Feature-Based Displacement Field Estimation for Visual Tracking By Using Coarse-to-Fine Block Matching*

Yong-Sheng Chen and Yi-Ping Hung

Institute of Information Science
Academia Sinica, 11529, Taipei, Taiwan
hung@iis.sinica.edu.tw

Abstract

Visual tracking is very important for many applications in computer vision, such as traffic control, intelligent surveillance, industry automation, human computer interaction, etc. In this paper, we propose a hierarchical framework for estimating displacement vectors at feature points with the coarse-to-fine block matching method. Feature points are chosen to be the image locations having large curvature, i.e., the corners. First, we calculate the pyramid for each image. At the toppest layer of the pyramid, corners and their displacement vectors are estimated with the block matching method. These displacement vectors are used to predict and narrow down the search region of the following lower layer of the pyramid. Early jump-out technique is also used to enhance the efficiency of block matching. Reliable displacement fields are obtained efficiently at the lowest layer of the image pyramid.

1 Introduction

Visual tracking is very important for many applications in computer vision, such as traffic control, intelligent surveillance, industry automation, human computer interaction, etc. Useful information including 3D motion parameters and scene geometry can also be obtained by tracking moving objects in an image sequence.

There are three major approaches for tracking: differential methods (optical flow estimation), phase correlation methods, and block matching methods. Computation of optical flow field by differential methods [4, 6] is sensitive to noise and needs careful design of filter to remove unreliable flow vectors. The other drawback of this type of methods is that they can only achieve short-term optical flow. Phase correlation methods utilize the discrete Fourier transform to obtain the phase information. Displacement field can be estimated by phase correlation. These methods are also very sensitive to noise because noise will contaminate high frequency component, thus reduce the accuracy of the estimated displacement field. Block matching methods produce more reliable displacement fields, but they are very time-consuming. This kind of methods will be more attractive if some accelerator, such as the early jump-out (EJO) technique or the hierarchical method proposed in this paper, can be used to improve the efficiency.

The displacement vectors are more accurate at salient feature points because these points contain more information due to their larger intensity variation. Although features such as edges and lines are important visual information, they may suffer the aperture problem because the image patch belonging to a straight edge is similar to others along the edge direction. Corners at the intersection of edges will not suffer the aperture problem because the image patch at the corner is different from others along the edge direction. In this paper, we use a fast curvature estimation method proposed by Cooper et al. [3] to detect corners, and compute the displacement vectors only at these corner feature points.

This paper proposes a feature-based displacement field estimator with coarse-to-fine EJO block matching, to meet the requirements of efficiency and reliability. First, a pyramid is calculated for each image. At the toppest layer of the pyramid, corner features are detected at the position with large curvature along the edge. Then the coarse displacement vectors at these corner points are estimated with an adaptive EJO block matching. The displacement vectors estimated at corner feature points are more accurate because corners contain rich

*This research work was supported in part by the National Science Council of Taiwan, under Grants NSC 85-2213-E-001-016.

information on intensity images. The displacement vectors are refined at the following lower layer of the pyramid by using the coarse displacement vectors obtained from the upper layer to predict and restrict the search area. Experimental results show that reliable displacement vectors can be obtained at the rate of about 3 frames per second using Sparc 20 without optimization.

We will briefly review some approaches of tracking with block matching methods in section 2. Creation of image pyramid, corner detection, and coarse-to-fine searching are described in section 3. Some experimental results and comparisons are shown in section 4. Finally, we will give some brief conclusions in section 5.

2 Tracking with Block Matching

The simplest block matching is to find the corresponding pair of pixels that minimize the sum of square differences of the two image blocks centered at the two corresponding pixels. Early in 1972, Barnea and Silverman [1] introduced a class of sequential similarity detection algorithms for expediting the similarity detection between two structured data sets. Their contribution was to propose a monotonically-increasing threshold sequence algorithm where a threshold sequence could be defined such that if, at any accumulation stage in the computation of the mean square error or the mean absolute error, the partial result was greater than the corresponding threshold in the sequence, one could jump out of the similarity test. Recently, Cooper et al. applied this sequential algorithm to the dissimilarity test in corner detection, and called it the Early Jump-Out technique [3].

While accumulating the intensity differences in the block, when the sum is large enough according to a predetermined threshold sequence, we can judge these two blocks are not similar and stop accumulating intensity differences of other pixels in the blocks. Cooper et. al [3] proposed this early jump-out concept. Time consumption is significantly reduced if most of block matchings are terminated after test first several pixels in the block. Apparently, the efficiency is depending on the values of the threshold sequence.

