MVA '96 IAPR Workshop on Machine Vision Applications, November. 12-14, 1996, Tokyo, Japan

Visual Tracking using Active Search for Color

V V Vinod * Hiroshi Murase†
NTT Basic Research Laboratories
Atsugi-shi, Japan

Abstract

A novel technique for tracking arbitrary colored objects in complex environments is proposed in this paper. Active search and prediction of search area is used for improving speed. The system is capable of accurately tracking at the rate of around 10 frames per second without any special hardware.

1 Introduction

Visual tracking of objects in a sequence of frames is an important task in computer vision. The major approaches for this task include Kalman filters for tracking geometric features, nearest neighbor methods under small frame to frame motion, statistical method for target tracking etc. [1, 3]. Such methods are often limited to rigid objects whose motion closely approximate some underlying mathematical model. Often, in applications such as virtual studios [2, 5], arbitrary non-rigid objects have to be tracked. In the absence of efficient methods, head mounted cameras and other such artifacts are still commonly employed in virtual studio systems. A method for efficiently tracking arbitrary objects in complex environments would be very useful in such situations. We propose a novel approach for tracking arbitrary, rigid or non-rigid objects based on color.

Several recent works [4, 10, 9] have shown that color histogram matching constitutes a powerful technique for object identification, image retrieval, indexing etc. Vinod and Murase [8, 9] proposed focused color intersection for efficiently detecting colored objects in complex scenes. In this paper we employ focused color intersection for detecting objects based on their color distributions. Active search is employed for efficiently searching for the position and size of the object in a frame. This results in huge speed improvements without affecting accuracy. The search area for a frame is statistically predicted based on previous frames to achieve higher frame rates. In the present paper, only the object's color is used to distinguish it. Other features can be easily incorporated in the proposed framework.

*Address: 3-1 Morinosato-Wakamiya, Atsugi-shi, 243-01 Japan. E-mail: vinod@eye.brl.ntt.jp
†Address: 3-1 Morinosato-Wakamiya, Atsugi-shi, 243-01 Japan. E-mail: murase@eye.brl.ntt.jp

The tracking system is presented in section 2, followed by the experimental results in section 3 and conclusions in 4.

2 Tracking System

The problem considered, may be stated as *determine the part of each frame which, at some scale, has the highest similarity with the model.* Let $S^k_{xy}(t)$ denote the similarity of the model with a part of the frame at time t, where k denotes the size of the image part and x, y its position. Then, the problem of tracking may be specified as

$$\forall t \text{ determine } q,\ i,\ j \text{ such that} S^q_{ij}(t) = \max_{k,x,y} S^k_{xy}(t)$$

In section 2.1, we discuss the definition of $S(.)$. Active search, presented in section 2.2, efficiently determines the maximum in a given frame. The area of a frame on which active search is performed is determined based on previous frames in section 2.3.

2.1 Focused Color Intersection (FCI)

A brief overview of FCI is presented here with respect to a $N \times N$ image. Extensions to rectangular frames are straightforward. A detailed discussion on FCI may be found in [8]. FCI [8] evaluates the similarity of the model against focus regions in the image considering different sizes and positions. The focus regions are derived by resizing the image and scanning each resized image with square window of $w \times w$ pixels. The focus regions obtained from an $N \times N$ frame may be characterized as:

$$\begin{aligned} \{R^k_{xy}\} \quad & \text{where } k = w, w + \Delta k, \ldots N \\ & x, y = \frac{w}{2}, \frac{w}{2} + s, \ldots, k - \frac{w}{2} \\ R^k_{xy} \quad = \quad & p^k_{ij} \text{ where } x - \tfrac{w}{2} \le i < x + \tfrac{w}{2}, \\ & y - \tfrac{w}{2} \le j < y + \tfrac{w}{2} \end{aligned}$$

where Δk and s are given parameters and p^k_{xy} denotes a pixel belonging to the image resized to $k \times k$ pixels. Thus, k indicates size and x, y the position.

The normalized color histogram intersection [6] is used as a measure of similarity between a focus region and the model. Let H^k_{xy} and H^M denote the normalized color histograms of focus region R^k_{xy} of

frame at time t and the model respectively. Then the similarity is evaluated as

$$S^k_{xy}(t) = \sum_{i=1}^{b} \min\{H^k_{xy}(i), H^M(i)\}$$

where b is the number of histogram bins. FCI efficiently evaluates $S^k_{xy}(t)$ in $O(w^2)$ time from an indexed representation of the image and the model histogram. The focus region with the highest similarity denotes the object, if the similarity is above a given threshold θ.

