Panoramic Appearance-Based Recognition of Video Contents Using Matching Graphs

Chu-Song Chen, Wen-Teng Hsieh, and Jiun-Hung Chen

Abstract—**This paper proposes a general scheme for recognizing the contents of a video using a set of panoramas recorded in a database. In essence, a panorama inherently records the appearances of an omni-directional scene from its central point to arbitrary viewing directions and, thus, can serve as a compact representation of an environment. In particular, this paper emphasizes the use of a sequence of successive frames in a video taken with a video camera, instead of a single frame, for visual recognition. The associated recognition task is formulated as a shortest-path searching problem, and a dynamic-programming technique is used to solve it. Experimental results show that our method can effectively recognize a video.**

Index Terms—**Appearance-based recognition, computer vision, image understanding, panorama, shortest path, video recognition.**

I. Introduction

VISUAL recognition is a central issue in researches on computer/robot vision and image understanding. In this paper, a systematic method is proposed for recognizing scenes captured with a video camera. More precisely, given a set of successive image frames from a video, the recognition task aims to perceive the scenes contained in these frames by generating high-level descriptions pertaining to the scenes. Although this recognition task can be achieved with a single image frame, the visual ambiguity tends to be more critical with the use of only a single image frame when more scenes have been recorded in the database. In this work, we particularly emphasize the use of a sequence of successive frames, instead of a single frame, for visual recognition. Such a recognition task is formulated as a shortest-path searching problem in our work, which can be effectively solved with standard algorithms in graph theory.

To recognize video contents, a database recording the appearances of the scenes must first be constructed, and in this work, panoramas were used to construct the database. A panorama is a type of static image with particularly successful applications for image-based virtual reality or telepresence [10], [15]. In our approach, panoramas are further employed as compact viewer-centered representations for learning the impressions of an environment. To recognize a series of image frames from a video can therefore be formulated as finding a set of continuously moving corresponding regions belonging to some panorama in the database, where each frame should match each region to a considerable extent.

A. Related Work

The recent decade of visual-recognition researches saw a gradual shift away from the three-dimensional reconstruction approaches [16], [22], [41] pioneered by Marr [25] toward view-based (or appearance-based) approaches that store snapshots of objects or scenes [9], [29], [30], [32], [33], [48]. An appearance-based representation of objects is constructed from a set of views of an object in a preprocessing (or learning) stage. Then, the collection of views is recorded in a compact way through an eigen-space representation [29], [32], [11] or neural networks [33], [48] for the purpose of detection and recognition. Generally, most appearance-based techniques were designed for recognizing objects [9], [29], [30], [33], [48]. In particular, most of them perform recognition based on isolated images [29], [30], [33], [48], whereas not many of them are based on image sequences or videos [9], [26], [27]. Appearance-based techniques have also shown their effectiveness for tracking long image sequences across views [3] or recognizing objects in cluttered environments [32], [33]. By purposefully controlling the cameras via maximizing an entropy measure, appearance-based recognition can be achieved dynamically in active vision as well [6].

Recently, encoding omni-directional appearances with panoramas has received considerable attention for the purpose of localization, navigation, or route recognition in robot vision [1], [17], [20], [26], [27], [47], [49]. Pioneering work using panoramic representation for route recognition by a mobile robot was done by Zheng and Tsuji [49]. In their work, two vertical stripes of each image were used to create mosaics (or called manifold mosaics latter [34]) of side views along a robot route by stitching the vertical stripes captured. Then, the recorded mosaics were used for robot route recognition. Recently, many approaches used a catadioptric visual sensor [2], [45] consisting of a video camera and a curved mirror for acquiring viewer-centered panoramas. The viewer-centered panoramas were used to memorize the environments at a number of reference points, and then, a mobile robot found its current local area by examining similarities between the current view and those recorded [1], [17], [20], [26], [27].

B. Overview of Our Approach

In this paper, we use viewer-centered panoramas to assist the recognition of videos captured with a common video camera. Unlike the approaches mentioned above that match an "entire" panorama to other panoramas, the problem faced in this paper is

Manuscript received March 25, 2002; revised September 25, 2002. This work is supported in part by the National Science Council of Taiwan, R.O.C., under Grant NSC 89-2218-E-001-004. This paper was recommended by Associate Editor X. Jiang.

C.-S. Chen and J.-H. Chen are with the Institute of Information Science, Academia Sinica, Taipei Taiwan, R.O.C. (e-mail: song@iis.sinica.edu.tw).

W.-T. Hsieh is with the Institute of Information Science, Academia Sinica, Taipei Taiwan, R.O.C. He is also with the Department of Computer Science and Information Engineering, National Taiwan University, Hsin-Chu, Taiwan, R.O.C.

Digital Object Identifier 10.1109/TSMCB.2003.811770

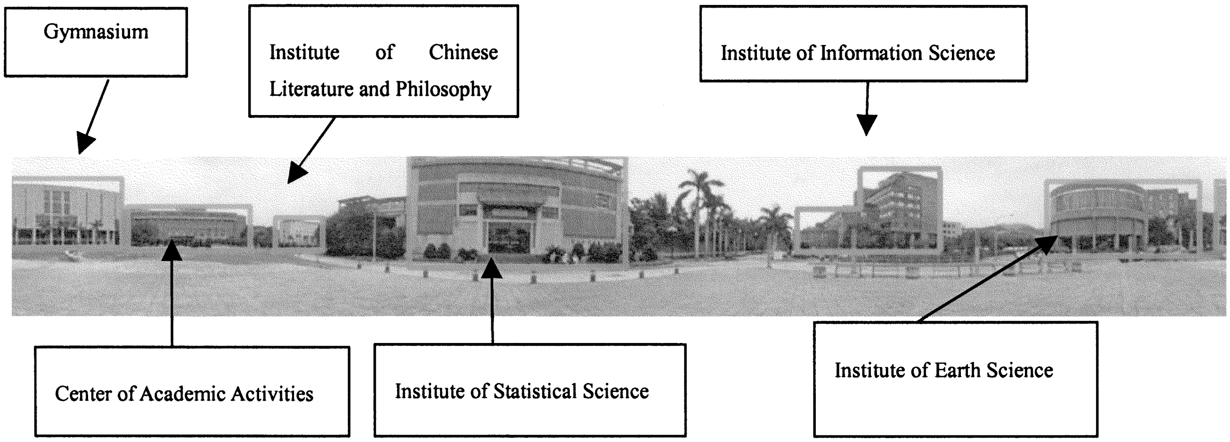

Gymnasium
Institute of Chinese Literature and Philosophy
Institute of Information Science
Center of Academic Activities
Institute of Statistical Science
Institute of Earth Science

Fig. 1. Example of the panorama that is augmented with some high-level descriptions.

to match each video frame to “part” of a panorama. Such a partial matching problem is more complex than a fully matching one and is closer to the situation of environment recognition in human vision. It therefore simulates the case of route recognition with a hand-held camera, a head-mounted camera [19], or a wearable camera [24]. In addition, almost all the previous approaches exploited the assumption that the imaging devices are placed vertical to the ground. Hence, the recognition performances will be seriously affected if the camera is vibrant (e.g., it may happen when robots move along a bumpy road). On the contrary, to tolerate possible shooting vibrations with hand-held cameras in our work, we use rotationally invariant features to deal with the matching problems, and thus, the camera has not necessarily to be exactly vertical. By using a coarse-to-fine approach for candidate selection of matching blocks, our method can tolerate about ±30° image rotations, which is sufficient for most applications when hand-held cameras are used (an experimental result is given in Section IV-B-3).

Based on the above scenario, our approach is divided into two phases: *panorama acquisition and authoring* (PA) and *panorama-guided visual recognition and tracking* (PGVRT).

- *PA phase:* Panoramas of the environments or scenes that are significant for a particular application are taken and stored in a database in this phase. Furthermore, if necessary, the recorded panoramas can be augmented by adding high-level descriptions associated with particular regions in the panoramas. More specifically, in the PA phase, all the panoramas were augmented with an environment name, and some particular regions of the panoramas were further augmented with other high-level descriptions such as the names and historical reviews of the observed buildings, landscapes, or roads, and so on. An example is shown in Fig. 1.
- *PGVRT phase:* Assume that the database contains a set of panoramas. When a video segment is taken within an acceptable range to the center of one panorama, it is desired to recognize the video content by correctly finding this panorama in the database and matching each frame in the video segment with an appropriate region in the panorama.

By using the high-level descriptions provided by the PA phase, the framework of this paper can be applied for automatic tour guidance, robot localization, and navigation, and it has potential to be extended for content-based panorama retrieval. Although a global positioning system may be used to achieve some of the same purposes, it requires four or more satellites to determine the position and time [44], [50] and may lose efficacy when insufficient satellite signals can be received (for example, the signals may be blocked by high buildings, mountains, or forests).

The PGVRT phase is very important to this approach and will be introduced in detail later. In our method, a matching graph is constructed for appearance-based recognition, and the recognition task is transformed into that of searching a shortest path in this graph, which can be solved with dynamic programming (DP). The remainder of this paper is organized as follows: Section II presents the PGVRT phase of our approach. Then, the method for dealing with partially recognizable cases is discussed in Section III. Section IV introduces some implementation details and shows several experimental results and discussions. Finally, Section V gives our conclusions.

II. PGVRT

Consider an environment database $\mathbf{P}$ containing a set of panoramas $\mathbf{P} = \{P_1 \ldots P_M\}$. Let a series of image frames contained in a video segment be $\mathbf{F} = \{f_1, \ldots, f_N\}$. Our purpose is to recognize the captured scene of these frames by matching them with a set of corresponding regions contained in some panorama belonging to $\mathbf{P}$. There are two common types of panoramas (*cylindrical* [10], [15] and *spherical* [10], [43]), both of which can be used in our framework for recognition of videos. The spherical type is actually better because its viewing range is larger than that of the cylindrical type. However, without lost of generality, panoramas of the cylindrical type will be used for discussions and experiments in this paper.

We formulate the problem of recognizing the series of frames as the problem of searching the optimal path in a specially designed matching graph. Our method for the PGVRT phase can be divided into three stages:

1) *candidate-selection* stage;
2) *graph-construction* stage;
3) *path-searching* stage;

which are introduced below.

A. Finding Matching Candidates

In the candidate-selection stage, to provide suitable lowpass filtering effects for increasing matching correctness, each image frame $f_i (i = 1, \dots, N)$ is smoothed and normalized to be $k \times k$ (in our implementation, the size of each frame is 240 × 240 and $k = 16$). The CIE L*u*v* space [18] is adopted in our work to represent an image block because it closely matches human perception of the discrimination between a pair of colors. Assume that a $k \times k$ image block I is represented as (L, U, V), where L, U, V are d-dimensional vectors $(d = k^2)$ formed by concatenating all corresponding l, u, and v values in the raster scanning order, respectively. To compensate for the problem caused by illumination changes, the L vector is normalized as $\underline{L} = (L - L_0)/||L - L_0||$, where L_0 is a d-dimensional vector with all of its values being $mean(L)$, which is the average of the d values contained in L. Note that normalization techniques, e.g., normalized cross-correlation [31], histogram equalization [36], [42], and normalized luminance component [22], [37], have been widely adopted in the past to accommodate the variations in illumination conditions. In our work, the latter approach was adopted. Such a normalization will cause the Euclidean distance between two normalized blocks $||\underline{L}_1 - \underline{L}_2||$ to be invariant to linear lighting variations. One can easily verify that $||\underline{L}_1 - \underline{L}_2||$ remains the same if L_1 and L_2 become $a_1 L_1 + b_1$ and $a_2 L_2 + b_2$, respectively, for all a_1, $b_1 a_2$, and b_2 with the requirement $a_1 > 0$, $b_1 \geq 0$, $a_2 > 0$, and $b_2 \geq 0$. Although the cross-correlation of L_1 and L_2 is also invariant to linear lighting variations, the Euclidean distance between two normalized blocks is adopted in our work because it satisfies the triangle inequality, which is a necessary property for a range search method [7] that is discussed in Section IV-C.

The purpose of this stage is to find, in each scaled panorama, all the blocks whose matching costs are smaller than a given threshold. Hence, several scaled versions of each panorama were kept for multiscale template matching. Let a set of ascending scaling factors be $\mathbf{S} = \{s_1, \dots, s_L | 0 < s_1 < s_2 < \dots < s_L\}$. Assume that a panorama P_j with the width and height W_j and H_j is contained in the database. Its lth scaled panorama P_j^l is an $(s_l W_j) \times (s_l H_j)$ image generated by linearly scaling P_j. Each frame in $\mathbf{F}$ is treated as a template for block matching to every scale of the panoramas in $\mathbf{S}$, and let $\Theta = \{p_j^{l;x,y} | 0 \leq x \leq (s_l W_j) - k, 0 \leq y \leq (s_l H_j) - k, l = 1, \dots, L, j = 1, \dots, M\}$, where $p_j^{l;x,y}$ is a $k \times k$ image block with its upper-left point being (x, y) in the scaled panorama P_j^l. To handle image rotations, we first use rotational moment invariants proposed in [14] to coarsely select candidate regions in Θ, as introduced in Section II-A1. Then, among them, some candidates regions are finely selected via template matching taking into account image rotations ranging from −30° to 30°, as introduced in Section II-A2.