Tsao et al. [7] have used some training methods to learn the EJO threshold sequence before tracking. Huang et al. [5] recently proposed an adaptive early jump-out algorithm to learn the threshold sequence on-line. This adaptive technique can be embedded into almost all the existing block matching algorithms of motion estimation and can speedup the computation further. This technique has been used in video coding to greatly reduce the computational cost of motion estimation with little degradation on video quality.

3 Hierarchical Feature-Based Displacement Field Estimation

This section describes the hierarchical framework of the coarse-to-fine block matching at corner points. Consider Figure 1. First, we create a pyramid for each image. Then a fast corner detection algorithm is applied on the top layer of the image pyramid. Initial displacement vectors at these corners are estimated with an adaptive EJO block matching. More accurate displacement vectors are estimated with local searching by using the initial displacement vectors obtained from the upper layer of the image pyramid.

3.1 Image Pyramid Creation

An image pyramid is a group of images whose size and resolutions are decreased at a regular step between neighboring levels. In this work, a pyramid is created for each image in an image sequence. The original image is copied to the bottom layer of the pyramid. The image of successively upward level is obtained by smoothing and sampling every other pixel of the lower level image. Let pixel $(r, c) \in I_l$, The intensity $I_{l-1}(\frac{r}{2}, \frac{c}{2})$ at level $l-1$ can be calculated by:

$$I_{l-1}(\tfrac{r}{2}, \tfrac{c}{2}) = \tfrac{1}{4}\big(I_l(r,c) + I_l(r+1,c) + I_l(r,c+1) + I_l(r+1,c+1)\big)$$

Once the image pyramid is created, the size of matching block B_{l-1} and search region S_{l-1} at layer $l-1$ can be determined by the following equation:

$$B_{l-1} = \frac{B_l + 1}{2} \quad | \quad 1$$

$$S_{l-1} = \frac{S_l + 1}{2} \quad | \quad 1$$

which | is the bitwise logical **OR** operator.

The reason of using the above formula for calculating B_{l-1} and S_{l-1} is to meet the following two requirements. First, the decreasing factor of image size is $\frac{1}{2}$. Second, the size of image block and search region must be odd

