Motion Fields to Predict Play Evolution in Dynamic Sport Scenes

Kihwan Kim[1] Matthias Grundmann[1] Ariel Shamir[2] Iain Matthews[3] Jessica Hodgins[3] Irfan Essa[1]

[1] {kihwan23, grundman, irfan}@cc.gatech.edu [2]arik@idc.ac.il [3]{iainm, jkh}@disneyresearch.com

[1]Georgia Institute of Technology, Atlanta, GA, USA [2]The Interdisciplinary Center, Herzliya, Israel [3]Disney Research, Pittsburgh, PA, USA

http://www.cc.gatech.edu/cpl/projects/playevolution

Abstract

Videos of multi-player team sports provide a challenging domain for dynamic scene analysis. Player actions and interactions are complex as they are driven by many factors, such as the short-term goals of the individual player, the overall team strategy, the rules of the sport, and the current context of the game. We show that constrained multi-agent events can be analyzed and even predicted from video. Such analysis requires estimating the global movements of all players in the scene at any time, and is needed for modeling and predicting how the multi-agent play evolves over time on the field. To this end, we propose a novel approach to detect the locations of where the play evolution will proceed, e.g. where interesting events will occur, by tracking player positions and movements over time. We start by extracting the ground level sparse movement of players in each time-step, and then generate a dense motion field. Using this field we detect locations where the motion converges, implying positions towards which the play is evolving. We evaluate our approach by analyzing videos of a variety of complex soccer plays.

1. Introduction

Understanding complex dynamic scenes in team sports is a challenging problem. This is partly because an event in a game involves not only the local behaviors of individual players but also structural global movements of players. We are interested in automated analysis of such complex scenes with multi-agent activities, and consider that the tracking of multiple agents can be used to analyze these scenes, extract interesting events, and even predict what is likely to happen. We draw motivation from a quote by Wayne Gretsky, the legendary hockey player, *"A good hockey player plays where the puck is. A great hockey player plays where the puck is going to be."* Our work is based on the assumption that the players themselves have the best view and clearest understanding of the development of a game during play. The players' movement on the field reflects their interpretation, and possibly their intent, based on their role in the game, which we should leverage for interpreting the state of the game.

Our hypothesis is that higher level information can be deduced by tracking and analyzing the players movements, not only individually but also as a group. In this paper we describe a method to build a global flow field from players ground-level motions. We propose the novel concept that the flow on the ground reflects the intentions of the group of individual players based on the game context, and use this for understanding and estimating future events.

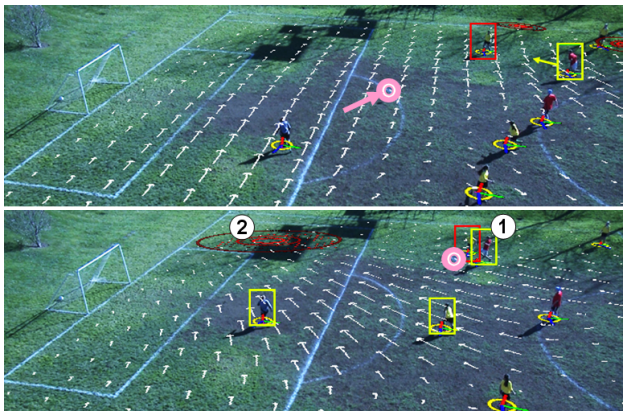


Figure 1. Examples of how players movement indicates play evolution in a dynamic sport scene. The motion field on the ground is denoted as white arrows, and the locations where play evolution is predicted are denoted as red iso-contours. Ball location is highlighted with a circle. **Top:** The goalkeeper passes the ball to the last defender (red box) while an offender (yellow box) is moving to intercept him. **Bottom:** One second (30 frames) later, at the moment of interception (position 1), the goalkeeper and another defender (yello boxes) are moving in the direction of position 2. This indicates the possible location of a future event.