2.2 Active Search

The number of focus regions in a frame may, in general, be quite large. Evaluating the similarity of the reference image against all the focus regions will be computationally very expensive. However, it may be observed that large parts of the frame will not contain the object and hence need not be examined in detail. Active search exploits this fact for determining the focus region with highest similarity without evaluating the similarity of all focus regions. Uninteresting focus regions are pruned by employing upper bounds on the similarity measure. The upper bounds are estimated as follows.

Consider two arbitrary regions A and B belonging to an image as shown in figure 1. Let the similarity of A be S_A. Then, we obtain an upper bound on S_B, the similarity of B, as:

$$S_B \leq \frac{\min(|A \cap B|, S_A|A|) + |B - A|}{|B|} \qquad (1)$$

where $|A|$, $|B|$, $|A \cap B|$, $|B - A|$ represent the number of pixels in region A, region B, number of pixels common to A and B and the number of pixels in B but not in A respectively. A detailed proof of this result may be found in [9]. Given two focus regions R^k_{ij} and R^p_{uv}, equation (1) is applied after projecting both the regions to the $N \times N$ image.

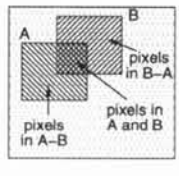

Figure 1: Two focus regions in an image

Active search will explicitly evaluate the similarity of a focus region only if the least among its upper bounds is higher than the upper bound cutoff at that time. The upper bounds and upper bound cutoff are updated during the course of the algorithm. The search starts with initial upper bounds of all focus regions set to 1.0 (the maximum similarity value) and the the initial upper bound cutoff set to the threshold θ. After each similarity evaluation, the upper bounds for all focus regions in the neighborhood are estimated based on the computed similarity value. Also the upper bound cutoff is updated to the best similarity value obtained till then. The algorithm terminates when all focus regions have been examined, i.e., either pruned or matched against the model. Now, strict upper bounds are used for pruning and the threshold θ will, in practice, be less than the highest similarity value. Therefore, active search will correctly determine the best matching focus region. Thus active search reduces computational cost without sacrificing accuracy.

From equation 1, it may be observed that the upper bound increases with two factors - the similarity of the neighboring focus region (S_A in the equation) and the number of pixels different between the two regions. Therefore, large neighborhoods of a focus region with low similarity will have low upper bounds. Consequently active search will prune large neighborhoods of focus regions with low similarity. That is, active search will explore only promising regions in the image, other regions are pruned. This effect is depicted in figure 2. Figure 2 shows the

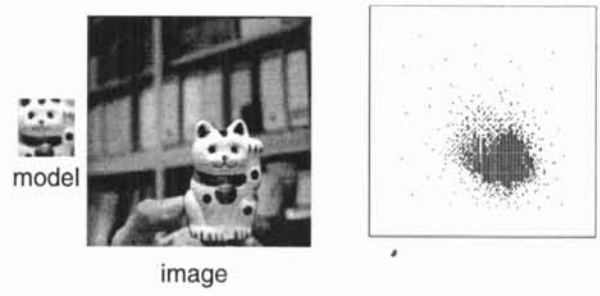

Figure 2: Similarity evaluations by active search

positions in the image at which active search evaluated the similarity value for the given image and reference. It may be observed that only the part of the image containing the object is densely searched. A comparison between exhaustive search and active search for a sequence of 128 × 128 pixel images is given in figure 3. The number of similarity evaluations obtained are plotted against the fraction of the image covered by the object. A threshold of 0.6 and $\Delta k = 1$, $s = 1$ and $w = 32$ were used. The number

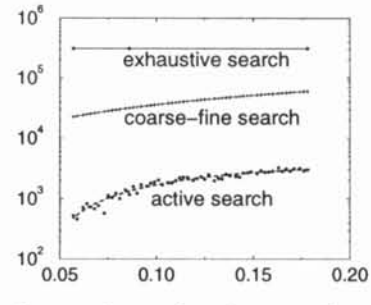

Figure 3: Comparison of active search and exhaustive search

of similarity evaluations are reduced by a factor of

1000 by using active search. In practice higher values of Δk and s will suffice, resulting in less number of focus regions. In practical situations with higher values of Δk and s, typically, a reduction in computation of around 10 times was observed. It may be noted that this gain in computation is achieved without sacrificing accuracy.

2.3 Predicting Search Area

Active search does not consider any focus region which has its initial upper bound less than the threshold θ (the initial upper bound cutoff). Hence we can predict the area to be searched in a frame by estimating the upper bounds and the threshold for that frame. Since the images change smoothly from frame to frame, especially at high frame rates, these quantities can be estimated from previous frames. For determining the search area for frame $t+1$ we estimate the following quantities from previous frames.

- θ_{t+1} - the threshold for frame $t+1$.
- $\tilde{S}^k_{xy}(t+1)$ - statistically estimated upper bound on the similarity of focus regions R^k_{xy} of frame $t+1$.