1) Coarse Selection of Matching Candidates: The complex moment $c_{pq}^{(f)}$ of order $(p + q)$ of the image $f(x, y)$ is defined as [14]

$$c_{pq}^{(f)} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x + iy)^p (x + iy)^q f(x, y) dx dy \tag{1}$$

where i denotes the imaginary unit, and $f(x, y)$ is an image function (or image). A discrete version for computing the complex moment is used in practice because images taken with a frame grabber are defined in a discrete domain:

$$c_{pq}^{(f)} = \sum_{x=1}^{k} \sum_{y=1}^{k} (x + iy)^p (x - iy)^q f(x, y) \tag{2}$$

In this work, we use the moment invariant $\Phi(2, 1)^{(f)}$ that is invariant to both convolution and rotation as defined below [14]:

$$\Phi(2, 1)^{(f)} = K(1, 2)^{(f)} K(2, 1)^{(f)} \tag{3}$$

where $K(p, q)^{(f)}$, p, and $q \in Z$ are recursively defined by

$$K(p, q)^{(f)} = \begin{cases} 0; & \text{if } (p + q) \text{ is even.} \\ c_{pq}^{(f)} - \frac{1}{c_{00}^{(f)}} \sum\limits_{n=0}^{p} \sum\limits_{\substack{m=0 \\ 0<n+m<p+q}}^{q} \binom{p}{n} \binom{q}{m} & \\ \qquad \times K(p - n, q - m)^{(f)} c_{nm}^{(f)}; & \text{otherwise} \end{cases} \tag{4}$$

An example of the moment-invariant values computed via (3) is shown in Fig.2 (a). Based on the above moment invariant, we define the three rotationally invariant measures $\text{RIM}_{\underline{L}}(I_1, I_2)$, $\text{RIM}_U(I_1, I_2)$, and $\text{RIM}_V(I_1, I_2)$ to evaluate the similarity between two image blocks I_1 and I_2 for L, U, and V color channels, respectively:

$$\text{RIM}_{\underline{L}}(I_1, I_2) = \frac{\left\| \Phi(2, 1)^{(\underline{L}_1)} - \Phi(2, 1)^{(\underline{L}_2)} \right\|}{\left\| \Phi(2, 1)^{(\underline{L}_1)} \right\|} \tag{5}$$

$$\text{RIM}_U(I_1, I_2) = \frac{\left\| \Phi(2, 1)^{(U_1)} - \Phi(2, 1)^{(U_2)} \right\|}{\left\| \Phi(2, 1)^{(U_1)} \right\|} \tag{6}$$

$$\text{RIM}_V(I_1, I_2) = \frac{\left\| \Phi(2, 1)^{(V_1)} - \Phi(2, 1)^{(V_2)} \right\|}{\left\| \Phi(2, 1)^{(V_1)} \right\|}. \tag{7}$$

Then, the purpose of coarsely finding matching candidates in our work can be formulated as finding Ψ defined below.

$$\Psi = \Psi_1 \cup \Psi_2 \cup \dots \cup \Psi_N, \tag{8}$$

where $\Psi_i = \{I | I \in \Theta, \text{RIM}_{\underline{L}}(f_i, I) < T_{\underline{L}}, \text{RIM}_U(f_i, I) < T_U, \text{RIM}_V(f_i, I) < T_V\}$ is the set of coarsely selected candidate blocks of the ith image frame, f_i, and $T_{\underline{L}}$, T_U, and T_V are threshold values. The coarsely selected blocks are then sent to the next step for fine selection.

2) Fine Selection of Matching Candidates: Although moment invariants can be used for finding some candidate regions, unrelated regions might also be included because two dissimilar blocks may have approximate moment-invariant values. Hence, a fine-selection step is further performed for choosing the candidate regions obtained from the previous step. For each image frame f_i, we rotate it according to a set of predefined degrees Γ (in our example, $\Gamma = \{-30°, -20°, -10°, 0°, 10°, 20°, 30°\}$) to form a set of templates $\{f_{i;\theta} | \theta \in \Gamma\}$, where $f_{i;\theta}$ is obtained by rotating f_i by a degree θ. Then, the purpose of finely selecting matching candidates in our work can be formulated as finding Ω as

$$\Omega = \Omega_1 \cup \Omega_2 \cup \dots \cup \Omega_N \tag{9}$$

where $\Omega_i = \{I | I \in \Psi_i, Sim(f_i, I) < T_m\}$ is the set of finely selected candidate blocks in association with the ith image

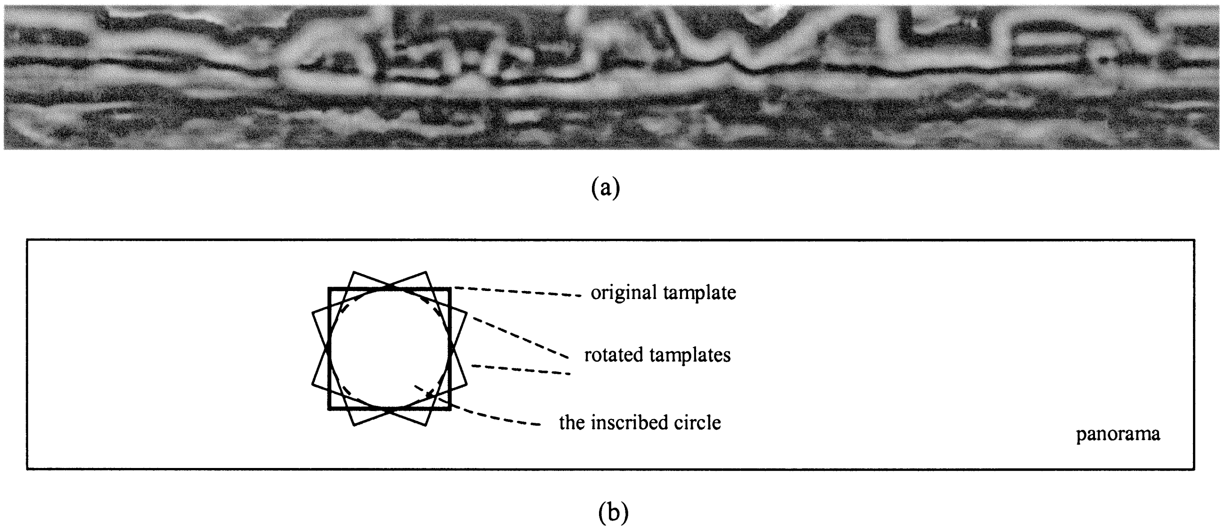


Fig. 2. (a) Moment-invariant values computed for the $\underline{L}$ components of all the blocks contained in the panorama shown in Fig. 1. This gray-level image is obtained by linearly scaling the moment-invariant value to [0, 255]. (b) Rotated templates and the inscribed circle. The pixels inside the inscribed circle of a template are used for matching.

frame, and the matching score between two blocks I_1 and I_2, $Sim(I_1, I_2)$ is defined to be

$$Sim(I_1, I_2) = \mathrm{Min}_{\theta\in\Gamma} \left(w_1 \cdot \left(\|\underline{L}_{1;\theta} - \underline{L}_2\|\right) \right. \\ \left. + w_2 \cdot \left(\|U_{1;\theta} - U_2\| + \|V_{1;\theta} - V_2\|\right)\right) \quad (10)$$

where $(\underline{L}_{1;\theta}, U_{1;\theta}, V_{1;\theta})$ and $(\underline{L}_2, U_2, V_2)$ are the images in association with the $\underline{L}$, U, and V components of $I_{1;\theta}$ and I_2, respectively, and w_1, $w_2 > 0$ are the weights for lighting and chromatic parts, respectively.

To find the set of matching candidates Ω, template matching is used in this work. Note that the template obtained via image rotation is a square block whose row is not horizontal. To simplify the implementation of matching, we only use the pixels inside the inscribed circle for matching, as illustrated in Fig.2(b). Nevertheless, we would like to emphasize that template matching is not the only way that can be used in this step. It is also possible to use other techniques, such as eigen-based approaches [29], [30], for finding the matching candidates—while note that our framework can still be used for sequence-based recognition employing interframe relationships that will be introduced below. Template matching is used in this work simply because it is easy to implement and requires less storage in our case since only the panoramas have to be recorded for it, whereas in an eigen-based approach, the coefficient vector used for linearly combining the eigenvectors has to be further stored for each image block for the recognition purpose.

B. Constructing Matching Graph

For each image frame contained in the video segment to be recognized, a set of candidate matches can be found from the panoramas in the environment database with the method introduced in Section II. It is hoped to further determine, for each image frame in the video segment, a unique candidate match that is supposed to be a correct (or approximately correct) recognition of this frame. This is difficult to achieve by treating the image frames as independent because no further visual clues can be used for identifying which candidate match is more suitable than the others. Therefore, an important issue is how to use the isolated visual recognition information provided by each image frame in an integrated manner, such that the whole video can be recognized more correctly. In this paper, we propose a method that can integrate inter-frame consistency of a video segment by constructing an associated *matching graph.*

Since, in this work, the video segment to be recognized is taken using a continuously moving and scaling camera with a zoom lens, it is reasonable to assume that the video segment is continuous in both motion and scaling. In the graph-construction stage, interframe relationships are used to increase the matching reliabilities. To construct a matching graph, the candidate blocks selected in the candidate-selection stage represent the nodes of this graph. The edges are constructed by linking those nodes associated with adjacent frames. There are directed edges coming from nodes associated with f_{i-1} for those associated with f_i for $i = 2, \ldots, N$. However, there are no edges among nodes belonging to different panoramas. In addition, if the distance between the centers of a pair of blocks associated with an edge is too long, then this edge will either not be constructed. Two additional nodes, the source node and the sink node, are built. Edges connecting the source node with layer 1 and the sink node with layer N are also constructed, respectively. Therefore, there are $N + 2$ layers in the matching graph, where layer 0 contains only a single source node, and layer $(N + 1)$ contains only a single sink node, respectively. Fig. 3 gives an illustration of a matching graph.

In a matching graph, each node is assigned with a cost, and so is each edge. The nodes in the ith layer are denoted as $V_{i;0}, V_{i;1}, \ldots, V_{i;\#(\Omega_i)-1}$, where $\#(\Omega_i)$ is the number of elements contained in Ω_i or, equivalently, the number of the finely selected candidate blocks in association with the ith image frame. Hence, the source and sink nodes are denoted as $V_{0;0}$ and $V_{N+1;0}$, respectively. Note that each node $V_{i;j}$ denotes a candidate match between f_i and a candidate block in a panorama, and let $p_{i;j}$ denote this candidate block. In addition, an edge connecting $V_{i;j}$ and $V_{i+1;k}$ is denoted as $E_{i;j;k}$ for $i = 0, \ldots, N$, $j = 0, \ldots, \#(\Omega_i) - 1$, *and*

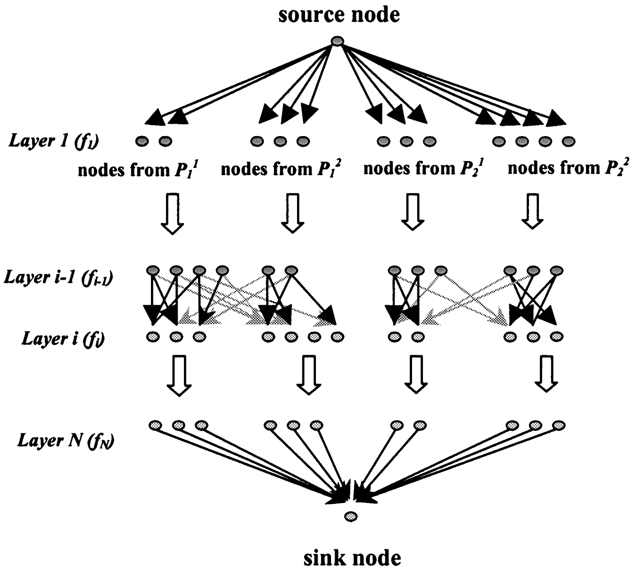


Fig. 3. Example of the matching graph where two panoramas are contained in the database, and two scales are used for each panorama. This graph contains N layers (in addition to the source and the sink node). Without loss of generality, we show a typical construction of nodes and edges for frames f_{i-1} and f_i in the middle part. There are no edges among nodes belonging to different panoramas. In addition, if the distance between the centers of a pair of blocks associated with an edge is too long, then this edge will also not be constructed. The edges across different scales of a panorama are drawn with gray color.

$k = 0, \ldots, \#(\Omega_{i+1}) - 1$. The node and edge costs are defined as follows.

- *Node Cost:* The cost of the source node Cost$(V_{0;0})$ is set to zero. The cost of each of the other nodes is recursively defined as

$$\text{Cost}(V_{ij}) = \begin{cases} \infty; \quad \text{if there is no edge between } (i-1,k) \\ \quad \text{and } (i,j) \text{ for all } k = 0,\ldots,\#(\Omega_{\text{i}-1}) - 1 \\ C(V_{i;j}) + \text{Min}\{\text{Cost}(V_{i-1;k}) \\ \quad + C(E_{i-1;k;j}) | k = 0, \ \ldots, \#(\Omega_{\text{i}-1}) - 1\}; \text{ otherwise} \end{cases}$$
$$\text{for } i = 1, 2, \ldots, N+1, \ j = 0, 1, \ldots, \ \#(\Omega_i) - 1 \qquad (11)$$

where $C(V_{i;j}) = \text{Sim}(f^i, p_{i;j})$ is the matching score defined in (10) of the two associated blocks of $V_{i;j}$, and $C(E_{i-1;k;j})$ is the edge cost that will be defined later. Note that cost of the sink node $C(V_{N+1;0})$ is also set to zero. The above definition inherits the spirit of dynamic programming.

- *Edge Cost:* The costs of the edges connecting with the source and the sink nodes are set to zero, i.e, $C(E_{0;0;k}) = C(E_{N;j;0}) = 0$ for all $k = 0, \ldots, \#(\Omega_1) - 1$, *and* $j = 0, \ldots, \#(\Omega_N) - 1$. The cost of each of the other edges is defined as a weighted sum of two components: the *motion-continuity* component, and the *scale-continuity* component. That is

$$C(E_{i;j;k}) = w_m C_m(E_{i;j;k}) + w_s C_s(E_{i;j;k}) \qquad (12)$$

where w_m and w_s are two positive weights. Each component is introduced as follows:

  - *Motion-Continuity Component* $C_m(E_{i;j;k})$: In our work, the camera is assumed to move in a continuous manner, and thus, the distance between consecutive matched blocks in a panorama must be small. The cost of this component is defined by

$$\text{motion\_cost} = \sqrt{(r_{i;j} - r_{i+1;k})^2 + (c_{i;j} - c_{i+1;k})^2} \qquad (13)$$

  where $(r_{i;j}, c_{i;j})$ and $(r_{i+1;k}, c_{i+1;k})$ are the centered positions of the blocks $p_{i;j}$ and $p_{i+1;k}$, which are the consecutive matched blocks of f_i and f_{i+1} contained in a panorama.

  - *Scale-Continuity Component* $C_s(E_{i;j;k})$: In addition, suppose the input video contains no shot changes, i.e., the effect of zooming in and out is smooth. Hence, if the consecutively matched blocks $p_{i;j}$ and $p_{i+1;k}$ are associated with panoramas of different scales s_{l_1} and s_{l_2}, $1 \leq l_1, l_2 \leq L$, the edge is assigned with a higher cost than those connecting the blocks of the same scales. The cost of this component is defined by

$$\text{scale\_cost} = \frac{|l_1 - l_2|}{L}. \qquad (14)$$

Each path starting from the source node and ending at the sink node represents a sequence of matches between the input video and the panoramas. The cost of the sink node cost$(V_{N+1;0})$ is then referred to as the *minimal cost*. The path associated with the minimal cost, or, equivalently, the shortest path from the source to the sink nodes, is then referred to as the *matching* (or *optimal*) *path* in this work. The candidate blocks associated with nodes in the matching path (except the source and sink nodes) are then treated as a sequence of matched blocks to the input frames.

