Affine Object Tracking with Kernel-based Spatial-Color Representation

[1,2]Haihong Zhang, [1]Weimin Huang, [2]Zhiyong Huang, [1]Liyuan Li
[1]Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore
{hhzhang,weimin, lyli}@i2r.a-star.edu.sg
[2]School of Computing, National University of Singpore
huangzy@comp.nus.edu.sg

Abstract

This paper presents a new visual tracking method that can achieve accurate estimation of affine transformation and precise spatial-color representation. The estimation of transformation provides more information than translation for better motion understanding and also helps maintain the precise representation; the precise representation enables tracking objects in highly-cluttered environment. The basis of the method is a kernel-based similarity measure called affine matching that describes the relationship between image regions with respect to affine transformation parameters. Based on the similarity measure, a mathematical solution is derived for estimating the transformation parameters for moving objects in videos. Various experiments have yielded positive results.

1. Introduction

Visual tracking is one of the most important disciplines in computer vision and play key roles in many scientific and engineering fields. A challenging issue in visual tracking is how to model an object (using *likelihood* functions or similarity measures) to precisely correlate a pictorial observation and the object's state [1], especially when the object is undergoing transformations.

In appearance-based approaches, the object model is usually set up over image regions in terms of spatial and color features. The literature has seen a great deal of relevant research on region modeling and tracking. Table 1 summarizes them into two rough categories – color models and spatial-color models.

The first category places emphasis on the color features of target objects. A very popular methodology in this category is to exploit color distributions in simple-shape regions such as blobs. For instance, faces are typical blobs that were represented by single Gaussian distributions in [2]. Some further endeavors to deal with mixtures of colors resort to multimodal Gaussians with Expectation-Maximization algorithm [3, 4], while it is still an open problem how to choose the right number of Gaussians.

In recent years, a number of non-parametric techniques have been suggested for color modeling and tracking. Unlike above methods they do not rely on presumptions about the probability distributions. In particular, color histograms have been widely used in tracking faces, hands and people [5, 6, 7, 8]. A remarkable work in this field is given by Comanicui et al. who combine spatial kernels and color histograms to obtain spatially-smooth similarity function which leads to a mean-shift [14] optimization procedure to tracking [9]. The method has demonstrated impressive performance in many challenging tasks [15], and in [10] it has also been extended to deal with scaling objects.

A drawback with color-based methods lies in their disadvantages in correlating spatial and color features. Due to the loss of spatial information, the color models may not be suited for handling objects/background with similar color distributions, nor for recovering accurate pose information such as rotation angles and deformation parameters.

The second category (spatial-color models) puts emphasis on the correlation between spatial and color features. A conventional approach called image templates uses an object's sample images or their high-level statistics as representations [11, 12, 16]. For tracking, one may search in a set of image windows and compare them with the template to tell if the relevant object is present. However, conventional templates often have difficulty in handling the deformations of images with complex spatial-color features.

More recently, Elgammal et al. suggested a new way to address the correlation between spatial and color features, by using kernel density techniques [13]. The key point there is to consider both feature values and feature locations in a joint probabilistic framework, thus providing an approach to precise spatial-color representation. The authors also introduced an entropy-based measure to describe the similarity between image regions, and developed an algorithm for tracking objects under translations. In [17], an alternative

Category	Method	Translation	Rotation	Deformation*	Accuracy	Ref.
Color Models	Blobs	$\surd$	$\surd$	$\times$	low	[2, 3, 4]
	Color-Hist.&Kernels	$\surd$	$\surd$	$\times$	average	[5, 6, 7, 8, 9, 10]
Spatial-Color Models	Image templates	$\surd$	$\times$	$\times$	high	[11, 12]
	Spatial-feature kernels	$\surd$	$\times$	$\times$	high	[13]
	Our method	$\surd$	$\surd$	$\surd$	high	

Table 1. Categorization of Appearance-based Methods for Visual Tracking.

* Deformation here refers to shearing and non-uniform scaling.

similarity measure was proposed based on the expectation of the density estimates over the model or target image. But our investigation suggests that these methods might not be suited for handling image transformations like scaling or shearing. A brief discussion will be given at the end of Section 2.

It is therefore important and challenging to develop an approach to tracking objects under transformations with precise spatial-color representations. Such an approach would have two general advantages: the precise representation allows to track objects in highly-cluttered environments; the estimation of transformations can provide more information than translation for motion understanding and also makes it possible to maintain the precise representation.

This paper presents a new tracking method to address the above issue, especially for objects under affine transformations. We first formulate a kernel-based similarity measure between image regions under affine transformations. From the similarity formulation we derive a mathematical solution for accurately estimating the transformation parameters for moving objects. We examined the proposed method under different levels of image noise. We also tested it on various real objects. Positive results have been obtained.

