Automatic Acquisition of Motion Trajectories: Tracking Hockey Players*

Kenji Okuma James J. Little David Lowe

Computer Science Dept.
University of British Columbia
Vancouver, BC, Canada V6T 1Z4
okumak@cs.ubc.ca

ABSTRACT

Computer systems that have the capability of analyzing complex and dynamic scenes play an essential role in video annotation. Scenes can be complex in such a way that there are many cluttered objects with different colors, shapes and sizes, and can be dynamic with multiple interacting moving objects and a constantly changing background. In reality, there are many scenes that are complex, dynamic, and challenging enough for computers to describe. These scenes include games of sports, air traffic, car traffic, street intersections, and cloud transformations.

Our research is about the challenge of inventing a descriptive computer system that analyzes scenes of hockey games where multiple moving players interact with each other on a constantly moving background due to camera motions. Ultimately, such a computer system should be able to acquire reliable data by extracting the players' motion as their trajectories, querying them by analyzing the descriptive information of data, and predict the motions of some hockey players based on the result of the query. Among these three major aspects of the system, we primarily focus on visual information of the scenes, that is, how to automatically acquire motion trajectories of hockey players from video. More accurately, we automatically analyze the hockey scenes by estimating parameters(i.e., pan, tilt, and zoom) of the broadcast cameras, tracking hockey players in those scenes, and constructing a visual description of the data by displaying trajectories of those players.

Many technical problems in vision such as fast and unpredictable players' motions and rapid camera motions make our challenge worth tackling. To the best of our knowledge, there have not been any automatic video annotation systems for hockey developed in the past. Although there are many obstacles to overcome, our efforts and accomplishments would hopefully establish the infrastructure of the automatic hockey annotation system and become a milestone for research in automatic video annotation in this domain.

Keywords: homography, trajectory, tracking, video indexing

1. INTRODUCTION

With the advance of information technologies and the increasing demand for managing the vast amount of visual data in video, there is a great potential for developing reliable and efficient systems that are capable of understanding and analyzing scenes. In order to design such systems that describe scenes in video, it is essential to compensate for camera motions by estimating a planar projective transformation (i.e., homography).[21–25] In this paper, we present our contributions for automatically estimating homography and tracking multiple moving objects on a constantly changing background.

First, we present our algorithm[19] for automatically computing homographies by combining the KLT tracking system,[26–28] RANSAC[29] and the normailzed Direct Linear Transformation (DLT) algorithm.[21] A new model-based correction system that fits projected images to the model is used to reduce projection errors produced by automatically computed homography. The system detects features that lie on line segments of projected images and minimize the difference between projected images and the model using the normalized DLT algorithm. Similarly, Koller *et. al*[30] uses line segments of moving vehicles to track them from road traffic scenes monitored by a stationary camera. Yamada *et. al*[31] uses line segments and circle segments of the soccer field to estimate camera parameters and mosaic a short sequence of video images in order to track players and a ball in the sequence.

*This research was supported by a grant from the Networks of Centres of Excellence Institute for Robotics and Intelligent Systems.

Automated tracking of multiple objects is still an open problem in many settings, including car surveillance,[10] sports[12, 13] and smart rooms[6] among many others.[5, 7, 11] In general, the problem of tracking visual features in complex environments is fraught with uncertainty.[6] To track a varying number of hockey players on a sequence of digitized video from TV we use a probabilistic method that relies on matching color histograms of the player's appearance combined with a method for tracking moving objects. Over the last few years, particle filters, also known as condensation or sequential Monte Carlo, have proved to be powerful tools for image tracking.[3, 8, 14, 15] We use a variation of particle filters, called a boosted particle filter (BPF),[20] which combines a mixture particle filter[17] with a detection method based on cascaded Adaboost detectors[18] to track multiple players.

Estimating homographies allows to transform a sequence of digitized video to the globally consistent rink map by compensating for camera motions of a broadcast camera. BPF tracks multiple hockey players and estimates their poistions in the coordinate of a original video sequence. Consequently, trajectories of tracked players are produced by combining estimated homographies and tracking results. These trajectories, as visual information of hockey scenes are entered into a database system that permits data mining[33, 34] and analyzes descriptive information of scenes in order to eventually recognize motions of hockey players being analyzed.

In the subsequent section, the theoretical background of the homography (also known as a plane projective transformation, or collineation) is described. The third section describes our algorithm for automatically computing homography between successive frames in image sequences. The fourth section explains our model-based correction system for compensating projection errors produced by automatic computation of homography. In the fifth section, we present how to track hockey players by BPF. The result of our experiments is presented in the sixth section. The final section concludes this paper and indicates future directions of our research.

2. HOMOGRAPHY

The definition of a homography[21] (or more generally *projectivity*) is an invertible mapping of points and lines on the projective plane $\mathbb{P}^2$. This gives a homography two useful properties. For a stationary camera with its fixed centre of projection, it does not depend on the scene structure (i.e., depth of the scene points) and it applies even if the camera "pans and zooms", which means to change the focal length of the camera while it is rotating about its centre. With these properties, a homography is applied under the circumstance which the camera pans, tilts, rotates, and zooms about its centre.

2.1. Representation of Homography

Homogeneous representation is used for a point $\mathbf{x} = (x, y, w)^\top$, which is a 3-vector, representing a point $(x/w, y/w)^\top$ in Euclidean 2-space $\mathbb{R}^2$. As homogeneous vectors, points are also elements of the projective space $\mathbb{P}^2$. It is helpful to consider the inhomogeneous coordinates of a pair of matching points in the world and image plane as $(x/w, y/w)^\top$ and $(x'/w', y'/w')^\top$, because points are measured in the inhomogeneous coordinates directly from the world plane. A homography[25]is a linear transformation of $\mathbb{P}^2$, which is expressed in inhomogeneous form as:

$$x'/w' = \frac{Ax + By + C}{Px + Qy + R},\ y'/w' = \frac{Dx + Ey + F}{Px + Qy + R} \tag{1}$$

where vectors $\mathbf{x}$ and $\mathbf{x}'$ are defined in homogeneous form, and a transformation matrix $\mathbf{M}$ as:

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ w \end{pmatrix} \qquad \mathbf{x}' = \begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} \qquad \mathbf{M} = \begin{bmatrix} A & B & C \\ D & E & F \\ P & Q & R \end{bmatrix}$$

where $\mathbf{x} \leftrightarrow \mathbf{x}'$ denotes a pair of 2D point correspondences. Normally the scale factor w is chosen in such a way that x/w and y/w have order of 1, so that numerical instability is avoided.