The threshold for frame $t+1$ is estimated as

$$\theta_{t+1} = \max(\theta, S^p_{uv}(t+1) \tag{2}$$

where R^p_{uv} is the focus region containing the object in frame t. For accurate results the threshold has to be less than the highest similarity in the frame. This requirement is trivially satisfied and accurate results are guaranteed. The gain in computation will, however, depend on how close the threshold is to the highest similarity in frame $t+1$. In most cases the position and size of the object will change smoothly from frame to frame. Therefore a large part of the object will be present in R^p_{uv} at time $t+1$. Thus in most cases θ_{t+1} will be close to the highest similarity in frame $t+1$.

$S^k_{xy}(t+1)$ may be written as

$$S^k_{xy}(t+1) = S^k_{xy}(t) + \Delta S^k_{xy}(t+1)$$

Since the color distribution of focus regions change smoothly from frame to frame, the frame to frame change in the similarity $\Delta S^k_{xy}(t)$ will be small. It can be approximated by a random variable with low variance. Let $\mu(\Delta S^k_{xy})$ denote the mean and $\mathrm{Var}(\Delta S^k_{xy})$ denote the variance of ΔS^k_{xy}. Then we have that

$$\mathrm{Prob}\left[\Delta S^k_{xy} \geq \mu(\Delta S^k_{xy}) + n\sqrt{\mathrm{Var}(\Delta S^k_{xy})}\right] \leq \frac{1}{n^2}$$

A probabilistic upper bound $\tilde{S}^k_{xy}(t+1)$ of $S^k_{xy}(t+1)$ may, then, be computed as

$$\tilde{S}^k_{xy}(t+1) = S^k_{xy}(t) + \mu(\Delta S^k_{xy}) + n\sqrt{\mathrm{Var}(\Delta S^k_{xy}(t))} \tag{3}$$

By the previous inequality $\tilde{S}^k_{xy}(t+1)$ is greater than $S^k_{xy}(t+1)$ with probability $(1 - 1/n^2)$. Since the estimate is a probabilistic upper bound, it is possible that the correct focus region is pruned. In general, there will be several focus regions containing whole or most part of the object and the adverse effects will be minimal. A higher value for n may be chosen to increase the probability of the estimate being an upper bound. Experimentally, a value of $n = 2$ was found to be satisfactory.

The mean and variance of ΔS^k_{xy} are recursively estimated from previous observations. A given frame is expected to be influenced more by the immediately preceding frame than by frames occurring much earlier. Therefore, in the recursive equations, a weight α is associated to the observation from the immediately preceding frame and the previous estimates of mean and variance are given a weight of $(1-\alpha)$. Higher α indicates that observations from frame t has a higher probability of occurring in frame $t+1$. It may be noted that $S^k_{xy}(t)$ will not be available for the focus regions of frame t pruned by active search. In such cases the upper bound estimate of S^k_{xy} for frame t is used in the calculations.

After estimating θ_{t+1} and $\tilde{S}^k_{xy}(t+1)$ for all focus regions active search is applied to frame $t+1$. Active search will restrict itself to only those focus regions which have the initial upper bound estimates higher than the initial upper bound cutoff. Thus the set of focus regions satisfying the relation $\tilde{S}^k_{xy}(t+1) > \theta_{t+1}$ constitute the predicted search area for frame $t+1$. This effect is schematically represented in figure 4.

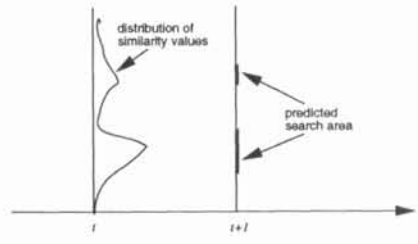

Figure 4: Predicting the search area for frame at time $t+1$ from frame at time t

The computational gain obtained by predicting the search area is shown in figure 5. The solid line de-