2. The Kernel-based Affine Matching

Consider a given object that appears as an image region $\Omega = \{(\mathbf{x}_i, \mathbf{u}_i)\}$ consisting of a set of points at positions $\{\mathbf{x}_i\}$ with colors $\{\mathbf{u}_i = \mathbf{u}(\mathbf{x}_i)\}$, $i = 1, \ldots, N$. A probabilistic way to represent the region is by modeling the joint position-color distribution [13]:

$$f(\mathbf{x}, \mathbf{u}|\Omega) = \frac{\alpha}{N} \sum_{i=1}^{N} k_s(||\mathbf{x} - \mathbf{x}_i||^2) k_u(||\mathbf{u} - \mathbf{u}_i)||^2) \tag{1}$$

Here k_s and k_u are kernel functions (we use Gaussians in the paper) with particular bandwidths for spatial and color features, and α is a normalization constant that gives $\int f d\mathbf{u} d\mathbf{x} = 1$.

In appearance-based tracking, it is important to formulate the similarity measure which can tell the likelihood of a candidate region Ω_p being an instance of the model Ω_q. Here we propose an l_2 norm measure on the basis of their kernel-based density functions (p and q)

$$\begin{aligned} D_0(p, q) &= -\int ||p - q||^2 d\mathbf{u} d\mathbf{x} \\ &= -\int pp d\mathbf{u} d\mathbf{x} - \int qq d\mathbf{u} d\mathbf{x} + 2 \int pq d\mathbf{u} d\mathbf{x} \end{aligned} \tag{2}$$

It is obvious that the highest similarity value D_0 would be zero, which happens if and only if the two density functions are exactly the same.

Now consider an object Ω_q undergoing an affine transformation which combines shearing, scaling, rotation and translation sequentially. Thereby, the points in the region will make the following movement

$$\mathbf{x}_i^{(T)} = M(\mathbf{a}, s, \theta)\mathbf{x}_i + \mathbf{x}_t \tag{3}$$

Here $\mathbf{x}_t$ represents the displacement of the whole region. The M is a matrix defining the region deformation by the scaling factors $\mathbf{a} = [a_x, a_y]^T$, shearing factor s and rotation angle θ:

$$M(\mathbf{a}, s, \theta) = \begin{bmatrix} a_x \cos\theta & -a_y \sin\theta + s a_x \cos\theta \\ a_x \sin\theta & a_y \cos\theta + s a_x \sin\theta \end{bmatrix} \tag{4}$$

Note that it is possible to consider M as a simple matrix without physical parameterization. From such an M we can derive a different form of similarity measure over affine transformation. Our study shows that, however, the consequent tracking algorithm would not be very robust because it does not take advantage of physical constraints for seeking the true affine parameters on rather complex hypersurfaces of similarity functions. On the other hand, as to be ellaborated in Section 4, the physically-modeled M will allow us to design a coarse-to-fine searching scheme for robust tracking.

Without loss of generality, we set the center of the object model at origin. Hence $\mathbf{x}_t$ in Eq. (3) will represent the center (referred to as the location) of the target object.

We assume that in affine transformation the color features are consistent while the positions are subject to change. Hence, the transformed region can be represented

by Eq. (1) with replaced locations by Eq. (3). With the new representation, we can calculate the integrals in Eq. (2) numerically, and derive the similarity between Ω_p and $\Omega_q(T)$ (the transformed object model) with respect to the transformation parameters $T = \{M, \mathbf{x}_t\}$.

$$
\begin{aligned}
&D_0(T) = \\
&-\tfrac{1}{N_p^2}\textstyle\sum_{i,j} k_s(\frac{||M(\mathbf{a},s,\theta)(\mathbf{x}_i^{(p)}-\mathbf{x}_j^{(p)})||^2}{2})k_u(\frac{||\mathbf{u}_i^{(p)}-\mathbf{u}_j^{(p)}||^2}{2}) \\
&+\tfrac{2}{N_pN_q}\textstyle\sum_{i,j} k_s(\frac{||M(\mathbf{a},s,\theta)\mathbf{x}_i^{(p)}+\mathbf{x}_t-\mathbf{x}_j^{(q)}||^2}{2})k_u(\frac{||\mathbf{u}_i^{(p)}-\mathbf{u}_j^{(q)}||^2}{2}) \\
&-\tfrac{1}{N_q^2}\textstyle\sum_{i,j} k_s(\frac{||(\mathbf{x}_i^{(q)}-\mathbf{x}_j^{(q)})||^2}{2})k_u(\frac{||\mathbf{u}_i^{(q)}-\mathbf{u}_j^{(q)}||^2}{2}) \\
&\triangleq -I_1(p_T,p_T) + I_2(p_T,q) - I_3(q,q)
\end{aligned}
\tag{5}
$$

where I_1, I_2 and I_3 are representations of the three addenda.

The similarity function Eq. (5) relates two regions under transformations in a much simpler way. As a result, it allows a direct and mathematical solution to the problem of similarity-maximization and tracking, as will be shown in the following.

Comparing Eq. (5) with those similarity functions introduced in [13, 17], it is interesting to note that the apparent form of our function has two extra terms (I_1, I_3). Our study (details omitted here due to space limitation) suggests that the two terms are crucial for relating model and target image over transformations – especially scaling and shearing.

