

Optimizing Distribution-based Matching by Random Subsampling

Alex Po Leung and Shaogang Gong
Department of Computer Science
Queen Mary, University of London
Mile End Road, London E1 4NS, UK
{alex,sgg}@dcs.qmul.ac.uk

Abstract

We boost the efficiency and robustness of distribution-based matching by random subsampling which results in the minimum number of samples required to achieve a specified probability that a candidate sampling distribution is a good approximation to the model distribution. The improvement is demonstrated with applications to object detection, Mean-Shift tracking using color distributions and tracking with improved robustness for low-resolution video sequences. The problem of minimizing the number of samples required for robust distribution matching is formulated as a constrained optimization problem with the specified probability as the objective function. We show that surprisingly Mean-Shift tracking using our method requires very few samples. Our experiments demonstrate that robust tracking can be achieved with even as few as 5 random samples from the distribution of the target candidate. This leads to a considerably reduced computational complexity that is also independent of object size. We show that random subsampling speeds up tracking by two orders of magnitude for typical object sizes.

1. Introduction

Suppose that we have a model distribution and a search space in which target candidates are compared with the model distribution. The target candidate closest to the model distribution is considered a match. This distribution matching problem appears in various forms such as matching for Time Series, Object Detection, Registration and Tracking. A recently proposed technique called Integral Histogram [1] can obtain histograms for all possible target candidates efficiently in Cartesian Space by assuming that an exhaustive search is required in the search space. After constructing an initial histogram, the technique recursively propagates the aggregated histogram and traverses through the rest of the search space along a scan-line. Dur-

ing each iteration, only a single bin is updated using the Integral Histogram at the previously visited neighboring data points. The histogram of any target candidate can be obtained efficiently with very few arithmetic operations given the integral histogram evaluated for the previous iteration. However, despite that this technique was designed for efficiency gain, it only works well when an exhaustive search is needed. Local optimization algorithms such as Mean Shift tracking do not require exhaustive searches to find the optimal solution. Thus, no or little efficiency gain can be achieved with Integral Histogram when there is no need for global optimization.

We propose a novel approach making use of the fact that not all samples from a target candidate are needed to obtain an sufficiently accurate approximation to the distribution of the target candidate. The distribution is randomly subsampled to form the approximate sampling distribution of the candidate. We estimate the minimum number of samples required for the sampling distribution as a good approximation to the original distribution according to a specified probability, which indicates the reliability of distribution matching with random subsampling. The proposed method leads to significant efficiency gain both for local and global optimization and the estimated minimum number of samples required specifies the resolution limit of the object for robust tracking. We demonstrate these through our experiments with applications to object detection (global optimization), Mean-Shift tracking using color distributions (local optimization) and robust tracking in low-resolution videos (local optimization). It is shown that the proposed method reduces the processing time of Mean-Shift tracking by two orders of magnitude for tracking typical object sizes. Thus, part of this work is devoted to the application to Mean-Shift tracking. Efficient distribution matching for Mean-Shift tracking provides a basis for robust real-time multi-object tracking, object tracking in low-resolution videos and real-time tracking on cheap mobile devices.

Much effort has been made in recent years to develop effective algorithms for real-time object tracking. Neverthe-

less, most tracking algorithms still suffer from fundamental problems including drifts away from targets [3] (partially due to change of viewpoint), inability to adapt to changes of object appearance, dependence on the first frame for template matching [4], instability to track objects under deformations (e.g. deformed contours), the inefficiency of Monte Carlo simulations for temporal tracking [5], and reliance on gradients by active contours [6], i.e. problems with similar intensities on the background and the object, or high gradient edges on the object itself. These problems are due to the complexity of the object dynamics. We also have to deal with difficult tracking conditions which include illumination changes, occlusions, changes of viewpoint, moving cameras and non-translational object motions like zooming and rotation. In particular, Mean-Shift tracking [2] has attracted much attention because of its efficiency and robustness to track nonrigid objects with partial occlusions, significant clutter and variations of object scale. However, as pointed out by Yang and Duraiswami [7], the computational complexity of traditional Mean-Shift tracking is quadratic in the number of samples, making real-time performance difficult. Although the complexity can be made linear with the application of Fast Gauss Transform [7], tracking in real-time remains a problem when large or multiple objects are involved. We propose to boost the efficiency of Mean-Shift tracking using random subsampling. The computational complexity of Mean-Shift tracking with random subsampling is independent of the object size. Therefore, it allows large or multiple objects to be tracked in real time. We also show that, instead of passing hundreds of samples to a traditional Mean-Shift tracker, only 5 random samples are required for the Mean-Shift object tracking with a relatively simple distribution and 15 samples with a typical distribution. In our experiments, random subsampling significantly reduces the processing time by two orders of magnitude for typical object sizes.