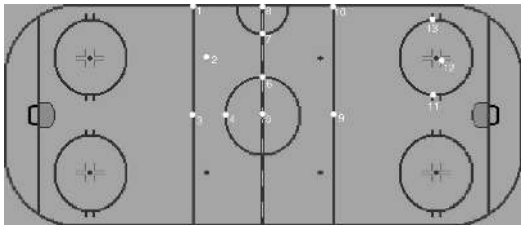
Now, Eq. (10) can be written as:

$$\mathbf{x}' = c\mathbf{M}\mathbf{x} \tag{2}$$

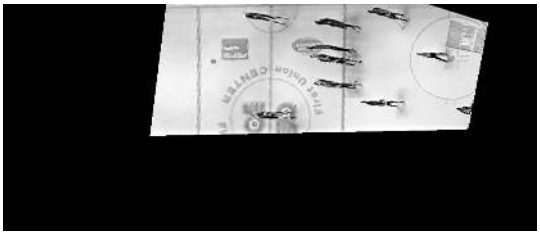
where c is an arbitrary nonzero constant. Homographies and points are defined up to a nonzero scalar c, and thus there are 8 degrees of freedom for homography. Often, $R = 1$ and the scale factor is set as $w = 1$. Eq. (2) can now be written simply as:

$$\mathbf{x}' = \mathbf{H}\mathbf{x}$$

(a) The original image



(b) the correspondences on the rink map



(c) The transformation result

Figure 1. Homography Transformation. (a) shows the original image (320×240) to be transformed. (b) shows manually selected points that are corresponding to those on the rink in the image, which are used only for the initial frame in a video sequence. (c) is the result (1000×425) of the transformation.

where $\mathbf{H}$ is 3×3 matrix called a homography. Every correspondence ($\mathbf{x}$, $\mathbf{x}'$) gives two equations. Therefore, computing a homography with this algorithm requires at least four correspondences. The normalized DLT algorithm[21] is used to compute frame-to-frame homographies. Figure 1 shows the result of homography transformation by the normalized DLT algorithm based on manually selected correspondences.

3. COMPUTATION OF THE HOMOGRAPHY

Given a sequence of images acquired by a broadcast camera, the objective is to specify a point-to-point planar homography map in order to remove the camera motion in images. Our algorithm has four major steps to automatically compute homographies.

3.1. Reduce vision challenges

Since the source of our data is video clips of broadcast hockey games, there are various vision problems to deal with, namely camera flashes that cause a large increase of image intensities and rapid motions of broadcast cameras for capturing highly dynamic hockey scenes.

3.1.1. Flash Detection

In order to deal with camera flashes in digitized hockey sequences, automatic detection of those flashes is necessary. Figure 2 shows the average intensity of over 2300 consecutive frames.

In the graph, there are several sudden spikes which indicate that there is a camera flash for that particular frame. With our observation of camera flashes, a simple flash detection method is derived by taking the difference of the average intensity from two successive frames.

3.1.2. Prediction

Broadcast cameras often make rapid motions to capture dynamic hockey scenes during a game. The amount of motion, however, can be reduced by *predicting* the current camera motion based on the previous camera motion. For instance, given a frame-to-frame homography $\mathbf{H}_{1,2}$ that represents the camera motion from Frame 1 to Frame 2, $\mathbf{H}_{1,2}$ is used as the estimation of $\mathbf{H}_{2,3}$ to transform Frame 2 so that we can minimize the amount of motion between Frame 2 and Frame 3. That is, to have the following assumption:

$$\mathbf{H}_{\mathrm{n,n-1}} \approx \mathbf{H}_{\mathrm{n+1,n}}$$

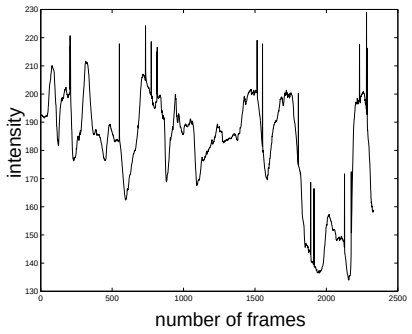


Figure 2. The average intensities over 2300 frames. The vertical axis indicates the number of the intensity ranging from 130 to 230 where 130 indicates a darker pixel and 230 is a brighter pixel. The horizontal axis is the number of the frame.

where every successive frame is processed and $\mathbf{H}_{n-1,n}$ means a homography from Frame $n-1$ to Frame n. This assumption holds only without skipping too many frames. In our experiments, our system processes every fourth frame of data sampled at 30 frames per second, and shows that it is capable of compensating a large motion of a camera.

3.2. Acquisition of correspondences

For successful homography computation, it is crucial to have a reliable set of point correspondences that gives an accurate homography. KLT[26–28] gives those correspondences automatically by extracting features and tracking them. That is, those features that are successfully tracked by KLT between images are ones that are corresponding to each other.

3.3. RANSAC: Elimination of outliers

Correspondences gained by KLT are yet imperfect to estimate a correct homography because they also include outliers. Though an initial set of correspondences selected by KLT contains a good proportion of correct matches, RANSAC[29] is used to identify consistent subsets of correspondences and obtain a better homography. In RANSAC, a putative set of correspondences is produced by a homography based on a random set of four correspondences, and outliers are eliminated by the homography.

