The Fourteenth International Symposium on Artificial Life and Robotics 2009 (AROB 14th '09),
B-Con Plaza, Beppu, Oita, Japan, February 5 - 7, 2009

Target-Adjusted Model for Kernel-Based Tracker

Jeehyun Goya Choe[1], Joon-Hong Seok[2] , Jeong-Jung Kim[2] and Ju-Jang Lee[2]

[1] Robotics Program, Korea Advanced Institute of Science and Technology, Daejeon, Korea
(Tel : +82-42-869-8032; E-mail: goya@kaist.ac.kr)

[2] Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, Daejeon, Korea
(Tel : +82-42-869-3432; E-mail: seokjh@kaist.ac.kr, jjlee@ee.kaist.ac.kr)

Abstract:
Kernel-based tracker shows robust performances in various object tracking technologies. Due to its robustness and accuracy, Kernel-based tracker using Mean-shift algorithm is regarded as one of the best ways to apply in object tracking technology in computer vision fields. However, it has a drawback. It fails tracking when faced with a speedy object moving beyond its window size within one image frame interval time. These failures are reduced with the use of Target-adjusted kernel models proposed in this paper. These models are designed to reflect the previously known target information. Same to the conventional kernel model in kernel-based object tracker, this model also contains the target information of both the color and the distance. Experimental results show that the time required for the calculations in tracking process is lessened so that causing lesser failures by applying this Target-adjusted model. Building a Target-adjusted model requires the previously known color data of the target. The model is designed to focus on the colors in the target object by using look-up tables. These look-up tables enable to reduce the size of the color bins in a model so that unnecessary trivial computations can be regarded.

Keywords: Kernel-based object tracking, Target-adjusted model, LUT(Look Up Table)

1. INTRODUCTION

There are many tracking algorithms which perform well in finding targets accurately in the sequence of image frames. However, most of them require expensive processing of the entire image pixels. Such algorithms do not solve the focus of attention problem, where the only place of interest to the observer lies around the object itself.

To address the focus of attention problem, using only local pixels around the previous target area by Kernel-based Mean-shift tracker has been proposed by Comaniciu and Ramesh (see [2]). An obvious advantage of the Mean-shift tracker over the conventional methods is the elimination of a brute force search, and the computation of the translation of the object patch in a small number of iterations (see [4]).

The original Kernel-based object tracking, analyzed in Section 2, performs real-time target image tracking without any prior information of the target. In this paper, tracking of the specific target--a metal cup--is implemented using a specified target model, the Target-adjusted model. The way to design a Target-adjusted model is described in detail in Section 3.

There are many well-known methods that elaborately represent object information. Those methods show good performance in object recognition and detection but are not appropriate for tracking. Delicate and fine representation of target object models, used in recognition or detection, costs too much computation. It gives poor tracking performance especially when the motions are unreliable and unexpected.

In this paper, Target-adjusted models with newly generated bins is implemented with the advantage of specially designed LUT and proved to be suitable for tracking.

2. KERNEL-BASED TRACKER

The Kernel-based tracker specifies how to combine a set of color values of pixels in local neighborhood with a proper kernel profile k(x) to produce a new location that tracks the center of the target in the image.

Prior to the computation of Mean-shift vector, color models that present the current window and the previously saved original window should be established. Those models should reflect both the color value and the color intensity rate of each color channel. The target model and the target candidate are represented by each pdf which can be estimated from the data in color space (see [2]). The target model with the color u=1…m is obtained as $\hat{\boldsymbol{q}} = \{\hat{q}_u\}_{u=1\ldots m}$,

$$\hat{q}_u = C\sum_{i=1}^{n} k(\|x_i^*\|^2)\delta[b(x_i^*) - u],$$

where x_i^* is the normalized pixel locations of the target model, $b(x_i^*)$ is the index of bin in the quantized color space. The target candidate with the color u=1…m at location y is computed as $\hat{\boldsymbol{p}}(y) = \{\hat{p}_u(y)\}_{u=1\ldots m}$

$$\hat{p}_u(y) = C_h\sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right)\delta[b(x_i) - u],$$

where x_i are the normalized pixel locations of the target candidate, centered at y in the current frame.