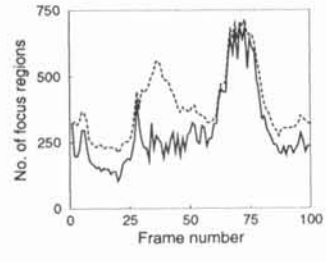

Figure 5: Computational gain by search area prediction

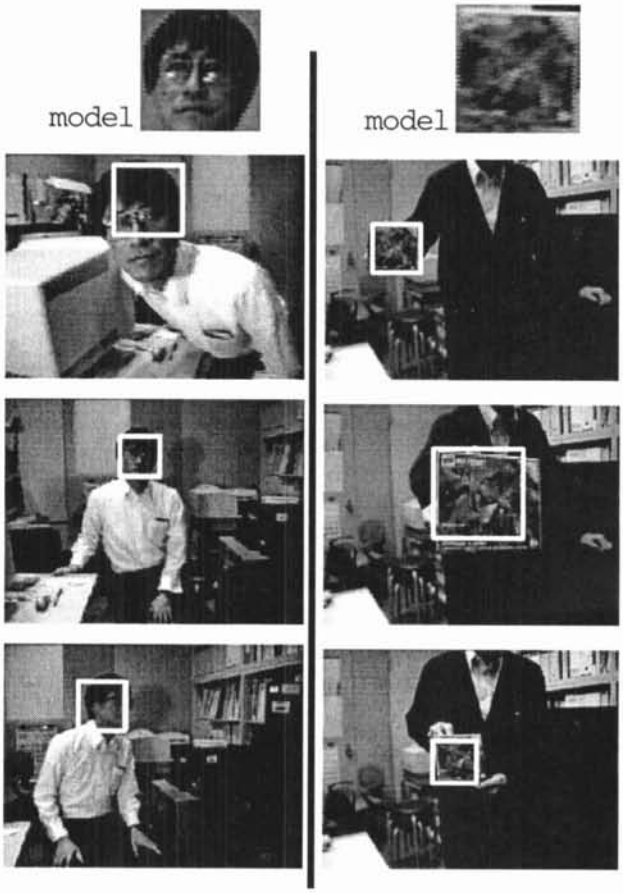

Figure 6: Sample results of tracking

notes the number of focus regions examined by active search after search area prediction. The dashed line shows the results without search area prediction. Predicting search area does not give high reductions in computation when the frame to frame variations are high as seen from the peak in curves where the camera is zooming in on the object which finally covers most of the scene. In other frames search area prediction reduces the computations by about 50%.

3 Experimental Results

In this section we evaluate the performance of the proposed visual tracking system. The experiments were conducted on a SGI Indy workstation with no special hardware. The experiments were performed with $s = 8$, $w = 32$ and 8 different image sizes. The initial upper bounds for predicting the search area were estimated using $\alpha = 0.05$ and $n = 2$. Sample results from two sequences of frames are shown in figure 6. It may be observed that the proposed system tracks the object in spite of large changes in object size, orientation etc. The average tracking speed was around 10 frames per second without any special hardware. With dedicated hardware or parallel architectures the processing speed could easily be increased to 30 frames per second or higher.

4 Conclusion

In this paper we have presented a novel method for tracking arbitrary objects based on similarity of color distribution. A fast search technique for detecting the best matching part of the image has been presented. The search area for a frame is determined based on the statistical properties of previous frames. The proposed method is able to track moving objects at the rate of 10-12 frames per second without using any special hardware. Use of parallel machines or other special hardware real-time processing rates of 30 frames per second can be easily achieved.

Combining a position prediction scheme with the search area prediction proposed in this paper could process at higher frame rates. In critical applications higher accuracy could be obtained by multi stage matching using features other than color histogram also [7]. These aspects are under investigation.

Acknowledgements The authors wish to thank Dr. T Izawa, Dr. K Ishii, Dr. N Hagita and Dr. S Naito of NTT Basic Research Labs for their help and encouragement in conducting this research.

References

[1] T Bar-Shalom and T E Fortman. *Tracking and Data Association.* Academic Press, 1988.

[2] L Blonde, et al. A virtual studio for live broadcasting: The monalisa project. *IEEE Multimedia*, Vol. 3, pp. 18–29, Summer 1996.

[3] R Deriche and O Faugeras. Tracking line segments. In *Proc. of ECCV'90*, 1990.

[4] M Flickner, et al. Query by image and video content : The QBIC system. *IEEE Computer*, Vol. 28, No. 9, pp. 23–32, Sept. 1995.

[5] J Ohya, et al. Virtual space teleconferencing: Real-time reproduction of 3-d human images. *Journal of Visual Communications and Image Representation*, Vol. 6, No. 1, pp. 1–25, 1995.

[6] M J Swain and D H Ballarad. Indexing via color histograms. In *Proc. Image Understanding Workshop*, pp. 623–630, 1990.

[7] V V Vinod and Murase. Object location using complementary color features: histogram and DCT. In *Proc. of ICPR'96*, August 1996.

[8] V V Vinod and H Murase. Focussed color intersection for object extraction from cluttered scenes. In *Proc. of Vision Interface*, May 1996.

[9] V V Vinod, H Murase, and C Hashizume. Focussed color intersection with efficient searching for object detection and image retrieval. In *Proc. of IEEE Conference on Multimedia Computing Systems*, June 1996.

[10] J K Wu, A D Narasimhalu, B M Mehtre, C P Lam, and Y J Gao. CORE: a content-based retrrieval engine for multimedia information systems. *ACM Multimedia Systems*, Vol. 3, No. 1, pp. 25–41, 1995.