A Dynamic Programming Approach for Appearance-based Recognition of Environments

Chu-Song Chen, Jiun-Hung Chen, and Wen-Teng Hsieh
Institute of Information Science, Academia Sinica, Taipei, Taiwan
Email: song@iis.sinica.edu.tw

Abstract

A systematic method for recognizing the contents of a video through the help of some panoramas recorded in a database is proposed in this paper. A panorama inherently records the appearances of an omin-directional scene from its center point to arbitrary viewing directions, and thus can be served as a compact representation of an environment. The associated recognition task is formulated to as a shortest-path searching problem, and a dynamic-programming technique is used to solve it. Experimental results show that our method can recognize a video effectively.

1. Introduction

Appearance-based methods for recognizing or tracking objects have aroused much attention in recent years due to their convincing performances [5][8][1]. Methods of this type emphasize on the use of view-based representations for objects, which are learned from a set of views of an object in a pre-processing stage. Then, the collection of views is recorded in a compact way through an eigen-space representation or neural networks for the purpose of object detection. Appearance-based techniques have shown their effectiveness for tracking long image sequences or sequences across views [1]. They can also be used for recognizing objects in a cluttered environment [5][8].

In this paper, appearance-based techniques are extended for recognizing environments or scenes instead of objects. In particular, an image-sequence-based method is proposed for appearance-based recognition, instead of single-image-based ones. 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 some high-level descriptions [1] about the scenes. Such a recognition task might be achieved with a single image frame. However, 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 on the use of a sequence of successive frames, instead of a single one, 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.

In order to recognize video contents, a database recording the appearances about the scenes has to be constructed in priori. Another characteristic of this work is that panoramas were used for constructing the database. Panorama is a type of static image with particularly successful applications for image-based virtual reality or tele-presence [6], while it is served in our approach as a compact viewer-centered representation for learning the impressions of an environment. Notice that a panorama is essentially an image with the field of view being omni-directional. To recognize a series of image frames from a video can therefore be formulated as finding a set of continuously moving regions belonging to some panorama in the database, where each frame should match with each region to a considerable extent, respectively.

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

- The preprocessing phase: Panoramas of the environments or scenes that are interesting for a particular application are taken and stored in a database in this phase. If necessary, the panoramas can be augmented by adding high-level descriptions associated with their particular regions.
- The PGVRT phase: Assume that the database contains a set of panoramas. When a video segment is taken nearby the center of one panorama, it is desired to recognize the video content by correctly finding this corresponding panorama in the database and matching each frame in the video segment with an appropriate region in the panorama.

2. PGVRT

Let a series of image frames contained in a video segment be $\mathbf{F} = \{f_1, ..., f_N\}$ and define the set of

[1] For example, the high-level descriptions may include the location of shooting, the viewing direction, names of the observed buildings or landscapes, and so on.

1051-4651/02 $17.00 (c) 2002 IEEE

panoramas contained in the environment database to be $\mathbf{P} = \{P_1 \ldots P_M\}$. 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 **P.** [2] In this phase, we formulate the problem of recognizing the series of frames as the problem of searching the optimal path in a specially designed graph. Our method can be divided into three stages, the *candidate-selection* stage, the *graph-construction* stage, and the *path-searching* stage, respectively, as subsequently introduced in the following.

A. Candidate-Selection Stage

In the candidate-selection stage, template-matching technique is used for the selection of several candidate regions in the panoramas contained in **P**. Several scaled versions of each panorama were kept for multi-scale template matching. Let a set of ascending scaling factors be $\mathbf{S} = \{s_1 \ldots s_L \mid 0<s_1 <s_2<\ldots<s_L\}$. Assume that a panorama, P_j, with the width and height being respectively W_j and H_j, is contained in the database. Its i-th scaled panorama, P_j^i, is an $(s_iW_j)\times(s_iH_j)$ image generated by linearly scaling P_j. Each frame in **F** is treated as a template for block matching to every scale of the panoramas in **S**. The purpose of this stage is to find, in each scaled panorama, all the blocks whose matching costs are smaller than a given threshold.