The heart of the Mean-shift algorithm is computation of a new location $\hat{y}_1$ from location $\hat{y}_0$ according to the

Mean-shift vector

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)},$$

where $g(x) = -k^{`}(x)$, h is the bandwidth of $k(x)$, w_i is given by

$$w_i = \sum_{u=1}^{m} \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}}\,\delta[b(x_i) - u],$$

where w_i is sample weight[2].

3. TARGET-ADJUSTED MODEL

The two most important factors affecting the efficiency of any algorithm using *color space* processing are the color space being chosen and its *quantization*[4]. In this paper, color space and quantization level are fixed for consistency. For the color space, RGB color channel, the most frequently used color space, was used instead of many other available color spaces.

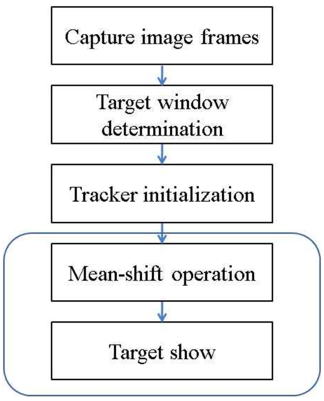

Figure 1. Flow chart of Kernel-based tracker

The chart above shows the main 5 steps in implementing Mean-shift tracker to a real-time video. The target model is built once in Tracker initialization process, while target candidate is periodically built in Mean-shift operation process updating the current window's color distribution. Among 5 steps, the last two steps--Mean-shift operation and Target Show processes--are the ones operated in every video frame. It is clear that improvement of speed is closely related to the time taken in building a target candidate and deriving a Mean-shift vector.

In Mean-shift algorithm, image pixels distributed in a 2-dimensional space are transformed to 1-dimensional bins. In this modeling process, two kinds of information of the target are delivered-color information and the distance information. In implementation, the color models, target model and target candidate, are represented as 1-dimensional arrays.

The number of length of the array is *(number of color-bins)*^{3} for 3 RGB channels. A usual image frame with 256 color-levels for each RGB channel would have 256^3 color-bins. This is a tremendous amount and when trying to directly implement in a practical use, the computer memory easily gets out of control. So in practice, the color-levels are uniformly quantized into 16-levels or less (see [6]).

Performance gets better with higher color-levels (see [1]). Models with larger number of bins represent the same target with more information so that tracking can be more accurate. A uniform quantization with smaller color-levels causes lowering of performance since lesser information is used for tracking. However, its computation time is short. In this paper, 16-level quantization is used for every experiment.

Following figure is an example of a kernel model. This target mainly consists of two colors, pink and white.

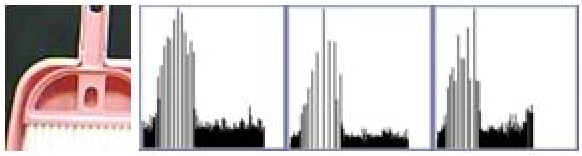

Figure 2. Target image and its RGB plot

The reason that we concentrate on the kernel model rather than the RGB channels is that because it is not easy to extract the common feature from the RGB channel plots. RGB color histograms do not reflect the distance information but only the color frequencies. Therefore they are very sensitive to light, as well as background colors. By observing the sequence of RGB plots in real-time tracking system, it is discovered that the characteristics of histograms vary from time to time (see [1]). Kernel model, on the other hand, usually gives specific distributions regardless of the various poses, orientations or lights. The kernel model of the object shown in Figure 2 is shown below.