In this work, we specifically focus on the sport of soccer (i.e. football). For example, consider the soccer scene in Fig. 1, which demonstrates play evolution. The goalkeeper passes the ball to a nearby defender (top), but one of the offensive players sees an opportunity to intercept/steal the ball. One second later (bottom) we see the goalkeeper and another defender start moving to location 2 to prepare to respond to an offensive interception. The players are tracked on the ground plane to generate a flow field (shown by the white vector field) which in turn is used to infer possible locations of future events, noted by red circles. Our primary contributions in this work are: (1) *Extracting ground-level motion* from individual players movement from multiple-views. (2) Generating a dense flow field from a sparse set of individual players motions, a *motion fields* on the ground. And, (3) Detecting the locations where the motion field converges and inferring the *play evolution*.

In the next few sections we present the technical details

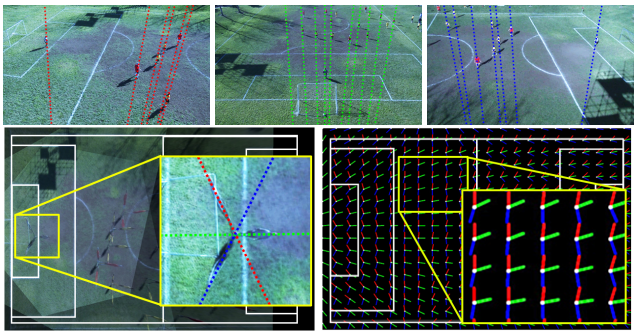


Figure 2. **Geometric Constraints**: (Upper row) Each view ($\mathbf{I}_k$) has a vertical vanishing points ($\mathbf{v}_k$). (Bottom left) In the top-view warped image ($\mathbf{I}_k^{top}$), players are distorted in the direction of the projected vanishing points ($\hat{\mathbf{v}}_k$) of each view. (Bottom right) The location of a player on the ground is identified by the intersection of these projections on the ground plane.

of our approach followed by a brief overview of some of the related work. Then we present a variety of results, conduct a series of evaluations, and discuss some applications of our approach.

2. Motion Field Construction from Video

The first step in constructing a motion field is extracting tracks of individual players on the field. While single camera tracking is possible for soccer [15], and could be useful for our approach, for this work, we have decided to rely on more robust multiple player tracking using multiple views.

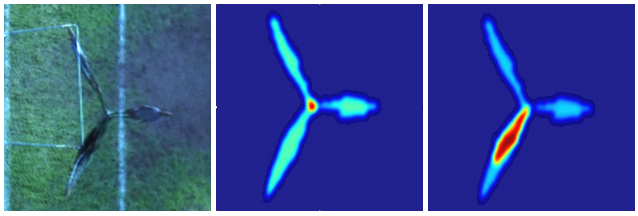
View dependent analysis for player tracking using multiple camera suffers from a data fusion problem. In addition, flow analysis may be sensitive to the perspective distortion of different views. To address these issues, our approach is to analyze the game scenes from a *top-view* warped image of the ground plane. The top-view is constructed by combining the warped footage of each of the multiple cameras. We then extract a multi-view consistent player location in the top-view by optimizing the geometric constraints shown in Fig. 2. This allows us to create the individual players' ground level motion. (Sec. 2.1).

Through spatial and temporal interpolation we combine the tracked player motions to create our *motion field on the ground-plane* (Sec. 2.2). Finally, we analyze this motion field to detect and localize important regions (Sec. 2.3).

We first define some notations. Assume that we have N cameras. Let $\mathbf{I}_k$ $(1 \leq k \leq N)$ refer to a frame of each camera and $\mathbf{I}_k^{top}$ is a top-view image where each $\mathbf{I}_k$ is warped through the homography $\mathbf{H}_k^{top}$. Additionally, $\mathbf{x} \in \mathbf{I}^{top}$ denotes that $\mathbf{x}$ is in the coordinate space of a top-view (ground field).