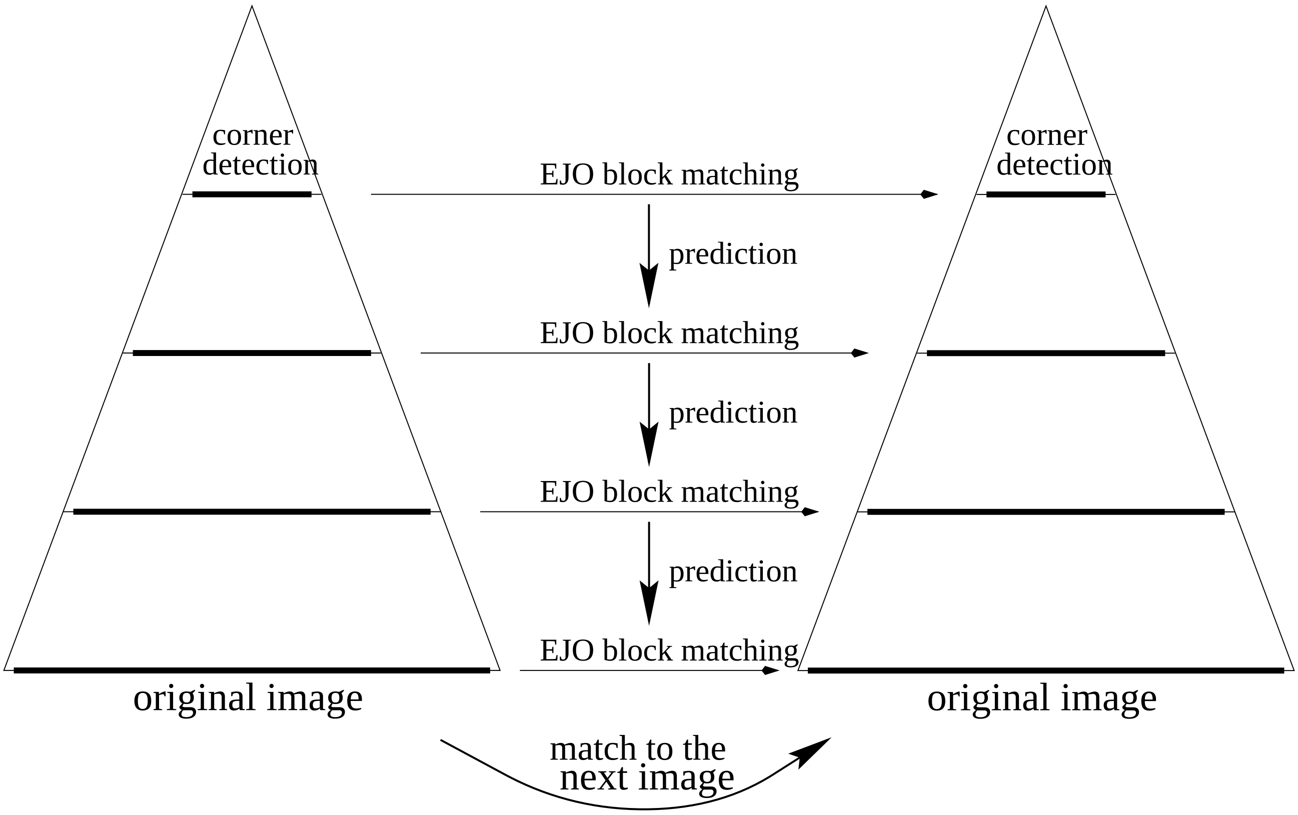


Figure 1: Illustration of the hierarchical block matching method.

example 1			
level	image size	block size	search area
1	64	3	3
2	128	5	3
3	256	7	5
4	512	11	9

example 2			
level	image size	block size	search area
1	64	5	5
2	128	9	7
3	256	15	13
4	512	27	23

Table 1: Two examples illustrate the relation between the image size, block size, and search area for a pyramid with four levels. In example 1, the size of image block, B_4, at level 4 is 11, search area, S_4, is 9. In example 2, B_4 is 27 and S_4 is 23. Computational time of block matching is significantly reduced because of the large decreasing rate.

number for symmetry. Table 1 gives two examples which illustrate the decrease of the image size, image block size, and search area for a pyramid with four levels.

3.2 Corner Detection

Salient image feature point contains more information because of larger intensity variation. This leads to better reliability of displacement vector estimation at salient point. In this work, a fast method proposed by Cooper et al. [3] is used to detect the corner position. First, Sobel operator is applied to the image to estimate edge magnitude and orientation. Curvature of each point on the edge is calculated according to the orientation of each point along the edge. Corner position is reported if both the edge magnitude and the edge curvature are larger than corresponding thresholds. These two thresholds control the amount of corner points. Sometimes there are multiple responses in the neighborhood of corner points. We use an additional suppression algorithm to eliminate all the feature points except the most salient one within its suppression region.

3.3 Coarse-to-Fine Searching

On the top layer of image pyramid, the coarse displacement vectors at the corner points are estimated by using the adaptive EJO block matching within a preset search region as shown in table 1. Coarse displacement vectors obtained in the upper layer are used as an image prediction in the next lower layer. Refinement is achieved by using local search within a 3x3 area centered at the predicted displacement vector. Let the area of original search region at the bottom layer be S^2 and local search area is 3x3, the total search area of coarse-to-fine search with l pyramid layer is reduced from S^2 to:

$$(\frac{S}{2^{l-1}})^2 + 9 \times (l-1)$$

When pyramid level l is 3, the search area is reduced from S^2 to $\frac{S^2}{16} + 18$.

4 Experimental Results

This section presents some experimental results obtained by using the coarse-to-fine block matching methods. The experiments are executed on a Sparc 20 workstation. In the first image sequence as shown in Figure 5(e), there is a CocaCola can moving from right to left on a conveyor belt in a static environment. The image pyramid is created, as shown in Figures 5(a)–5(d). At the toppest layer of the pyramid (see Figure 5(a)), corner features are detected and marked by black boxes. Displacement vectors are estimated at these corners via block matching in a small search region. Only nonzero displacement vectors are drawn in this figure. These displacement vectors are used to predict and narrow down the search region to 3x3 at the lower (finer) layer of the image pyramid.

method	corner detection	motion estimation	total
with hierarchical	0.192 sec	0.1425 sec	0.3345 sec
without hierarchical	0.903 sec	13.96775 sec	14.87075 sec

Table 2: The average computation time for corner detection and motion estimation with and without the hierarchical coarse-to-fine method. The hierarchical coarse-to-fine method is about 44 times faster.

By repeating this process, we can get the desired displacement field at the bottom layer of the pyramid as shown from Figures 5(b) to 5(d).

The image size of the first image sequence is 512 by 512. Table 2 shows the average computation time for corner detection and displacement field estimation with and without the hierarchical coarse-to-fine method. In this experiment, we use 7x7 matching block and 19x19 search area. The pyramid level of the hierarchical method is 3 and the local search area is 3x3. We can see that our hierarchical coarse-to-fine method is about 44 times faster.

