Robust Tracking of Athletes Using Multiple Features of Multiple Views

Toshihiko Misu Seiichi Gohshi Yoshinori Izumi Yoshihiro Fujita Masahide Naemura*

NHK (Japan Broadcasting Corporation)

1-10-11 Kinuta, Setagaya, Tokyo 157-8510, Japan

{misu.t-ey, gohshi.s-fu, izumi.y-kk, fujita.y-ic}@nhk.or.jp, naemura@atr.co.jp

ABSTRACT

This paper presents a robust and reconfigurable object tracker that integrates multiple visual features from multiple views. The tandem modular architecture stepwise refines the estimate of trajectories of the objects in the world coordinates using many plug-ins that observe various features such as texture, color, region and motion in 2D images acquired by the cameras. One of the most important features of our proposed method is that each plug-in innovates the trajectories not only by back-projecting 2D observations of the features, but also by weighting them adaptively to their self-evaluated reliability. In the paper, the architecture of the system and that of the plug-ins are formulated. The behavior and robustness against occlusion are also shown through experiments with football-game sequences.

Keywords

tracking, data fusion, multiocular measurement, sports

1. INTRODUCTION

Automated tracking of moving objects have been a key technology in various fields including video surveillance, scene analysis, metadata production, etc. In any cases above, the robust strategies, which are intended to overcome problems of occlusion, deformation, noise and/or illumination changes, are intensely studied [Jan00a][Mey94a][Isa98a].

Based on decomposed Kalman filtering, we developed a tandem tracker that gradually refines the estimated trajectories on an image plane by a series of tracking plug-ins with various measurement strategies [Mis02a]. The algorithm has following two major advantages that are required to multi-purpose object trackers: (1) heterogeneous measurements are integrated adaptively to their self-evaluated reliability to reinforce the robustness, and (2) the observation strategies can easily be assembled/reordered by plugging-in/-out the modules.

Journal of WSCG, Vol.12, No.1-3, ISSN 1213-6972

WSCG'2004, February 2-6, 2004, Plzen, Czech Republic.

The algorithm, however, was designed to track objects on the image plane using monocular sensory system, and was not capable of detecting 3D world coordinates.

To integrate spatially diverse observations, we extended the algorithm to estimate the trajectories in a world coordinate system. The Observations from the cameras are gathered together through the parameters of position, attitude, and focal length. A variety of silhouette extraction, matching, and position prediction are provided as plug-ins, which expand the multimodality of the platform. In this paper, the system architecture and constituent plug-ins are illustrated and formulated. Experimental results with football sequences show the robustness of the algorithm.

2. ARCHITECTURE OF TRACKER

The tandem architecture of our proposed tracker, as illustrated in Fig. 1, stepwise updates the estimates of positions. Each the step detects objects by one specific strategy, and is in charge of one specific viewpoint. It also automatically updates the tracking templates if necessary. The boxes in Fig. 1 are implemented as software plug-ins (dynamic link libraries), which can be classified into three major categories: silhouette extraction (EXTR), template

* He is currently working for Advanced Telecommunication Research Institute International, Japan.

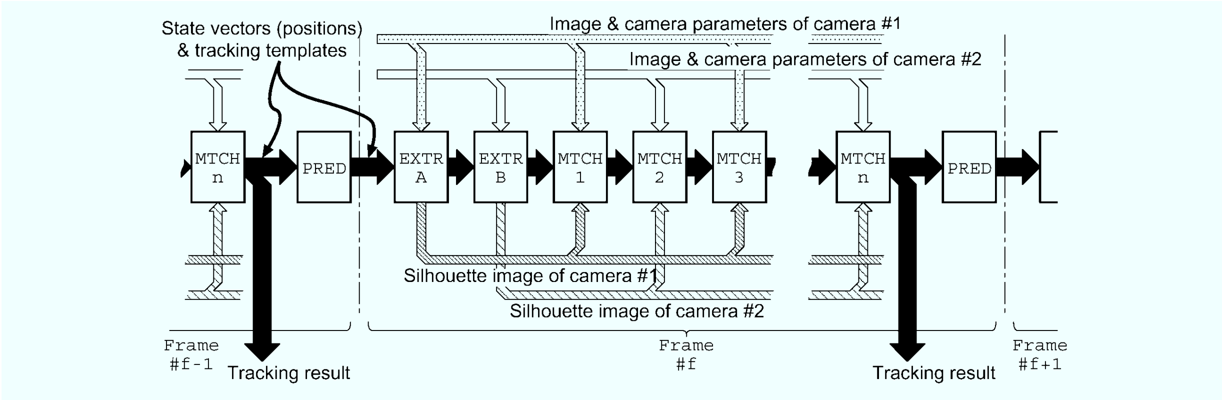


Figure 1. Architecture of Tracker

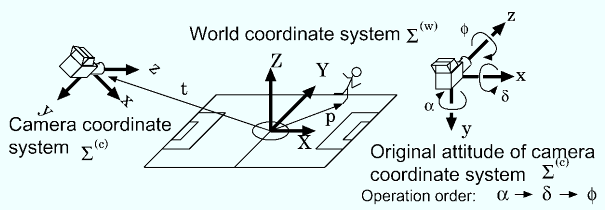


Figure 2. Coordinate System

matching (MTCH), and position prediction (PRED). An EXTR plug-in assigns 1/0 flags to each pixel to get a silhouette image by judging whether the pixel belongs to an object region or not. An MTCH plug-in, which matches observed visual features with templates, searches the input image for the target objects to update their estimated positions. A PRED plug-in predicts the positions of the target objects based on a dynamics model from those of the previous time frame.