The CIE $L^*u^*v^*$ space [4] is adopted in our work for block matching because the Euclidean distance in this color space considerably matches human perception about the differences 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. The L vector is then normalized as $\underline{L} = (L-L_0)/\|L\|$, where L_0 is a d-dimensional vector with all of its values being *mean(L)*, the average of the d values contained in L. Such a normalization is to make the Euclidean distance between two normalized blocks, $\|\underline{L}_1-\underline{L}_2\|$, 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 $a_1L_1+b_1$ and $a_2L_2+b_2$, respectively for all a_1, b_1 a_2, $b_2>0$. 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 required property for the range search part that will be introduced later.

The matching cost between two blocks I_1 and I_2 is defined to be $S(I_1,I_2) =$

$$w_1\cdot(\|\underline{L}_1-\underline{L}_2\|) + w_2\cdot(\|U_1-U_2\| + \|V_1-V_2\|), \quad (1)$$

where $w_1>0$ and $w_2>0$ are the weights for the lighting and chromatic matching costs, respectively.

Given an image frame f_i, all the blocks contained in the scaled panoramas in **P** are considered and those have matching costs smaller than a threshold, T_m, are served as the candidate blocks for f_i. To provide suitable low-pass filtering effects for increasing matching correctness, the frames and blocks are smoothed and normalized to be 32×32 in our implementation. To improve the matching efficiency, a range-search approach is adopted in our work [2]. In this approach, a geometrical near-neighbor access tree (GNAT) can be constructed for improving the matching efficiency about finding similar blocks within a range, say, T_m. A metric space with an associated distance function obeying the triangle inequality has to be used as a distance or cost measure to ensure the correctness of searching via GNAT. The definition (1) used in our approach is a distance measure satisfying the above requirement. A GNAT is designed to have a data structure that reflected the intrinsic geometry of the underlying data. This is achieved as a hierarchical, Dirichlet domain based structure. This hierarchical structure can be used to prune the nodes impossible to be contained in a range of f_i. The search time, compared to that of directly performing template matching of f_i to each scaled panorama, can thus be reduced. In our experience, using the GNAT approach can save more than a half time for candidate selections than using direct template matching.

B. Graph-Construction Stage

In the graph-construction stage, inter-frame relationship is used to increase the matching reliabilities. In the matching graph, the candidate regions selected in the first stage represent nodes. The edges are constructed by linking those nodes associated with adjacent frames. That is, there are directed edges coming from nodes associated with f_{i-1} to 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 a pair of blocks associated with an edge is too long, then this edge will not be constructed either. Two additional nodes, the source node and sink node, are built and the edges respectively connecting each of them with the layer *1* and the layer N are also constructed. Fig. 1 shows an example of

[2] In essence, this work assumes that the image frames are taken with a video camera whose optical center is roughly parallel to the ground plane (that is, its up-vector is roughly vertical to the ground). This assumption is suitable for most applications and is very helpful for reducing the complexity of matching.

the matching graph.

In a matching graph, each node and edge is assigned with a score, respectively. The cost of a node is assigned with the matching cost defined in (1). The costs of the edges connecting with the source and the sink nodes are set to zero. The cost of each of the other edges is defined as a weighted sum of the three components, the *motion-continuity* component, the *scale-continuity* component, and the *scene-consistent* component, as introduced in the following:

- The motion-continuity component:

In our work, the camera is assumed to move in a continuous manner, and thus the distance between consecutive matched blocks in a panorama has to be small. The cost of this component is defined by:

$$motion\_cost = \sqrt{(r - r')^2 + (c - c')^2}\ , \qquad (2)$$

where (r, c) and (r', c') are the centered positions of the consecutive blocks.