The other two well-know image sequences are acquired from the FTP site at the university of Western Ontario, Canada. The first one, as shown in Figure 6, contains a plate rotating counterclockwise with a cubic on top of it. The surface of the plate has no feature point at all. Hence no motion is detectable in this region. On the other hand, both the side of the plate and the cubic contain many feature points. Displacement vectors can be estimated accurately on these points.

Figure 7 shows another image sequence in which there are three moving cars: the white taxi turning right around the street corner, the black sedan at the left side of the road moving rightward, and the truck at the right side of the road moving leftward. The black sedan at the left side is not detected because its color is too similar to the the color of the road.

To compare the efficiency of the three displacement field estimation algorithms: hierarchical algorithm, adaptive EJO algorithm, and hierarchical+adaptive EJO, we synthesize a sequence of seven images with five moving blocks having graylevel values of 80, 120, 160, 200, and 250. In Figure 3, we use matching block of different sizes, ranging from 3x3 to 29x29, to test these three algorithms. The image pyramid level is 3. The increase of computation time for the adaptive EJO algorithm is large. When the size of matching block is smaller than seven, the computation time of the hierarchical algorithm, with or without adaptive EJO, decreases as the size of matching block increases. From Table 1, we can see that when the pyramid level is 3, the most efficienct estimation is achieved when the block size is equal to or larger than 7. When the size of matching block is larger than 7, the computation time increases very slowly, especially when using the hierarchical method together with adaptive EJO algorithm.

In Figure 4, we use different sizes of search area range from 3x3 to 29x29. We can obtain similar result. From Figure 3 and Figure 4, the computation time of the hierarchical method increase slower when the search area increases than when the block size increases. This agrees with our intuition that the hierarchical method is especially useful when the search area is larger, i.e., the motion of object in images is larger.

5 Conclusions and Discussions

In this paper, a hierarchical block matching method for feature-based displacement field estimation is developed to estimate the displacement vectors between two consecutive images efficiently and accurately. Reliability is achieved because corner points are rich of information and the block matching results at the corners are more accurate. We check the curvature along the edge to detect corner points. Efficiency is achieved by using the adaptive EJO technique and the hierarchical framework.

We have used averaging smoothing for image pyramid creation in this paper. The Gaussian pyramid maybe more suitable but requires more computation. By using the coarse-to-fine technique to narrow down the search area, we can reduce the computation time tremendously. However, if the coarse matching result is wrong, there is no way to get the correct displacement vector at the finer matching, which uses the prediction obtains from the coarse matching. According to Table 2, the computation time of corner detection dominates the total computation time of our hierarchical coarse-to-fine method. We shall use faster method to detect corners in order to further speedup the computation of displacement vectors.