2. Distribution Matching

Each bin of the histogram estimate of a sampling distribution follows the Poisson Distribution. Hence, the χ^2 statistic can be used to compute the probability that a model distribution and a sampling distribution are different. To estimate the minimum numbers of samples required from the target candidate and the target model, it is necessary to calculate the χ^2 statistic for the probability, Pr , that a candidate sampling distribution, \hat{p} , is closer to the model distribution, \hat{q} than the background distribution \hat{r} . $1 - Pr$ is the probability for the condition that the similarity measure fails. Specifically, let $\rho[a, b]$ be a given similarity measure. The probability that $\rho[\hat{p}, \hat{q}] \leq \rho[\hat{r}, \hat{q}]$ should be lower than a specified value $1 - Pr$ (our requirement of the reliability),

i.e. the probability that

$$\sum_{i=1}^m \sqrt{\hat{p}_i \hat{q}_i} \leq \sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i} \quad (1)$$

where the Bhattacharyya Coefficient is used as the similarity measure with $\rho[\hat{p}, \hat{q}] = \sum_{i=1}^m \sqrt{\hat{p}_i \hat{q}_i}$ and m is the number of bins for the histogram estimates. However, our derivation of the minimum number of samples required for distribution matching in this work is general enough for any similarity measure to be used. When both of the candidate and model distributions, \hat{p} and \hat{q} , are randomly subsampled and compared, we use the χ^2 statistic [8]

$$\chi^2 = \sum_i \frac{(\sqrt{S/RR}R_i - \sqrt{R/SS}S_i)^2}{R_i + S_i} \quad (2)$$

where R_i and S_i are the numbers of events in bin i from two separate distributions and

$$R = \sum_i R_i, \quad S = \sum_i S_i \quad (3)$$

are the numbers of samples. With the normalized histogram estimate of the candidate and model sampling distributions, \hat{p} and \hat{q} ,

$$\begin{aligned} \chi^2 &= \sum_i \frac{(\sqrt{S/RR}\hat{q}_i - \sqrt{R/SS}\hat{p}_i)^2}{R\hat{q}_i + S\hat{p}_i} \\ &= RS \sum_i \frac{(\hat{q}_i - \hat{p}_i)^2}{R\hat{q}_i + S\hat{p}_i} \end{aligned} \quad (4)$$

where R becomes the number of samples for the model sampling distribution, \hat{q} , and S becomes the number of samples for the candidate sampling distributions, \hat{p} . Both of the candidate and the model distributions, \hat{p} and \hat{q} , can be subsampled to achieve efficient gain. Notice that the distribution of the model never changes during tracking while the candidate distribution has to be evaluated repeatedly in order to find the subwindow closest to the model. As it is not necessary to subsample the model distribution for efficiency gain, instead of subsampling both distributions, only the candidate distribution is subsampled and the model distribution is estimated once during initialization. If only the target candidate is subsampled, the χ^2 statistic becomes

$$\chi^2 = \sum_i \frac{(N_i - n_i)^2}{n_i} \quad (5)$$

$$\chi^2 = \sum_i \frac{(N\hat{p}_i - N\hat{q}_i)^2}{N\hat{q}_i} = N \sum_i \frac{(\hat{p}_i - \hat{q}_i)^2}{\hat{q}_i} \quad (6)$$

where N is the number of samples for the candidate sampling distribution, \hat{p} , when only the candidate distribution,

\hat{p} , is subsampled. The probability that two distributions are not different can be obtained using the χ^2 probability function $Q(\chi^2|\nu) = Q(\frac{\nu}{2}, \frac{\chi^2}{2})$, where $Q(\frac{\nu}{2}, \frac{\chi^2}{2})$ is the incomplete Gamma function and ν is the number of degree of freedom [8]:

$$Pr = Q(\frac{1}{2}\nu, \frac{1}{2}\chi^2_N) \quad (7)$$

3. A Constrained Optimization Problem



Figure 1. Experiments 1, 2, 3 and 4: The four rows of images represent four separate video sequences produced by four experiments respectively. In the first experiment (first row), only 5 random samples are picked from each of the candidate and model images, 10 samples in the second experiment (second row), 15 samples in the third (third row) and 150 samples in the fourth (fourth row).