Parameterization of Track (State Vector)

The state of each target object i is parameterized by its position p_i, velocity $\dot{p}_i$, and acceleration $\ddot{p}_i$ in a view-independent world coordinate system $\Sigma^{(w)}$ as shown in Fig. 2. We define the following 9-dimensional state vector $\mathbf{x}_i$ for each object i:

$$\mathbf{x}_i = \begin{bmatrix} p_i^T & \dot{p}_i^T & \ddot{p}_i^T \end{bmatrix}^T, \tag{1}$$

where superscript T denotes the transpose of the matrix.

Camera Model (Observation)

Our proposed system has one or more camera(s) to observe projected image coordinates of the target position p_i. Each camera coordinate system $\Sigma^{(c)}$ is modeled with the position vector $\mathbf{t} = \begin{bmatrix} t_X & t_Y & t_Z \end{bmatrix}^T$ and the attitude angles (pan α, tilt δ, and roll ϕ) in $\Sigma^{(w)}$. In this paper, the position vector, the attitude angles, plus the focal length f are referred to as "camera parameters" (see Fig. 2 for the definitions of and its original attitude). We assume that the camera parameters are calibrated/measured by an image-based algorithm [Tsa87a] or by a tripod with rotary encoders.

The pinhole model with the above-mentioned camera parameters yields the following perspective mapping function $\mathbf{h}(\mathbf{x}_i)$ that gives the ideal image coordinates $\hat{\mathbf{y}}_i$ of the object i:

$$\hat{\mathbf{y}}_i = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \end{bmatrix} \ell \Big/ \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \ell \overset{\Delta}{=} \mathbf{h}(\mathbf{x}_i) \tag{2}$$

$$\ell = \begin{bmatrix} s\alpha s\delta s\phi + c\alpha c\phi & c\alpha s\delta s\phi - s\alpha c\phi & -c\delta s\phi \\ s\alpha s\delta c\phi - c\alpha s\phi & c\alpha s\delta c\phi + s\alpha s\phi & -c\delta c\phi \\ s\alpha c\delta & c\alpha c\delta & s\delta \end{bmatrix} \cdot (\mathbf{p}_i - \mathbf{t}), \tag{3}$$

where $\sin\theta$ and $\cos\theta$ are abbreviated to $s\theta$ and $c\theta$, respectively.

We modeled the error of each MTCH as a zero-mean white Gaussian noise w with a covariance of R, which leads to:

$$\mathbf{y}_i = \mathbf{h}(\mathbf{x}_i) + \mathbf{w} \tag{4}$$

$$E[\mathbf{w}] = \mathbf{0}, \quad E[\mathbf{w}\mathbf{w}^T] = R, \tag{5}$$

where $\mathbf{y}_i$ is the observation of image coordinates of object i.

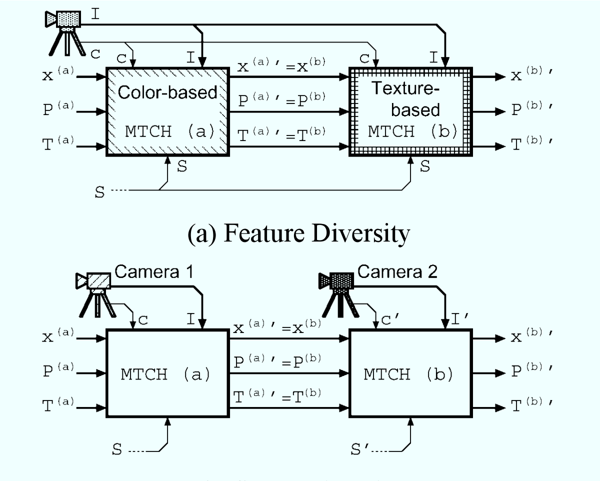
Model of Dynamics

We employed the following simple transition as a dynamics model of each state vector $\mathbf{x}_i$:

$$\mathbf{x}_i' = F\mathbf{x}_i + \mathbf{v}, \tag{6}$$

where $\mathbf{x}_i$, $\mathbf{x}_i'$, F and $\mathbf{v}$ are the current state vector, that of the next time frame, transition matrix, and the process noise, respectively. We assume that $\mathbf{v}$ is a zero-mean white Gaussian noise with covariance Q:

$$E[\mathbf{v}] = \mathbf{0}, \quad E[\mathbf{v}\mathbf{v}^T] = Q. \tag{7}$$



(a) Feature Diversity

(b) Space Diversity

Figure 3. Feature and Space Diversity

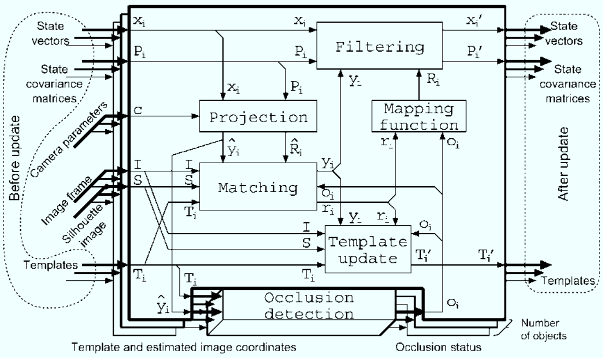


Figure 4. Structure of Matching Plug-in

The matrix F will be defined in Section 4.