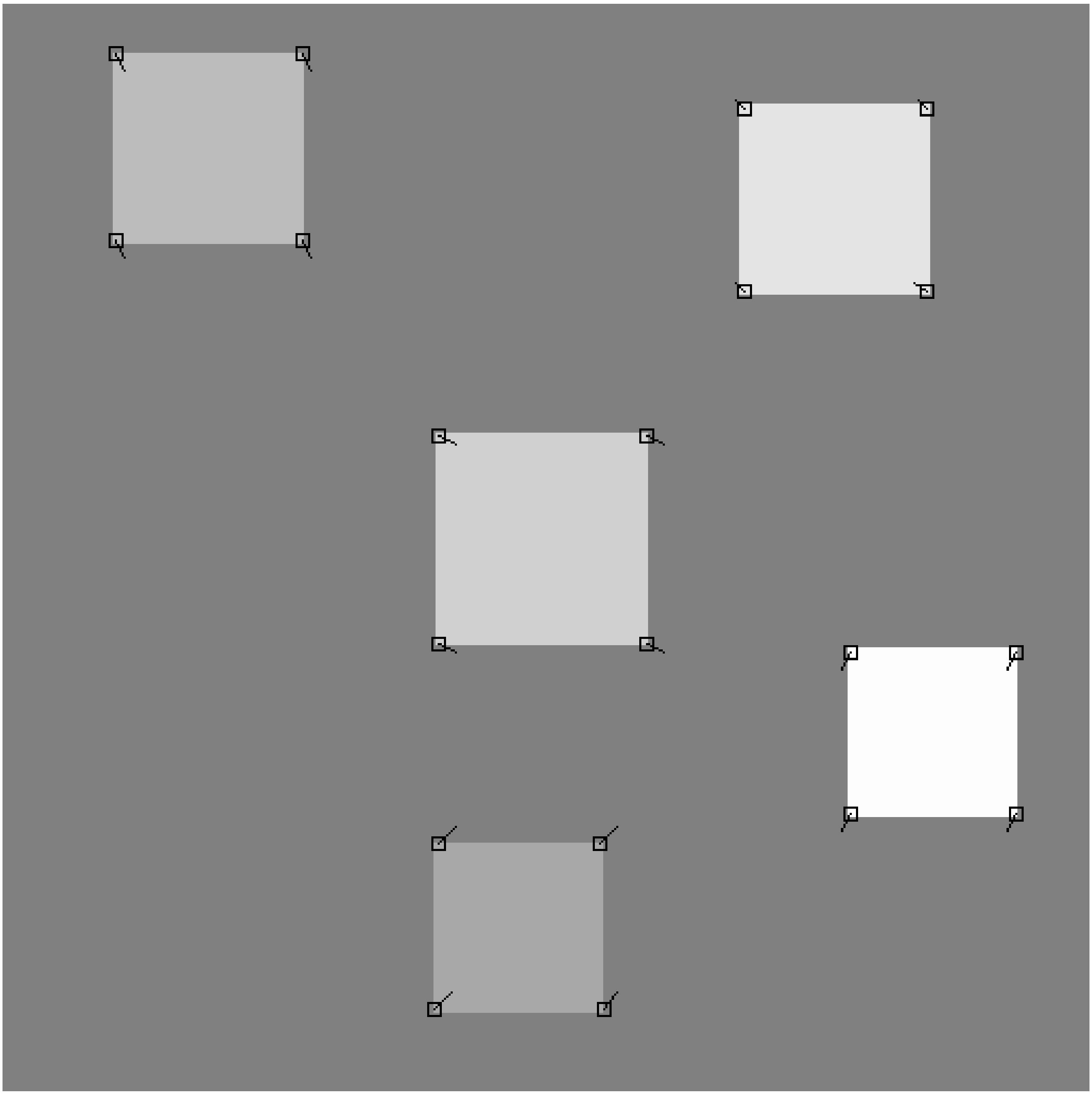


Figure 2: A synthesized image with five moving blocks.