C. Finding Optimal Path

In the path-searching stage, the dynamic-programming (DP) technique is used to find the optimal path, i.e. the path from the source node to the sink node with the lowest accumulated cost of the nodes and edges passed by it. In our work, to avoid recursive programming, the Dijkstra algorithm [13] is used to find the optimal path. For each node, an incoming edge with the lowest accumulated cost is retained in our approach. After finding the best incoming choice for all nodes, our process backtracks, from the sink node to the source node, to obtain an optimal path. Except for the source and sink nodes, each node passed by the optimal path then represents a match between an image frame in the video and a region in a panorama. Note that the cost of the optimal path cost$(V_{N+1;0})$ is ∞ if either of the following cases holds. First, there are some adjacent layers by which no edges connect. Second, no candidate blocks can be found for some frame, and thus, matching graphs cannot be constructed.

D. Time Complexity Analysis of PGVRT

The time complexity of the proposed algorithm for the PGVRT phase is briefly analyzed below. Note that the analysis focuses on the online recognition time and discard the procedures that can be done offline. For example, computation of the moment invariants of all the blocks contained in the panoramas can be done offline and, thus, is not considered in the time-complexity analysis shown in the following.

- *Coarse candidate selection via moment invariants*: In this step, the computation of the moment invariants of the image frame takes O(Nd), where d is the number

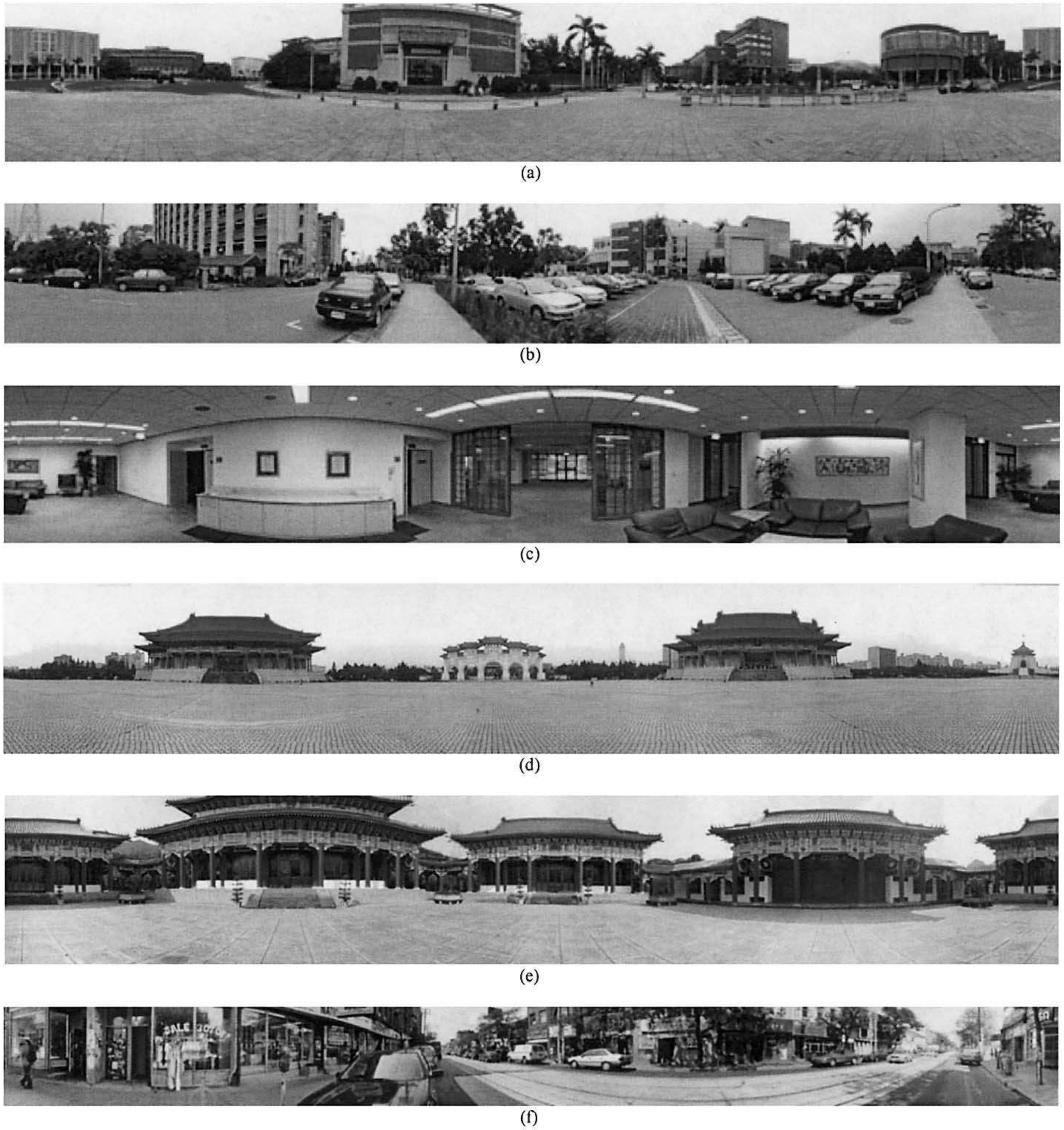


Fig. 4. Six of the 20 panoramas contained in the database used in our experiment. (a) Third panorama: Taichi Plaza (taken on September 17, 2001). (b) Fourth panorama: Hu-Shih Memorial Hall (taken on April 4, 2001). (c) Sixth panorama: lobby of Institute of Information Science (taken on September 17, 2001). (d) Seventh panorama: Chung-Cheng Memorial Hall (taken on January 5, 2002). (e) Eighth panorama: Military Cemetery (taken on January 7, 2002). (f) Twentieth panorama: a street scene (downloaded from Internet).

of pixels contained in a block, and N is the number of frames in the test video, as defined in Section II-A. Finding the candidates takes $\mathrm{O}(N\#(\Theta))$, where $\#(\Theta)$ is the number of elements contained in Θ. Hence, the time complexity is $\mathrm{O}(N(d+\#(\Theta))) \approx \mathrm{O}(N\#(\Theta))$ because d is far smaller than $\#(\Theta)$.

- *Matching with rotated templates*: This step requires the taking of block matching $\mathrm{O}(\#(\Gamma)\#(\Psi))$ times for each image frame, where $\#(\Gamma)$ and $\#(\Psi)$ are the numbers of rotation degrees and coarsely selected blocks contained in Γ and Ψ, respectively. Each block matching takes $\mathrm{O}(d)$. Hence, this step takes $\mathrm{O}(Nd\#(\Gamma)\#(\Psi))$.
- *Graph-construction and path-searching stages*: These stages take $\mathrm{O}(\sum_{i=1}^{N+1}\#(\Omega_{i-1})\#(\Omega_i))$ to find the shortest path, which is approximately $\mathrm{O}(\sum_{i=1}^{N+1}n^2) = \mathrm{O}(Nn^2)$, where n is the average number of the finely-selected candidate blocks for each frame.

To sum up, the time complexity of the PGVRT algorithm is $\mathrm{O}(N(\#(\Theta)+d\#(\Gamma)\#(\Psi)+n^2))$.

III. Dealing With Partially Recognizable Cases

In the above discussion, a video segment is taken as a whole for recognition. More specifically, the video segment is supposed to be either fully recognized as taken from an environment that has been recorded in the database or fully rejected. It therefore cannot deal with cases that are only partially recognizable, that is, some image frames are taken from a known environment, but the others are not. For example, when a video is taken by someone who goes into a known environment from outside, initial frames in the beginning of the video would be taken from an unknown environment, but the rest of the frames would be taken from a known one. In addition, it is desired that the task of environmental recognition can also be done incrementally. That is, recognition can be achieved when sufficient initial frames of a video, but not necessarily all frames, were presented.

To achieve this purpose, a video is segmented into several nonoverlapping *episodic videos* (EVs) in our approach. Consider a video containing A image frames $f_1, \ldots, f_A$. Assume that an EV contains N successive image frames. Let $\Xi_1, \ldots, \Xi_T$ denote the T EVs, where $T = \lceil A/N \rceil$. Then, $\Xi_1 = (f_1, \ldots, f_N), \ldots, \Xi_i = (f_{(i-1)N+1}, \ldots, f_{iN}), \ldots$, and $\Xi_T = (f_{(T-1)N+1}, \ldots, f_W)$. Given an EV Ξ, assume that Ξ is processed with the method introduced in Section II, and an average cost $C = cost(V_{N+1;0})/N$ can thus be obtained. If C is smaller than a threshold T_R, then Ξ is called *recognizable* and set $\text{Recognize}(\Xi) = (P_j; p_j^{l(1);x(1),y(1)}, p_j^{l(2);x(2),y(2)}, \ldots, p_j^{l(N);x(N),y(N)})$, where P_j is the recognized panorama, and $p_j^{l(1);x(1),y(1)}, \ldots, p_j^{l(N);x(N),y(N)}$ is the sequence of recognized blocks contained in P_j associated with the optimal path. Otherwise, if C is larger than T_R, Ξ is treated as *unrecognizable* and $\text{Recognize}(E) = \phi$.

A pair of recognition results $(P_i; p_i^{l(1);x(1),y(1)}, p_i^{l(2);x(2),y(2)}, \ldots, p_i^{l(N);x(N),y(N)})$ and $(P_j; p_j^{l'(1);x'(1),y'(1)}, p_j^{l'(2);x'(2),y'(2)}, \ldots, p_j^{l'(N);x'(N),y'(N)})$ is called *continuously consistent* if $i = j$ and $l(N) \approx l'(1)$, $x(N) \approx x'(1)$, $y(N) \approx y'(1)$, where "$\approx$," *approximately equal*, is defined via thresholds. In essence, it is desired that the recognized video satisfies the following constraint: If a pair of consecutive EVs Ξ_i and Ξ_{i+1} are both recognizable, then their recognition results must be continuously consistent. This constraint holds because in our framework, the environments recorded in a database are restricted to be independent to each other.

The developed algorithm, which can deal with partial recognition cases incrementally based on EVs, is shown at the bottom of the page. It ensures that recognition results satisfying the above continuously consistent constraint are obtained.

In essence, the above algorithm can recognize a video incrementally, but there is at most 1.5 EV delay of the recognition response. In this algorithm, a function $\text{Verify}(\Xi; j, s, x, y)$ is used to verify whether or not an EV Ξ can be recognized as a continuous sequence of image blocks approximately starting from the $x - y$ position of the jth panoramas, whose scales are approx-

```
Episode-Based Incremental Recognition Algorithm
0. Input new image frames in turn, until that the first EV $\Xi_1$ is formed. Set all
   frames of $\Xi_1$ as recognized based on Recognize($\Xi_1$); Set $i = 2$.
1. Continuously input new image frames in turn, until a new EV $\Xi_i$ is formed.
2. If $\Xi_i$ is unrecognizable, set all frames of $\Xi_i$ as unrecognizable, and go to Step 4.
3. If there is any unrecognizable frame in $\Xi_{i-1}$,
     set all frames of $\Xi_i$ as recognized based on Recognize($\Xi_i$).
   Else (i.e., all frames in $\Xi_{i-1}$ are recognizable),
     assume that Recognize($\Xi_{i-1}$) $= (P_j; p_j^{l(1);x(1),y(1)}, p_j^{l(2);x(2),y(2)}, \ldots, p_j^{l(N);x(N),y(N)})$.
     If the recognized result of $\Xi_i$ is continuously consistent with that of $\Xi_{i-1}$,
       set all frames of $\Xi_i$ as recognized based on Recognize($\Xi_i$).
     Else (i.e., they are not continuously consistent),
       let $\Xi$ be a new EV composed of the latter half frames of $\Xi_{i-1}$ and the
         former half frames of $\Xi_i$.
       Let $s = l(\lceil (N+1)/2 \rceil)$, $x = x(\lceil (N+1)/2 \rceil)$, $y = y(\lceil (N+1)/2 \rceil)$, the scale and $x - y$
       position of the middle block in Recognize($E_{i-1}$).
       If Verify($\Xi; j, s, x, y$) $= \phi$ (i.e., unrecognizable),
           modify the later half frames of $\Xi_{i-1}$ as unrecognizable. Set the
           former half frames of $\Xi_i$ as unrecognizable and set its later half frames
           as recognized based on Recognize($\Xi_i$).
       Else (i.e., Verify($\Xi'; j, s, x, y$) $\neq \phi$)
           Let $\Xi'$ be a new EV composed of $\Xi$ and the later half frames of $\Xi_i$.
           If Verify($\Xi; j, s, x, y$) $= \phi$,
           set the former half frames of $\Xi_i$ as recognized (in the $j$th panorama)
           based on Verify($\Xi; j, s, x, y$), and set its later half frames as
           unrecognizable.
         Else
           set all frames of $\Xi_i$ and the later half frames of $\Xi_{i-1}$ as recognized
           (in the $j$th panorama) based on Verify($\Xi'; j, s, x, y$).
4. If no further image frames are available in the video, stop.
   Else, set $i \leftarrow i + 1$, and go to Step 1.
```

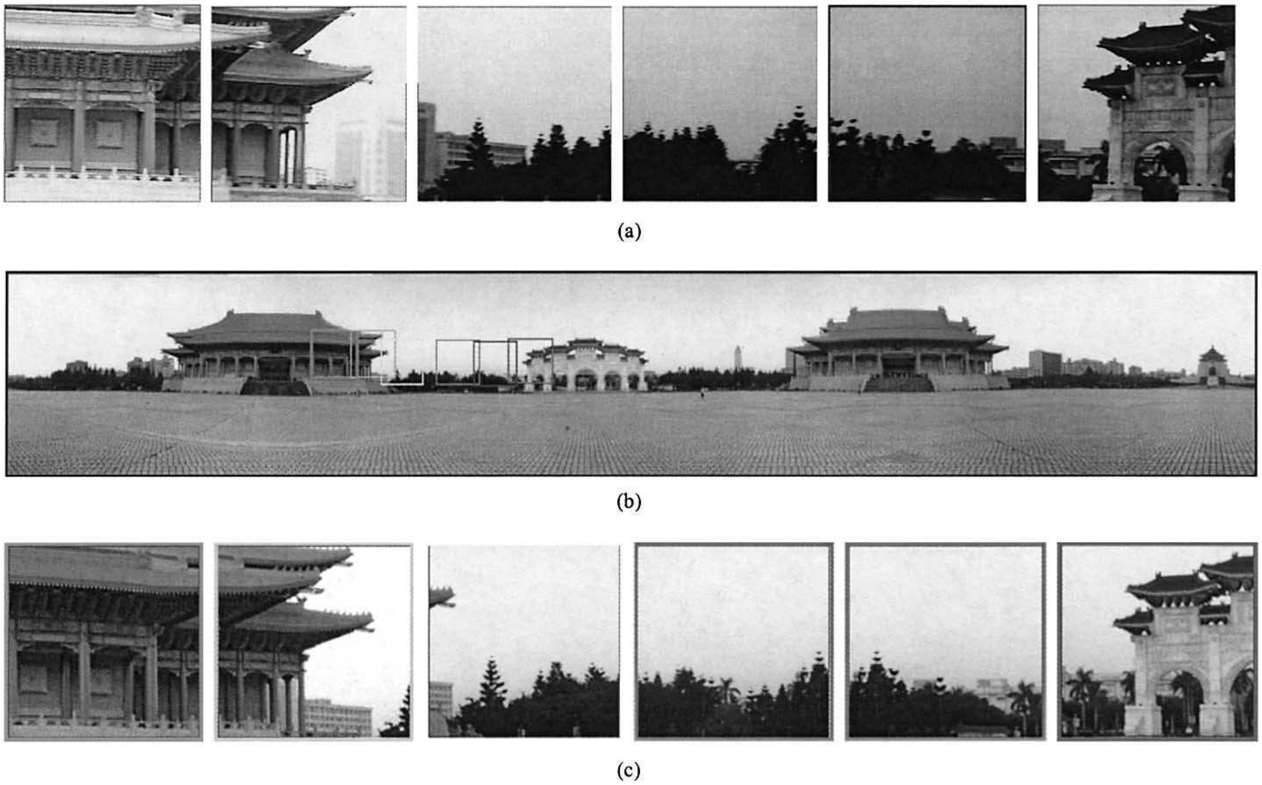


Fig. 5. (a) Test video used in our experiment. This video contains 76 image frames, and the image frames shown in this figure are the 0th, 15th, 30th, 45th, 60th, and 75th frames, respectively. This video was recognized with the method introduced in Section II. It was then correctly found that the seventh panorama—Chung-Cheng Memorial Hall—is the correct match, and (b) shows the matched panorama in association with the matched blocks corresponding to the images frames shown in (a). (c) These matched blocks more clearly. Note that the video was taken about 10 m away from the center of the panorama, and thus, there are detectable disparities between them. In addition, the video and the panorama were taken at different dates and times, and thus, their weather conditions are not the same (the video was taken on a cloudier day). However, the proposed method can still find approximately correct matches for the video frames.

