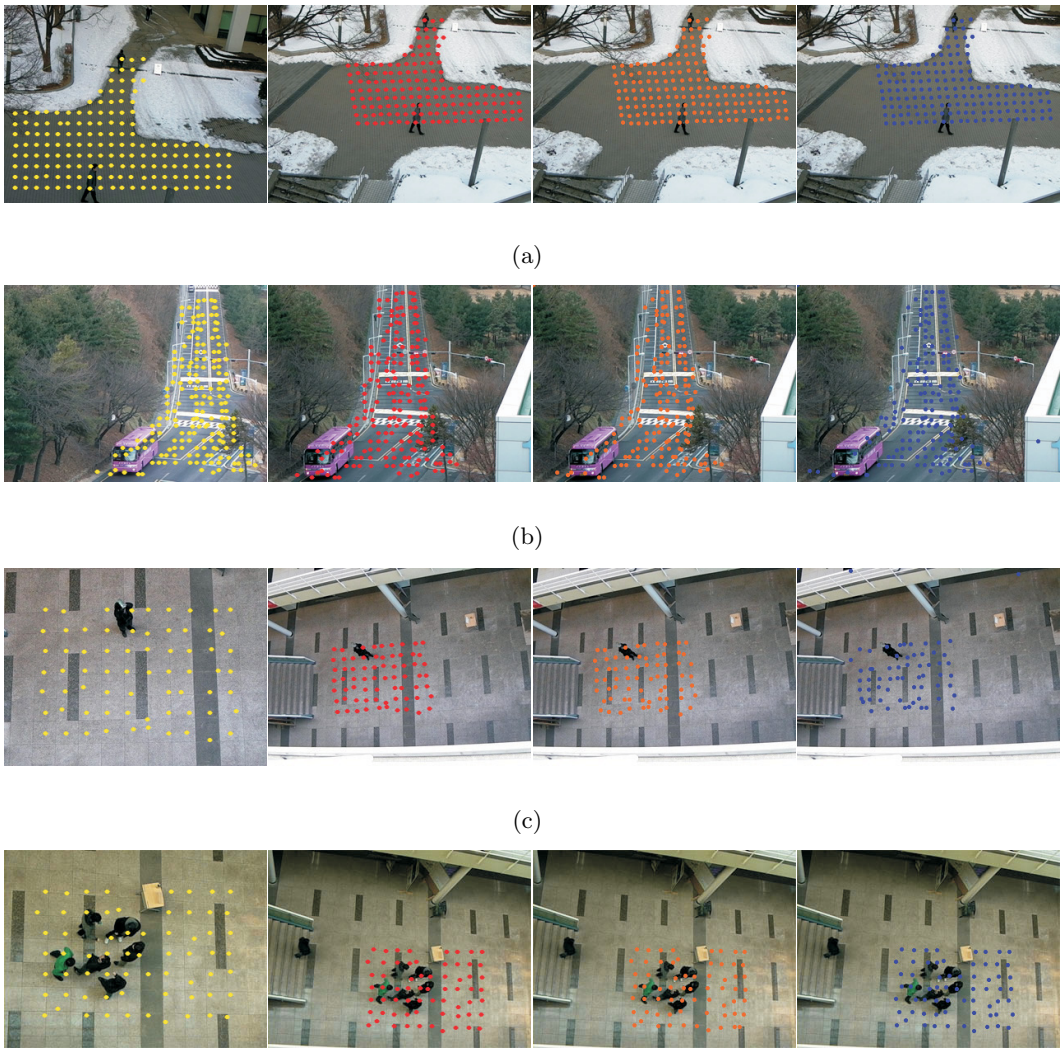
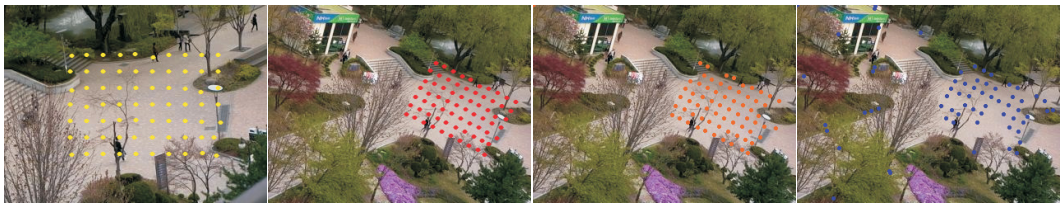


Figure 4.14: Matching results on the (a) 'Library,' (b) 'Road,' (c) 'Hall,' (d) 'Desk,' (e) 'ArtCollege' sequences.The leftmost column shows consistent pixel positions in one view. The second and third columns show the matching positions in the other view, obtained by the proposed algorithm with and without the MRF based optimization, respectively. The last column shows the matching results by the conventional algorithm [2].