3.3.1. Sample Selection

Distributed spatial sampling is used to avoid choosing too many collinear points to produce degenerate homography. In the sampling, a whole image is divided into four sub-regions of an equal size so that each correspondence is sampled from a different sub-region. Once four point correspondences are sampled with a good spatial distribution, a homography is computed based on those correspondences and use the homography to select an initial set of inliers. For inlier classification, we use the symmetric transfer error $d^2_{transfer}$, defined[21]:

Let $\mathbf{x} \leftrightarrow \mathbf{x}'$ be the point correspondence and $\mathbf{H}$ be a homography such that $\mathbf{x}' = \mathbf{H}\mathbf{x}$, then

$$d^2_{transfer} = d(\mathbf{x}, \mathbf{H^{-1}}\mathbf{x}')^2 + d(\mathbf{x}', \mathbf{H}\mathbf{x})^2 \tag{3}$$

where $d(\mathbf{x}, \mathbf{H^{-1}}\mathbf{x}')$ represents the distance between $\mathbf{x}$ and $\mathbf{H^{-1}}\mathbf{x}'$. After the symmetric transfer error is estimated from each point correspondence, we then calculate the standard deviation of the sum of the symmetric errors from all correspondences, which is denoted by σ_{error} and defined as follows:

Suppose there are N point correspondences and each one of them has the symmetric transfer error $\{d^2_{transfer}\}_{i=1\ldots N}$, then:

$$\sigma_{error} = \sqrt{\frac{\sum_{1 \le i \le N} (\{d^2_{transfer}\}_i - \mu)^2}{N-1}} \tag{4}$$

where μ is the mean of the symmetric errors. Now we can classify an outlier as any point $\mathbf{x}_i$ that satisfies the following condition:

$$\gamma(\mathbf{x}_i) = \begin{cases} 0 & \{d^2_{transfer}\}_i \ge \sqrt{5.99} * \sigma_{error} \quad \text{(outlier)} \\ 1 & \text{Otherwise} \quad \text{(inlier)} \end{cases} \tag{5}$$

where γ is a simple binary function that determines whether the point $\mathbf{x}_i$ is an outlier. The distance threshold is chosen based on a probability of the point being an inlier. The constant real number, $\sqrt{5.99}$, is, therefore, derived by computing the probability distribution for the distance of an inlier based on the model of which this case is the homography matrix.[21]

3.3.2. Adaptive termination of sampling

After sampling four spatially distributed correspondences and classifying inliers and outliers, the termination of sampling needs to be determined in order to save unnecessary computation. An adaptive algorithm[21] for determining the number of RANSAC samples is implemented for that purpose. The adaptive algorithm gives us a homography that produces the largest number of inliers by adaptively determining the termination of the algorithm with respect to the probability of at least one of the random samples being free from outliers and that of any selected data point being an outlier.

3.4. Selection of best inliers

The set of inliers selected by RANSAC sometimes contains a large number of matches. This set is further refined by eliminating points with a large amount of the symmetric transfer error in Eq.(3) and making a set of better inlying matches. The aim of this further estimation is, therefore, to obtain an improved estimate of a homography with better inliers selected by randomly selected 100 point correspondences, instead of being selected by only randomly selected four point correspondences in RANSAC. We limit the number of point correspondences to 100 since least squares solution of more than 100 correspondences is too costly. If the set of inliers contains less than 100 matches, then this process is skipped.

The process of the further estimation is that at each iteration, a homography is estimated with a set of 100 randomly selected point correspondences that are considered to be inliers, classify a set of all correspondences based on our simple classifier in Eq.(5) and update a set of inliers. the process is repeated until the symmetric error of all the inlier becomes less than $\sqrt{5.99} * \sigma_{error}$. An important remark of this estimation process is to take an initial set of correspondences into account without eliminating any one of them, and to consider some outliers being re-designated as inliers.

4. MODEL FITTING

In order to reduce projection errors from automatic computation of a homography, model fitting is applied to the result of the homography transformation. The rink dimensions and our model are strictly based on the official measurement presented in.[32] Our model consists of features on lines and circles of the rink. There are 296 point features in total on lines and circles: 178 features on four End-Zone circles, 4 features on centre ice face-off spots around the centre circle, and 114 features on lines.

4.1. Edge search

This section describes how to fit projected images to our model of the rink and reduce projection errors produced by automatic computation of homography. In order to fit the projected images to the model, a local search is performed on each model point appearing within the region of each projected image. The local search is conducted to find the nearest edge pixel in the image. Figure 3 shows how to fit the projected image to our rink model.

For edge detection, the search is performed locally only on high gradient regions in the original sequence where there are most likely edges in order to save on the computational time. In the search on high gradient regions, edge orientation is considered to find a most likely edge pixel. Given an image, I, the image gradient vector $\mathbf{g}$ is represented as: $\mathbf{g}(\mathbf{x}) = (\frac{\partial}{\partial x}(I), \frac{\partial}{\partial y}(I))$. The gradient vector represents the orientation of the edge. The orientation is perpendicular to the direction of the local line or circle segment. Since lines and circles of the hockey rink are not single edges but thick lines, they give two peaks of gradients. The image gradient vector $\mathbf{g}$ is computed from the original image because the projected image may not give accurate gradients due to resampling effects. Figure 4 shows how the edge search is conducted.