imately s. This verification procedure can be treated as a special recognition problem in which only one panorama is used to construct a database $\boldsymbol{P}_j = \{P_j^l | l = 1, \ldots, L\}$, and the nodes in the first layer of its matching graph are restricted to those associated with blocks whose position and scale are approximately (x, y) and s. Hence, the verification can also be performed with the method introduced in Section II, and Verify$(\Xi; j, s, x, y)$ is then set equal to Recognize(Ξ) when the specific recognition problem mentioned above is solved.

IV. Implementation Issues and Experimental Results

Twenty panoramas, most of which were taken from an academic campus, were used to construct an environment database for use in our experiment. These panoramas include both indoor and outdoor scenes, six of which are shown in Fig. 4.

A. *Implementation Issues*

To speed up the candidate-selection process, a useful property about the structure of the constructed matching graph can be used. Remember that no edges are constructed between the nodes associated with adjacent image frames during the graph-construction stage; therefore, the candidate blocks of an image frame f_i need only to be found within a neighbor region and range of approximate scales of those have been found for f_{i-1}. By using this property, only the candidate blocks of the first frame of a video segment were matched with all the blocks contained in Θ (the set of images blocks contained in all scaled panoramas in the implementation), whereas those of the other frames are matched only within neighborhoods of the candidate blocks found for their previous frames. Note that the time complexity of candidate-selection stage presented in Section II-D is analyzed at the case when candidate blocks are independently found for each image frame. Hence, the time complexity mentioned above can be lowered by using this fast implementation, which can be seen as an upper bound of that of the fast implementation.

To further increase the computational efficiency and obtain a smooth trajectory of matched regions during recognition in our implementation, a video is uniformly subsampled every t frames, and only those sampled frames are used to construct the matching graph. Then, the matched positions and scales of the other frames are obtained by linear interpolation from those obtained with the subsampled video.

In the following, seven experiments are shown to demonstrate the effectiveness of our method. In the first three experiments, a simple version of our algorithm that does not consider image rotations (i.e., $\Gamma = \{0°\}$) was employed, and only the first ten panoramas contained in the database were used. In the other four experiments, image rotations were considered $(\Gamma = \{-30°, -20°, -10°, 0°, 10°, 20°, 30°\})$ and all 20 panoramas contained in the database were used.

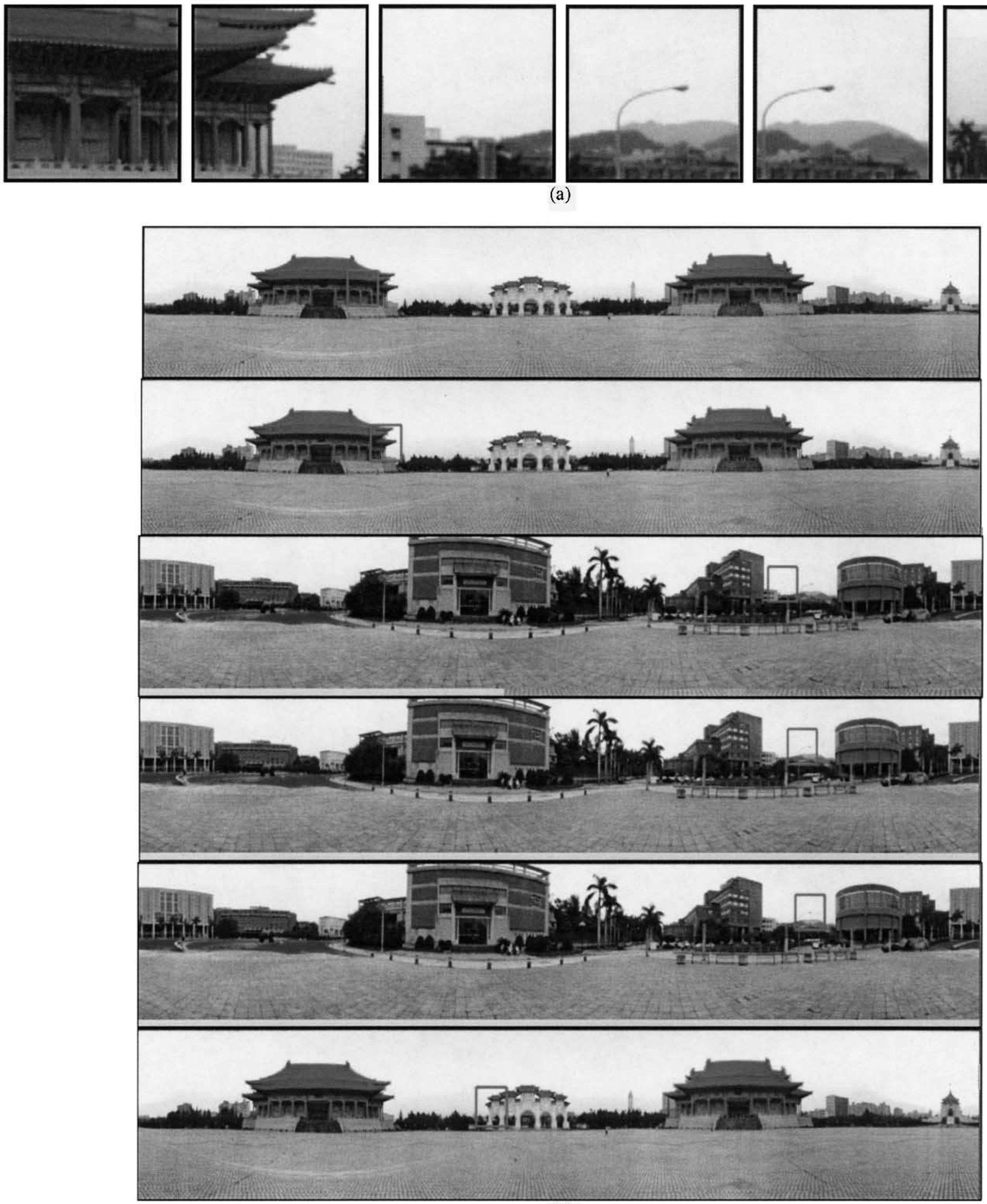
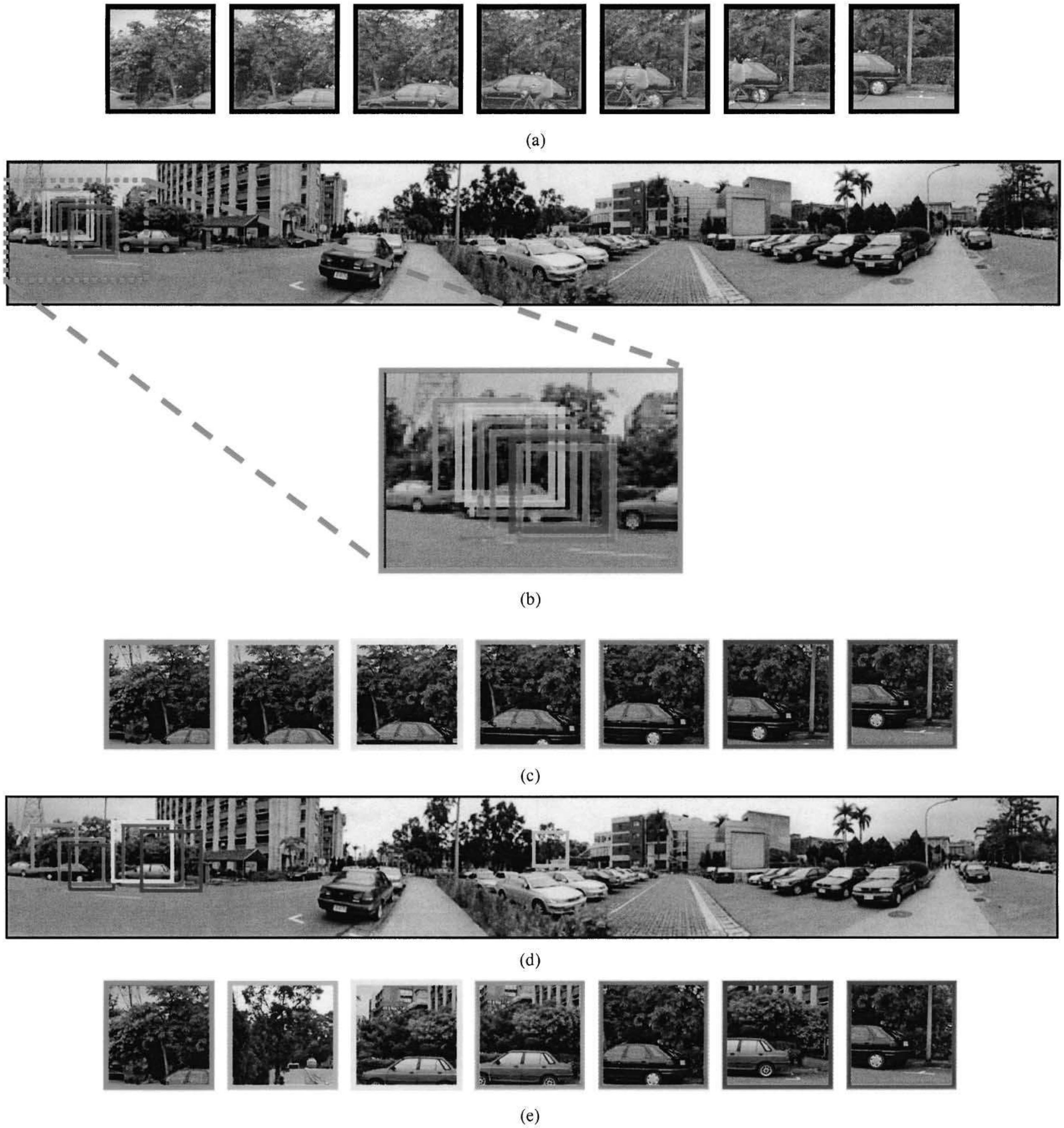


Fig. 6. Comparison of the results obtained using our method with that using the best-only strategy (i.e., associating each image frame with the corresponding candidate block with the smallest matching cost). (a) Matched blocks when best-only strategy was used for recognizing the same video shown in Fig.5(a). (b) Matched panoramas associated with the matched blocks. Comparing them with the recognition results shown in Fig. 5, it is found that motion and scale continuities integrated with DP are very successfully used in our image sequence-based approach for removing the ambiguities of individual matches.

B. Experimental Results[1]

A test video containing 76 image frames, some of which are shown in Fig.5 (a), was used for testing the performance of our method. The sample interval t is set to 5 in this experiment. By using the method introduced in Section II, a matching graph was first constructed for the subsampled video. Then, the shortest path associated with this graph was found for recognition. This experiment was done using a PC with 1.8-GHz CPU and 512 MB memory, where it took 4.096 s for the candidate-selection and graph-construction stage, and 0.04 s for finding the optimal path. The average recognition is about $(4.096 + 0.04)/76 = 0.055$ s/frame. More specifically, finding the candidate blocks of the first frame takes 1.630 s, and finding those of the other frames takes 0.165 s/frame, which shows that the implementation strategy described in Section IV-A can improve the efficiency for candidate selection. In this experiment, it was correctly found that the seventh panorama

[1] Demo videos associated with all the experimental results can be found in [51].