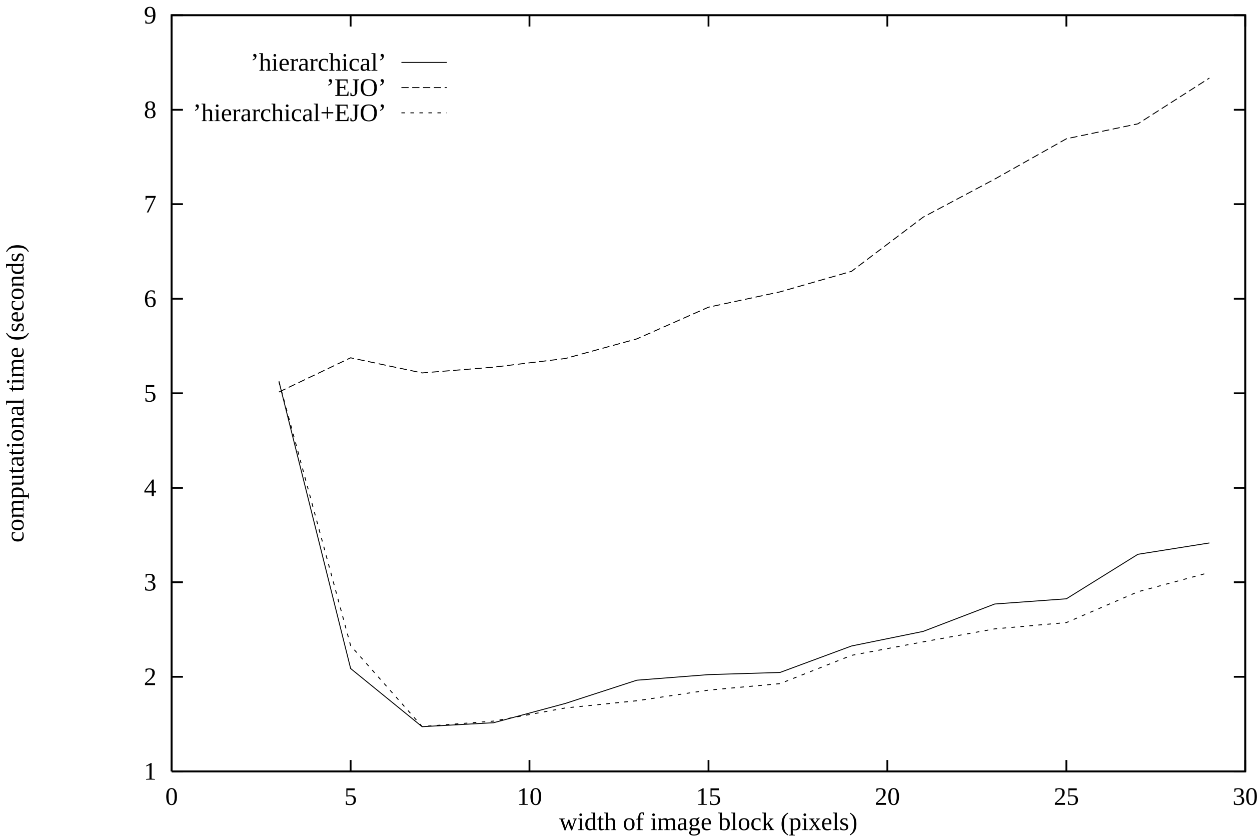


Figure 3: The computation time for the three algorithms: hierarchical algorithm, adaptive EJO algorithm, and hierarchical with adaptive EJO algorithm. The vertical axis is the total computational time of the displacement field estimation over seven images. The horizontal axis is the size of matching block.

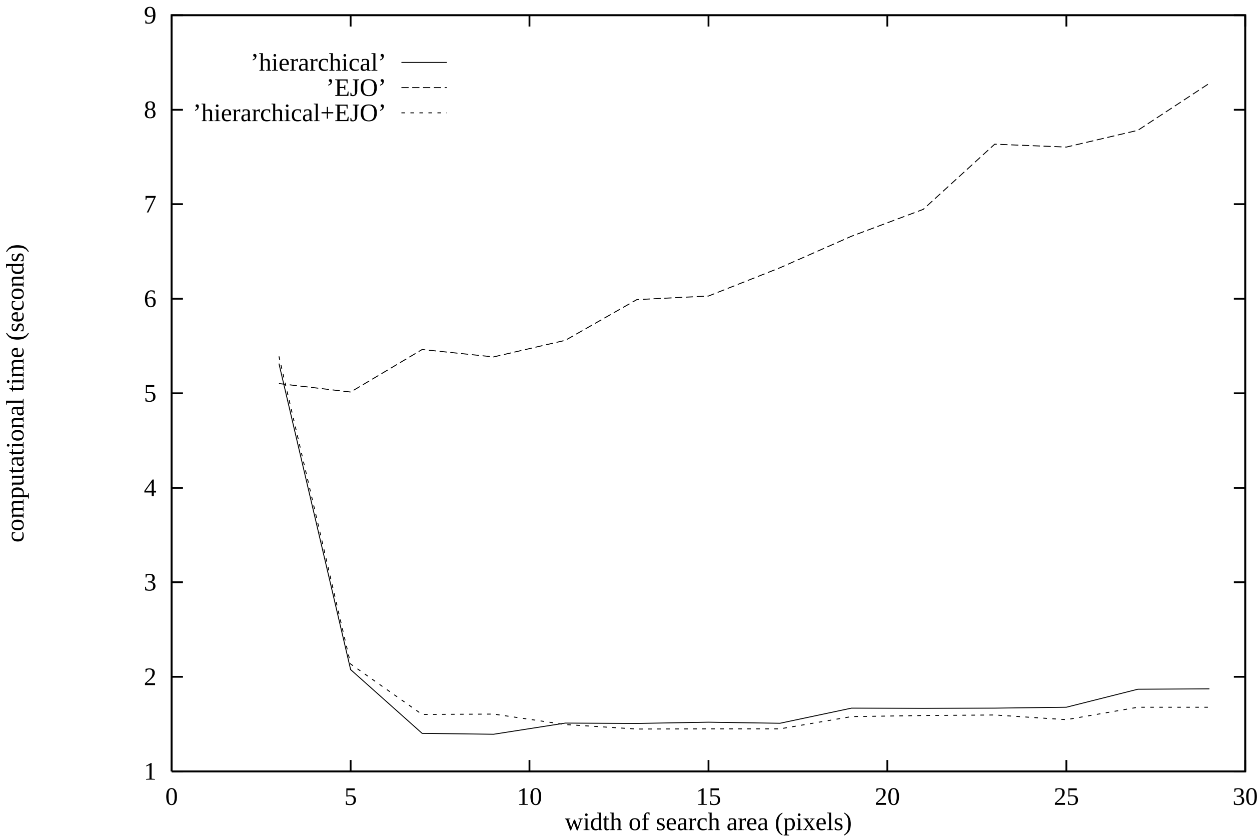


Figure 4: The computation time for the three algorithms: hierarchical algorithm, adaptive EJO algorithm, and hierarchical with adaptive EJO algorithm. The vertical axis is the total computational time of the displacement field estimation over seven images. The horizontal axis is the size of search area.

References

[1] D. I. Barnea and H. F. Silverman, "A Class of Algorithms for Fast Digital Image Registration," *IEEE Trans. Comput.*, Vol. 21, pp. 179-186, 1972.