(a)

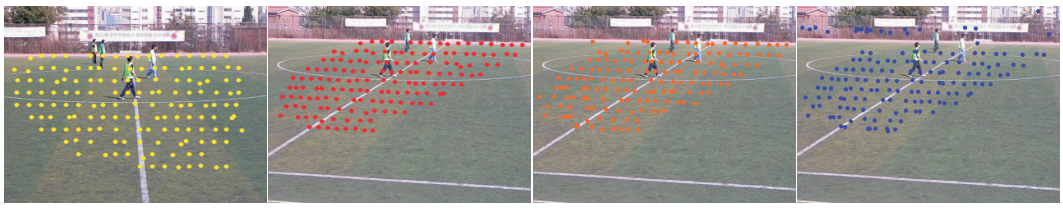


(b)

(c)

Figure 4.15: Matching results on the (a) 'ArtCollege,' (b) 'Jahayeon,' (c) 'ParkingLot' sequences. The leftmost column shows consistent pixel positions in one view. The second and third columns show the matching positions in the other view, obtained by the proposed algorithm with and without the MRF based optimization, respectively. The last column shows the matching results by the conventional algorithm [2].

(a)



(b)

(c)

Figure 4.16: Matching results on the (a) 'Stair,' (b) 'Soccer,' and (c) 'Crossroad' sequences. The leftmost column shows consistent pixel positions in one view. The second and third columns show the matching positions in the other view, obtained by the proposed algorithm with and without the MRF based optimization, respectively. The last column shows the matching results by the conventional algorithm [2].

respectively. We see that the MRF optimization provides more reliable matching results by enforcing the smoothness constraint between neighboring pixels, especially on the 'Hall,' 'Desk,' 'ArtCollege,' and 'Jahayeon' sequences in Fig. 4.14(c) and (d), Fig. 4.15(a) and (b).

In addition, Fig. 4.14∼4.16 also compares the performance of the proposed algorithm with that of the conventional activity-based matching algorithm [2]. In both algorithms, we apply the same foreground detection method in [36] to generate binary activity maps and use the same set of consistent pixels to determine the correspondence matching. Note that the conventional algorithm uses rectangular bottom areas of object regions to derive activity vectors and does not refine the matching results using the MRF optimization. The last column in Fig. 4.14∼4.16 shows the results of the conventional algorithm. Compared with the proposed algorithm in the second column, the conventional algorithm yields a larger number of incorrect matching positions. In particular, whereas the proposed algorithm is robust to the rotational transformations between cameras, the conventional algorithm provides severely erroneous matching results on the 'Jahayeon,' 'ParkingLot,' and 'Stair' sequences in Fig. 4.15(b) and (c) and Fig. 4.16(a). It is because the conventional algorithm always uses bottom areas of foreground objects to form activity vectors and generates unreliable activity information under severely rotated cameras. Moreover, due to sudden illumination changes and shadows, many errors occur in the background subtraction, especially on outdoor scenes with large fast moving objects, such as the 'Soccer' and 'Crossroad' sequences in Fig. 4.16(b and (c). In such cases, the conventional algorithm provides poor matching results or simply fails. On the contrary, the proposed algorithm is very robust to the noise in activity vectors and therefore yields faithful correspondence matching results on