(a)

(b)

(c)

(d)

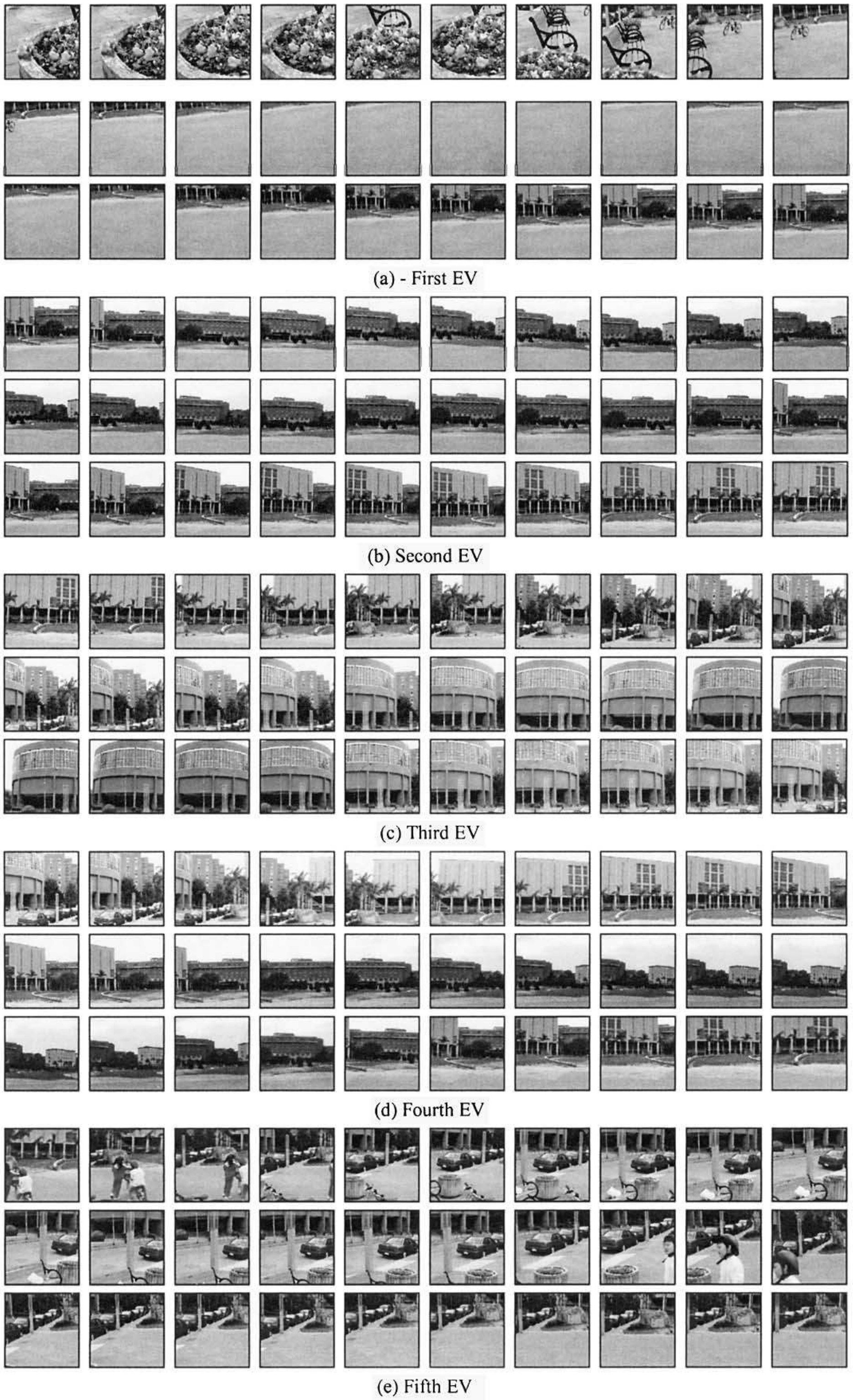
(e)

Fig. 7. (a) Video shot with a slight partial occlusion showing its 0th, third, sixth, ninth, 12th, and 15th image frames. (b), (c) Recognized panorama of this video and matched image blocks of the image frames shown in (a) obtained with our method. (d), (e) Recognized panorama of this video and matched image blocks of the image frames shown in (a) obtained with the best-only strategy.

(Chung-Cheng Memorial Hall) is the right match. Fig.5(b) shows the matched panorama, in association with the matched blocks, corresponding to the images frames shown in Fig.5(a). To be clear, Fig.5(c) further shows these matched blocks individually. It can be observed that our method successfully found very convincing matches for the test video.

It is worth noting that the video was taken in about 10 m away from the center of the panorama; therefore, there are detectable disparities between them. In addition, the video and the panorama were taken at different dates and times (the video was taken on January 20, 2002, and the panorama was taken on January 5, 2002); therefore, their weather conditions are different (the video was actually taken on a cloudier day). However, our method can still find approximately correct matches of the video frames. We owe this to the following. First, the L component used in the matching measure (10) has been normalized; therefore, linear lighting variations can be compensated. Second, a sequence of image frames, but not individual ones, was used for video content recognition, which helped to remove ambiguities or illusions occurring in individual matching results.

To clarify the second reason, let us compare our method with the best-only strategy that associates each image frame with



(a) - First EV

(b) Second EV

(c) Third EV

(d) Fourth EV

(e) Fifth EV

Fig. 8. Some images from a video containing 750 image frames. This video is segmented into five EVs, and each EV contains 150 frames. (a)–(e) Some image frames contained in the first, second, third, fourth, and fifth EVs, respectively.

the corresponding candidate block with the smallest matching cost. Fig.6(a) shows the matched blocks when the best-only strategy is used for recognizing the same video shown in Fig.5(a). Fig.6(b) further shows the matched panoramas in association with the matched blocks. Comparing the recognition results shown in Fig. 5, it is found that the motion and scale continuities (integrated with DP) are very successfully exploited and integrated in our image sequence-based approach for removing the ambiguities of individual matches. (A demo video exp_1-fig5-fig6.zip can be found in [51].)

In another experiment, a video with slight partial occlusion was employed as input, as shown in Fig.7(a). The video con-

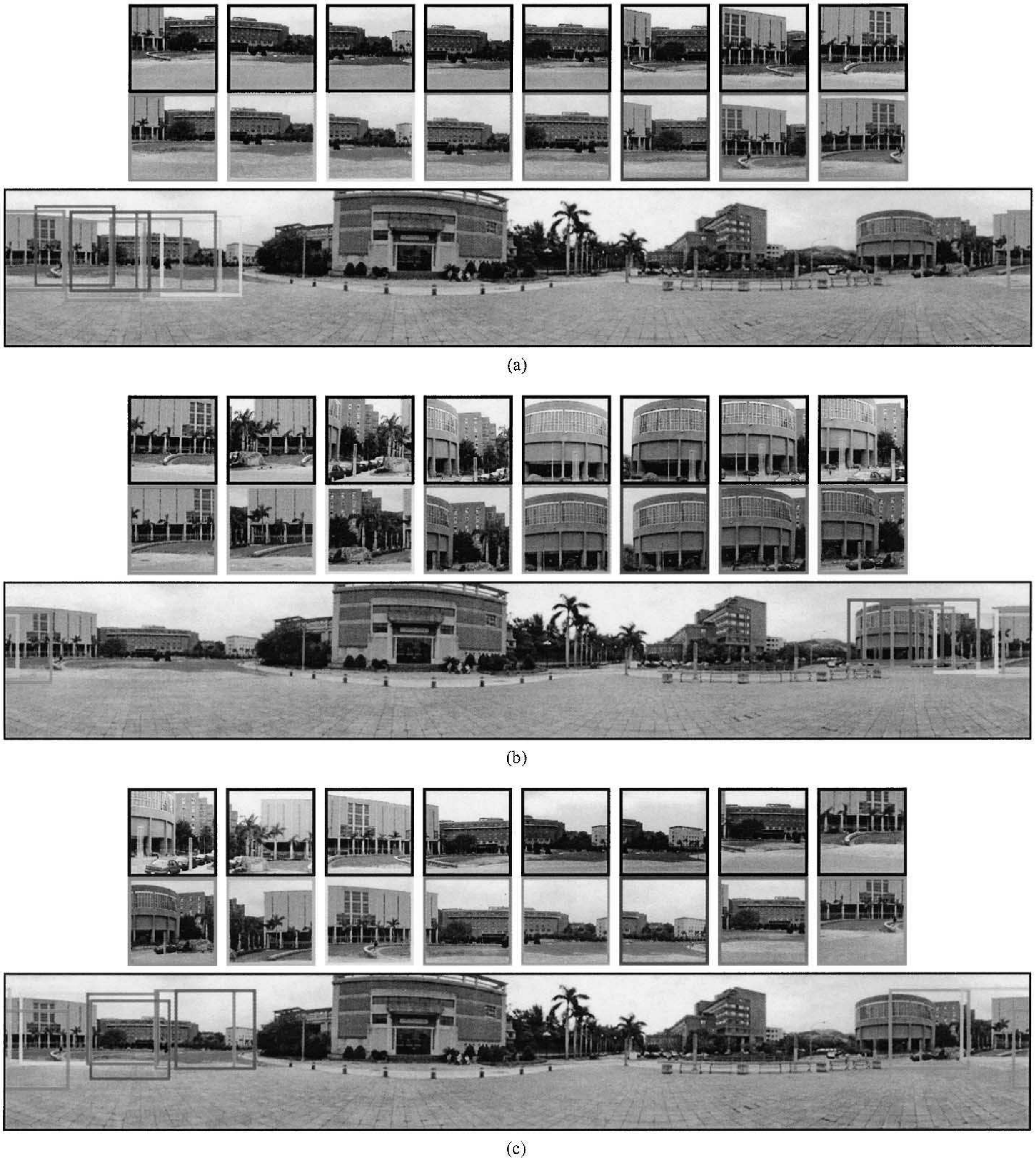


Fig. 9. (a) Upper row shows some image frames contained in the second EV, and the middle and lower rows show their matched blocks in the third panorama shown in Fig.4 (a). (b) Upper row shows some image frames contained in the third EV, and the middle and lower rows show their matched blocks in the third panorama shown in Fig.4 (a). (c) Upper tow shows some image frames contained in the fourth EV, and the middle and lower rows show their matched blocks in the third panorama shown in Fig.4 (a).

tained 22 image frames, and t, which is the sample interval, was set to 3. After being processed with the simple version of our method, the recognized panorama and matched sequence of blocks are shown in Figs.7(b) and (c), respectively, which reveals that our method still found quite convincing matches. Compared with the matches found with the best-only strategy, as shown in Figs.7(d) and (e), it can be seen that interframe consistencies among consecutive image frames are also very helpful for identifying correct matches when partial occlusions occurred. The average recognition time is 0.30 s/frame. (A demo video exp_2-fig7.zip can be found in [51].)

In the third experiment, we investigated partially recognizable situations. Fig. 8 shows a video containing 750 frames, which was grabbed with a handheld video camera taken by a person who went into the Taichi Plaza [shown in Fig.4 (a)] from its periphery. Hence, a period of its initial video segment should be unrecognizable. The test video is divided into five nonoverlapping EVs, as illustrated in Fig. 8, and each EV consists of 150

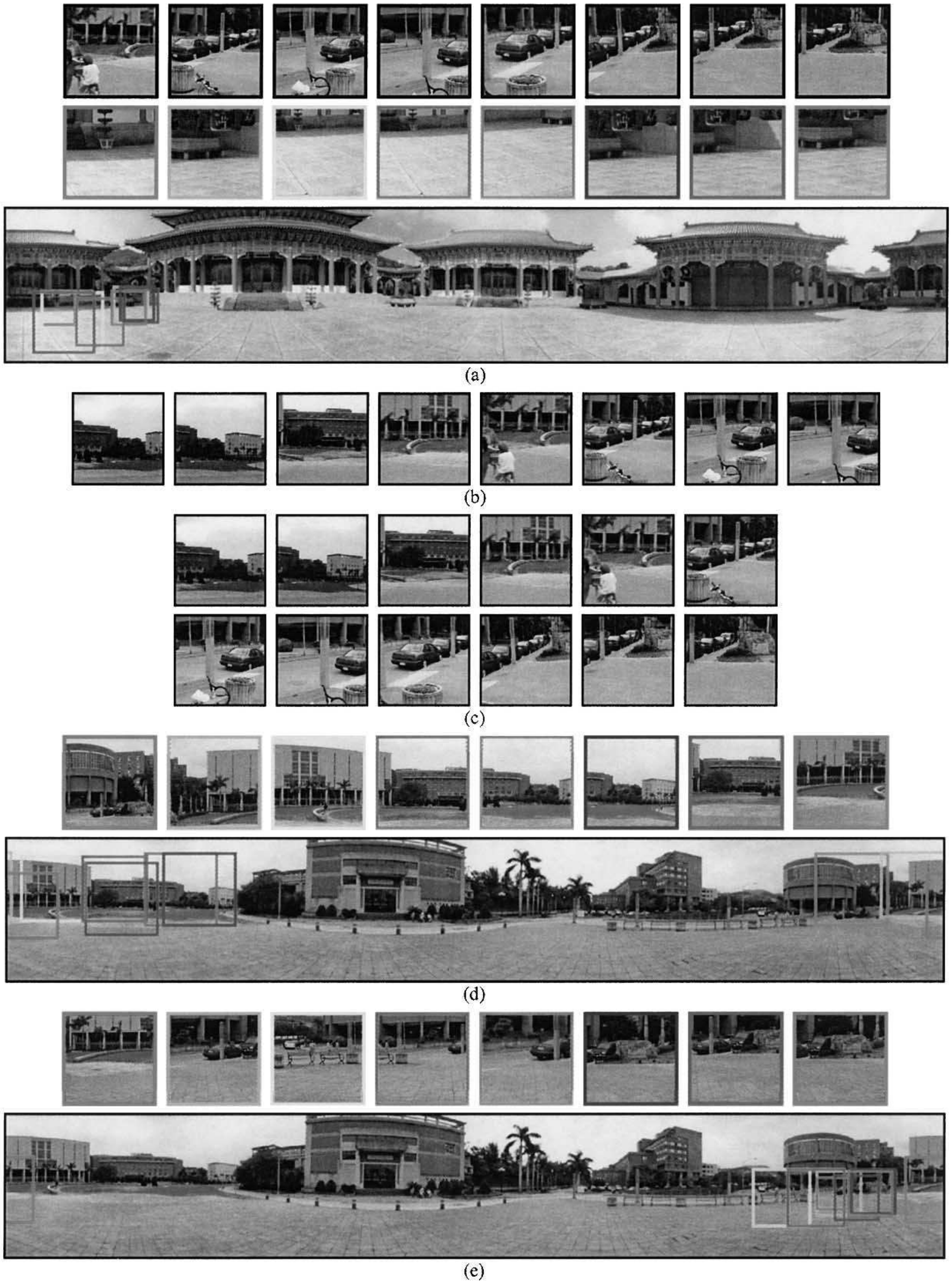


Fig. 10. (a) Initial recognition result of the fifth EV, which was recognized as in the eigth panorama shown in Fig.4 (e). The upper row shows some frames in the fifth EV, the middle row shows their matched blocks, and the lower row shows the recognized panorama. (b) Some frames contained in the EV Ξ that is used for verification as described in the algorithm shown in Section III. (c) Some frames contained in the EV Ξ' that is used for verification. (d) Final recognition result of the fourth EV. The matched blocks (in the third panorama) associated with the upper row of Fig.9 (c) are shown here. (e) Final recognition result of the fifth EV. The matched blocks (in the third panorama) associated with the upper row of (a) are shown.

frames. Because t, which is the sample interval, is set to 5, 30 frames are uniformly sampled in each EV and are used to construct a matching graph for recognition. The algorithm shown in Section III (while image rotation is not considered and therefore $\Gamma = \{0°\}$) was used to process each EV in turn to maintain continuity between consecutive EVs, and the results obtained are shown in the following.