3 The Optimization Procedure

To recover the transformation relationship between a model region Ω_q and a candidate region Ω_p, it is essential to find such a transformation configuration that maximizes the affine similarity given by Eq. (5). It can be seen that for a given model, the term I_3 in Eq. (5) is constant. Therefore the similarity D_0 to be maximized can be substituted by

$$D(p_T,q) = -I_1(p_T,p_T) + I_2(p_T,q) \tag{6}$$

Thanks to the smooth and differentiable nature of Gaussian kernel functions, the objective function can be written as

$$\nabla D(p_T,q) = \frac{\partial D}{\partial \mathbf{x}_t}\Delta\mathbf{x}_t + \frac{\partial D}{\partial \theta}\Delta\theta + \frac{\partial D}{\partial \mathbf{a}}\Delta\mathbf{a} + \frac{\partial D}{\partial s}\Delta s = 0 \tag{7}$$

which implies that $\nabla_{\mathbf{x}_t} = 0$, $\nabla_{\mathbf{a}} = 0$, $\nabla_\theta = 0$ and $\nabla_s = 0$ are to be satisfied simultaneously. The corresponding solutions are given below.

3.1 Computing Translation Vector $\mathbf{x}_t$

Consider the similarity measure Eq. (5). It can be shown that $I_1(p_T,q)$ is independent upon $\mathbf{x}_t$. So there is

$$
\begin{aligned}
\nabla_{\mathbf{x}_t} D(p_T,q) &= -\nabla_{\mathbf{x}_t} I_1(p_T,p_T) + \nabla_{\mathbf{x}_t} I_2(p_T,q) \\
&= 0 + 2\sum_{i,j} w_{ij}(\mathbf{x}_t + M(\mathbf{a},s,\theta)\mathbf{x}_i^{(p)} - \mathbf{x}_j^{(q)})
\end{aligned}
\tag{8}
$$

where the weight w_{ij} is given by

$$w_{ij} = \tfrac{2}{N_pN_q}\, g_s(\frac{||M(\mathbf{a},s,\theta)\mathbf{x}_i^{(p)}+\mathbf{x}_t-\mathbf{x}_j^{(q)}||^2}{2})k_u(\frac{||\mathbf{u}_i^{(p)}-\mathbf{u}_j^{(q)}||^2}{2}) \tag{9}$$

with $g_s(\cdot)$ representing the derivative $\partial k_s(x)/\partial x$. If k_s is a Gaussian kernel function, g_s would also take a Gaussian form.

The equation $\nabla_{\mathbf{x}_t} D(p_T,q) = 0$ leads to the following solution.

$$\mathbf{x}_t^* = \frac{\sum_{i,j} w_{ij}(\mathbf{x}_j^{(q)} - M(\mathbf{a},s,\theta)\mathbf{x}_i^{(p)})}{\sum_{i,j} w_{ij}} \tag{10}$$

Since $\mathbf{x}_t$ is involved in the weights $\{w_{ij}\}$ (see Eq. (9)) on the right side, this is indeed an iterative solution for computing the translational vector $\mathbf{x}_t$.

3.2 Computing Rotation Angle θ

Consider the partial derivative of $I_1(p_T,p_T)$ with respect to θ. Denote

$$
\begin{aligned}
f_{ij}^{(1)} &\triangleq ||M(\mathbf{a},s,\theta)(\mathbf{x}_i^{(p)} - \mathbf{x}_j^{(p)})||^2 \\
&= a_x^2(x_i^{(p)} + sy_i^{(p)} - x_j^{(p)} - sy_j^{(p)})^2 \\
&\quad + a_y^2(y_i^{(p)} - y_j^{(p)})^2
\end{aligned}
\tag{11}
$$

which is independent upon θ. From Eq. (5), we have

$$\frac{\partial I_1(p_T,p_T)}{\partial\theta} = \frac{1}{N_p^2}\sum_{i,j}\frac{\partial k_s(f_{ij}^{(1)}/2)}{\partial\theta}k_u(\frac{||\mathbf{u}_i^{(p)} - \mathbf{u}_j^{(p)}||^2}{2}) = 0 \tag{12}$$

So $I_1(p_T,p_T)$ is also independent upon θ.

Consider the partial derivative of $I_2(p_T,q)$ with respect to θ. Let

$$\hat{\mathbf{x}}_j^{(q)} = \mathbf{x}_j^{(q)} - \mathbf{x}_t, \;\; \mathbf{x}_i^{(a)} = M(\mathbf{a},s,\theta)\mathbf{x}_i^{(p)} \tag{13}$$

and

$$f_{ij}^{(2)} = ||\mathbf{x}_i^{(a)} - \hat{\mathbf{x}}_j^{(q)}||^2 \tag{14}$$

Then there is

$$\begin{aligned} f_{ij}^{(2)} &= (x_i^{(a)} - \hat{x}_j^{(q)})^2 + (y_i^{(a)} - \hat{y}_j^{(q)})^2 \\ &= (2a_y y_i^{(p)} \hat{x}_j^{(q)} - 2a_x(x_i^{(p)} + sy_i^{(p)})\hat{y}_j^{(q)})\sin\theta \\ &\quad +(-2a_x(x_i^{(p)} + sy_i^{(p)})\hat{x}_j^{(q)} - 2a_y y_i^{(p)}\hat{y}_j^{(q)})\cos\theta \\ &\quad +(a_x^2(x_i^{(p)} + sy_i^{(p)})^2 + a_y^2 y_i^{(p)2} + \hat{x}_j^2 + \hat{y}_j^2) \\ &\equiv a_{ij}sin\theta + b_{ij}cos\theta + c_{ij} \end{aligned} \quad (15)$$

where c_{ij} is a variable independent upon θ.