- The scale-continuity component:

In addition, assume that the effect of zooming in and out is fluent. Hence, if the consecutive blocks are from different scales s_i and s_j, $1 \le i, j \le L$, $i \ne j$, 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:

$$scale\_cost = | i - j | / L \qquad (3)$$

- The scene-consistent component:

Assume that the contents of the consecutive blocks are similar. The cost of this component is also defined based on (1):

$$consistent\_cost = S(B_1, B_2), \qquad (4)$$

where B_1 and B_2 are two consecutive blocks.

C. Path-Searching Stage

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 the recursive programming, the Dijkstra algorithm is used to find the optimal path [3]. For each node, an incoming edge with the highest accumulated score is kept in our approach. After finding the best incoming choice for all nodes, our process tracks back, from the sink to the source nodes, to obtain an optimal path. These nodes in the optimal path are then treated as a sequence of matched blocks to the input frames.

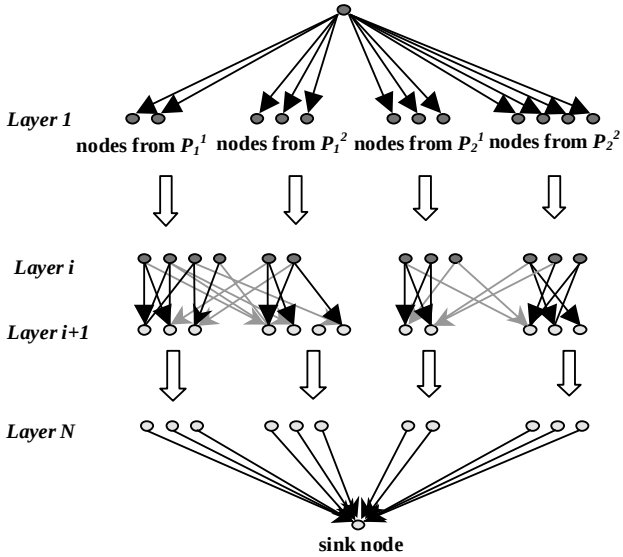

Figure. 1. This figure illustrates an example of the matching graph where 2 panoramas are contained in the database and 2 scales are used for each panorama. This graph contains N layers (in addition to the source and the sink node). Without loss of generality, we show a typical construction of nodes and edges for frames f_{i-1} and f_i in the middle part of this figure. 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 not be constructed either. Those edges across different scales of panorama are drawn with gray color.

3. Experimental Results

In our experiment, twelve panoramas taken from a campus were used to construct an environment database. These panoramas include both indoor and outdoor scenes, and four of them are shown in Fig. 2. In the pre-processing phase, all the panoramas were augmented with an environment name (e.g., the plaza or square names), and some particular regions of the panoramas were further augmented with corresponding high-level descriptions (e.g., building or road names), as shown in Fig. 3. A test video containing 180 frames was taken and some of them are shown in Fig. 4(a). Twenty frames in this video were used to build a matching graph, and the shortest path was then found. The matching regions of the other frames were obtained by interpolating the positions and scales of those found with the twenty frames. After using our approach, it was successfully recognized that the test video was roughly taken in the same environment of the 7^{th} panorama in the database as shown in Fig. 2(b), and a series of matched regions in this panorama was also obtained, as shown in Fig. 4(b). The recognition time is 0.56 seconds per frame. From this example, it is observed that quite convincing recognition results can

be obtained with our approach. Moreover, the high-level descriptions about the matched regions can then be identified, as shown in Fig. 4(d).

Figure 2. Four of the twelve panoramas contained in the database. (a) the 2nd panorama: Gymnasium 2F, (b) the 7th panorama: Taichi Plaza, (c) the 8th panorama: Hu Shih Memorial Hall, (d) the 10th panorama: Institute of Information Science 1F.

4. Conclusions and Discussion

This paper shows a framework for recognizing the scenes captured with a video camera. Contributions of this paper are listed as follows:

1. A scenario is proposed about using an image sequence, instead of single images, for appearance-based recognition and tracking. It demonstrates 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. In fact, existing appearance-based learning methods suffer from that multiple images have to be taken for each target. In addition, a multiple-image representation can only sample finite views of a target. However, a panorama inherently records infinite many viewer-center images in a single omni-directional image. It is therefore a suitable compact representation for appearance-based visual recognition and tracking.