Feature Diversity and Space Diversity

The most important feature of our proposed architecture is that the cascade of plug-ins (with different extraction/matching/prediction strategies) serially refines the estimates of target positions integrating heterogeneous visual features (feature diversity) and/or those from different viewpoints (space diversity).

In case different MTCH plug-ins are connected (feature diversity configuration), as shown in Fig. 3a, the plug-ins update the state vectors (also with templates) to incorporate features/strategies.

The series of MTCH plug-ins that receives images from different cameras, as shown in Fig. 3b, unifies the observation from the multiple viewpoints into the state vectors in a single world coordinate system through the camera parameters. Not only does the space-diversity configuration resolve the occlusion problem, but the moderate interaction of the plug-ins with the state vectors implicitly performs triangulation to compensate the inherent ambiguity along the line-of-sight of each view.

3. MATCHING PLUG-INS

All matching plug-ins (MTCHs) have the same basic structure as shown in Fig. 4. The variety of implementation of constituent blocks enables their polymorphism.

The plural layers in the figure update their own targets taking others' positions (i.e., occlusion status) into account. Firstly, the input state vector $\mathbf{x}_i$ is projected onto the image plane using Equation (2) to obtain estimated image coordinates $\hat{\mathbf{y}}_i$ of the target i. The state covariance matrix P_i — a measure of the error on $\mathbf{x}_i$ (see Equations (15) and (25)) — is also mapped onto the image plane to get $\hat{R}_i$:

$$\hat{R}_i = H(\mathbf{x}_i) P_i H(\mathbf{x}_i)^T \tag{8}$$

$$\text{where, } H(\mathbf{x}) = \partial \mathbf{h}(\mathbf{x}) / \partial \mathbf{x}. \tag{9}$$

As $\hat{\mathbf{y}}_i$ and $\hat{R}_i$ can be interpreted as an estimated center and radii of the error ellipsoid around the target's image, we can reduce the search area $\mathbf{A}_i$ to be a disk of radius ρ in Mahalanobis metric [Mcl97a]:

$$\mathbf{A}_i = \{\mathbf{y} \mid \mathrm{dist}_{R_i}(\mathbf{y}, \hat{\mathbf{y}}_i) \le \rho\} \tag{10}$$

$$dist_R(\mathbf{x}, \boldsymbol{\mu}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})}, \tag{11}$$

where $\mathrm{dist}_\Sigma(\mathbf{x}, \boldsymbol{\mu})$ denotes the Mahalanobis distance of input vector $\mathbf{x}$ from the distribution of mean $\boldsymbol{\mu}$ and covariance Σ.

Secondly, the "matching" step searches the input image $\mathbf{I}$ within a search area $\mathbf{A}_i$ for a similar region to the template(s) in $\mathbf{T}_i$. The matching step outputs the image coordinates $\mathbf{y}_i$ and their reliability $\mathbf{r}_i$. In case of MSE-based block matching, for example, the minimum MSE can be a criterion for the reliability $\mathbf{r}_i$.

In order to determine the observation covariance matrix R_i (supposed to be diagonal in this paper), we introduce empirically designed look-up functions $\tau_x(\mathbf{r}_i, \mathbf{o}_i)$ and $\tau_y(\mathbf{r}_i, \mathbf{o}_i)$ that map the reliability $\mathbf{r}_i$ and the occlusion status $\mathbf{o}_i$ to the diagonal components of R_i (see subsequent subsections for the individual definitions):

$$R_i = \mathrm{diag}\{\tau_x(\mathbf{r}_i, \mathbf{o}_i), \tau_y(\mathbf{r}_i, \mathbf{o}_i)\}. \tag{12}$$

The functions return large values when the reliability $\mathbf{r}_i$ lowers or the occlusion status $\mathbf{o}_i$ represents overlap with other similar objects.

As examples of the occlusion status $\mathbf{o}_i$, we define the following two criteria $\mathbf{o}_i^{(d)}$ and $\mathbf{o}_i^{(a)}$ which are calculated from the arrangement of the projected objects' bounding regions $\mathbf{B}_i$ s (see Figs. 5 and 6):

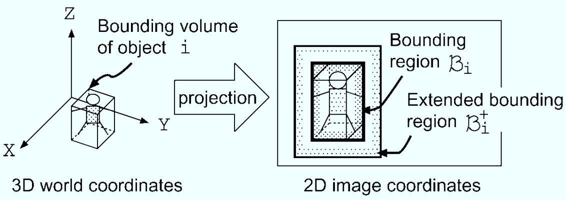
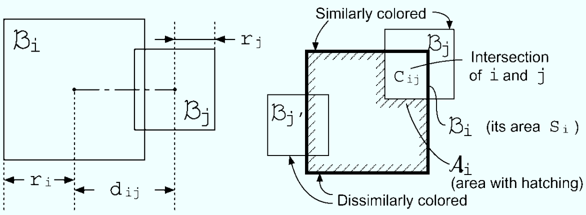