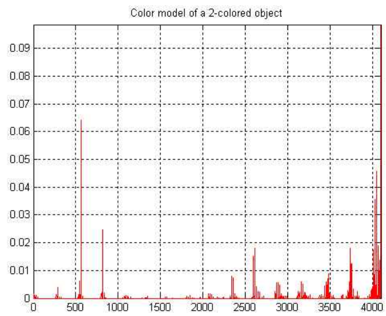

Figure 3. Kernel model with 16^3 color bins

As the Figures 2 and 3 show, the target image usually can be specified with certain number of dominant colors. Some dominant color-bins as well as empty bins are discovered. If we can design the model bins to focus on the meaningful information and discard the unnecessary ones, the implementation of the tracking system would be much simple and efficient.

3-1. Designing a Target-adjusted model

If a uniform quantization of the color model is implemented, there are many unused bins causing memory waste. These unnecessary bins are likely to have noise values, and cause unnecessary calculations.

Extracting an essential data from the target image can be efficiently realized by using the prior knowledge of the target model.

Figure 4 Target selection in the image frame

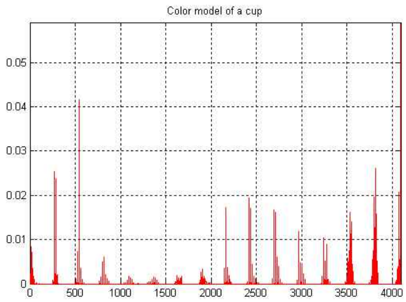

Figure 5. Kernel model of a cup with 16^3 color bins

The image given in Figure 4 is the raw image from the camera, and the selected target region is shown at the right top. Mean-shift kernel model is displayed together with the selected local image. Since this snapshots are from uniform quantization program with 16 *bins per channel*(bpc) the number of bins in x-axis of the Mean-shift model graph is $16^3 = 4096$.

To have a non-uniform quantization of the color-levels, there should be prior model of the target.

As can be seen from the figures, there are some dominant colors, weak colors and very weak colors, practically without any value in its bin, are detected. In the operation that is held for every single frame, every bin value in target candidate should work both as a divider and a multiplier for every bin value in the previously saved target model.

To ignore the bins that have no meaning in tracking the target object, bins with zero color values were mostly discarded and reduced to a small amount of bins.

In the Mean-shift tracker, each color value in target model and target candidate is always used together. Therefore when either target model value or target candidate value is zero, the values calculated from target model and target candidate are always zero. This means if we know that some color levels in target model or target candidate have poor possibility to emerge in the target, discarding that color levels makes no difference to the resulting Mean-shift vectors.

The implementation of establishing a Target-adjusted model is done by observing the target models and target candidates from many samples. Target models--in the form of kernel model in Mean--shift tracker-are sampled from various poses and motions. From the stored sample values, the color levels with frequent zero values can be noticed.

By memorizing only the value-existing color bins, a lookup table can be made. The size of a lookup table is one array with the length same as that of the number of bins in original color model.

The effect of reducing the number of bins is great since there are always more than one iteration per a frame, and usually are more than 6 iterations especially when the target movement is large.

4. EXPERIMENTS

The proposed approach was implemented in the C programming language. The experiment was held in the environment of real-time video system, so the consistency of speed and motion could have had some slight uneven cases. For all experiments, the same set of parameter values, as shown below, were used.

1) The size of each image frame is 320*240 pixels
2) The window size for mean-shift tracking is fixed to 90*104 pixels, thus the bandwidth of the local image is all the same in these experiments.
3) BPC is fixed to 16 bins per one color channel
4) 3 color channels used (red, green and blue)
5) A metal cup was used as a target object

The metal cup used as the target here basically contains three colors. However, the color of the metal surface is very sensitive to the pose, orientation and light.

A. Uniform quantization

5 different speeds of motions were experimented for each type of motion.

Table 1 Experimental variables

Variables	
Types of motions	Linear Rectangular Circular
Speeds of motions	Use of relative motions from 1 to 5

B. Same experiment using the Target-adjusted model

Since the number of bins is critical factor in mean-shift process, we can discard much of the bins from the target model if the bin is empty. Target-adjusted model is built as a Lookup table of important bin numbers. It allocates new indexes to selected bins reducing the amount of bins to a smaller quantity.