As shown in the figure, the edge search does not perfectly detect all the edge pixels on the rink surface. For instance, in (b) of Figure 4, there is one edge pixel that does not belong to any lines in the left bottom face-off circle. Furthermore, there are not many edge points detected on the centre circle since there are many gradient peaks detected on the line of the circle, the edges of the logo, and the edges of the letters. In order to avoid finding edge points that are not on the edge of the circles or lines on the rink, our edge search ignores ambiguous regions with many edges by detecting multiple gradient peaks in the search region. Given n edge points found by our edge search, these points can be used to compute

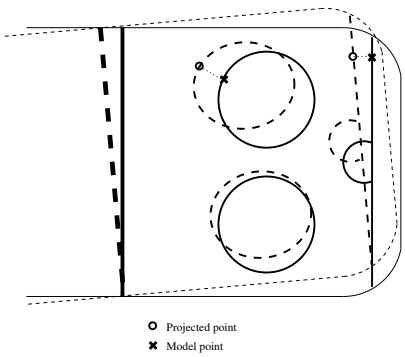
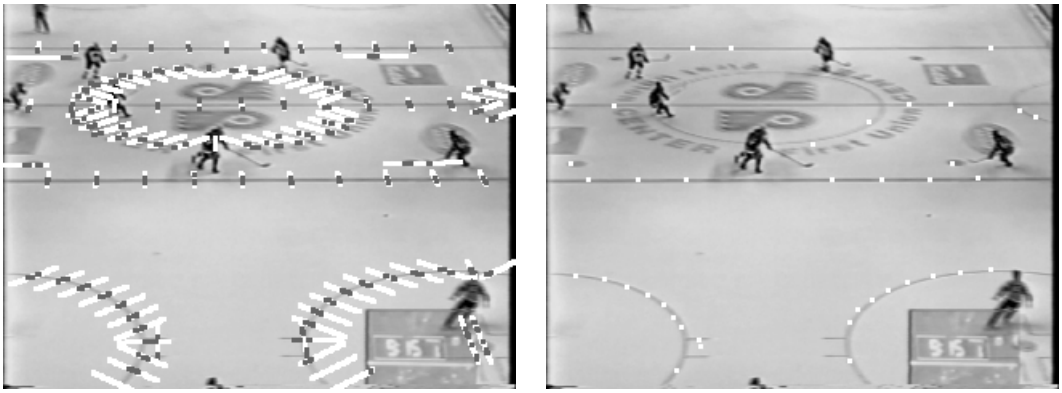


Figure 3. Fitting a projected image to our model of the rink. Dotted lines represent the projected image and solid lines represent the model. Although only two examples of matching a projected point to a model point are presented in this image, a local search is performed for finding the nearest edge pixel (i.e., a projected point) from all model points appearing within the projected image.



(a) local edge search (b) Edge points found by the search

Figure 4. Searching edges. (a) shows the search regions (lighter points) and high gradient regions (darker points). It is shown that edges lie on high gradient regions. (b) is the result of the edge search. It shows how successfully our search detects edge points for each model point.

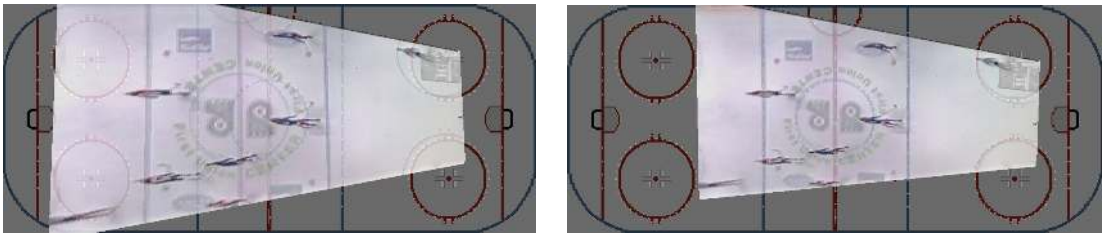
a transformation, $\mathbf{H}_{corr}$, to rectify a projected image to the model. The normalized DLT algorithm is used to compute $\mathbf{H}_{corr}$ based on 2D to 2D point correspondences $\{\mathbf{x}_i^{Edge} \leftrightarrow \mathbf{x}_i^{Model}\}_{i=1\ldots n}$ where $\{\mathbf{x}_i^{Edge}\}_{i=1\ldots n}$ denote n edge points detected by our edge search and $\{\mathbf{x}_i^{Model}\}_{i=1\ldots n}$ are n corresponding model points. Overall, our edge search gives us reliable performance and can prove that our model fitting system works well. Figure 5 shows how effective our model fitting is for reducing accumulative projection errors over a sequence of frames.

5. TRACKING HOCKEY PLAYERS

We use a particle filter method, Boosted Particle Filter (BPF),[20] based on the multiple target method of,[17] augmented by the cascaded Adaboost algorithm.[18] Adaboost provides a detection step that can re initialize tracks when they fail. When one or more new objects appear in the scene, they are detected by Adaboost and automatically initialized with an observation model. Using a different color-based observation model allows us to track different colored objects.

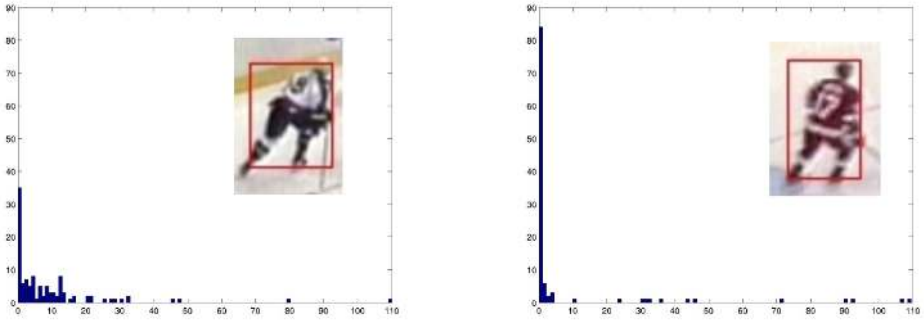
5.1. Observation Model

We adopt a multi-color observation model[14] based on Hue-Saturation-Value (HSV) color histograms. Since HSV decouples the intensity (i.e., value) from color (i.e., hue and saturation), it is reasonably insensitive to illumination effects. An HSV histogram is composed of $N = N_h N_s + N_v$ bins and we denote $b_t(\mathbf{d}) \in \{1, \ldots N\}$ as the bin index associated with the color vector $\mathbf{y}_t(\mathbf{k})$ at a pixel location $\mathbf{d}$ at time t. Figure 6 shows two instances of the color histogram.