Hence we have

$$\begin{aligned} \frac{\partial I_2}{\partial\theta} &= \frac{2}{N_p N_q}\sum_{i,j}\frac{\partial k_s(f_{ij}^{(2)}/2)}{\partial\theta}k_u(\frac{||\mathbf{u}_i^{(p)} - \mathbf{u}_j^{(q)}||^2}{2}) \\ &= \sum_{ij} w_{ij}\frac{\partial f_{ij}^{(2)}}{\partial\theta} \\ &= \beta_1 cos\theta + \beta_2 sin\theta \end{aligned} \quad (16)$$

Here w_{ij} has been given in Eq. (9), and β_1, β_2 are given by

$$\beta_1 = \sum_{i,j} w_{ij}a_{ij}\ ,\ \beta_2 = \sum_{i,j} -w_{ij}b_{ij}$$

Therefore, $\nabla_\theta D(p_T, q) = 0$ would lead to $\beta_1 cos\theta + \beta_2 sin\theta = 0$. And the solution is

$$\theta^* = -sin^{-1}(\frac{\beta_1}{\sqrt{\beta_1^2 + \beta_2^2}}) \quad (17)$$

Since in fact $\beta_{1,2}$ on the right side indirectly involve (via w_{ij} in Eq. (9)) the parameter θ, the solution implies an iterative procedure to achieving $\nabla_\theta D(p_T, q) = 0$. Besides, because of the property of arcsine, Eq. (17) would suggest two values: θ and $\theta + \pi$. In practice we can choose the one that produces relatively larger similarity distance $D(p_T, q)$.

3.3 Computing Scaling Factors a

Let's first consider the partial derivatives of $I_1(p_T, p_T)$ with respect to $\mathbf{a}$. By introducing Eq. (11) to I_1 and I_2, we have

$$\begin{aligned} \nabla_{a_x} I_1 &= \nabla_{a_x}\left(\frac{1}{N_p^2}\sum_{i,j} k_s(\frac{f_{ij}^{(1)}}{2})k_u(\frac{||\mathbf{u}_i^{(p)} - \mathbf{u}_j^{(p)}||^2}{2})\right) \\ &= \sum_{i,j}\frac{1}{2}v_{ij}\frac{\partial f_{ij}^{(1)}}{\partial a_x} \\ &= \sum_{i,j} v_{ij}(x_i^{(p)} + sy_i^{(p)} - x_j^{(p)} - sy_j^{(p)})^2 a_x \end{aligned} \quad (18)$$

And similarly,

$$\nabla_{a_y} I_1 = \sum_{i,j} v_{ij}(y_i^{(p)} - y_j^{(p)})^2 a_y \quad (19)$$

where v_{ij} is given by

$$v_{ij} = \frac{1}{N_p^2} g_s(\frac{f_{ij}^{(1)}}{2})k_u(\frac{||\mathbf{u}_i^{(p)} - \mathbf{u}_j^{(p)}||^2}{2}) \quad (20)$$

For $I_2(p_T, q)$ we have

$$\begin{aligned} &\nabla_{a_x} I_2(p_T, q) \\ &= \nabla_{a_x}\left(\frac{2}{N_p N_q}\sum_{i,j} k_s(\frac{f_{ij}^{(2)}}{2})k_u(\frac{||\mathbf{u}_i^{(p)} - \mathbf{u}_j^{(p)}||^2}{2})\right) \\ &= \sum_{ij}\frac{1}{2}w_{ij}\frac{\partial f_{ij}^{(2)}}{\partial a_x} \\ &= \sum_{ij} w_{ij}\Big[(x_i^{(p)} + sy_i^{(p)})^2 a_x - \sin\theta(x_i^{(p)} + sy_i^{(p)})\hat{y}_j^{(q)} \\ &\quad -\cos\theta(x_i^{(p)} + sy_i^{(p)})\hat{x}_j^{(q)}\Big] \end{aligned} \quad (21)$$

and

$$\begin{aligned} &\nabla_{a_y} I_2(p_T, q) \\ &= \nabla_{a_y}\left(\frac{2}{N_p N_q}\sum_{i,j} k_s(\frac{f_{ij}^{(2)}}{2})k_u(\frac{||\mathbf{u}_i^{(p)} - \mathbf{u}_j^{(p)}||^2}{2})\right) \\ &= \sum_{ij}\frac{1}{2}w_{ij}\frac{\partial f_{ij}^{(2)}}{\partial a_y} \\ &= \sum_{ij} w_{ij}(y_i^{(p)2}a_y + \sin\theta y_i^{(p)}\hat{x}_j^{(q)} - \cos\theta y_i^{(p)}\hat{y}_j^{(q)}) \end{aligned} \quad (22)$$

where w_{ij} is given by Eq. (9).