Figure 5. Bounding Regions $\mathbf{B}_i$ and $\mathbf{B}_i^+$



(a) Distance-based $o_i^{(d)}$ (b) Area-based $o_i^{(a)}$

Figure 6. Criteria of Occlusion Status $\mathbf{o}_i$

$$\mathbf{o}_i = \begin{bmatrix} o_i^{(d)} \\ o_i^{(a)} \end{bmatrix} = \begin{bmatrix} \min_j\left((d_{ij} + r_i - r_j)/2r_i\right) \\ 1 - \sum_{j \in \{n \mid \mathrm{dist}_{\Sigma_i}(\boldsymbol{\mu}_n, \boldsymbol{\mu}_i) \le \theta_a\}} (C_{ij}/S_i) \end{bmatrix}, \quad (13)$$

where S_i and C_{ij} are the areas of $\mathbf{B}_i$ and $\mathbf{B}_i \cap \mathbf{B}_i$, respectively. Note that the Mahalanobis distance $\mathrm{dist}_{\Sigma_i}(\boldsymbol{\mu}_n, \boldsymbol{\mu}_i)$ measures the color-distance between objects n and i. The values of distance-based criterion $o_i^{(d)}$ can be categorized into: (1) totally occluded ($o_i^{(d)} \le 0$), (2) partially occluded ($0 < o_i^{(d)} \le 1$), and (3) isolated ($o_i^{(d)} > 1$). On the other hand, area-based criterion $o_i^{(a)}$ becomes smaller than 1 when the object i is occluded by other similarly colored object(s).

Lastly, the "filtering" step refines the state vector $\mathbf{x}_i$ and its covariance P_i based on the observed image coordinates $\mathbf{y}_i$ and the covariance R_i, which is determined by Equation (12), using the following formulae:

$$\mathbf{x}_i' = \mathbf{x}_i + K_i(\mathbf{y}_i - \mathbf{h}(\mathbf{x}_i)) \quad (14)$$

$$P_i' = P_i - K_i H(\mathbf{x}_i) P_i \quad (15)$$

$$K_i = P_i H(\mathbf{x}_i)[H(\mathbf{x}_i) P_i H(\mathbf{x}_i)^T + R_i]^{-1}, \quad (16)$$

where K_i is noted as "Kalman gain" [Kal60a].

Color Matching Plug-in

The color matching plug-in has statistic parameters of each object's color as a member of template set $\mathbf{T}_i$. Let $\boldsymbol{\mu}_i$ and Σ_i be the mean and the covariance matrix of the color vector of the target object, and $\overline{\boldsymbol{\mu}}_i$ and $\overline{\Sigma}_i$ be those of region near (but outside) the object. The plug-in observes the spatial center of gravity of similarity $s(\mathbf{y})$ as follows:

$$\mathbf{y}_i = \sum_{\mathbf{y} \in \mathbf{A}_i} \mathbf{y} \cdot s(\mathbf{y}) \Big/ \sum_{\mathbf{y} \in \mathbf{A}_i} s(\mathbf{y}) \quad (17)$$

$$s(\mathbf{y}) = \exp(-\rho_{\mathrm{fg}} \mathrm{dist}_{\Sigma_i}(\mathbf{I}(\mathbf{y}), \boldsymbol{\mu}_i)) \cdot \{1 - \exp(-\rho_{\mathrm{bg}} \mathrm{dist}_{\overline{\Sigma}_i}(\mathbf{I}(\mathbf{y}), \overline{\boldsymbol{\mu}}_i))\}, \quad (18)$$

where ρ_{fg} and ρ_{bg} are empirically predefined constants. The similarity $s(\mathbf{y})$ increases when the pixel's color is similar to $\boldsymbol{\mu}_i$ but dissimilar to $\overline{\boldsymbol{\mu}}_i$. As shown by hatching in Fig. 6b, the search area $\mathbf{A}_i$ is determined basically by a dilated area $\mathbf{B}_i^+$ of the bounding-region $\mathbf{B}_i$, but the region interfered by other similarly colored object(s) is eliminated from the search area $\mathbf{A}_i$.

To determine the observation covariance matrix R_i, the following τ_x and τ_y are used:

$$\tau_x(\mathbf{r}_i, \mathbf{o}_i) = \tau_y(\mathbf{r}_i, \mathbf{o}_i) = \begin{cases} 10^{-9} & (o_i^{(a)} = 1) \\ \infty & (\text{otherwise}) \end{cases}, \quad (19)$$

which mean that the state vector $\mathbf{x}_i$ is updated if and only if no similarly colored objects are overlapping.