For all \hat{p} satisfying

$$\sum_{i=1}^m \sqrt{\hat{p}_i \hat{q}_i} \leq \sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i}, \quad (8)$$

we find \hat{p} with the lowest probability that the sampling distribution, \hat{p} , and the model distribution, \hat{q} , are different. In other words, we estimate, after random subsampling, how likely would $\sum_{i=1}^m \sqrt{\hat{p}_i \hat{q}_i}$ be greater than $\sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i}$, given the number of samples, N . The problem to find the probability that a candidate sampling distribution is closer to the model distribution than the background distribution is a constrained optimization problem with the objective function as the probability. The χ^2 probability function is a monotonic function of χ^2 given by Equation 7. Thus, to minimize the probability, $1 - Pr$, i.e.

$$\min_{\hat{p}} (1 - Pr_N\{\hat{p}\}),$$



Figure 2. Experiments 5, 6, 7 and 8: The four rows of images represent four separate video sequences produced by four experiments respectively. In Experiment 5 (first row), only 2 random samples are picked from each of the candidate and model images, 3 samples in Experiment 6 (second row), 5 samples in Experiment 7 (third row) and all samples from the candidate and model images (traditional Mean-Shift tracking) in Experiment 8 (fourth row).

we simply minimize the χ^2 :

$$\begin{aligned} \min_{\hat{p}} \chi^2_N\{\hat{p}\} &= \min_{\hat{p}} N \sum_i \frac{(\hat{p}_i - \hat{q}_i)^2}{\hat{q}_i} = N \min_{\hat{p}} \sum_i \frac{(\hat{p}_i - \hat{q}_i)^2}{\hat{q}_i} \\ \text{subject to} \quad &\sum_{i=1}^m \sqrt{\hat{p}_i \hat{q}_i} \leq \sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i}, \\ &\sum_{i=1}^m \hat{p}_i = 1, \text{ and } 0 \leq \hat{p}_i \leq 1 \quad \forall i \end{aligned} \quad (9)$$

3.1. Incomplete Gamma Function and Its Inverse

The incomplete Gamma function gives us the corresponding probability for a given χ^2 and ν , the degree of freedom:

$$Q(a, x) = \frac{1}{\Gamma(a)} \int_x^\infty e^{-t} t^{a-1} dt \quad (a > 0) \quad (10)$$

where

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt. \quad (11)$$

However, in our case, a specified probability, Pr , and the degree of freedom, ν are given and we need to know χ^2 . That is to say the inverse of the incomplete Gamma function needs to be computed. There does not exist any analytic solution to the inverse of the incomplete Gamma function. We adopt a numerical method by DiDonato and Alfred [9] that requires only a few iterations to obtain the χ^2 for the specified probability.

3.2. Lagrangian Multipliers

Lagrangian Multipliers are used to solve constrained optimization problems with differentiable functions. The method maximizes an associated function called the Lagrangian, L , with an additional control variable, λ_i for each constraints. To deal with inequality constraints, Lagrangian Multipliers can be used as if we are dealing with equality constraints before introducing the complementary slackness condition to eliminate solutions for the inequality constraints. Hence, the inequality constraint in Equation 9

$$\sum_{i=1}^m \sqrt{\hat{\mathbf{p}}_i \hat{q}_i} \leq \sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i}$$

can be transformed to the equality constraint

$$\sum_{i=1}^m \sqrt{\hat{\mathbf{p}}_i \hat{q}_i} = \sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i}. \quad (12)$$