Therefore, for $\nabla_{\mathbf{a}} D(p_T, q) = \nabla_{\mathbf{a}}(-I_1(p_T, q) + I_2(p_T, q)) = 0$ we have

$$\begin{aligned} a_x^* &= \frac{\sum_{ij} w_{ij}(\sin\theta(x_i^{(p)} + sy_i^{(p)})\hat{y}_j^{(q)} + \cos\theta x_i^{(p)}\hat{x}_j^{(q)})}{\left(\sum_{ij} w_{ij}(x_i^{(p)} + sy_i^{(p)})^2\right) - \sum_{ij} v_{ij}(x_i^{(p)} + sy_i^{(p)} - x_j^{(p)} - sy_j^{(p)})^2} \\ a_y^* &= \frac{\sum_{ij} w_{ij}(\cos\theta y_i^{(p)}\hat{y}_j^{(q)} - \sin\theta y_i^{(p)}\hat{x}_j^{(q)})}{\left(\sum_{ij} w_{ij} y_i^{(p)2}\right) - \sum_{ij} v_{ij}(y_i^{(p)} - y_j^{(p)})^2} \end{aligned} \quad (23)$$

Since $a_{x,y}$ is involved in both w_{ij} and v_{ij} on the right sides, the equations imply an iterative procedure to computing scaling factors.

3.4 Computing Shearing Factor s

Consider now the partial derivatives of $I_1(p_T, p_T)$ with respect to s.

$$\begin{aligned}
& \nabla_s I_1(p_T, p_T) \\
= \; & \nabla_s \left(\tfrac{1}{N_p^2} \textstyle\sum_{i,j} k_s(\tfrac{f_{ij}^{(1)}}{2}) k_u(\tfrac{||\mathbf{u}_i^{(p)} - \mathbf{u}_j^{(p)}||^2}{2}) \right) \\
= \; & \textstyle\sum_{i,j} \tfrac{1}{2} v_{ij} \tfrac{\partial f_{ij}^{(1)}}{\partial s} \\
= \; & \textstyle\sum_{i,j} v_{ij} \left[(y_i^{(p)} - y_j^{(p)})^2 s - a_x^2 (x_i^{(p)} - x_j^{(p)})(y_i^{(p)} - y_j^{(p)}) \right] \\
\equiv \; & z_1 s + z_2
\end{aligned} \tag{24}$$

Now consider $I_2(p_T, q)$. Let's rewrite $f_{ij}^{(2)}$ in Eq. (15) as

$$\begin{aligned}
f_{ij}^{(2)} = \; & (a_x^2 y_i^{(p)2}) s^2 + (2 a_x^2 x_i^{(p)} x_y^{(p)} - 2 a_x y_i^{(p)} \hat{y}_j^{(q)} \sin\theta - 2 \cdot \\
& a_x y_i^{(p)} \hat{x}_j^{(q)} \cos\theta) s + \Big[(a_y y_i^{(p)} \hat{x}_j^{(q)} - 2 a_x x_i(p) \hat{y}_j^{(q)}) \sin\theta \\
& + (2 a_x x_i^{(p)} \hat{x}_j^{(q)} - 2 a_y y_i(p) \hat{y}_j^{(q)}) \cos\theta + a_x^2 x_i^{(p)2} + \\
& a_y^2 y_i^{(p)2} + \hat{x}_j^{(q)2} + \hat{y}_j^{(q)2} \Big] \equiv \mu_1 s^2 + \mu_2 s + \mu_3
\end{aligned} \tag{25}$$

It follows that $\nabla_s I_2(p_T, q)$ would become

$$\begin{aligned}
\nabla_s I_2 = & \nabla_s \left(\frac{2}{N_p N_q} \sum_{i,j} k_s(\frac{f_{ij}^{(2)}}{2}) k_u(\frac{||\mathbf{u}_i^{(p)} - \mathbf{u}_j^{(p)}||^2}{2}) \right) \\
= & \sum_{ij} \frac{1}{2} w_{ij} \frac{\partial f_{ij}^{(2)}}{\partial s} = \sum_{ij} w_{ij} [\mu_1 s + \frac{1}{2} \mu_2] \\
\equiv & \; z_3 s + z_4
\end{aligned} \tag{26}$$

Therefore, the solution to $\nabla_s D(p_T, q) = \nabla_s(-I_1(p_T, p_T) + I_2(p_T, q)) = 0$ can be derived as

$$s^* = -\frac{z_2 - z_4}{z_1 - z_3} \tag{27}$$

This solution implies an iterative procedure to computing shearing factors because both w_{ij} and v_{ij} on the right side involve the shearing factor s.

The above equations (10,17,23,27) provide an iterative approach to seeking the optimal transformation state T, which represents a local maximum in the state space. Because of the smooth nature of the similarity measure (see Eq. (5)), the gradient-based optimization technique would converge to that solution, provided the initial guess is within a sufficiently small neighborhood of the maximum.