The first EV did not match any panoramas contained in the database and, hence, was set as unrecognizable. The second, third, and fourth EVs were recognized as a sequence of successively matched blocks contained in the third panorama (Taichi Plaza), as shown in Fig.4(a), and each consecutive pair of them is also continuously consistent. Fig. 9 shows some of the image frames contained in the second, third, and fourth EVs, as well as their matched blocks. The fifth episodic video was initially recognized as in the eigth panorama, as shown in Fig. 10(a), so it is not continuously consistent with the fourth one because their recognized panoramas are not the same. Hence, according to the method introduced in Section III, a new EV Ξ was formed as composed of the latter half segment of the fourth EV and the former half segment of the fifth one. Then, Ξ was verified, whether or not it can be recognized as a sequence of image blocks continuously consistent with the recognition result of the former half segment of the fourth EV. In our experiment, Ξ was successfully verified, and hence, another EV Ξ' composed of Ξ and the later half of the fifth one was further verified, according to Step 3 in the algorithm, and Ξ' was successfully verified again in this experiment. The final recognition results of the fourth and fifth EVs are shown in Figs.10(d) and (e), respectively. From the results shown above, it can be observed that the test video was initially unrecognizable and was successfully recognized as having been taken in the environment recorded in the third panorama in our database. This shows that a series of continuously consistent recognition results can be obtained with our method. The average recognition time of this experiment is about 0.30 s/frame. (A demo video exp_3-fig8-fig9-fig10.zip can be found in [51].)

To verify the applicability of our approach under different conditions such as image rotation, scaling, viewpoint variation, and illumination changes in a detailed way, we have done a series of experiments as shown below, where the sample interval t was set to 5.

1) Influences of the Distance From the Video-Shooting Position to the Panorama Center: In the fourth experiment, we further investigate the influence of the distance between the video camera and the panorama center. This distance is referred to as the *baseline length* because the panorama and the image taken with the video camera essentially form a "stereo pair." The environment to be recognized is the same as that of experiment 1, where a south building is about 90 foot steps away from the panorama center, and a west arch is from this center about 150 foot steps away (a foot step is roughly equal to 67 cm). In this experiment, seven videos away from the center 0, 15, 30, 45, 60, 75, and 90 foot steps to the west arch were taken, respectively, by panning the camera from south to west. An illustration of the shooting conditions is shown in Fig.11(a), and Fig. 11 (b) shows the other seven videos taken for recognition in this experiment. Note that these videos were all taken on a different date (July 28, 2002) from that of the panorama (January 5, 2002), and it can be observed that the environment has a little change with occlusions (the video captures some additional tents, cars, and people). Fig. 12 shows the matched panorama in association with the matched blocks for each video. As can be seen, the larger the baseline length in terms of foot steps, the more severe the image distortions. In this experiment, the videos of 0, 15, 30, 45, and 60 foot steps were treated as successfully recognized, whereas the matching results of the videos of 75 and 90 foot steps are not quite good but still in the same correct environment [the panorama shown in Fig. 4(d)]. The average recognition time of each video in this experiment is 0.97, 0.93, 0.83, 0.72, 0.78, 0.75, and 1.46 s/frame, respectively. (Demo videos exp_4-fig11(b)-1.zip–exp_4-fig11(b)-7.zip, which are in associated with 0–90 foot steps, can be found in [51].)

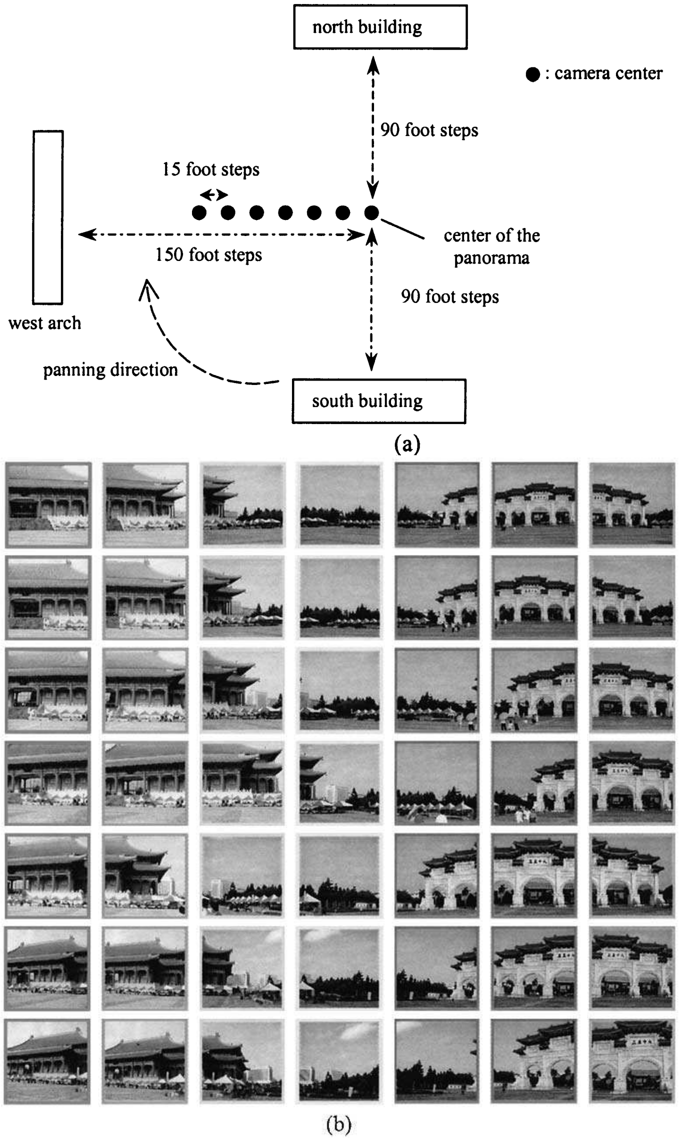


Fig. 11. (a) Shooting condition of the seven videos taken on July 28, 2002. The seven black spots are the camera centers where the videos are taken. These videos were taken away from the center 0, 15, 30, 45, 60, 75, and 90 foot steps to the west arch, respectively, by panning the camera from south to west. (b) Seven rows show some sample frames of these seven videos taken away from the panorama center 0, 15, 30, 45, 60, 75, and 90 foot steps, respectively.

In principle, when the baseline length is large or the objects in the scene are close to the camera, the matching performance diminishes with our approach. This is because the stereo pair formed by the panorama and the image taken with the video camera may has nonuniform image disparities and partial occlusions if the baseline is long and the objects contained in the scene are not co-planar. Since our method is a view-based approach, nonuniform distortions of an image block caused by large disparities between the corresponding points may affect the matching accuracies. Nevertheless, it should be noted that not only our approach but also all the methods belonging the type of using panoramas or videos for encoding the appearances of environments or objects [17], [20], [26], [27] suffer from this problem.

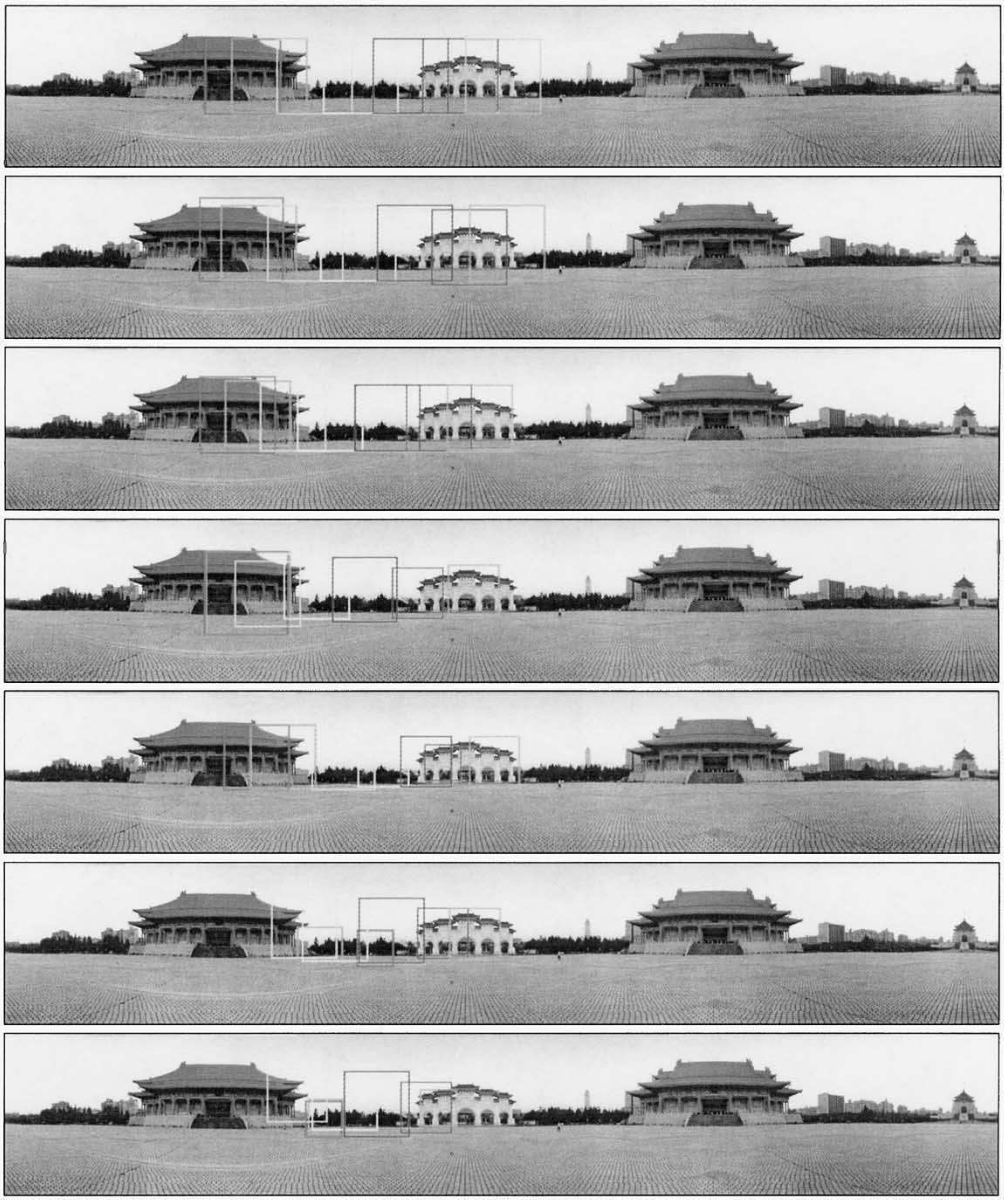
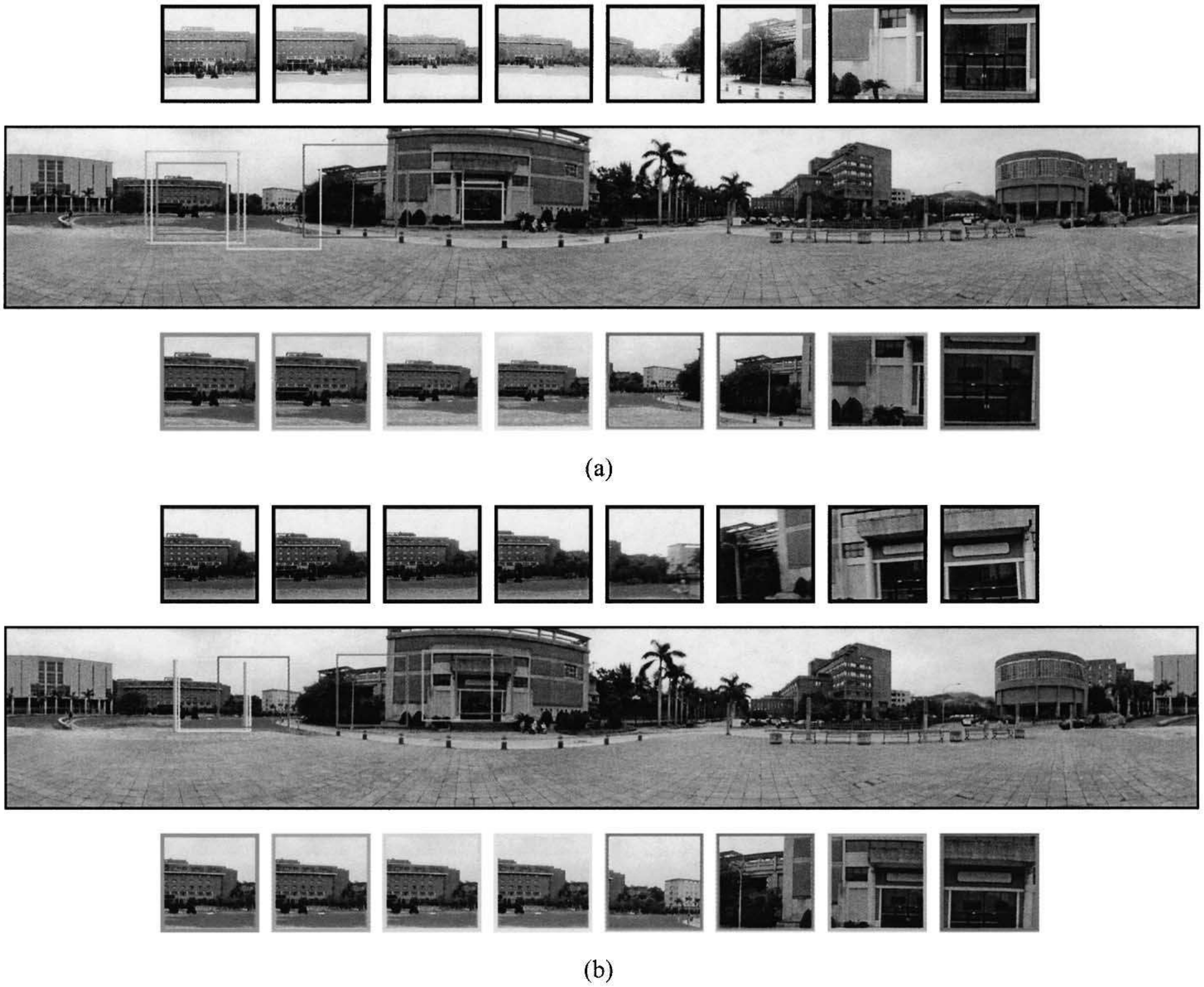


Fig. 12. From top to bottom, the seven images show the recognized panorama and the matched blocks, respectively, for the sampled frames of the videos of 0, 15, 30, 45, 60, 75, and 90 foot steps shown in Fig. 11(b).

2) Influences of Scales: The intrinsic parameters of a video camera are modified when auto-focusing is turned on or zoom-in/out is tuned, which may affect the matching results also. In the fifth experiment, a video containing 126 frames was taken by panning and zooming in/out in a scene of our campus with a Sony DCR-TRV11 DV in an auto-focus mode, as shown in the upper row of Fig. 13(a). After processed by our method, the video was correctly recognized, and it was found that the third panorama [Taichi Plaza shown in Fig. 4(a)] is the right match. The matched panorama and matched sequence of blocks are shown in the middle and lower rows of Fig. 13(a), respectively. It can be observed that the scale of each matched block varies according to the zooming condition of each frame. The average recognition time for this video is 3.29 s/frame. (A demo video exp_5-fig13(a).zip can be found in [51].)