(a) The result without fitting the model (b) The result with fitting the model

Figure 5. The result of our model fitting. (a) is the result after 323 frames without using the model fitting. (b) is the result after 323 frames with the model fitting. (b) clearly shows a more accurate projection over 300 frames.



(a) Color histogram of a player(white uniform) (b) Color histogram of a player(red uniform)

Figure 6. Color histograms: This figure shows two color histograms of selected rectangular regions, each of which is from a different region of the image. The player on left has uniform whose color is the combination of dark blue and white and the player on right has red uniform. One can clearly see concentrations of color bins due to limited number of colors. In (a) and (b), we set the number of bins, $N = 110$, where N_h, N_s, and N_v are set to 10.

If we define the candidate region in which we formulate the HSV histogram as $R(\mathbf{x}_t) \triangleq \mathbf{l}_t + s_t W$, then a kernel density estimate $\mathbf{K}(\mathbf{x}_t) \triangleq \{k(n;\mathbf{x}_t)\}_{n=1,\ldots,N}$ of the color distribution at time t is given by[1, 14]:

$$k(n;\mathbf{x}_t) = \eta \sum_{\mathbf{d}\in R(\mathbf{x}_t)} \delta[b_t(\mathbf{d}) - n] \tag{6}$$

where δ is the delta function, η is a normalizing constant which ensures k to be a probability distribution, $\sum_{n=1}^{N} k(n;\mathbf{x}_t) = 1$, and a location $\mathbf{d}$ could be any pixel location within $R(\mathbf{x}_t)$. Eq. (6) defines $k(n;\mathbf{x}_t)$ as the probability of a color bin n at time t.

If we denote $\mathbf{K}^* = \{k^*(n;\mathbf{x}_0)\}_{n=1,\ldots,N}$ as the reference color model and $\mathbf{K}(\mathbf{x}_t)$ as a candidate color model, then we need to measure the data likelihood (i.e., similarity) between $\mathbf{K}^*$ and $\mathbf{K}(\mathbf{x}_t)$. As in,[1, 14] we apply the Bhattacharyya similarity coefficient to define a distance ξ on HSV histograms. The mathematical formulation of this measure is given by[1]:

$$\xi[\mathbf{K}^*, \mathbf{K}(\mathbf{x}_t)] = \left[1 - \sum_{n=1}^{N} \sqrt{k^*(n;\mathbf{x}_0)k(n;\mathbf{x}_t)}\right]^{\frac{1}{2}} \tag{7}$$

Statistical properties of near optimality and scale invariance ensure that the Bhattacharyya coefficient is an appropriate choice of measuring similarity of color histograms.[1] Once we obtain a distance ξ on the HSV color histograms, we use the following likelihood distribution given by[14]:

$$p(\mathbf{y}_t|\mathbf{x}_t) \propto e^{-\lambda\xi^2[\mathbf{K}^*,\mathbf{K}(\mathbf{x}_t)]} \tag{8}$$

where $\lambda = 20$. λ is suggested in,[14, 17] and confirmed also on our experiments. Also, we set the size of bins N_h, N_s, and N_v as 10.

The HSV color histogram is a reliable approximation of the color density on the tracked region. However, a better approximation is obtained when we consider the spatial layout of the color distribution. If we define the tracked region as the sum of r sub-regions $R(\mathbf{x}_t) = \sum_{j=1}^{r} R_j(\mathbf{x}_t)$, then we apply the likelihood as the sum of the reference histograms $\{k_j^*\}_{j=1,\ldots,r}$ associated with each sum-region by[14]:

$$p(\mathbf{y}_t|\mathbf{x}_t) \propto e^{\sum_{j=1}^{r} -\lambda\xi^2\left[\mathbf{K}_j^*, \mathbf{K}_j(\mathbf{x}_t)\right]} \tag{9}$$

Eq. (9) shows how the spatial layout of the color is incorporated into the data likelihood. In Figure 7, we divide up the tracked regions into two sub-regions in order to use spatial information of the color in the appearance of a hockey player. For hockey players, their uniforms usually have a different color on their jacket and their pants and the spatial relationship

Figure 7. Multi-part color likelihood model: This figure shows our multi-part color likelihood model. We divide our model into two sub-regions and take a color histogram from each sub-region so that we take into account the spatial layout of colors of two sub-regions.

of different colors becomes important.

5.2. The Filtering Distribution

We denote the state vectors and observation vectors up to time t by $\mathbf{x}_{0:t} \triangleq \{\mathbf{x}_0 \ldots \mathbf{x}_t\}$ and $\mathbf{y}_{0:t}$. Given the observation and transition models, the solution to the filtering problem is given by the following Bayesian recursion[3]:

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{0:t}) &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{0:t-1})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)\int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1})d\mathbf{x}_{t-1}}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})\,d\mathbf{x}_t} \end{aligned} \tag{10}$$

To deal with multiple targets, we adopt the mixture approach of.[17] The posterior distribution, $p(\mathbf{x}_t|\mathbf{y}_{0:t})$, is modelled as an M-component non-parametric mixture model:

$$p(\mathbf{x}_t|\mathbf{y}_{0:t}) = \sum_{j=1}^{M} \Pi_{j,t} p_j(\mathbf{x}_t|\mathbf{y}_{0:t}) \tag{11}$$

where the mixture weights satisfy $\sum_{m=1}^{M} \Pi_{m,t} = 1$. Using the the filtering distribution, $p_j(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1})$, computed in the previous step, the predictive distribution becomes

$$p(\mathbf{x}_t|\mathbf{y}_{0:t-1}) = \sum_{j=1}^{M} \Pi_{j,t-1} p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1}) \tag{12}$$

where $p_j(\mathbf{x}_t|\mathbf{y}_{0:t}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p_j(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1})d\mathbf{x}_{t-1}$. Hence, the updated posterior mixture takes the form