References

[1] M.J. Black and A. Jepson, "EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation," *IJCV*, 26(1), pp. 63-84, 1998.

[2] S. Brin, "Near Neighbor Search in Large Metric Spaces," *Proc. 21st Very Large Database (VLDB) Conference*, 1995.

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

[4] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall International, London, 1989.

[5] H. Murase and S. K. Nayar, "Visual Learning and Recognition of 3-D objects from Appearances," *IJCV*, Vol. 14, pp. 5-24, 1995.

[6] H. C. Huang, Y. P. Hung, "Panoramic Stereo Imaging System with Automatic Disparity Warping and Seaming," *Graphical Models and Image Processing*, Vol. 60, No. 3, pp. 196-208, 1998.

[7] K. Ohba and K. Ikeuchi, "Detectability, Uniqueness, and Reliability of Eigen Widows for Stable Verification of Partially Occluded Objects," *IEEE Trans. PAMI*, Vol. 19, pp. 1043-1048, 1998.

[8] C. Papageorgious and T. Poggio, "A Pattern Classification Approach to Dynamic Object Detection," *Proc. ICCV'99*, Corfu, Greece, pp. 1223-1228, 1999.

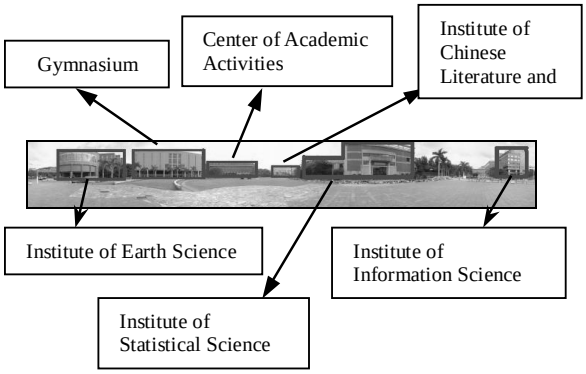

Figure 3. An example of the panorama that is augmented with some high-level descriptions.

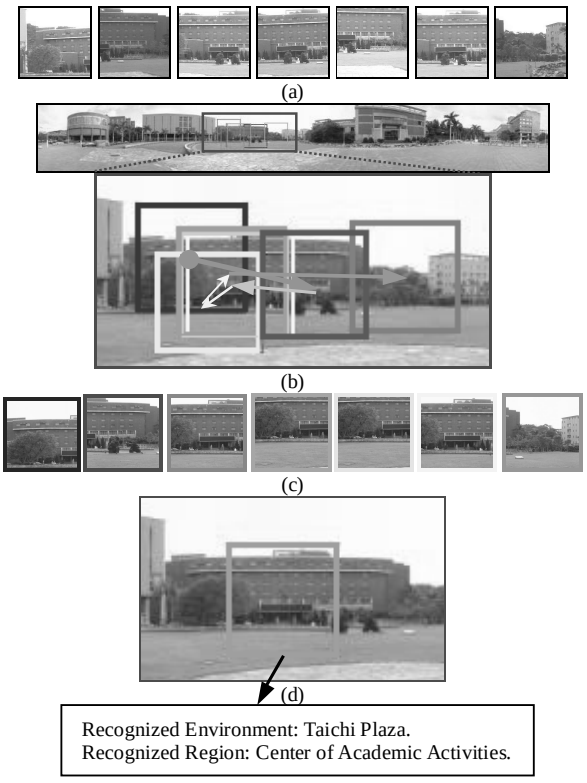

Figure 4. (a) Seven of the 180 frames of the test video that was taken by roughly panning around the camcoder in front of a building. These frames are respectively the 0th, 27th, 54th, 81th, 108th, 135th, and 162th frames. (b) The 7th panorama in the database has been recognized for the test video with our method. (c) The matched regions in the panorama. (d) High-level descriptions about the recognized views.