5. DISCUSSIONS AND COMPARISONS

A. Uniform quantization

For each case, the total time taken in Mean-shift process was checked and recorded. The results are shown below.

Table 2 Experiment using uniform quantization

Approx. Relative Speed	Computation time for 150 frames (sec)		
	linear motion	rectangular motion	circular motion
1	2.246	2.281	4.374
2	2.841	3.18	4.67
3	3.549	3.531	5.578
4	4.743	4.254	5.8
5	4.9	4.967	6.162

As the speed gets higher, the more calculation time is needed. Also, more time is needed for tracking big motions than tracking small motions. It is noticeable that along with the number of bins, amount of motion path is also a critical factor for calculation speed.

B. Target-adjusted model

The same experiments done in uniform quantization were implemented using the Target-adjusted model.

Table 3 Experiment using LUT

Approx. Relative Speed	Computation time for 150 frames (sec)		
	In linear motion	In rectangular motion	In circular motion
1	1.802	1.977	4.003
2	2.471	2.527	4.511
3	3.096	2.935	5.003
4	3.937	3.992	5.492
5	4.016	3.998	5.588

The result of linear motions is plotted below. Dashed line is the result of using the LUT, and the solid line is the result of uniform quantization. The graph clearly shows that the time needed for computing in the uniform quantization case is longer than that of the LUT case.

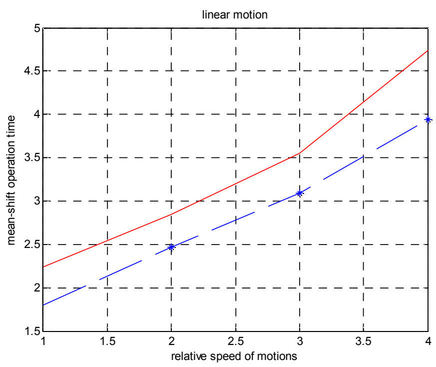

Figure 6 Linear motion with different speeds

These results show the tendency of computing time. It depends on the existence of using Target-adjusted model. Target information helps to reduce the tracking process.

At the relative speed of about 7 in linear motion, the Target-adjusted tracking system had no problem in tracking the target but the conventional Kernel-based tracker repeatedly failed.

6. CONCLUSION

In Target-adjusted tracking system, the time (or the number of calculation) required for tracking process within a frame was reduced without any significant effect on performance.

Extensive simulation and experimental results verified the efficacy and robustness of the proposed algorithm. Tracking failures were reduced. The reduction in processing time also substantiates this conclusion.

Further, this adjusted model scheme can be applied to many other non-rigid objects that have few dominant color values.

REFERENCES

[1] J. G. Choe, J. Seok, J. Lee, "Mean-shift Tracker with Face-adjusted Model", *International Conference on Control, Automation and Systems*, October 2008.

[2] D. Comaniciu, V. Ramesh, P. Meer, "Kernel-based object tracking", IEEE Trans. *Pattern Analysis and Machine Intelligence*, vol25 (5), pp. 564-577, August 2003.

[3] R. T. Collins, "Mean-shift Blob Tracking through Scale Space", IEEE Computer Society Conference on *Computer Vision and Pattern Recognition*, 2003.

[4] A. Yilmaz, O. Javed, M. Shah, "Object Tracking: A Survey", ACM Computing Surveys, 2006.

[5] F. Ennesser, "Finding Waldo, or Focus of Attention Using Local Color Information", IEEE Trans. *Pattern Analysis and Machine Intelligence*, August 1995.

[6] Iain Wallace, *A mean-shift tracker: implementations in C++ and Hume*, 2005.

[7] S. Jeong, M. Lee, Computer Vision programming using OpenCV (in Korean), Hongneung Science Publisher, 2007.