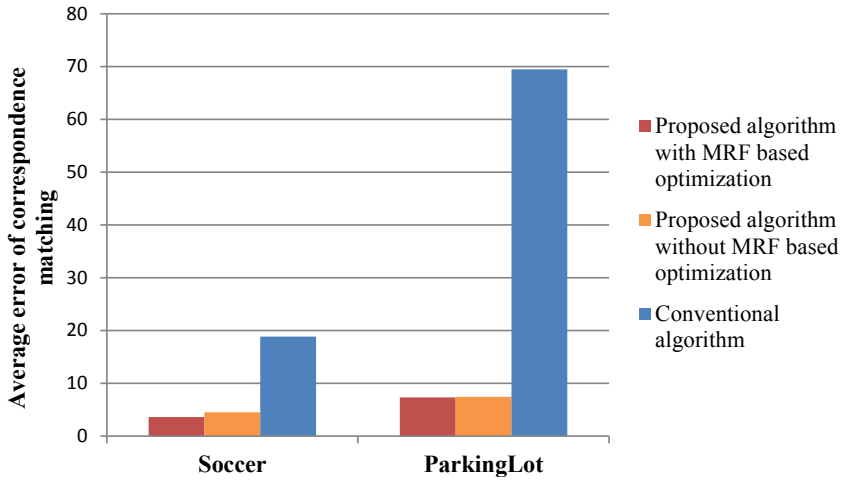


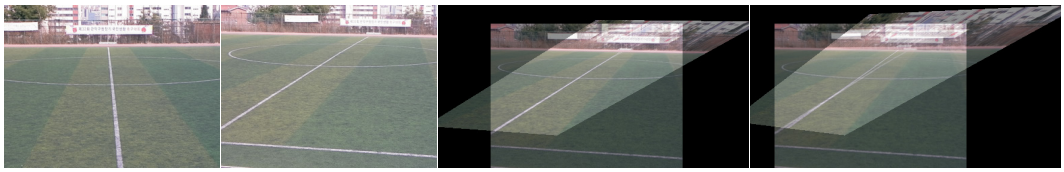
Figure 4.17: Quantitative comparison of the correspondence matching performance of the proposed algorithm with that of the conventional algorithm [2]. The average Euclidean distances between computed matching positions and the ground truth ones are measured.

these challenging sequences.

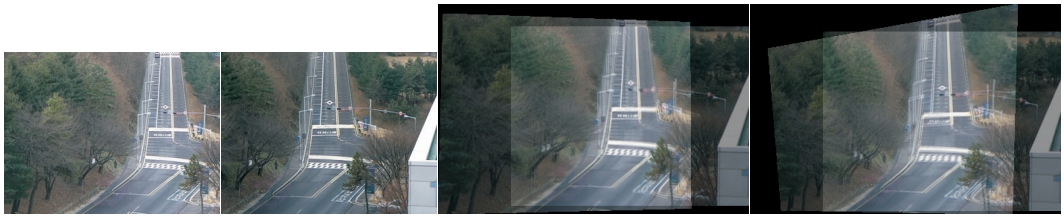
Fig. 4.17 assesses the matching performance on the 'Soccer' and 'ParkingLot' sequences quantitatively. The proposed algorithm yields much smaller matching errors than the conventional algorithm [2]. More specifically, the matching errors of the proposed algorithm are 5.2 times smaller on the 'Soccer' sequence and 9.5 times smaller on the 'ParkingLot' sequence, respectively, than those of the conventional algorithm. Note that the MRF optimization further reduces the matching errors.

4.4.3 Application to Panoramic View Synthesis

As an exemplar application of the proposed correspondence matching algorithm, let us synthesize a panoramic view from two different views. We first estimate the homography between two views using multiple pairs of corresponding pixels,



(a)



(b)

(c)

Figure 4.18: Panoramic view synthesis for the (a) 'Soccer,' (b) 'Road,' and (c) 'Jahayeon' sequences. The first and second columns represent the two views, respectively. The third and fourth columns show the resultant panoramic views obtained by the proposed algorithm and the conventional algorithm [2], respectively.

(a)

fail

(b)

Figure 4.19: Panoramic view synthesis for the (a) 'Desk' and (b) 'Stair' sequences. The first and second columns represent the two views, respectively. The third and fourth columns show the resultant panoramic views obtained by the proposed algorithm and the conventional algorithm [2], respectively.

and then project one view onto the other view. Fig. 4.18 and Fig. 4.19 shows the synthesized panoramic views. The first and the second columns present the two views in $\mathcal{I}$ and $\mathcal{J}$, respectively. The third and the fourth columns are the panoramic views, obtained by the proposed correspondence matching algorithm and the conventional algorithm [2], respectively. To clearly visualize the accuracy of an estimated homography, we multiply each color channel of the two views by 0.5 and then add the transformed view of $\mathcal{I}$ to the view of $\mathcal{J}$. The overlapped regions are hence brighter than the other regions, which are observed from only one of the two cameras. In general, when the correspondence matching is not accurate, the synthesized panoramic view tends to be blurred. We see that the proposed algorithm faithfully blends two different views without severe blurring artifacts. However, the conventional algorithm causes noticeable misalignment between the white center lines in the 'Soccer' sequence and blurs the lanes and crosswalk in the 'Road' sequence, as shown in Fig. 4.18(a) and (b). Moreover, the conventional algorithm suffers from severe ghost artifacts around the trees in the 'Jahayeon' sequence and blurred stripe patterns on the floor in the 'Desk' sequence, as shown in Fig. 4.18(c) and Fig. 4.18(a). The panoramic view synthesis is challenging for the 'Stair' sequence, since there is the camera rotation of almost 180 degrees as shown in Fig. 4.19(c). Whereas the conventional algorithm fails to estimate the homography for the 'Stair' sequence, the proposed algorithm successfully estimates the homography and provides a relatively accurate panoramic view.