4 The Tracking Algorithm

For a given candidate region Ω_p, the above optimization procedure will exactly recover its relation in terms of affine transformation to the model region Ω_q. Therefore, the tracking means to seek a proper candidate, i.e. the true image of the object, for finding the true state of the object model. In real tracking situations, however, the practical candidates may be corrupted by various image noises, plus they may include many background pixels due to inaccurate information about the object's state. In this section we propose a tracking algorithm to address the problem.

According to our empirical study, the imperfect candidates have different levels of impact on different types of transformations. In specific, the robustness of the affine matching varies in a decreasing order from that on translation to shearing/rotation and then to scaling.

Based on the findings we propose a coarse-to-fine tracking scheme, by computing the translation, rotation-shearing, and scaling parameters in a sequential and recursive manner. The scheme uses a flexible candidate extraction method: for the object model Ω_q with estimated transformation T, it extracts the candidate region by

$$\Omega_p : \{\mathbf{x}_i, \mathbf{u}_i | \left(\underset{\mathbf{x}_j \in \Omega_q}{min} ||T^{-1}(\mathbf{x}_i) - \mathbf{x}_j||^2 \right) < \epsilon \} \tag{28}$$

Here ϵ is a relaxation factor. Since the T may not be very accurate, an Ω_p with larger ϵ is more likely to cover the true object image, at the cost of including more background pixels, and vice versa.

The coarse-to-fine scheme uses different ϵ in computing different parameters. When a new frame is first presented, the available information about the present state is limited. So we use a relatively larger ϵ to obtain a candidate that would better cover the object. As per the above findings the candidate can be used for computing the translation vector. With the new information then recovered, we can obtain better candidates with smaller ϵ for computing other transformation parameters. In order to increase the robustness and accuracy, the whole process may repeat for a few times.

The following details the proposed tracking algorithm. The particular ϵ should be selected according to the true situations such as the object's size and uncertainties in motion, and our experience suggests that the system is usually not very sensitive to it.

1. Initialization: create the object model Ω_q using a sample image; set frame no. $k = 0$ and the object's initial state T_0;
2. Proceed to next frame: $k = k + 1$;
3. Obtain the prediction of the transformation state $T_k = \{\mathbf{x}_0^{(k)}, \theta^{(k)}, s^{(k)}, \mathbf{a}^{(k)}\}$; here we may just set them to $T_k = T_{k-1}$;
4. Estimating the transformation parameters:
   (a) Estimating translation vector $\mathbf{x}_0^{(k)}$:
      i. Extract a candidate region Ω_p according to T_k, with appropriate ϵ (Eq. (28));
      ii. Update $\mathbf{x}_0^{(k)}$ by Eq. (10);

iii. If the iteration converges, i.e. $d(\mathbf{x}_0^{(k)}) < \epsilon_x$, where ϵ_x is a preset small value and $d(\mathbf{x}_0^{(k)})$ is the change of $\mathbf{x}_0^{(k)}$ in current iteration step, proceed to Step (b); otherwise go back to Step 4.a.i;

(b) Estimating rotation angle and shearing factor $\{\theta^{(k)}, s^{(k)}\}$:

i. Extract a candidate region Ω_p according to T_k, with appropriate ϵ (Eq. (28));

ii. Update $\theta^{(k)}$ by Eq. (17);

iii. Update $s^{(k)}$ by Eq. (27);

iv. If $d(\theta^{(k)}) < \epsilon_\theta$ and $d(s^{(k)}) < \epsilon_s$, proceed to Step (c); otherwise return to Step 4.b.i;

(c) Estimating scaling factors $\mathbf{a}^{(k)}$:

i. Extract a candidate region Ω_p according to T_k, with appropriate ϵ (Eq. (28));

ii. Update $\mathbf{a}^{(k)}$ by Eq. (23);

iii. If $d(\mathbf{a}^{(k)}) < \epsilon_a$, proceed to Step 5; otherwise go to Step 4.c.i.

5. If (a)(b)(c) yield no change in T_k, go to Step 2 for next frame; otherwise go to Step 4 to further optimize the estimations.

It can be seen that the total computational complexity largely depends on the cost of each iteration. In computing translation vector, each iteration costs $O(N_p N_q)$ computational time. Similarly, the computational cost on rotation angle is also about $O(N_p N_q)$. But for scaling factors and shearing factor, additional $O(N_q^2)$ time is needed to complete each iteration. Hence, we conclude that the overall computational cost is approximately $O(N_p N_q + N_q^2)$.

In addition, our experiments show that the approach can converge to the objective function maximum in a few iterations (typically within 5 iterations in the step 4.(a), 4.(b) or 4.(c)). In a real implementation on a conventional 1.3GHz Pentium-M PC, the tracking system could process two frames per second for a target object of 1000 pixels in *rgb* colors. We believe that the employment of fast Gauss transform (see [18]) will produce a significant speed-up that enables real time applications.