$$
\begin{aligned}
p(\mathbf{x}_t|\mathbf{y}_{0:t}) &= \frac{\sum_{j=1}^{M} \Pi_{j,t-1} p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{\sum_{k=1}^{M} \Pi_{k,t-1} \int p_k(\mathbf{y}_t|\mathbf{x}_t) p_k(\mathbf{x}_t|\mathbf{y}_{0:t-1})\, d\mathbf{x}_t} \\
&= \sum_{j=1}^{M} \left[\frac{\Pi_{j,t-1} \int p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1})\, d\mathbf{x}_t}{\sum_{k=1}^{M} \Pi_{k,t-1} \int p_k(\mathbf{y}_t|\mathbf{x}_t) p_k(\mathbf{x}_t|\mathbf{y}_{0:t-1})\, d\mathbf{x}_t} \right] \\
&\times \left[\frac{p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{\int p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1})\, d\mathbf{x}_t} \right] \\
&= \sum_{j=1}^{M} \Pi_{j,t} p_j(\mathbf{x}_t|\mathbf{y}_{0:t})
\end{aligned}
\tag{13}
$$

where the new weights (independent of $\mathbf{x}_t$) are given by:

$$
\Pi_{j,t} = \left[\frac{\Pi_{j,t-1} \int p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1})\, d\mathbf{x}_t}{\sum_{k=1}^{M} \Pi_{k,t-1} \int p_k(\mathbf{y}_t|\mathbf{x}_t) p_k(\mathbf{x}_t|\mathbf{y}_{0:t-1})\, d\mathbf{x}_t} \right]
$$

Unlike a mixture paritcle filter by,[17] we have M different likelihood distributions, $\{p_j(\mathbf{y}_t|\mathbf{x}_t)\}_{j=1\ldots M}$. When one or more new objects appear in the scene, they are detected by Adaboost and automatically initialized with an observation model. Using a different color-based observation model allows us to track different colored objects.

5.3. Boosted Particle Filtering

In standard particle filtering, we approximate the posterior $p(\mathbf{x}_t|\mathbf{y}_{0:t})$ with a Dirac measure using a finite set of N particles $\{\mathbf{x}_t^i\}_{i=1\cdots N}$. To accomplish this, we sample candidate particles from an appropriate proposal distribution $\tilde{\mathbf{x}}_t^i \sim q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t})$ (In the simplest scenario, it is set as $q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$, yielding the bootstrap filter[3]) and weight these particles according to the following importance ratio:

$$
w_t^i = w_{t-1}^i \frac{p(\mathbf{y}_t|\tilde{\mathbf{x}}_t^i) p(\tilde{\mathbf{x}}_t^i|\mathbf{x}_{t-1}^i)}{q(\tilde{\mathbf{x}}_t^i|\mathbf{x}_{0:t-1}^i, \mathbf{y}_{0:t})} \tag{14}
$$

where, in BPF, the expression for the proposal distribution is given by the following mixture instead of the transition priors:

$$
q(\tilde{\mathbf{x}}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) = \alpha q_{ada}(\tilde{\mathbf{x}}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) + (1-\alpha) p(\tilde{\mathbf{x}}_t|\mathbf{x}_{t-1}) \tag{15}
$$

where q_{ada} is assumed to be a Gaussian distribution that consists of a set of Adaboost detections. The parameter α can be set dynamically without affecting the convergence of the particle filter (it is only a parameter of the proposal distribution and therefore its influence is corrected in the calculation of the importance weights). When $\alpha = 0$, our algorithm reduces to the MPF of.[17] By increasing α we place more importance on the Adaboost detections. We can adapt the value of α depending on tracking situations, including crossovers, collisions and occlusions.

The mixture proposal is applied only when q_{ada} is overlaid on a transition distribution modeled by autoregressive state dynamics. However, if these two different distributions are not overlapped, there is a distance between the mean of these distributions. If there is no overlap between the Monte Carlo estimation of a mixture particle filter for each mixture component and clusters given by the Adaboost detection, then we set $\alpha = 1$ so that our proposal distribution takes only a transition distribution of a mixture particle filter.

We resample the particles using their importance weights to generate an unweighted approximation of $p(\mathbf{x}_t|\mathbf{y}_{0:t})$. In the mixture approach of,[17] the particles are used to obtain the following approximation of the posterior distribution:

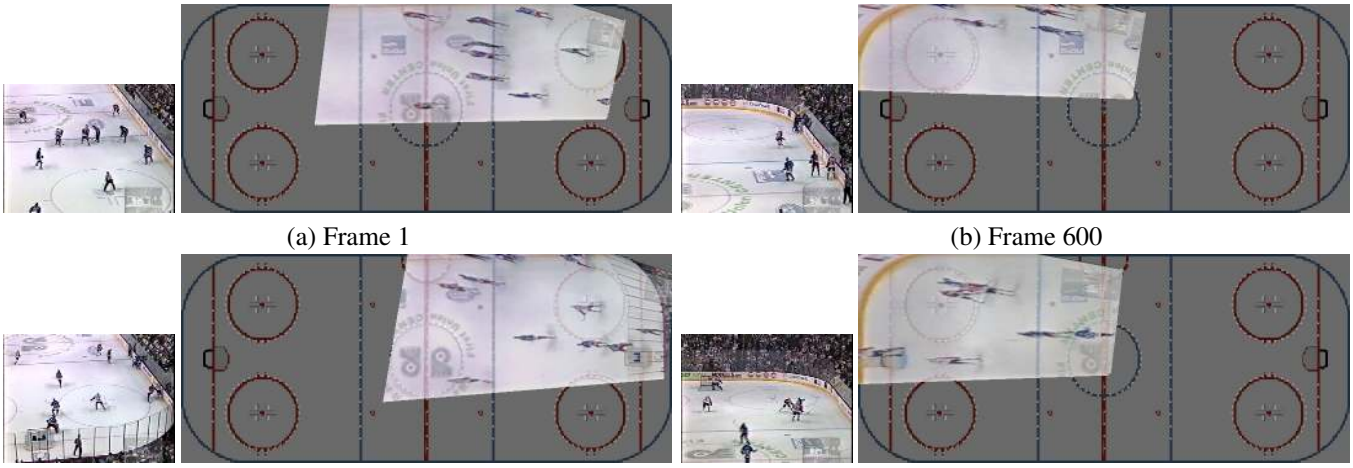
$$
p(\mathbf{x}_t|\mathbf{y}_{0:t}) \approx \sum_{j=1}^{M} \Pi_{j,t} \sum_{i \in \mathcal{I}_j} w_t^i \delta_{\mathbf{x}_t^i}(\mathbf{x}_t)
$$

where $\mathcal{I}_j$ is the set of indices of the particles belonging to the j-th mixture component.