4.5 Conclusion

In this chapter, we proposed a framework of finding correspondence between multiple video sequences. We employed the MIBS measure to describe the similarity between position pairs. To enhance the performance of the MIBS measure which is based on the activity information of moving foreground objects, we employed the adaptive activity areas representing actual bottom areas of the objects touching the ground planes. Also, the proposed algorithm selected the consistent pixel positions using the iterative bidirectional matching to evaluate the reliable matching pairs between two views. Moreover, we optimized the correspondence matching results for multiple pixel positions by minimizing the cost function based on the MRF framework. Experimental results demonstrated that the proposed algorithm finds the correspondence matching between two different views more accurately and reliably than the conventional state of the art method. Therefore, the proposed algorithm is a very promising technique for various multi-view video applications such as visual surveillance and panoramic view generation.

Chapter 5

Conclusions

In this dissertation, a new similarity measure based on mutual information and a framework of correspondence matching algorithm for multi-view surveillance video sequences were presented.

First, a noble correspondence matching algorithm for multiple video sequences was proposed. We exploited an activity vector for the correspondence matching, which is the temporal occurrence pattern of foreground objects at a specific pixel position. In order to compare two activity vectors efficiently, we considered them as the realization of random variables and measured their similarity using the mutual information of the random variables. Since the mutual information describes the joint and individual behavior of the activity vectors faithfully, the proposed mutual information based similarity (MIBS) measure provides more accurate and reliable matching results than the conventional Hamming distance measure. The experimental results with additive noises also demonstrated the MIBS measure is robust to additive random noises. Furthermore, we showed that the MIBS measure outperformed other similarity measures for binary vectors.

Next, the framework for finding a dense correspondence matching between multi-view video sequences was proposed. In order to find reliable and accurate matching pairs, three practical techniques was developed. The proposed algorithm refines moving foreground areas into adaptive activity areas coinciding with actual regions on the ground, which are touched by moving objects. And we suggested consistent pixel positions where the MIBS measure can be reliably evaluated. Finally, the proposed algorithm utilize the MRF optimization to refine the matching results of multiple source pixel positions by minimizing a matching cost function based on the Markov random field. The experimental results demonstrated that the proposed algorithm can estimate the correspondence matching results of consistent pixel positions over the entire view areas faithfully.

The main features of our approach can be summarized in following.

- Robustness to noise: The MIBS measure yields stable correspondence matching results under additive noises as shown in Fig. 3.7∼3.10. Noisy components are easily added in various causes such as CCD heat noises in cameras, foreground detection, and geometric dissimilarity. Compared to other similarity measures, the proposed measure shows more reliable matching performances even when the activity vectors are severely collapsed by random noises.

- Without a supervision: The proposed algorithm does not need the knowledge of camera locations, camera parameters, and illumination conditions. Moreover, the proposed method does not require calibration or rectification between cameras. The only requirement is the timely synchronized multi-view video sequences. Therefore, it is applicable to the challenging situations in which the users hardly access the cameras such as skyscraper or highway surveillances.

The several future works can be considered as

- Additional photometry information for similarity measure: In proposed algorithm, only binary activity information is considered for evaluating a similarity measure. This approach is efficient due to the small computational complexity, however, it disregards a plenty of color or intensity information. As in traditional stereo matching algorithms [10, 12], utilizing photometry information can reduce an ambiguous similarity measurement and improve the matching performance.

- Synthesizing a panoramic view from multiple view videos: In this dissertation, we have shown simple example results of a panoramic view synthesis with two different views. However, for many applications such as surveillance systems and virtual street views, more effective panoramic view generation technique is required [46–49]. The reliable correspondence matching from proposed algorithm is essential to preserve a visual consistency of the panoramic views.

- Object removal and completion of multi-view video sequences: The inpainting technique, which removes the selected objects and fills the area without visual artifacts, has been widely researched [50–54]. Since the correspondence matching can provides a initial seed position of the source blocks, it will be helpful in developing a reliable inpainting algorithm for video sequences.