[2] P. J. Burt and E. H. Adalson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Trans. on Communications*, COM-31(4), pp. 532–540, April, 1983.

[3] J. Cooper, S. Venkatesh, and L. Kitchen, "Early Jump-Out Corner Detectors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, No. 8, pp. 823–828, August 1993.

[4] B. K. P. Horn and B. G. Schunck, "Determining Optical Flow," *Artificial Intelligence*, vol. 17, no. 1–3, pp. 185–203, August 1981.

[5] H. C. Huang, Y. P. Hung, and W. L. Hwang, "Adaptive Early Jump-Out Technique for Fast Motion Estimation in Video Coding," *the Proceedings of ICPR*, Volume II, Track B, pp. 864–868, Vienna, Austria, August 1996.

[6] B. G. Schunck and B. K. P. Horn, "Constraints on Optical Flow Computation," *Proc. Pattern Recognition and Image Processing Conference*, pp. 205–210, Dallas, Texas, August 1981.

[7] A.T. Tsao, Y. P. Hung, C. S. Fuh, and H. Y. M. Liao, "On Learning the Threshold Sequence for the Early Jump-Out Template Matching," *Proceedings of TAAI Artificial Intelligence Workshop*, Taipei, Taiwan, pp. 186–192, 1995.

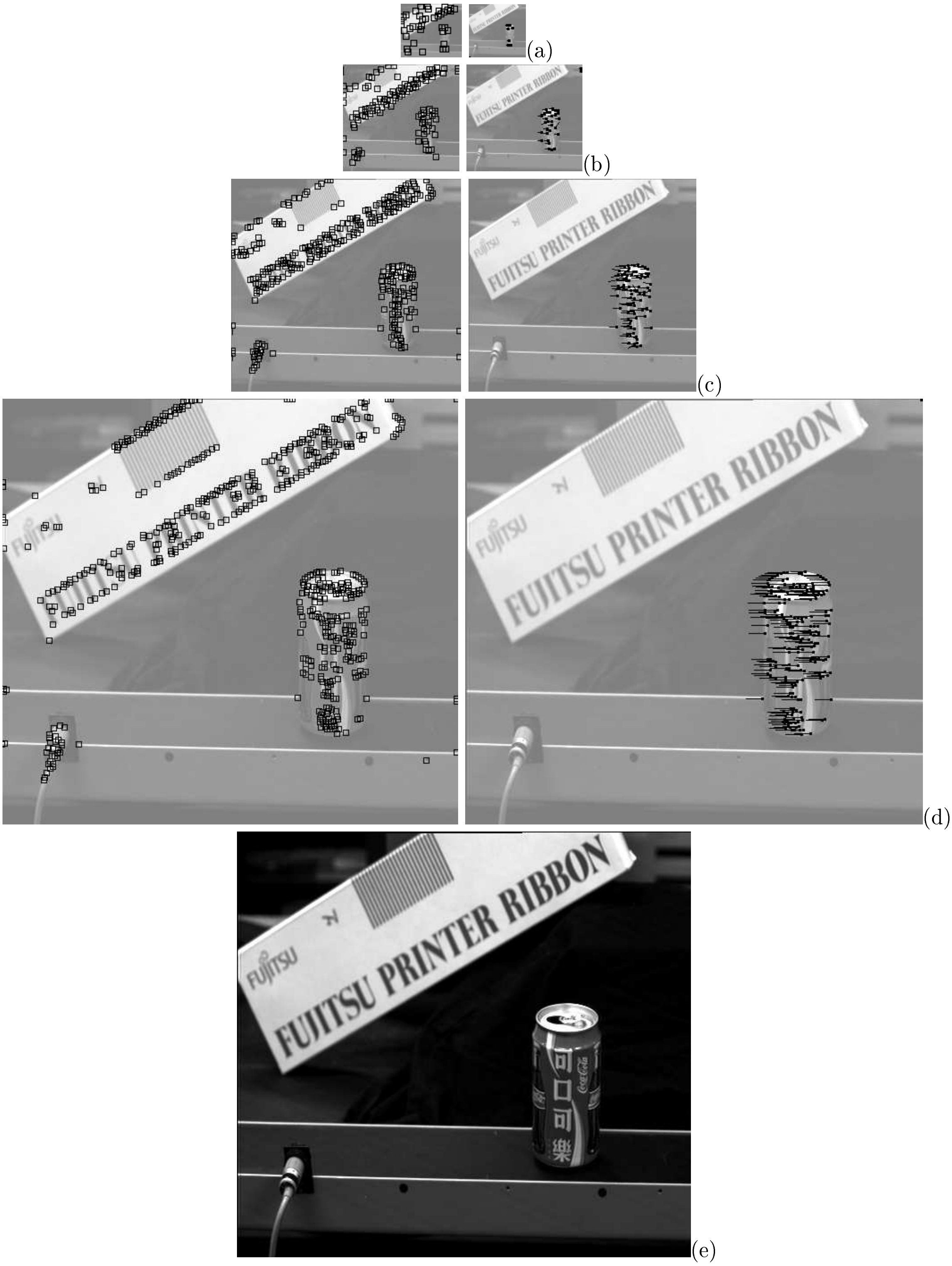
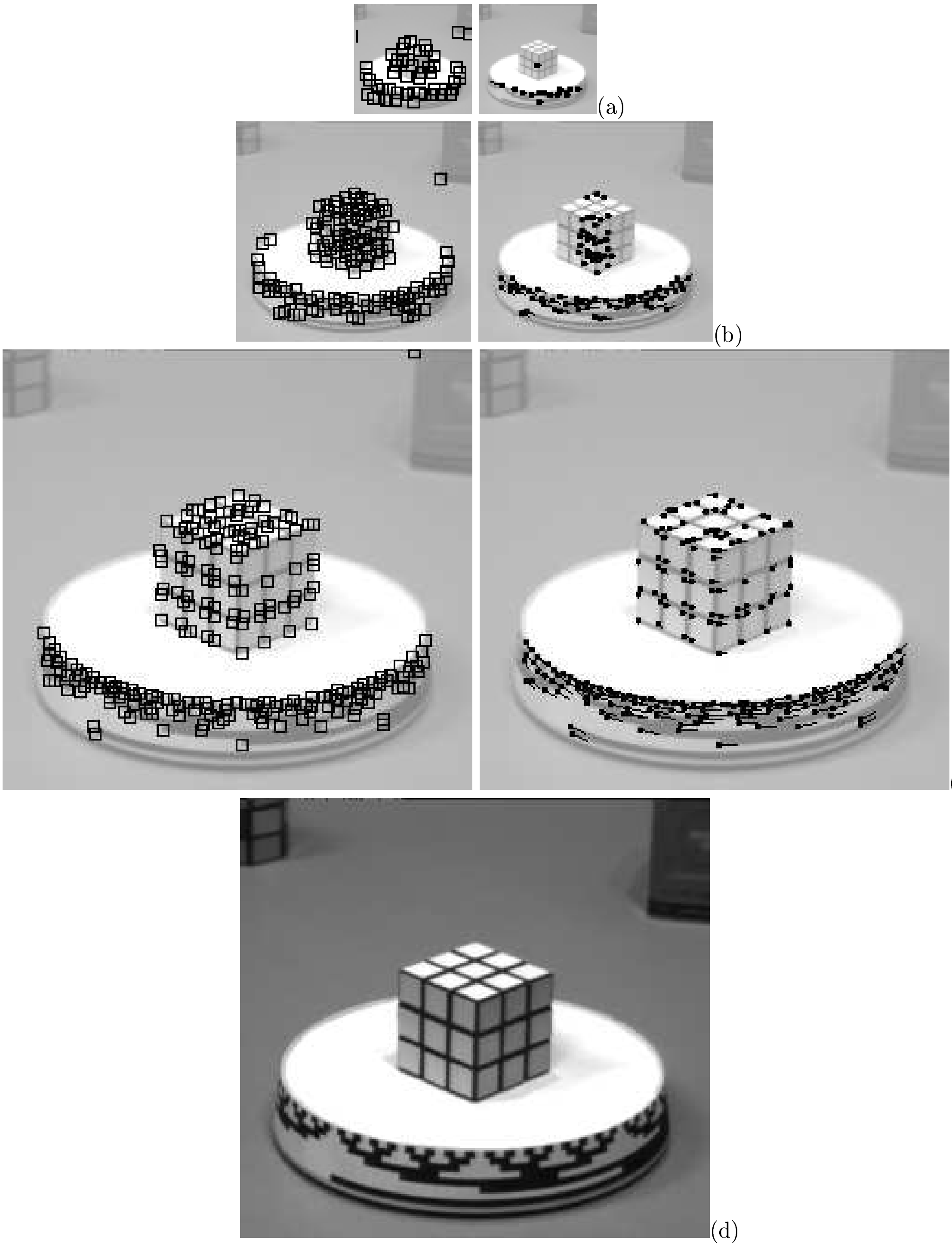


Figure 5: CocaCola can image sequence: containing a moving CocaCola can with displacement vectors obtained at corners.



(a)

(b)

(c)

(d)

Figure 6: Cubic image sequence: containing a rotating plate and a cubic with displacement vectors obtained at corners.

Figure 7: Taxi image sequence: containing moving cars with displacement vectors obtained at corners.