We use Lagrangian Multipliers to solve the constrained optimization problem with the multipliers λ_1 and λ_2 . The Lagrangian is

$$\begin{aligned} L = & \sum_i \frac{(\hat{\mathbf{p}}_i - \hat{q}_i)^2}{\hat{q}_i} - \\ & \lambda_1 \left(\sum_{i=1}^m \sqrt{\hat{\mathbf{p}}_i \hat{q}_i} - \sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i} \right) - \\ & \lambda_2 \left(\sum_{i=1}^m \hat{\mathbf{p}}_i - 1 \right) \end{aligned} \quad (13)$$

with constraints

$$\sum_{i=1}^m \sqrt{\hat{\mathbf{p}}_i \hat{q}_i} = \sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i}, \text{ and } \sum_{i=1}^m \hat{\mathbf{p}}_i = 1. \quad (14)$$

To find the minimum of L , we obtain the partial derivative $\partial L / \partial \hat{\mathbf{p}}_i$:

$$\frac{\partial L}{\partial \hat{\mathbf{p}}_i} = 2 \left(\frac{\hat{\mathbf{p}}_i - \hat{q}_i}{\hat{q}_i} \right) - \frac{\lambda_1}{2} \sqrt{\frac{\hat{q}_i}{\hat{\mathbf{p}}_i}} - \lambda_2 \quad (15)$$

and let

$$\frac{\partial L}{\partial \hat{\mathbf{p}}_i} = 0. \quad (16)$$

We have the following set of nonlinear equations to be solved:

$$\frac{4}{\hat{q}_i} \hat{\mathbf{p}}_i^{\frac{3}{2}} + 2(\lambda_2 - 2) \hat{\mathbf{p}}_i^{\frac{1}{2}} - \lambda_1 \sqrt{\hat{q}_i} = 0 \quad \forall i \quad (17)$$

$$\sum_{i=1}^m \sqrt{\hat{\mathbf{p}}_i \hat{q}_i} - \sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i} = 0 \quad (18)$$

$$\sum_{i=1}^m \hat{\mathbf{p}}_i - 1 = 0 \quad (19)$$

3.3. Constraint Satisfaction

Instead of solving the $m + 2$ nonlinear equations using numerical methods (m could be very large), we obtain an analytic solution to the optimization problem on a case-by-case basis.

When \hat{r} and \hat{q} do not overlap in the color space (Case 1),

$$\sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i} = 0 \quad (20)$$

which, to satisfy the constraint in Equation 9, requires

$$\sum_{i=1}^m \sqrt{\hat{\mathbf{p}}_i \hat{q}_i} = 0. \quad (21)$$

It is impossible to satisfy Equation 21 because $\hat{\mathbf{p}}$ is a sampling distribution of \hat{q} . $\hat{\mathbf{p}}$ and \hat{q} must overlap in the color space. Therefore, the constraint cannot be satisfied when \hat{r} and \hat{q} do not overlap and the candidate sampling distribution must be closer to the model distribution than the background distribution regardless of the number of samples used, N . In this case, theoretically, we can use any number of samples larger than 0.

When \hat{r} and \hat{q} do not largely overlap (Case 2), $\sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i}$ is small. To check if there exists any $\hat{\mathbf{p}}$ satisfying the constraint in Equation 9, we can compare the minimum of $\sum_{i=1}^m \sqrt{\hat{\mathbf{p}}_i \hat{q}_i}$ with $\sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i}$. If the condition

$$\min_{\hat{\mathbf{p}}} \sum_{i=1}^m \sqrt{\hat{\mathbf{p}}_i \hat{q}_i} \leq \sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i} \quad (22)$$

is not satisfied, there does not exist any $\hat{\mathbf{p}}$. We can obtain the smallest $\sum_{i=1}^m \sqrt{\hat{\mathbf{p}}_i \hat{q}_i}$ possible when $\hat{\mathbf{p}}_s = 1$ where s is the index for the bin of \hat{q} with the lowest value except 0. Thus, we have

$$\min_{\hat{\mathbf{p}}} \sum_{i=1}^m \sqrt{\hat{\mathbf{p}}_i \hat{q}_i} = \sqrt{\hat{q}_s}. \quad (23)$$

And, we check if there exists any $\hat{\mathbf{p}}$ satisfying the constraint in Equation 9 using

$$\sqrt{\hat{q}_s} \leq \sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i}. \quad (24)$$

The number of samples, N , can be again any number larger than 0 if Equation 24 is not satisfied.

When \hat{r} is close to \hat{q} (Case 3), i.e. there exists any $\hat{\mathbf{p}}$ satisfying the constraint in Equation 9, we use an approximation to the solution derived in the following section.