To summarize, the proposed algorithm presents a new approach to find the correspondence matching for the multi-view video sequences. The proposed algorithm utilizes an activity information to feature the pixel positions of multi-view videos. The proposed MIBS measure efficiently represents the similarity between two activity vectors and outperforms other conventional similarity measures. Then,

the proposed algorithm suggested the framework for finding dense correspondences. Adaptive activity area refinement enhances the performance of activity vectors to represent a ground surface. And the proposed algorithm selects consistent pixel positions where the MIBS measure is reliably evaluated. Finally, the correspondence matching at multiple pixel positions are refined by a MRF-based energy minimization technique. Therefore, it is believed that the proposed correspondence matching algorithm yields reliable matching results enough to be applicable to computer vision and surveillance applications.

Bibliography

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, pp. 91–110, Nov. 2004.

[2] E. B. Ermis, P. Calrot, P.-M. Jodoin, and V. Saligrama, "Activity based matching in distributed camera networks," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2595–2613, Oct. 2010.

[3] J.-W. Kang, S.-H. Cho, N.-H. Hur, C.-S. Kim, and S.-U. Lee, "Graph theoretical optimization of prediction structure in multiview video coding," in *Proc. IEEE ICIP*, Oct. 2007, pp. 429–432.

[4] R.-S. Wang and Y. Wang, "Multiview video sequence analysis, compression, and virtual viewpoint synthesis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 397–410, Apr. 2000.

[5] W. Matusik and H. Pfister, "3D TV: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes," *ACM Trans. Graph.*, vol. 23, pp. 814–824, 2004.

[6] K. Obraczka, R. Manduchi, and J. Garcia-Luna-Aveces, "Managing the information flow in visual sensor networks," in *Proc. Int. Symposium on Wireless Personal Multimedia Communications*, vol. 3, Oct. 2002, pp. 1177–1181.

[7] S. Soro and W. Heinzelman, "A survey of visual sensor networks," *Adv. Multimed.*, vol. 2009, Article ID 640386, pp. 1–21, May 2009.

[8] B. Rinner and W. Wolf, "An introduction to distributed smart cameras," *Proc. IEEE*, vol. 96, no. 10, pp. 1565–1575, Oct. 2008.

[9] M. D. Levine, D. A. O'Handley, and G. M. Yagi, "Computer determination of depth maps," *Comput. Graph. Image Process.*, vol. 2, no. 2, pp. 131–150, Apr. 1973.

[10] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: theory and experiment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 9, pp. 920–932, Sep. 1994.

[11] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, Jun. 2007, pp. 1–8.

[12] Y. S. Heo, K. M. Lee, and S. U. Lee, "Robust stereo matching using adaptive normalized cross-correlation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 807–822, Apr. 2011.

[13] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: speeded up robust features," in *Proc. European Conf. Computer Vision*, Jul. 2006, pp. 404–417.

[14] P. Sand and S. Teller, "Video matching," *ACM Trans. Graph.*, vol. 23, pp. 592–599, Aug. 2004.

[15] G. P. Stein, "Tracking from multiple view points: self-calibration of space and time," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, vol. 1, no. 2, Jun. 1999, pp. 637–642.

[16] L. Lee, R. Romano, and G. Stein, "Monitoring activities from multiple video streams: establishing a common coordinate frame," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 758–767, Aug. 2000.

[17] S. Khan and M. Shah, "Consistent labeling of tracked objects in multiple cameras with overlapping fields of view," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1355–1360, Oct. 2003.

[18] Y. Caspi and M. Irani, "Spatio-temporal alignment of sequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 11, pp. 1409–1424, Nov. 2002.

[19] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, vol. 1, Sep. 1999, pp. 666–673.

[20] S. N. Sinha, M. Pollefeys, and L. McMillan, "Camera network calibration from dynamic silhouettes," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, Jun. 2004, pp. 195–202.

[21] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Addison Wesley, 1993, pp. 191–195.

[22] S. Choi, S. Cha, and C. C. Tappert, "A survey of binary similarity and distance measures," *Journal of Systemics, Cybernetics and Informatics*, vol. 8, no. 1, pp. 43–48, Jan. 2010.

[23] J. D. Tubbs, "A note on binary template matching," *Pattern Recogn.*, vol. 22, no. 4, pp. 359–366, Aug. 1989.

[24] B. Zhang and S. N. Srihari, "Binrary vector dissimilarities for handwriting identification," *Proc. SPIE-IS&T Electronic Imaging*, vol. 5010, pp. 28–38, 2003.