Furthermore, different video cameras usually have different intrinsic parameters because the lens systems and the CCD chips are different. Therefore, we have also used another video camera (a Canon XL1s DV) to shoot the same scene with panning and zooming in/out, as shown in the upper row of Fig.13 (b). The video contains 84 frames, and it was also successfully recognized as shown in Fig. 13(b). From the above experimental results, it can be seen that the influence of intrinsic parameters is less than that of the baseline lengths since our method has taken into account scale changes by storing a set of scaled panoramas in the database. (A demo video exp_5-fig13(b).zip can be found in [51].)

3) Influences of Illumination Conditions and Image Rotations: To show how different lighting conditions affect the recognition results in practice, several videos were taken in the sixth experiment for recognition under different weather conditions in the same position with consistent way of shooting, which includes drizzling, cloudy, cloudless, sunny, and shower conditions, as shown in the upper rows of Figs. 14(a)–(e).



(a)

(b)

Fig. 13. Videos taken with different cameras and their matching results. Both videos were taken at the same time (July 29, 2002). (a) Upper row shows some of the image frames of a video taken with a Sony DCR-TRV11 DV. The video was taken by panning and zooming in/out. The middle row and the lower row show the recognized panorama and the matched blocks, respectively. (b) Upper row shows some of the image frames of a video taken with a Canon XL1s DV by panning and zooming in/out. The middle row and lower row show the recognized panorama and the matched blocks, respectively.

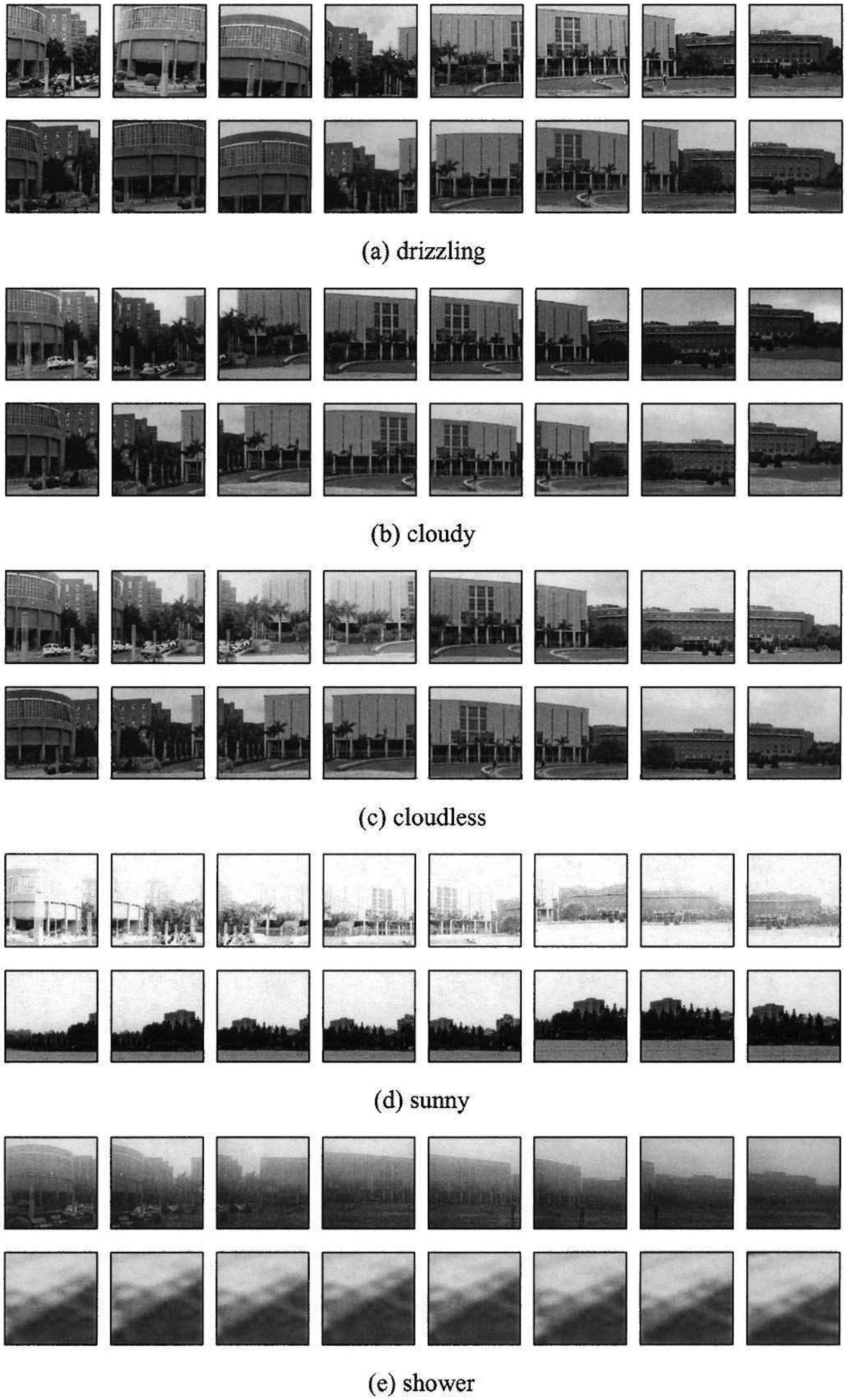
The videos taken in drizzling, cloudy, and cloudless weathers were successfully recognized, as shown in the middle and the lower rows of Figs. 14(a)–(c). However, our method failed to recognize the video taken in the showery weather or in a sunny day with a harsh reflection of the sunlight on the buildings, as shown in the middle and the lower rows of Figs. 14(d) and (e). The average recognition time of each video in Figs. 14 (a)–(e) is 1.16, 3.46, 5.61, 5.53, and 12.65 s/frame, respectively. In fact, existing methods based on panoramic appearances suffer from the same problem because matching images with significantly nonuniform illumination variations is still very difficult. (Demo videos exp_6-fig14(a).zip (drizzling), exp_6-fig14(b).zip (cloudy), exp_6-fig14(c).zip (cloudless), exp_6-fig14(d).zip (sunny), and exp_6-fig14(e).zip (shower) can be found in [51].)

In addition, to show the ability of our method to handle the matching problem with image rotation, a video containing 110 frames with significant image rotations is taken in the seventh experiment for recognition, as shown in Fig. 15(a). The complete version of our algorithm successfully found convincing matches for the input image frames, as shown in Fig. 15(b), and the average recognition time of this video is 1.18 s/frame. (A demo video exp_7-fig15.zip can be found in [51].)

Finally, we show that the augmented panoramas that contain high-level descriptions associated with particular regions can be used to generate more useful information for recognition. Fig. 16 shows some image frames and their matched blocks in the third panorama, whic was obtained in the third experiment. By using the augmented high-level information of this panorama shown in Fig. 1, useful descriptions about the matched regions can be further generated and serve as part of recognition results. Similarly, Fig. 17 shows the generated high-level descriptions in association with the fourth experiment.

C. Discussions

In this paper, a framework including three stages (candidate-selection, graph-construction, and path-searching) is proposed to recognize a video based on an environmental database con-



(a) drizzling

(b) cloudy

(c) cloudless

(d) sunny

(e) shower

Fig. 14. Upper rows of (a)–(e) show five videos taken in drizzling, cloudy, cloudless, sunny, and showery days, respectively. Lower rows of (a)–(e) show the matched blocks, respectively. In this experiment, the videos taken in drizzling, cloudy, and cloudless days were successfully recognized, whereas the videos taken in sunny and showery days failed to be recognized with our approach.

structed with multiscale panoramas. Among them, the candidate-selection stage is the most flexible, and there are many choices to build it. In this paper, template matching based on full search is adopted for this stage, but it is possible to be further speeded up with some fast algorithms by constructing a particular multilevel structure associated with each image [12], [21] or by constructing a specific tree structure of the whole database [7]. In fact, we have ever implemented the method proposed in [7] for speeding up the candidate-selection process. However, its efficiency was worse than full-search when the same experiments described in Section IV-B were performed. We attribute this to the following: First, there is extra overhead when matching in a tree structure, which makes it possible to be even slower than full search. Second, the memory access becomes quite irregular when matching with a tree structure, which is not suitable to be implemented in modern computers because

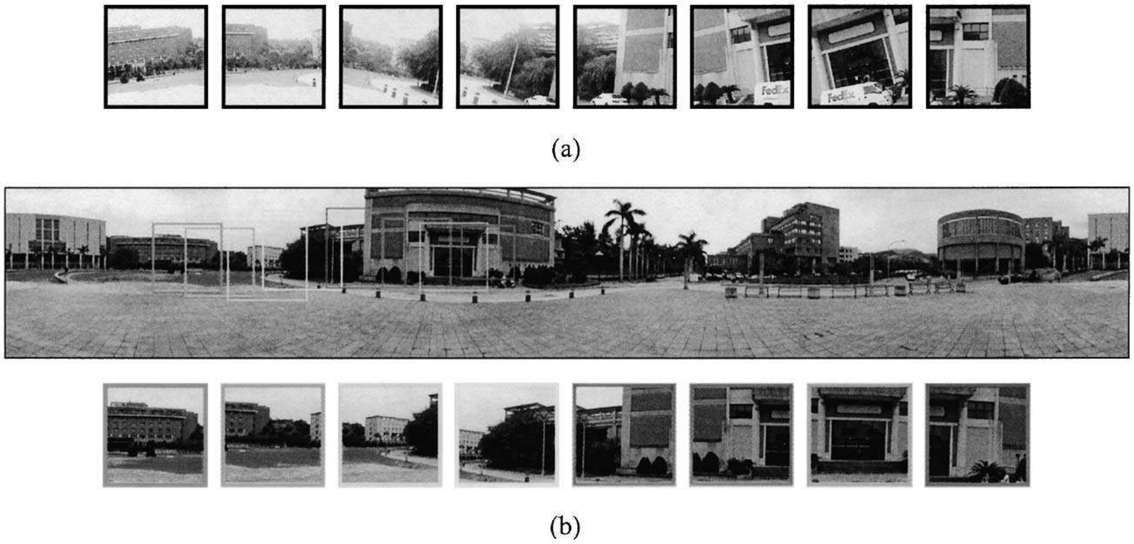


Fig. 15. (a) Video with explicit image rotations. (b) Recognized panorama and the matched blocks.

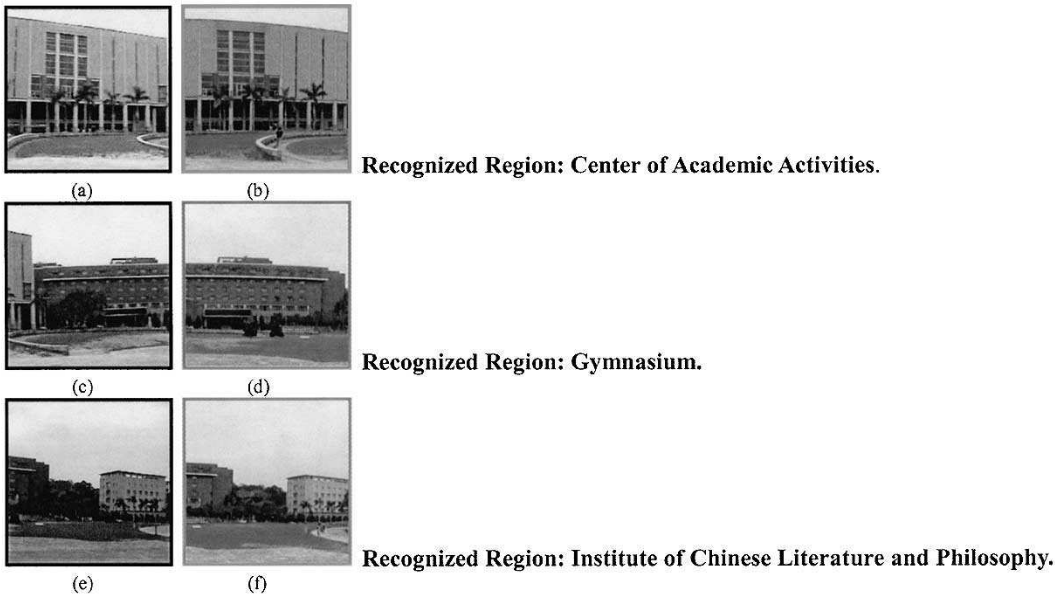

Recognized Region: Center of Academic Activities.
(a)
(b)
Recognized Region: Gymnasium.
(c)
(d)
Recognized Region: Institute of Chinese Literature and Philosophy.
(e)
(f)

Fig. 16. By using the augmented high-level descriptions of the third panorama shown in Fig. 1, useful information about the matched regions can be further generated and can serve as part of recognition results. (a), (c), and (e) show the 490th, 520th, and 540th frames contained in the test video, respectively. (b), (d), and (f) show their matching blocks in the third panorama obtained by the Episode-Based Incremental Recognition Algorithm. The corresponding high-level descriptions generated are shown on the right side, respectively.

they are usually designed to be particularly fast for consecutive memory access. However, the number of key operations to be performed can be reduced with such approaches, and hence, there is still the chance for it to be more efficient if different experimental settings and implementation environments are encountered.

Template-matching can also be modified to deal more exactly with partial occlusions, although our experimental results show that slight partial occlusion did not cause explicit difficulties for recognition. It can be improved by using robust estimation techniques such as M-estimators [28], [40], which pass a least square measurement [such as (10)] in a robust loss function to reduce the influence of outliers. In the sixth experiment presented in Section IV-B3, it was shown that large variation of weather or lighting conditions remains a difficult issue. It may be improved in our approach by including more training examples, i.e., more panoramas, which represent various lighting conditions in the environment database [4], [5], or by building rough 3-D models and lighting parameters of environments [46]. We will consider this issue in our future work.

To completely handle the problem caused by image disparities, occlusions, or intrinsic parameters, one possible way is to upgrade the view-based matching approach to a dense-point matching one in the candidate-selection stage.[2] In fact, if a dense matching has been done, then the 3-D structure of the scene can also be established via projective reconstruction or Euclidean reconstruction. However, as a well-sensed phenomenon in the computer-vision community, dense point matching considering partial occlusions is a very difficult problem that can still not be solved reliably. On the other hand, view-based approaches are

[2] The graph-construction and the path-searching stages in our framework can still be applied as well.