3.4. Analytic Solution

To minimize $\sum_i (\hat{\mathbf{p}}_i - \hat{q}_i)^2 / \hat{q}_i$, $\hat{\mathbf{p}}$ has to be close to \hat{q} while satisfying $\sum_{i=1}^m \sqrt{\hat{\mathbf{p}}_i \hat{q}_i} \leq \sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i}$. Hence, $\hat{\mathbf{p}}$ should be

very close to the hyperplane

$$\sum_{i=1}^m \sqrt{\hat{p}_i \hat{q}_i} = \sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i} \quad (25)$$

when \hat{p} is closest to \hat{q} . In addition, it can be observed that although different similarity measures give different values (quantitative differences) when the difference of two distributions is measured, they exhibit little qualitative differences. With the little qualitative differences,

$$\sum_i \frac{(\hat{p}_i - \hat{q}_i)^2}{\hat{q}_i} \approx \sum_i \frac{(\hat{r}_i - \hat{q}_i)^2}{\hat{q}_i} \quad (26)$$

if $\sum_{i=1}^m \sqrt{\hat{p}_i \hat{q}_i} = \sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i}$. Therefore, $\min_{\hat{p}} \sum_i (\hat{p}_i - \hat{q}_i)^2 / \hat{q}_i$ with \hat{p} on the hyperplane (25) using Equation 26 equals $\sum_i (\hat{r}_i - \hat{q}_i)^2 / \hat{q}_i$. The final analytic solution to Equation 9 is approximately

$$\min_{\hat{p}} \chi_N^2\{\hat{p}\} \approx N \sum_{i \in T} \frac{(\hat{r}_i - \hat{q}_i)^2}{\hat{q}_i} \quad (27)$$

where $T = \{i : \hat{q}_i \neq 0\}$.

3.5. Numerical Solution by Monte Carlo methods

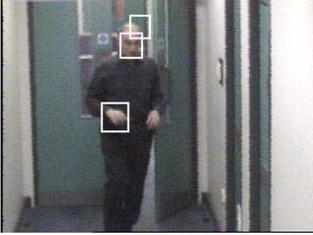


Figure 3. Image for the Calculation of the Number of Samples Required for Reliable Distribution Matching

A numerical solution to the constrained optimization problem can be obtained using the acceptance-rejection method. In order to find the minimum of $\sum_i (\hat{p}_i - \hat{q}_i)^2 / \hat{q}_i$, we consider \hat{p} as a random vector. N_p samples of \hat{p} are generated by rejecting \hat{p} when the constraint $\sum_{i=1}^m \sqrt{\hat{p}_i \hat{q}_i} \leq \sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i}$ in Equation 9 is not satisfied. Note that the number of samples, N_p , generated for \hat{p} does not include the rejected samples. With \hat{p} , the distribution of $\sum_i (\hat{p}_i - \hat{q}_i)^2 / \hat{q}_i$ can also be obtained. When N_p is large enough, the solution equals the smallest $\sum_i (\hat{p}_i - \hat{q}_i)^2 / \hat{q}_i$ from the set of samples \hat{p} .

The image in Figure 3 is used to give us an idea how many samples are required for reliable distribution matching. In the image, the face is the target for the estimation of the model distribution, \hat{q} . One of the hands and the background with part of the face are the negative examples for

the background distributions, \hat{r} . With the hand as the negative example used for the estimation of \hat{r} , the numerical solution is 2.01 while, with the background and part of the face used for the estimation of \hat{r} , the numerical solution is 4.25. Both of the solution values are obtained twice by repeating the Monte Carlo methods to see the effects of statistical fluctuations. For the repeated run, the solution values are respectively 2.06 and 4.17. We can compare the numerical solution with our analytic solution shown in Table 1. Although the analytic solution is not very accurate, it can be used for real-time applications because of its simplicity. A safer approach is to multiply the number of samples required by a factor such as 2 so as to compensate for the lack of accuracy when the approximate analytic solution is used. The time required to obtain the numerical solution on Matlab depends on the number of rejected samples. It is 69.52 seconds with the hand as the negative example and 561.97 seconds with the background and part of the face in Figure 3. In addition, the solution for Case 1 and Case 2 (when \hat{q} and \hat{r} do not largely overlap) is not an approximation so the numerical solution is not necessary at all for Case 1 and Case 2.