Texture Matching Plug-in

As a template, the plug-in utilizes an arbitrary-shaped colored image $\mathbf{E}_i(\boldsymbol{\eta}) \in \mathbf{T}_i$ $(\boldsymbol{\eta} \in \mathbf{B}_i)$ of the object observed from each viewpoint. Updating the template automatically, the plug-in detects the target based on the following block-matching algorithm:

$$\mathbf{y}_i = \arg\min_{\mathbf{y} \in \mathbf{A}_i} \varepsilon_i(\mathbf{y}) \quad (20)$$

$$\varepsilon_i(\mathbf{y}) = \sqrt{\sum_{\boldsymbol{\eta} \in \mathbf{B}_i} \left\| \mathbf{I}(\mathbf{y} + \boldsymbol{\eta}) - \mathbf{E}_i(\boldsymbol{\eta}) \right\|^2 \Big/ \sum_{\boldsymbol{\eta} \in \mathbf{B}_i} 3}, \quad (21)$$

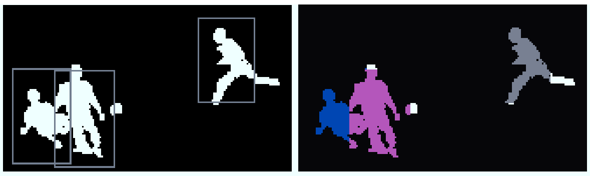
where $\varepsilon_i(\mathbf{y})$ is the RMS error between the template $\mathbf{E}_i$ centered at $\mathbf{y}$ and the input image $\mathbf{I}$. The observation covariance $\mathbf{R}_i$ is controlled based on the following equations:

$$\tau_x(\mathbf{r}_i, \mathbf{o}_i) = \tau_y(\mathbf{r}_i, \mathbf{o}_i) = \begin{cases} 10^{-8} & (r_i < 0.1) \\ 10^{-7} & (0.1 \le r_i < 0.3) \\ \infty & (\text{otherwise}) \end{cases} \quad (22)$$

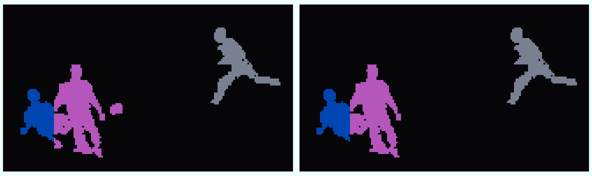
$$r_i = \min_{\mathbf{y} \in \mathbf{A}_i} \varepsilon_i(\mathbf{y}). \quad (23)$$

Local Feature Matching Plug-in

Using a plural number of 5×5-pixel templates around visually distinct points (with local maxima of variance of the texture inside $\mathbf{B}_i$), block matching (same as the texture matching plug-in) is performed. All the templates are updated every frame. Both $\tau_x(\mathbf{r}_i, \mathbf{o}_i)$ and $\tau_y(\mathbf{r}_i, \mathbf{o}_i)$ constantly return 10^{-7}'s.



(a) Silhouette Image and $\mathbf{B}_i$ s (b) After Filling $\mathbf{B}_i$ s



(c) After Region Gowing (d) After Noise Reduction

Figure 7. Silhouette Matching

Silhouette Matching Plug-in

First, the plug-in labels the silhouette image $\mathbf{S}(\mathbf{y})$ (Fig. 7a) based on the objects' bounding regions $\mathbf{B}_i$ s and on their depths to get a labeled image (Fig. 7b). The undetermined pixels without any labels (white pixels in Fig. 7b) are region-grown from neighboring determined pixels resulting in Fig. 7c. Then, an area-filter eliminates small blotches to get a refined labeled image as shown in Fig. 7d.

Based on the center $\mathbf{y}_i$ of re-calculated bounding region of each label in Fig. 7d, the state vector $\mathbf{x}_i$ is updated using Equation (14). The following are used to determine observation covariance R_i :

$$\begin{bmatrix} \tau_x(\mathbf{r}_i, \mathbf{o}_i) \\ \tau_y(\mathbf{r}_i, \mathbf{o}_i) \end{bmatrix} = \begin{cases} \begin{bmatrix} 10^{-10} & 10^{-10} \end{bmatrix}^T & (o_i^{(a)} = 1) \\ \begin{bmatrix} \max\{W^2, 10^{-8}\} \\ \max\{H^2, 10^{-8}\} \end{bmatrix} & \text{(otherwise)} \end{cases}, \quad (24)$$

where W and H are the width and the height of the region labeled with i .

4. PREDICTION PLUG-INS

A prediction plug-in (PRED) estimates a state vector $\mathbf{x}'_i$ and its covariance P'_i at the next frame from he current state $\mathbf{x}_i$ and the covariance P_i assuming a model of dynamics such as constant acceleration model.

Based on Kalman filtering technique, the following prediction equations are obtained:

$$\mathbf{x}'_i = F\mathbf{x}_i, \qquad P'_i = FP_iF^T + Q. \qquad (25)$$

The process covariance matrix Q , which has been modeled in Equations (6) and (7), should be given in order to reflect modeling ambiguities and/or disturbance.

As an example of PRED plug-ins, we implemented the Singer model, in which the models of constant acceleration and of constant velocity are smoothly unified through a parameter of smoothness λ :

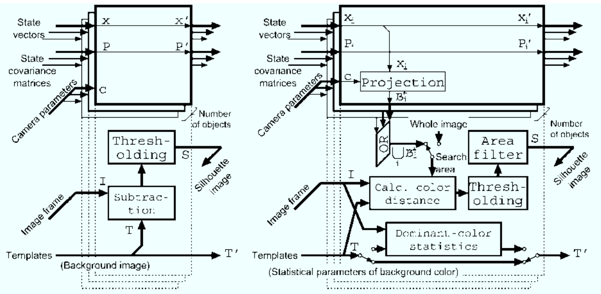
$$d\ddot{\mathbf{p}}_i(t)/dt = -\lambda\ddot{\mathbf{p}}_i(t) + [0,0,0,0,0,0,u(t)^T]^T, \quad (26)$$