6. EXPERIMENTS

This section presents the result of our experiments on the two aspects of our work: homography estimation and player tracking.

6.1. Homography Experiments



(a) Frame 1 (b) Frame 600

(c) Frame 1200 (d) Frame 1850

Figure 8. Automatic rectification result. The figure shows the result of our algorithm on over 1800 frames on hockey data. The left column shows the original image (320×240) to be transformed and on the right, it shows a rectified image that is superimposed on the model of the rink map.

In Figure 8, our system is demonstrated on a sequence of 1900 frames that is digitized from a video clip of NHL hockey games on TV. The system processes every fourth frame and rectifies them by computing 1200 KLT features from which the best inliers are selected. Once a set of correspondences are manually selected only on the very first frame of the sequence to compute the transformation between the image and rink mapping, homographies between the rest of the sequence and rink mapping are automatically computed by our algorithm. Our non-optimized implementation in C on a 2.8 GHz Pentium IV takes about an hour to process 1900 frames of data. Figure 8 shows the successful automatic rectification. Although our system is demonstrated on Hockey data at this time, our algorithm is also applicable to other domains of sports such as soccer and football or any other planar surface scenes with identifiable features.

6.2. Tracking Experiments

This section shows tracking results on hockey players in a digitized video sequence from TV. In our experments, we used digitized data of video sequences and all experiments are done by our non-optimized implementation in C on a 2.8 GHz Pentium IV.

Figure 9 shows the result of BPF over 95 frames of a digitized video sequence. As it is clear on the figure, BPF is capable of tracking a different number of objects: i.e., it has an ability of successfully detecting objects that appear or disappear from the scene. Figure 10 is the result of combining homography estimation in Figure 8 and tracking result from Figure 9. BPF tracking gives players' positions, and homographies can transform their positions to the rink map.

7. CONCLUSIONS

This paper describes an automatic system of computing homographies over a long image sequence and rectifying the sequence by compensating for the panning, tilting and zooming of the cameras. Since our model-based correction system performs a local search of both straight and circular line segments and distinguishes them by their orientation, it does not require direct methods of conic detection or line detection. It achieves robustness by combining a number of different methods that would not be sufficient on their own.

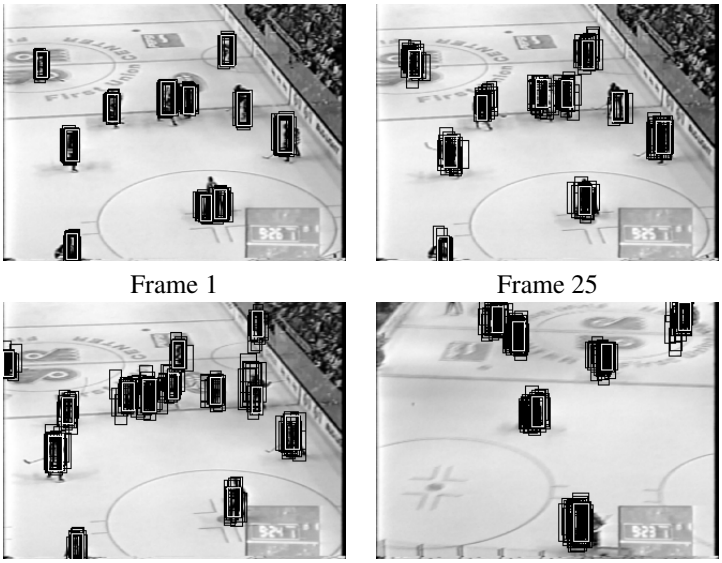


Figure 9. BPF tracking result: The results of BPF tracking are shown over 80 frames of a digitized video sequence.

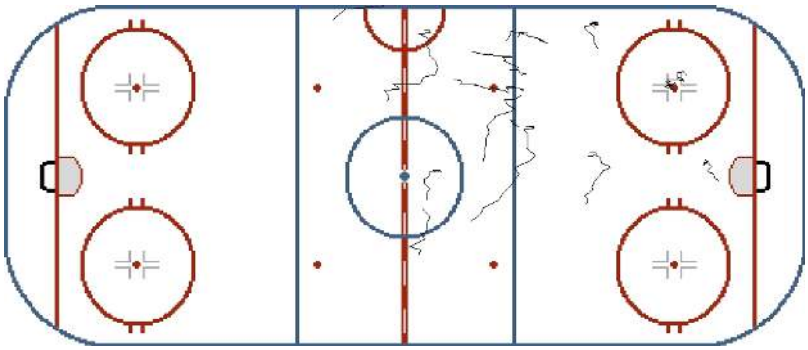


Figure 10. Combined trajectories of several hockey players in Figure 9

Our system is easily applicable to different scenes such as soccer, football, or many other scenes that have a planer surfaces with identifiable features and line segments.

We have also devised an approach to combining the strengths of Adaboost for object detection with those of mixtures of particle filters for multiple-object tracking. We have experimented with this boosted particle filter in the context of tracking hockey players in video from broadcast television. The results show that most players are successfully detected and tracked, even as players move in and out of the scene.