	Background	Hand
First Run	4.25	2.01
Second Run	4.17	2.06
Analytic Solution	2.62	2.31

Table 1. The Obtained Numerical Solution and Analytic Solution

With the solution in Table 1 which is $\min_{\hat{p}} \sum_i (\hat{p}_i - \hat{q}_i)^2 / \hat{q}_i$, we need \hat{p} the specified probability that Equation 8 is satisfied and the degree of freedom of the histogram for the candidate sampling distribution, ν , to calculate the number of samples required using the inverse of the incomplete Gamma function. $\nu + 1$ equals the number of bins of the histogram for the model distribution with $\hat{q}_i \neq 0$. Let us set $Pr = 1 - 10^{-4}$ (our requirement for the reliability) and we have $\nu = 12$. With the numerical method mentioned in Section 3.1, we obtain $\chi_N^2 = 39.14$. As $\chi_N^2 = N \min_{\hat{p}} \sum_i (\hat{p}_i - \hat{q}_i)^2 / \hat{q}_i$, the number of samples required, $N = \chi_N^2 / \min_{\hat{p}} \sum_i (\hat{p}_i - \hat{q}_i)^2 / \hat{q}_i$, with the hand as the negative example is 19.47. N , with the background and part of the face as the negative example is equal to 9.21.

4. Improved Mean-Shift Tracking

The probabilities of color u in the target model and the target candidate are given by

$$\hat{q}_i = C \sum_{i=1}^n k(\|x_i^*\|^2) \delta[b(x_i^*) - u], \text{ and} \quad (28)$$

$$\hat{p}_i(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y-x_i}{h}\right\|^2\right) \delta[b(x_i) - u] \quad (29)$$

where C and C_h are normalization factors, x_i^* and x_i are the pixel locations of the target model and the target candidate, and k is a convex and monotonic decreasing kernel profile. The distance between the two discrete distributions is defined as [2]

$$d(y) = \sqrt{1 - \rho[\hat{p}(y), \hat{q}]} \quad (30)$$

where

$$\hat{\rho}(y) = \rho[\hat{p}(y), \hat{q}] = \sum_{i=1}^m \sqrt{\hat{p}_i(y) \hat{q}_i}, \quad (31)$$

the sample estimate of the Bhattacharyya coefficient between p and q .

Our derived minimum number of samples required, N , with the specified probability, $1 - Pr$, that

$$\sum_{i=1}^m \sqrt{\hat{p}_i \hat{q}_i} \leq \sum_{i=1}^m \sqrt{\hat{r}_i \hat{q}_i} \quad (32)$$

can be applied to estimate the reliability of Mean-Shift tracking with random subsampling. When random subsampling is utilized, the reliability of tracking can be estimated by our specified probability, $1 - Pr$, that the similarity measure for Mean-Shift tracking fails (the condition 32).

5. Experiments

For Mean-Shift tracking with random subsampling, we evaluate the efficiency of the proposed method and obtained the surprising result that Mean-Shift tracking requires only very few samples. Robust tracking can be achieved with as few as even 5 random samples from the image of the object. With random subsampling, the computational complexity of Mean-Shift tracking is independent of object size. Near real-time performance is obtained even in our Matlab implementation because, instead of passing hundreds of samples to a traditional Mean-Shift tracker, only 5 random samples are required for the Mean-Shift tracker to track objects with a relatively simple distribution and 15 samples for a typical distribution. The speed of our tracker is 2.17 fps (frames per second) to track the head of a person in a given video sequence while the speed of a traditional implementation used in [7] is 0.011 fps (197 times slower) with the same tracking sub-window size of 24x25 (600 samples without random subsampling).