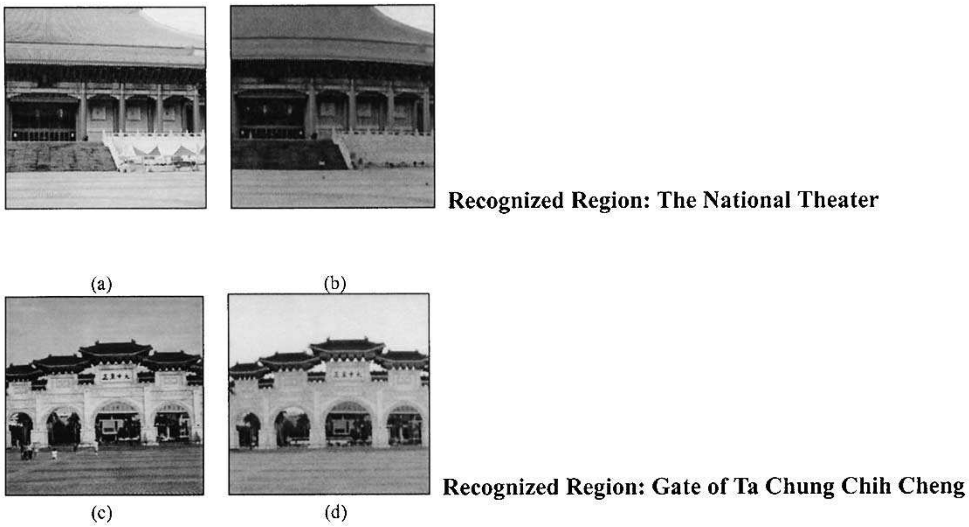

Recognized Region: The National Theater
(a)
(b)
Recognized Region: Gate of Ta Chung Chih Cheng
(c)
(d)

Fig. 17. Generated high-level descriptions for two frames of the video of 0 foot step in Fig.11. (b). (a) and (c) show the first and 80th frame in the test video, respectively. (b) and (d) show their matching blocks in the seventh panorama and their corresponding high-level descriptions, respectively.

more reliable, easy to implement, and were thus indeed widely adopted in many applications such as motion estimation, object/face recognition, and robot localization—although such approaches may be affected by the factors mentioned above. Nevertheless, note that our approach can still handle or compensate the influences caused by these problems to a considerable extent because we emphasize to use a sequence of views, instead of a single view, for visual recognition. In essence, our method only needs a "rough" matching result for each individual view in the candidate-selection stage. After gathering some roughly similar image blocks as the matching candidates for every image frame in the sequence, we then make a decision by matching the whole sequence, instead of a single image frame, by finding the shortest path in the matching graph. By using our sequence-based approach, the matching performances can be considerably more refined than those of the "best-only" approach (which matches based on isolated views), as shown in our experiments.

V. Conclusions and Future Work

This paper presents a framework for recognizing scenes captured with a video camera. The framework developed in this paper cannot only be used for recognizing environments represented in a viewer-centered way but can also be used for recognizing objects represented in an object-centered manner—if an appropriate method is exploited for the candidate-selection stage. For example, an eigen-space structure of multiple images of an object can serve as a basis for finding matching candidates in the candidate-selection stage. Then, by constructing a matching graph in the same way introduced in Section II-B, interframe relationships of a video can then be exploited for appearance-based object recognition. Since a recognition strategy has been proposed for video, the framework developed in this paper also has potential to serve as a basis for content-based video retrieval.

The contributions of this paper are summarized as follows.

1) A generally useful scenario is proposed using an image sequence, instead of a single image, for appearance-based recognition and tracking. We demonstrate that this problem can be transformed into a shortest-path searching problem associated with a well-organized matching graph, and DP can be used for finding the optimal sequence of matches.
2) A single panorama is used, instead of multiple images, for learning the appearances of an environment. Existing appearance-based learning methods have the drawback that multiple images have to be taken for a target. In addition, a multiple-image representation can only sample finite views of a target. In this paper, the recognition targets are environments instead of objects, and we note that panoramas are compact representations particularly suitable for appearance-based visual recognition and tracking of environments because a panorama inherently records infinitely many viewer-centered images of an environment. However, no such complete and compact way can be used to represent all the object-centered images of an object.

In this paper, a single panorama was used to record each environment. As discussed in Section IV-C, although dense matching or 3-D reconstruction is a method that can completely cope with the problem caused by image disparities, it is still difficult to be solved reliably. In fact, using more panoramas for a single scene is also helpful in solving this problem. To achieve this, a better way is to use the structure of concentric mosaics [38] because it inherently consists of infinite panoramas centered within a circle and, hence, is suitable to record the environments from multiple viewpoints. However, the extension from using panoramas for environment recognition to using concentric mosaics is not trivial and remains to be done in the future.

References

[1] M. Artac, M. Jogan, and A. Leonardis, "Mobile robot localization using an incremental eigenspace model," in *Proc. Int. Conf. Robotics Automation*, Washington DC, 2002, pp. 1025–1030.

[2] S. Baker and S. K. Nayar, "A theory of catadioptric image formation," in *Proc. Int. Conf. Computer Vision*, Bombay, India, 1998, pp. 35–42.

[3] M. J. Black and A. Jepson, "Eigentracking: robust matching and tracking of articulated objects using a view-based representation," *Int. J. Comput. Vision*, vol. 26, pp. 63–84, 1998.

[4] P. N. Belhumeur and D. J. Kriegman, "What is the set of image of an object under all possible illumination conditions?," *Int. J. Comput. Vision*, vol. 28, pp. 245–260, 1998.

[5] P. N. Belhumeur, D. J. Kriegman, and A. S. Georghiades, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, pp. 643–660, June 2001.

[6] H. Borotschnig *et al.*, "Active object recognition in parametric eigenspace," in *Proc. British Machine Vision Conf.*, Southampton, U.K., 1998, pp. 629–638.

[7] S. Brin, "Near neighbor search in large metric spaces," in *Proc. 21st Very Large Database Conf.*, Zurich, Switzerland, 1995, pp. 574–584.

[8] R. Brunelli and T. Poggio, "Face recognitions: features versus templates," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1042–1052, Oct. 1993.

[9] H. H. Bulthoff, C. Wallraven, and A. Graf, "View-based dynamic object recognition based on human perceptions," in *Proc. Int. Conf. Pattern Recognition*, Quèbec City, QC, Canada, 2002.

[10] S. E. Chen, "Quick time VR—an image-based approach to virtual environment navigation," in *Proc. SIGGRAPH*, Los Angeles, CA, 1995, pp. 29–38.

[11] Y. S. Chen *et al.*, "Video-based eye tracking for autostereoscopic displays," *Opt. Eng.*, vol. 40, pp. 2726–2734, 2001.

[12] Y.-S. Chen, Y.-P. Hung, and C.-S. Fuh, "Fast block matching algorithm based on the winner-update strategy," *IEEE Trans. Image Processing*, vol. 10, pp. 1212–1222, Sept. 2001.

[13] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1990.

[14] J. Flusser and B. Zitova, "Combined invariants to linear filtering and rotation," *Int. J. Pattern Recognit. Artificial Intell.*, vol. 13, pp. 1123–1136, 1999.

[15] H. C. Huang and Y. P. Hung, "Panoramic stereo imaging system with automatic disparity warping and seaming," *Graphical Models Image Process.*, vol. 60, pp. 196–208, 1998.

[16] H. Hügli, Ch. Schütz, and D. Semitekos, "Geometric matching for free-form 3D object recognition," in *Proc. Second Asian Conf. Computer Vision*, vol. 3, Singapore, 1995, pp. 819–823.

[17] H. Ishiguro and S. Tsuji, "Image-based memory of environment," in *Proc. IEEE/RSI Conf. Intelligent Robots Syst.*, Osaka, Japan, 1996, pp. 634–639.

[18] A. K. Jain, *Fundamentals of Digital Image Processing*. London, U.K.: Prentice-Hall Int., 1989.

[19] T. Jebara, B. Schiele, N. Oliver, and A. Pentland, "DyPERS: "dynamic personal enhanced reality system"," in *Proc. Image Understanding Workshop*, Monterey, CA, 1998, pp. 1043–1048.

[20] M. Jogan and A. Leonardis, "Robust localization using panoramic view-based recognition," in *Proc. Int. Conf. Pattern Recognition*, vol. 4, Barcelona, Spain, 2000, pp. 136–139.

[21] C.-H. Lee and L.-H. Chen, "A fast motion estimation algorithm based on the block sum pyramid," *IEEE Trans. Image Processing*, vol. 6, pp. 1587–1591, Nov. 1997.

[22] Q. Li, M. Zhou, and J. Liu, "Multi-resolution mesh based 3D object recognition," in *Proc. IEEE Workshop Computer Vision Beyond Visible Spectrum: Methods and Applications, in Conjunction With CVPR 2000*, 2000, pp. 37–43.

[23] H. Z. Long and W. K. Leow, "Perceptual texture space for content-based image retrieval," in *Proc. Int. Conf. Multimedia Modeling*, Nagano, Japan, 2000, pp. 167–180.

[24] S. Mann, "Humanistic intelligence: "wearcomp" as a new framework and application for intelligent signal processing," *Proc. IEEE*, vol. 86, pp. 2123–2151, Nov. 1998.

[25] D. Marr and H. K. Nishihara, "Representatin and recognition of the spatial organization of three-dimensional shapes," in *Proc. Royal Soc. Lond.*, vol. 200, 1978, pp. 269–294.

[26] Y. Matsumoto, M. Inaba, and H. Inoue, "Visual navigation using view-sequenced route representation," in *Proc. Int. Conf. Robotics Automation*, vol. 1, Minneapolis, MN, 1996, pp. 83–88.

[27] Y. Matsumoto *et al.*, "Visual navigation using omnidirectional view sequence," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Syst.*, Kyongju, Korea, 1999, pp. 317–322.

[28] P. Meer, D. Mintz, A. Rosenfeld, and D. Y. Kim, "Robust regression methods for computer vision: A review," *Int. J. Comput. Vision*, vol. 6, pp. 59–70, 1991.

[29] H. Murase and S. K. Nayar, "Visual learning and recognition of 3-D objects from appearances," *Int. J. Comput. Vision*, vol. 14, pp. 5–24, 1995.

[30] S. K. Nayar and T. Poggio, *Early Visual Learning*. New York: Oxford Univ. Press, 1996.

[31] T. S. Newman and A. K. Jain, "A survey of automatic visual inspection," *Comput. Vision Image Understanding*, vol. 61, pp. 231–262, 1995.

[32] K. Ohba and K. Ikeuchi, "Detectability, uniqueness, and reliability of Eigen widows for stable verification of partially occluded objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 1043–1048, Sept. 1997.

[33] C. Papageorgious and T. Poggio, "A pattern classification approach to dynamic object detection," in *Proc. Int. Conf. Computer Vision*, Corfu, Greece, 1999, pp. 1223–1228.

[34] S. Peleg *et al.*, "Mosaicing on adaptive manifolds," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 1144–1154, Oct. 2000.

[35] D. Roobaert and M. M. V. Hulle, "View-based 3D object recognition with support vector machines," in *Proc. IEEE Int. Workshop Neural Networks Signal Processing*, Madison, WI, 1999, pp. 77–84.

[36] H. A. Rowely, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 23–38, Jan. 1998.

[37] T. Shakunaga and K. Shigenari, "Decomposed eigenface for face recognition under various lighting conditions," in *Proc. Int. Conf. Computer Vision Pattern Recognition*, vol. 1, Kauai, HI, 2001, pp. 864–871.

[38] H. Y. Shum and L. W. He, "Rendering with concentric mosaics," in *Proc. SIGGRAPH*, Los Angeles, CA, 1999, pp. 299–306.

[39] T. Sim *et al.*, "Memory-based face recognition for visitor identification," in *Proc. Fourth IEEE Int. Conf. Automatic Face Gesture Recognition*, Grenoble, France, 2000, pp. 214–220.

[40] C. V. Stewart, "Robust parameter estimation in computer vision," *SIAM Rev.*, vol. 41, pp. 513–537, 1999.

[41] M. Suk and S. M. Bhandarkar, *Three-Dimensional Object Recognition From Range Images*. Tokyo, Japan: Springer-Verlag, 1992.

[42] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 39–51, Jan. 1998.

[43] R. Szeliski and H. Y. Shum, "Creating full view panoramic image mosaics and environment maps," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, 1997, pp. 251–258.

[44] H. Wellenhof, B. H. Lichtenegger, and J. Collins, *GPS: Theory and Practice*, 3rd ed. New York: Springer-Verlag, 1994.

[45] K. Yamazawa, Y. Yagi, and M. Yachida, "Obstacle avoidance with omnidirectional image sensor hyperomni vision," in *Proc. Int. Conf. Robotics Automation*, Nagoya, Japan, 1995, pp. 1062–1067.

[46] Y. Yu and J. Malik, "Recovering photometric properties of architectural scenes from phogographs," in *Proc. SIGGRAPH*, Orlando, FL, 1998, pp. 207–217.

[47] D. Yuan and B. MacDonald, "Natural landmark based localization system using panoramic images," in *Proc. Int. Conf. Robotics Automation*, Washington, DC, 2002.

[48] C. Yuan and H. Niemann, "An appearance based neural image processing for 3-D object recognition," in *Proc. Int. Conf. Image Processing*, Vancouver, BC, Canada, 2000, pp. 344–347.

[49] J. Y. Zheng and S. Tsuji, "Panoramic representation for route recognition by a mobile robot," *Int. J. Comput. Vision*, vol. 9, pp. 55–76, 1992.

[50] [Online]. Available: http://www.ncgia.ucsb.edu/education/curricula/giscc/units/u017/u017.html

[51] [Online]. Available: http://smart.iis.sinica.edu.tw/projects/PGVRT

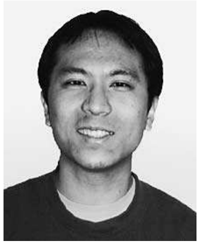
Chu-Song Chen received the B.S. degree in control engineering from National Chiao-Tung University, Hsin-Chu, Taiwan, R.O.C., in 1989 and the M.S. and Ph.D degrees in 1991 and 1996, respectively, from the Department of Computer Science and Information Engineering, National Taiwan University, Taipei.

His research interests include pattern recognition, computer vision, signal/image processing, and computer graphics. From 1997 to 1999, he was a postdoctoral fellow with the Institute of Information Science, Academia Sinica, Taipei, and has been an assistant research fellow since 1999. He has been an adjunct assistant professor at the Department of CSIE, National Taipei University of Technology, since 2000. He has also been a joint appointment assistant professor with the Department of Graphic Arts and Communication, National Taiwan Normal University, Taipei, since 2001.

Dr. Chen has received both the Outstanding Paper Award of the Image Processing and Pattern Recognition (IPPR) Society and the Best Paper Award of the Image Processing and Application Association (IPAA) of Taiwan, R.O.C., in 1997. He received the Outstanding Paper Award in the field of computer applications in ICS'2000, Chiayi, Taiwan.

Wen-Teng Hsieh received the B.S. and M.S. degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 2000 and 2002, respectively.

From 1999 to 2002, he was a part-time research assistant with the Institute of Information Science, Academia Sinica, Taipei. Currently, he is in obligatory military service. His research interest lies in computer vision.



Jiun-Hung Chen received the B.S. and M.S. degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1997 and 1999, respectively.

Since 1999, he has been a research assistant with the Institute of Information Science, Academia Sinica, Taipei. His research interests include pattern recognition, image processing, and computer vision.