Trajectories of hockey players being tracked are automatically generated by using estimated homographies and transforming positions of hockey players to the globally consistent rink map. Such visual information of scenes are then inserted into a database system that permits querying and analysis of the motion trajectories. Our automatic system of generating visual information of hockey scenes introduced in this work is therefore a critical part of our automatic hockey annotatin system that unltimately analyzes scenes of hockey games, recoginizes and predicts motions of hockey players.

REFERENCES

1. Comaniciu, D., Ramesh, V., Meer, P.: Real-Time Tracking of Non-Rigid Objects using Mean Shift. IEEE Conference on Computer Vision and Pattern Recognition, pp. 142-151 (2000)
2. Deutscher, J., Blake, A., Ried, I.: Articulated body motion capture by annealed particle filtering. IEEE Conference on Computer Vision and Pattern Recognition, (2000)

3. Doucet, A., de Freitas, J. F. G., N. J. Gordon, editors: Sequential Monte Carlo Methods in Practice. Springer-Verlag, New York (2001)
4. Freund, Y., Schapire, R. E.: A decision-theoretic generalization of on-line learning and an application to boosting. Computational Learning Theory, pp. 23-37, Springer-Verlag, (1995)
5. Hue, C., Le Cadre, J.-P., Pérez, P.: Tracking Multiple Objects with Particle Filtering. IEEE Transactions on Aerospace and Electronic Systems, 38(3):791-812 (2002)
6. Intille, S. S., Davis, J. W., Bobick, A.F.: Real-Time Closed-World Tracking. IEEE Conference on Computer Vision and Pattern Recognition, pp. 697-703 (1997)
7. Isard, M., MacCormick, J.: BraMBLe: A Bayesian multiple-blob tracker. International Conference on Computer Vision, pp. 34-41(2001)
8. Isard, M., Blake, A.: Condensation – conditional density propagation for visual tracking. International Journal on Computer Vision, 28(1):5-28 (1998)
9. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems Transactions of the ASME–Journal of Basic Engineering, vol.82 Series D pp.35-45 (1960)
10. Koller, D., Weber, J., Malik, J.: Robust Multiple Car Tracking with Occlusion Reasoning. European Conference on Computer Vision, pp. 186-196, LNCS 800, Springer-Verlag (1994)
11. MacCormick, J., Blake, A.: A probabilistic exclusion principle for tracking multiple objects. International Conference on Computer Vision, pp. 572-578 (1999)
12. Misu, T., Naemura, M., Wentao Zheng, Izumi, Y., Fukui, K.: Robust Tracking of Soccer Players Based on Data Fusion IEEE 16th International Conference on Pattern Recognition, pp. 556-561 vol.1 (2002)
13. Needham, C. J., Boyle, R. D.: Tracking multiple sports players through occlusion, congestion and scale. British Machine Vision Conference, vol. 1, pp. 93-102 BMVA (2001)
14. Pérez, P., Hue. C, Vermaak, J., Gangnet, M.: Color-Based Probabilistic Tracking. European Conference on Computer Vision, (2002)
15. Rui, Y., Chen, Y.: Better Proposal Distributions: Object Tracking Using Unscented Particle Filter. IEEE Conference on Computer Vision and Pattern Recognition, pp. 786-793 (2001)
16. van der Merwe, R., Doucet, A., de Freitas, J. F. G., Wan, E: The Unscented Particle Filter. Advances in Neural Information Processing Systems, vol. 8 pp 351-357 (2000)
17. Vermaak, J., Doucet, A., Pérez, P.: Maintaining Multi-Modality through Mixture Tracking. International Conference on Computer Vision (2003)
18. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE Conference on Computer Vision and Pattern Recognition (2001)
19. Okuma, K., Little, J., Lowe, D., Automatic rectification of long image sequences, submitted to the Asian Conference on Computer Vision, 2004.
20. Okuma, K., Taleg, A., de Freitas, N., Little, J., Lowe, D., A Boosted Particle Filter: Multitarget Detection and Tracking, submitted to the European Conference on Computer Vision, 2004.
21. Richard Hartley and Andrew Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, June 2000.
22. M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu, "Efficient representations of video sequences and their applications," 1996.
23. K. Kanatani and N. Ohta, "Accuracy bounds and optimal computation of homography for image mosaicing applications," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV-99)*, Los Alamitos, CA, Sept. 20–27 1999, vol. I, pp. 73–79, IEEE.
24. R. Szeliski, "Image mosaicing for tele-reality applications," in *WACV94*, 1994, pp. 44–53.
25. I. Zoghiami, O.D. Faugeras, and R. Deriche, "Using geometric corners to build a 2d mosaic from a set of images," in *CVPR97*, 1997, pp. 420–425.
26. Stan Birchfield, *Depth and motion discontinuities*, Ph.D. thesis, Stanford University, 1999.
27. Jianbo Shi and Carlo Tomasi, "Good features to track," Technical Report TR93-1399, Cornell University, Computer Science Department, Nov. 1993.
28. Carlo Tomasi and Takeo Kanade, "Detection and tracking of point features," Technical Report CMU-CS-91-132, Carnegie Mellon University, Computer Science Department, 1991.
29. Martin A. Fischler and Robert C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.
30. D. Koller, K. Daniilidis, and H.H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *IJCV*, vol. 10, no. 3, pp. 257–281, June 1993.
31. A. Yamada, Y. Shirai, and J. Miura, "Tracking players and a ball in video image sequence and estimating camera parameters for 3D interpretation of soccer games," in *ICPR02 VOL I*. IEEE, 2002, pp. 303–306.
32. USA Hockey Inc., "The official rules of ice hockey," 2001.
33. R. Yang, "Multi-Scale Summaries of Temporal Trajectories," Univ. of British Columbia, MSc thesis, 2003.
34. M. Dimitrijevic, "Mining for co-occurring motion trajectories: sport analysis," Univ. of British Columbia, MSc thesis, 2002.