Our first four experiments, as shown in Figure 1, evaluate the number of random samples required to track a typical object which is a human face in the experiments. In the first experiment, only 5 random samples are picked from each of the candidate and model images, 10 samples in the

second experiment, 15 samples in the third and all samples from the candidate and model images (traditional Mean-Shift tracking) in the fourth. Tracking fails with too few samples. The tracker fails with 5 samples from the image of the object. With 10 samples, as shown in the second row of Figure 1, the tracker tracks the object successfully but the trajectory is not very stable when compared with the tracker with 150 samples (fourth row). There is no difference between the tracking performance of the Mean-Shift tracker with 15 samples (third row) and that of the tracker with 150 samples. A larger number of samples more than 15 does not make any difference to the tracking performance of the tracker. On Matlab, the time required for the tracking in Experiments 1, 2, 3 and 4 are 2.95 fps (frames per second), 2.63 fps, 2.18 fps and 0.04 fps respectively. Experiments 5, 6, 7 and 8, as shown in Figure 2, evaluate the number of random samples required to track an object with a relatively simple distribution which is the head of a person. In Experiment 5, only 2 random samples are picked from each of the candidate and model images, 3 samples in Experiment 6, 15 samples in Experiment 7 and all samples from the candidate and model images (traditional Mean-Shift tracking) in Experiment 8. The tracker fails with 2 samples from the image of the object. With 3 samples, as shown in the second row of Figure 2, the tracker tracks the object successfully but the trajectory is not very stable when compared with the original Mean-Shift tracker (fourth row). There is no difference between the tracking performance of the Mean-Shift tracker with 5 samples (third row) and that of traditional Mean-Shift tracking. Therefore, for an object with a relatively simple color distribution, a larger number of samples more than 5 does not make any difference to the tracking performance of the tracker. Moreover, Figure 8 shows the Root-Mean-Square difference in the tracked positions against the number of samples used. Notice that we compared the tracked positions with the positions from the tracker using 100 samples. Both of the original Mean-Shift tracker [2] and a variant using a new similarity measure by Yang, Duraiswami and Davis [7] are evaluated with random subsampling. Although the latter method uses a kernel function instead of histograms to estimate the color distributions of the target candidate and target model, the performance is similar to that of the original tracker with any number of samples used for tracking. With our Matlab implementation, the time required for the tracking in Experiments 5, 6, 7 and 8 are 3.19 fps (frames per second), 3.12 fps, 2.17 fps and 0.011 fps respectively. Successful tracking with 5 random samples is 197 times faster than traditional Mean-Shift tracking with the same tracking sub-window size of 24x25 (600 samples without random subsampling). The computational complexity of traditional Mean-Shift tracking is quadratic in the number of samples. In our experiments, random subsampling significantly reduces the pro-



Figure 4. A person dances in front of the camera.



Figure 5. A hockey player dribbles the hockey puck.

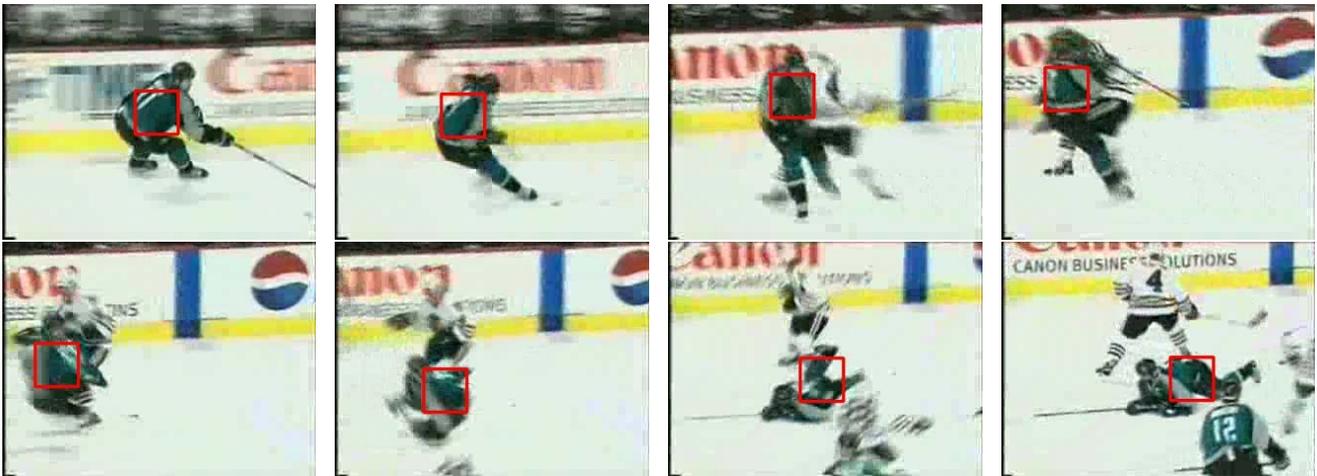


Figure 6. A hockey player crashes into the other and is pushed over.

cessing time by two orders of magnitude for typical object sizes. The near real-time performance in our Matlab implementation demonstrates that Mean-Shift tracking with random subsampling runs much faster than 30 frames per second as a Matlab implementation is typically at least two orders of magnitude slower than an implementation with C.