which leads to:

$$F = \begin{bmatrix} I_{3\times3} & I_{3\times3} & \frac{\lambda-1+e^{-\lambda}}{\lambda^2} I_{3\times3} \\ O_{3\times3} & I_{3\times3} & \frac{1-e^{-\lambda}}{\lambda} I_{3\times3} \\ O_{3\times3} & O_{3\times3} & e^{-\lambda} I_{3\times3} \end{bmatrix} \qquad (27)$$

$$Q = \begin{bmatrix} O_{6\times6} & O_{6\times3} \\ O_{3\times6} & E[\mathbf{u}\mathbf{u}^T]\Delta t \end{bmatrix}, \qquad (28)$$

where $\mathbf{u}$ and Δt are the 3D white Gaussian noise on acceleration and the time interval between two successive frames, respectively.



(a) Background Subtraction Plug-in (b) Chroma-key Plug-in

Figure 8. Structure of Extraction Plug-ins

(a) Original Image $\mathbf{I}(\mathbf{y})$ (b) Silhouette Image $\mathbf{S}(\mathbf{y})$ (Extracted by Chroma-key Plug-in)

Figure 9. Structure of Extraction Plug-ins

5. EXTRACTION PLUG-INS

An extraction plug-in (EXTR), examples of whose internal structure are illustrated in Fig. 8, extracts silhouettes of objects as shown in Fig. 9. We developed two EXTRs: background subtraction and chroma-key. The EXTRs are not mandatory, but will give a powerful constraint on object positions or a mask against outliers.

As shown Fig. 8a, the background subtraction plug-in refers a pre-calculated background image as a template to judge whether each pixel in the input image $\mathbf{I}$ belongs to the athletes' region or not. This strategy is applicable to the images acquired by fixed cameras.

(a) Frame 0 (b) Frame 17 (c) Frame 29

Figure 10. Tracked Athletes Superimposed on Camera Images (Scene 1)

Step	Type	Strategy
1	EXTR	Chroma-key
2	MTCH	Color Matching
3	MTCH	Texture Matching
4	MTCH	Local Feature Matching
5	MTCH	Silhouette Matching
6	PRED	Prediction by Singer Model

Table 1. Processing Order (Scene 1)

Step	Type	Strategy	
1-3	EXTR	Background Subtraction	(Cams. 1-3)
4-6	MTCH	Color Matching	(Cams. 1-3)
7-9	MTCH	Texture Matching	(Cams. 1-3)
10-12	MTCH	Local Feature Matching	(Cams. 1-3)
13-15	MTCH	Silhouette Matching	(Cams. 1-3)
16	PRED	Prediction by Singer Model	

Table 2. Processing Order (Scene 2)

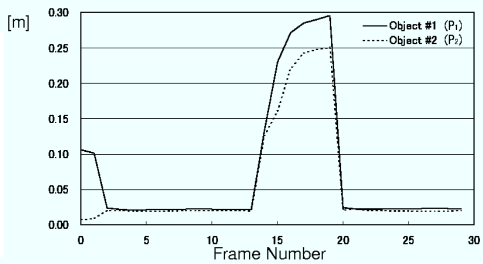


Figure 11. State Covariance of Scene 1

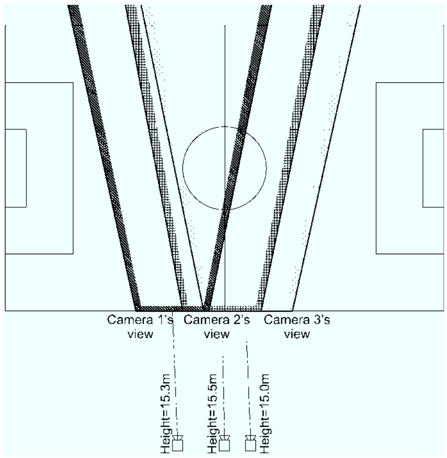


Figure 12. Camera Positions (Scene 2)

For the scenes with uniformly colored background (e.g., turf of football field), color-based algorithms such as [Nae00a] would be suitable for finding objects. Fig. 8b is the block-diagram of our designed chroma-key plug-in based on the Mahalanobis distance from the background color statistics: the mean color and the covariance (which is assumed to be a measure of granularity).