5. Evaluations with Synthetic Videos

This section aims to evaluate the tracking algorithm's performance against image noises. The target is a synthetic diamond-shaped object that randomly moves through synthetic image sequences (examples shown in Figure 5), with each sequence corrupted by an additive zero-mean Gaussian noise with (σ_n^2). It can be seen that high levels of noises pose a serious problem to accurate tracking.

The first test compares the proposed method with the mean-shift tracker [9] on tracking a translational object. In order to obtain accurate evaluation, we ran 8 independent trials at each noise level and averaged the results. Figure 2 plots the tracking errors as functions over the noise level. The proposed method could accurately track the object under $\sigma \leq 70$, while the mean-shift tracker worked well only when the noise level is much lower ($\sigma \leq 30$).

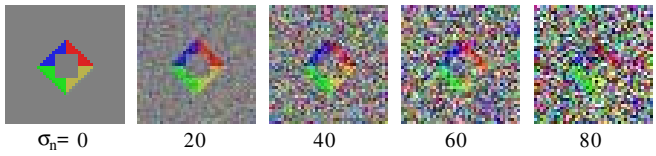

Figure 1. Synthetic objects under various levels of noise.

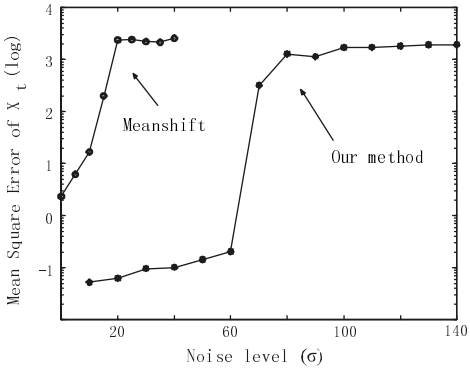

Figure 2. Comparative results on tracking synthetic objects, with our method and the mean-shift tracker [9].

Another test further examines the proposed method on tracking objects under fully affine transformation. As above, the results were also obtained on the basis of eight independent trials at each noise level. Figure 3 plots the results, where circles denote the true state and the curves represent tracking results in individual trials. Clearly, the method could accurately determine the object's state despite heavy noise corruptions in the images.

6 Tracking Real-World Objects

The goal of this section is to examine the proposed method in tracking various real-world objects such as hands, faces, tanks and a special circular object. Note that some tracking videos can be viewed at *http://perception.i2r.a-star.edu.sg/AffineKernelTracking/KernelAffine.html.*

The hand video (Figure 4) was captured in a lab environment using a video camera. The test sequence has 160 frames at 360×288 pixels. The goal was to track the left hand, while the two hands in the video display same color features. The estimated regions (by the propopsed method)

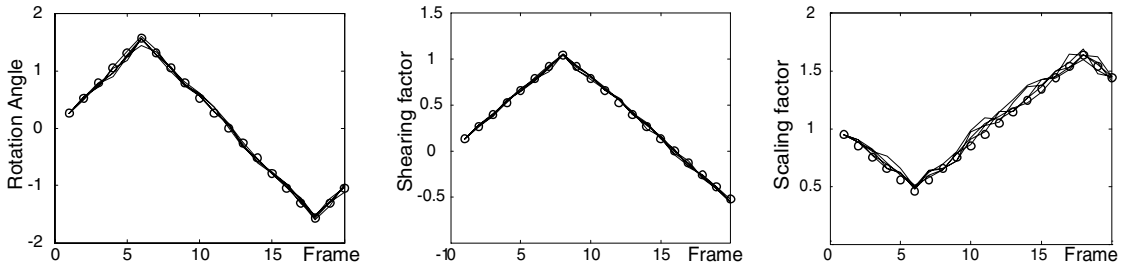

Figure 3. Tracking synthetic objects with affine transformation under image noise at $\sigma = 40$.

Figure 4. Hand tracking with the proposed method.

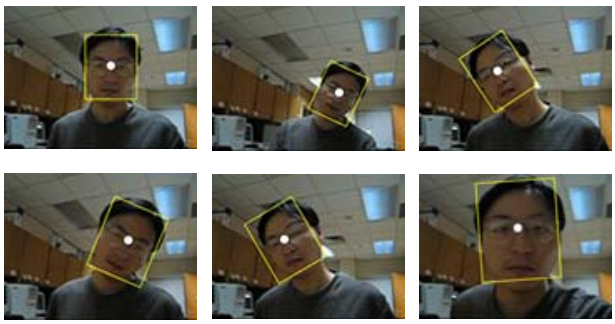

Figure 5. Face tracking.

of the target hand are outlined in the figure. We also examined the mean-shift tracker but it failed tracking the hand usually after a few frames.

The face video (Figure 5) has over 500 frames at 160×120 pixels. The face was continuously moving and rotating, resulting in large variations in size and pose angle. The tracking results are shown here using the estimated outlines of the moving face.

The tank video (Figure 6) has about 200 frames at 356×288 pixels. The tracking is made very challenging by the close similarity in appearance between the camouflaged tank and the meadow background. Note that due to the considerable out-of-plane rotation, some parts of the tanks disappear in the first frame but appear later, and they are not taken into consideration in the object modeling. The figure shows the tracking results using the outlines of estimated object regions.