We should note that the Mean-Shift vector from the original Mean-Shift tracking algorithm [2] suffers from numerical problems when the number of samples is too low

(usually when $N \leq 10$ for Experiments 1-8). The problems are due to the denominator of the vector. When the number of samples is low, the denominator can become 0. Mean-Shift tracking with a new similarity measure by Yang, Duraiswami and Davis [7] does not suffer from numerical problems even when the number of samples is very low (the algorithm works as long as the number of samples is larger than 1).

Further experiments evaluate Mean-Shift tracking with

random subsampling (using the new similarity measure [7]) to track fast moving objects in three different videos in the resolution of 320x240. The scenes in the videos include a person dancing in front of the camera in Figure 4, a hockey player dribbling the hockey puck in Figure 5 and a hockey player crashing into the other and pushed over in Figure 6. It is demonstrated that our tracker is able to track fast moving objects with the number of samples $N = 50$.

5.1. Efficient Object Detection and Object Tracking in Low-Resolution Videos

Significant efficiency gain with random subsampling for object detection using exhaustive searches has also been achieved as shown in Figure 7. Only 10 samples from the target candidate are needed for successful object detection instead of all 600 samples from each subwindow.

The estimated minimum number of samples also provides a basis to analysis the tracking performance of the tracker for low-resolution video sequences. A video sequences is down-sampled to create low-resolution video sequences to test the validity of our estimate. Figure 9 shows Mean-Shift tracking is able to track a subwindow with the size of 5x5 pixels. There are only 25 samples available to the tracker.

6. Conclusion

To conclude, we boost the efficiency and robustness of distribution-based matching by random subsampling with applications to object detection, Mean-Shift tracking using color distributions and tracking with improved robustness for low-resolution video sequences. The minimum number of samples required for distribution matching is derived to achieve a specified probability that a candidate sampling distribution is a good approximation to the model distribution. Random subsampling with the minimum of the samples required is shown to significantly speed up tracking by two orders of magnitude for typical object sizes.



Figure 7. Object Detection using Color Distributions with the Number of Samples 10

References

[1] F. Porkili. "Integral histogram: A fast way to extract histograms in cartesian spaces". *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

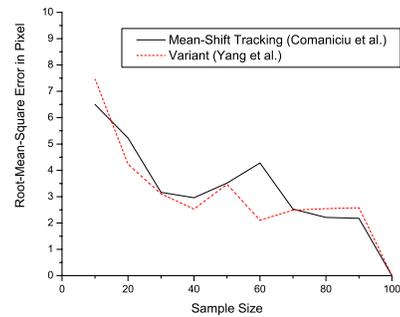


Figure 8. Root-Mean-Square Difference in Position against the Number of Samples



Figure 9. Low-Resolution Video Sequence

[2] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-Based Object Tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 5, May 2003.

[3] I. Matthews, T. Ishikawa and S. Baker, "The Template Update Problem," *PAMI(26)*, No. 6, 2004, pp. 810-815.

[4] F. Jurie and M. Dhome, "Hyperplane Approximation for Template Matching," *IEEE PAMI 24(7)*, 996-1000, 2002.

[5] S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *Transaction of Signal Processing*, 50(2):174-188, 2002.

[6] M. Isard and A. Blake, "CONDENSATION – conditional density propagation for visual tracking," *IJCV*, 29, 1, 5–28, (1998).

[7] Changjiang Yang, Ramani Duraiswami, Larry S. Davis, "Efficient Mean-Shift Tracking via a New Similarity Measure," *CVPR*, 1, 176-183, 2005.

[8] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, "Numerical Recipes in C: The Art of Scientific Computing". Cambridge: Cambridge University Press, 1992.

[9] Armido DiDonato, Alfred Morris, "Computation of the Incomplete Gamma Function Ratios and their Inverse", *ACM Transactions on Mathematical Software*, Volume 12, pages 377-393, 1986.