6. EXPERIMENTS

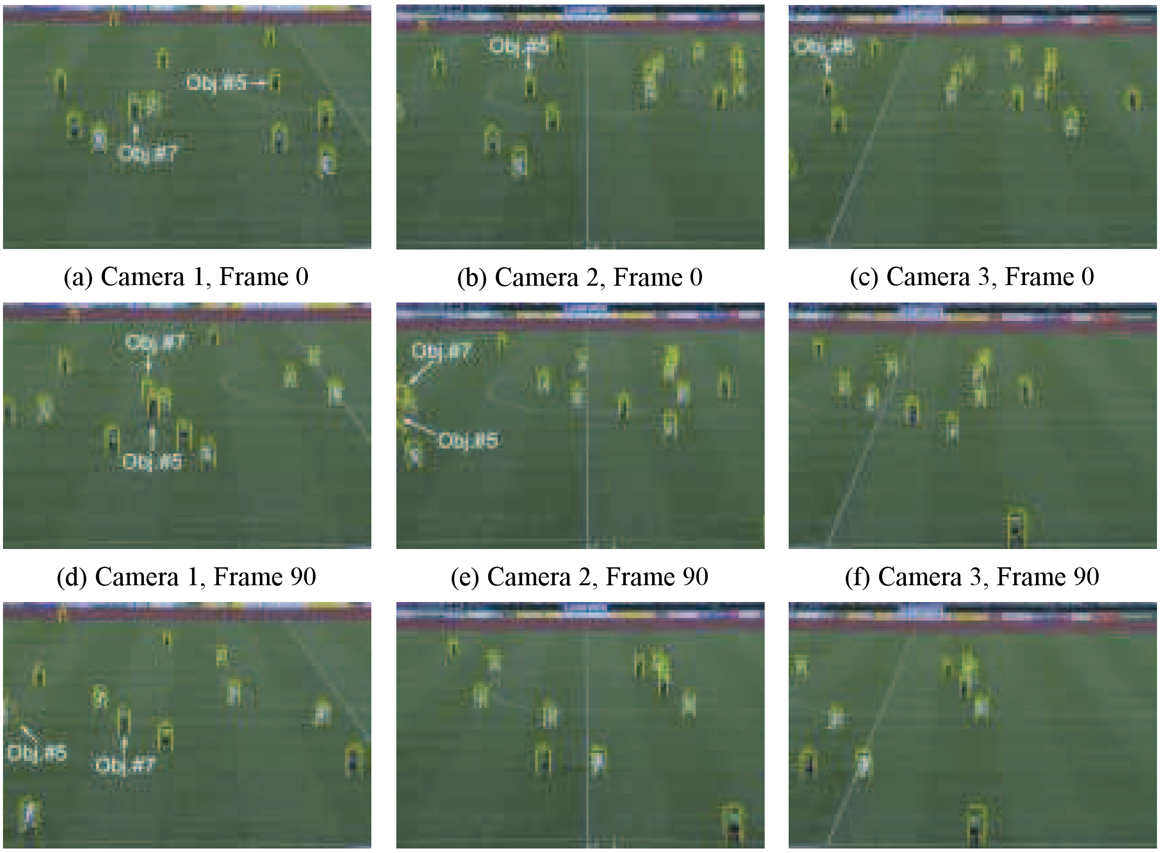
Fig. 10 shows a tracking result (projected onto the image plane) of simply crossing athletes (Scene 1) observed by a single camera using a tracker with the plug-ins listed in Table 1. The initial world coordinates of athletes are manually designated. Although both the athletes are wearing similar blue shirts, they are successfully tracked.

In Fig. 11, the state covariance matrices P_1 and P_2 are visualized[1]. The state variances become larger during their overlapping period, because more of the MTCHs are automatically invalidated to avoid unreliable observation. Athlete #2 is running on the camera side of #1, which is why the covariance P_2 has somewhat smaller values (i.e., more reliable estimates) compared to those of P_1.

As more complicated case, we tried tracking athletes in a real professional football sequence (Scene 2). We specified the tracking order as listed in Table 2 using three fixed cameras on the stand as illustrated in Fig. 12. The image processing of three views required 5.5 seconds per frame using a PC with a 1 GHz Pentium III processor. The camera parameters were visually calibrated using the white lines of the football field.

As shown in Fig. 13 and 14, almost all of the athletes (except #7) were correctly tracked

[1] The square root of sum of the first three diagonal components of P_i is plotted as a measure of the positional error.



(a) Camera 1, Frame 0 (b) Camera 2, Frame 0 (c) Camera 3, Frame 0

(d) Camera 1, Frame 90 (e) Camera 2, Frame 90 (f) Camera 3, Frame 90

(g) Camera 1, Frame 210 (h) Camera 2, Frame 210 (i) Camera 3, Frame 210

Figure 13. Tracked Athletes Superimposed on Camera Images (Scene 2)

throughout the 7-second sequence. The trajectory of athlete #7 became slightly unstable at around the 90^{th} frame. This was because there were two other athletes #5 and #18 overlapping with him, and because he was going to be visible from no more than one camera. The tracking failure, however, would easily be detected by thresholding the state covariance P_7, since it had larger values around the 90^{th} frame compared to other periods (see dotted line in Fig. 15). In contrast, athlete #5, who passed in front of #7, was sufficiently observed to be tracked confidently while passing #7.