[25] M. R. Anderberg, *Cluster Analysis for Applications.* Academic Press, 1973, pp. 83–89.

[26] P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy.* W. H. Freeman and Company, 1973, pp. 127–140.

[27] T. W. Kurczynski, "Generalized distance and discrete variables," *Biometrics*, vol. 26, no. 3, pp. 525–534, Sep. 1970.

[28] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973, pp. 276–282.

[29] P. Viola and W. M. Wells III, "Alignment by maximization of mutual information," *Int. J. Comput. Vis.*, vol. 24, no. 2, pp. 137–154, Sep. 1997.

[30] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal, "Automated multi-modality image registration based on information theory," in *Proc. Int. Conf. Information Processing in Medical Imaging*, vol. 3, no. 6, 1995, pp. 263–274.

[31] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," *IEEE Trans. Med. Imag.*, vol. 16, no. 2, pp. 187–198, Apr. 1997.

[32] C. Studholme, D. L. G. Hill, and D. J. Hawkes, "An overlap invariant entropy measure of 3D medical image alignment," *Pattern Recognit.*, vol. 32, no. 1, pp. 71–86, Oct. 1999.

[33] J. Kim, V. Kolmogorov, and R. Zabih, "Visual correspondence using energy minimization and mutual information," in *Proc. IEEE Int. Conf. Computer Vision*, Oct. 2003, pp. 1033–1040.

[34] Y. S. Heo, K. M. Lee, and S. U. Lee, "Mutual information-based stereo matching combined with sift descriptor in log-chromaticity color space," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, Jun. 2009, pp. 445–452.

[35] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley press, 1991.

[36] J. M. McHugh, J. Konrad, V. Saligrama, and P. M. Jodoin, "Foreground-adaptive background subtraction," *IEEE Signal Process. Lett.*, vol. 16, no. 5, pp. 390–393, May 2009.

[37] G. Strang, *Introduction to Linear Algebra*. Wellesley-Cambridge press, 2003, pp. 330–337.

[38] Y. Boykov, O.Veksler, and R.Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[39] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *ACM Trans. Commun.*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[40] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

[41] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields with smoothness-based priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 1068–1080, Jun. 2008.

[42] H.-Y. Jung, K.-M. Lee, and S.-U. Lee, "Stereo reconstruction using high order likelihood," in *Proc. IEEE Int. Conf. Computer Vision*, Nov. 2011, pp. 1211–1218.

[43] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *Int. J. Comput. Vis.*, vol. 70, no. 1, pp. 41–54, Oct. 2006.

[44] M. J. Wainwright, T. S. Jaakkola, and A. Willsky, "MAP estimation via agreement on trees: Message-passing and linear programming," *IEEE Trans. Information Theory*, vol. 51, no. 11, pp. 3697–3717, Nov. 2005.

[45] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1568–1583, Oct. 2006.

[46] A. Redert, E. Hendriks, and J. Biemond, "3-D scene reconstruction with viewpoint adaptation on stereo displays," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 4, pp. 550–562, Apr. 2000.

[47] S. Peleg, M. Ben-Ezra, and Y. Pritch, "Omnistereo: panoramic stereo imaging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 279–290, Mar. 2001.

[48] S. Tzavidas and A. K. Katsaggelos, "A multicamera setup for generating stereo panoramic video," *IEEE Trans. Multimed.*, vol. 7, no. 5, pp. 880–890, May 2005.

[49] Y. Wang, R. R. Schultz, and R. A. Fevig, "Panorama recovery from noisy UAV surveillance video," in *Proc. IEEE ICASSP*, Apr. 2009, pp. 1285–1288.

[50] K. A. Patwardhan, G. Sapiro, and M. Bertalmío, "Video inpainting under constrained camera motion," *IEEE Trans. Image Process.*, vol. 16, pp. 545–553, May 2007.

[51] T. Shiratori, Y. Matsushita, X. Tang, and S. B. Kang, "Video completion by motion field transfer," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, Jun. 2006, pp. 411–418.

[52] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, Mar. 2007.

[53] J. Jia, Y.-W. Tai, T.-P. Wu, and C.-K. Tang, "Video repairing under variable illumination using cyclic motions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 832–839, 2006.

[54] S.-Y. Lee, J.-H. Heu, C.-S. Kim, and S.-U. Lee, "Object removal and inpainting in multi-view video sequences," *Int. J. Inovative Comput. Inf. Control*, vol. 6, no. 3(B), pp. 1241–1255, Mar. 2010.