The experimental results show that the proposed method can accurately recover various objects' states in challenging tasks. They also demonstrate that the kernel-based spatial-color representation model can well distinguish visual objects even though they have similar color features.

Another important experiment aims to test the method on a special task (Figure 7). The target is a red-white circular object that moves and rotates on a red-white chessboard. The tracking task appears to be very difficult (even for human vision system), since the circle looks quite similar to the background. Figure 7 shows the object model as well as the tracking results with the recovered outlines and directions. The inset images draw the details of the windows emcompassing the moving target. It is shown that the proposed method can accurately determine the object's state throughout the sequence. We also tested other methods including the mean-shift tracker or Condensation [19] (a contour tracker with particle filtering), but they failed tracking the circular object. This suggests that neither histogram or contour informationi might be well suited to distinguishing an object from background if both exhibit similar colors and rich textures.

7. Conclusion

We have presented a visual tracking method that can achieve accurate estimation of affine transformation and precise spatial-color representation. On the basis of kernel-based spatial-color representations we formulated a similarity measure called affine matching between image regions, and we derived a mathematical solution for the estimation of transformation parameters for moving objects. We also described an iterative algorithm for robust tracking. Various

Figure 6. Tank tracking images.

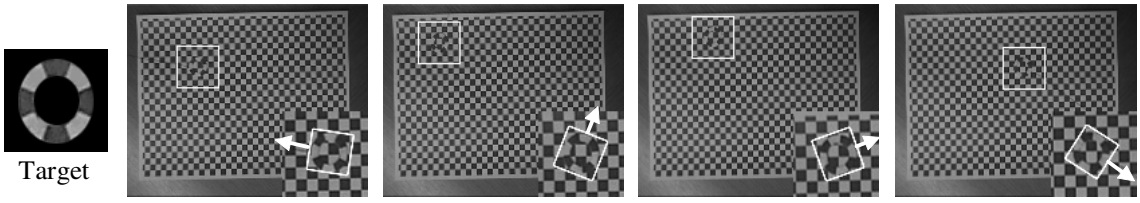

Figure 7. Tracking circle with the proposed method.

experiments on synthetic data and real-world videos were conducted, and the results attest to the method's excellent performance in challenging tasks.

References

[1] D.A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

[2] N. Oliver, A. Pentland, and F. Berard. Lafter: A real-time face and lips tracker with facial expression recognition. *Pattern Recognition*, 33:1369–1382, 2000.

[3] Y. Raja, S. J. Mckenna, and S. Gong. Colour model selection and adaptation in dynamic scenes. In *European Conference of Computer Vision*, pages 460–474, 1998.

[4] Y. Raja, S. J. Mckenna, and S. Gong. Tracking colour objects using adaptive mixture models. *Image Vision Computing*, 17(17):225–231, 1998.

[5] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *IEEE Conf. CVPR*, pages 232–237, 1998.

[6] J. Martin, V. Devin, and J. Crowley. Active hand tracking. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 573–578, 1998.

[7] P. Withagen, K. Schutte, and F. Groen. Likelihood-based object detection and object tracking using color histograms and **EM**. In *ICIP*, volume 1, pages 589–592, 2002.

[8] J. Lee, W. Lee, and D. Jeong. Object tracking method using back-projection of multiple color histogram models. In *International Symposium on Circuits and Systems*, volume 2, pages 668–671, 2003.

[9] D. Comanicui, V. Ramesh, and P. Peer. Real-time tracking of non-rigid objects using mean shift. In *IEEE Conf. CVPR*, volume 2, pages 142–149, 2000.

[10] R. T. Collins. Mean-shift blob tracking through scale space. In *IEEE Conf. CVPR*, pages 234–240, 2003.

[11] M. Black and A. Jepson. Eigentracking: Robut matching and tracking of articulated objects using a view-based representation. In *European Conference of Computer Vision*, pages 329–342, 1996.

[12] N. Papanikolopoulos, P. Khosla, and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision. *IEEE Trans. PAMI*, 1:14–35, 1993.

[13] A. Elgammal, R. Duraiswami, and L. Davis. Probabilistic tracking in joint feature-spatial spaces. In *IEEE Conf. CVPR*, pages 781–788, 2003.

[14] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. PAMI*, 17:790–799, 1995.

[15] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. PAMI*, 25(5):564–577, 2003.

[16] V. Ferrari, T. Tuytelaars, and L. van Gool. Real-time affine region tracking and coplanar grouping. In *IEEE Conf. CVPR*, volume 2, pages 226–233, 2001.

[17] C.J. Yang, R. Duraiswami, and L. Davis. Similarity measure for nonparametric kernel density based object tracking. In *Advances in Neural Information Processing Systems*, pages 1561–1568, 2005.

[18] A. Elgammal, R. Duraiswami, and L. Davis. Efficient non-parametric adaptive color modeling using fast gauss transform. In *IEEE Conf. CVPR*, volume 2, pages 563–570, 2001.

[19] M. Isard and A. Blake. Visual tracking by stochastic propagation of conditional density. In *European Conference of Computer Vision*, pages 343–356, 1996.