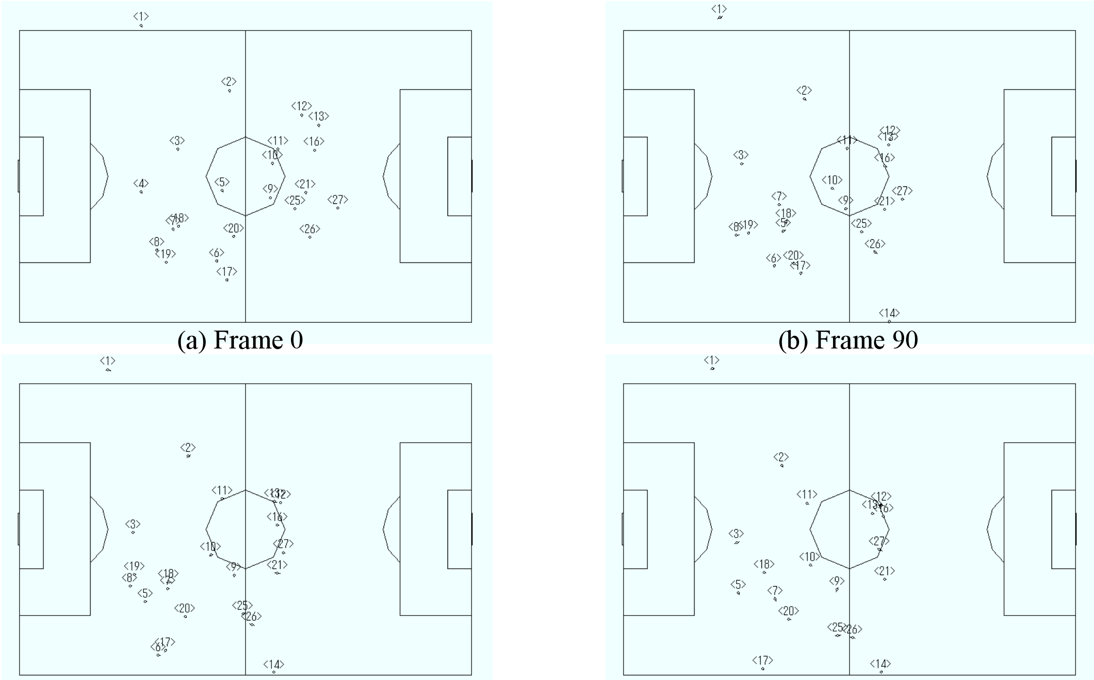
The color matching plug-in in Table 2 can find its target by color, even if he/she has been lost in the past, as long as the other similarly colored athletes are accurately tracked. Consequently, #7 could be recaptured successfully at the 105^{th} frame.

7. SUMMARY AND CONCLUSIONS

We proposed a reconfigurable architecture for tracking moving objects based on observation of multiple features acquired by multiple cameras. The matching plug-ins stepwise integrate the observations to refine the estimates of the target trajectories in world coordinates. The experiments with football sequences showed the robustness of the architecture against occlusion.

One of the most important and novel features of our proposed architecture is that the tracking algorithm can easily be modified just by plugging-in/-out some modules (EXTRs, MTCHs, and PREDs) upon demand. In the experiments, the chroma-key plug-in was employed in Scene 1, whereas the background subtraction plug-ins were used in Scene 2. The reconfigurable architecture will be a key technology to develop a multi-purpose automated athlete tracker for general sport scenes.

We are planning to apply this tracker to visualize invisible information in sports scenes (e.g., offside-lines of football, etc.). As a by-product of the silhouette matching plug-in, the system creats a labeled image as shown in Fig. 16. We are also developing a semantic scene analyzer that detects athletes' actions from track and shape information [Mal03a].



(a) Frame 0 (b) Frame 90

(c) Frame 150 (d) Frame 210

Figure 14. Top View of Athletes' Positions (Scene 2)

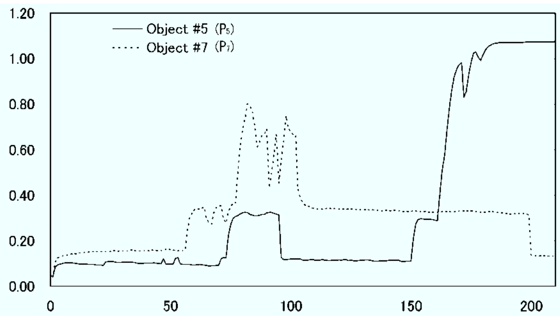


Figure 15. State Covariance of Scene 2

Figure 16. Labeled Image (Scene 2, Camera 1, Frame 210)

8. REFERENCES

[Isa98a] Isard, M. and Blake, A. Codensation — Conditional density propagation for visual tracking. International Journal of Computer Vision, Vol. 29, No. 1, pp. 5-28, 1998.

[Jan00a] Jang, D.-S., Jang, S.-W., and Choi, H.-I. Structured Kalman filter for tracking partially occluded moving objects. Lecture Notes in Cmputer Science, pp. 248-257, 2000.

[Kal60a] 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.

[Mal03a] Malerczyk, C., Klein, K., and Wiebesiek, T. 3D reconstraction of sports events for digital TV. Journal of WSCG, Vol. 11, No. 1, pp. 306-313, 2003.

[Mcl97a] McLachlan, G.J., Krishnan, T. The EM algorithm and extensions. Wiley Series in Probability and Statistics, John Wiley & Sons, Inc., 1997.

[Mey94a] Meyer, F.G. and Bouthemy, P. Region-based tracking using Affine motion models in long image sequences. CVGIP, Image Understanding, Vol. 60, No. 2, pp. 119-140, 1994.

[Mis02a] Misu, T., Naemura, M., Zheng, W., Izumi, Y., and Fukui, K. Robust tracking of soccer players based on data fusion. Proceedings of ICPR2002, Vol. 1, pp. 556-561, 2002.

[Nae00a] Naemura, M., Fukuda, A. Mizutani, Y., Izumi, Y., Tanaka, Y., and Enami, K. Morphological segmentation of sports scenes using color information. IEEE Transactions on Broadcasting, Vol. 46, No. 3, pp. 181-188, 2000.

[Tsa87a] Tsai, R.Y. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, pp. 323-344, 1987.