2.1. Extracting Individual Ground-Level Motion

To construct our flow field, we first extract the ground-level motion of individual players. At each time t this motion is



(a) (b) (c)

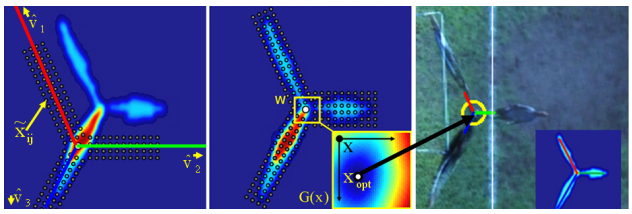
Figure 3. **Position Confidence in top-view**: (a) Overlapped top-view of warped views from each angle (each view of a player is warped over the direction of vertical vanishing points) (b) Normalized probability of the summation of the warped foreground probabilities. Note that the region which is close to the ground has higher probability (c) In the presence of shadow.

defined as the velocity vector $[u\ v]^T$ representing a player's movement on the ground at a 2D location $\mathbf{x}(x, y) \in \mathbf{I}^{top}$. To find the motion, our algorithm first detects the ground position (optimized location near the feet) of each player $\mathbf{x}$ at a given time t. Then, we search for a corresponding position in a previous frame at time $t - a$ (where $a > 1$ for stability and is usually set to 5 frames). The motion velocity at time t is the difference between these two positions. Note that the individual motion is reconstructed at each time separately and does not require explicit tracking since it is only used to construct our flow field.

To find the 2D location of players on the ground we make use of the fact that each view has its own vertical vanishing point (VVP) $\mathbf{v}_k$. We denote the projected VVP onto the ground view as $\hat{\mathbf{v}}_k = \mathbf{H}_k^{top}\mathbf{v}_k$ $(1 \leq k \leq N)$. Each $\hat{\mathbf{v}}_k$ gives us a unique direction in any location on the ground (see Fig. 2). Using background subtraction we define for each pixel in each view a confidence measure of that pixel being part of the foreground (i.e. player) or background (i.e. grass field) [12]. We combine all measures from all views on the ground plane by summing their projections and normalizing to get the *position confidence* map $\mathbf{PC} : \mathbf{I}^{top} \rightarrow [0, 1]$, where $\mathbf{PC}(\mathbf{x})$ is the probability that $\mathbf{x} \in \mathbf{I}^{top}$ is part of the foreground (Fig. 3).

Since the the region around each player's foot is located on the ground plane where the homographies for each view are extracted, the probability of foreground in those regions will be higher than in other regions (Fig. 3(b)) [10]. However, if there are cast shadows, the shadow region will also have high foreground probability (Fig. 3(c)). Therefore, we consider the highest $\mathbf{PC}$ position only as an initial location of the player. We define a window $\mathbf{W}_{init}$ around the initial position and refine it based on geometric constraints.

The geometric constraints are included by searching for the intersection point of the foreground projection of all N directions, where N is the number of views. This intersection is the weighted centroid of all foreground samples ($\mathbf{x}$ s.t. $\mathbf{PC}(\mathbf{x}) \neq 0$) along each projected VVP in all N directions (Fig. 4(b)). We define the player ground loca-



(a) (b) (c)

Figure 4. **Finding the optimal location using Geometric Constraints**: (a) Samples ($\tilde{\mathbf{x}}_{ij}$) along lines from the evaluating point $\mathbf{x}$ and each projected VVPs $\hat{\mathbf{v}}_k$. Since the number of foreground samples is small, the window moves to top-right. (b) Evaluate Eq. 1 at each point $\mathbf{x}_{ij}$ inside $\mathbf{W}_{init}$. (c) Optimized location.

tion cost function $\mathbf{G}(\mathbf{x})$ and search for its minimum inside $\mathbf{W}_{init}$. $\mathbf{G}(\mathbf{x})$ is the weighted summation of the distance between a set of foreground sample points $\tilde{\mathbf{x}}_{i,k}$ and a line axis established by $\mathbf{x} \in \mathbf{W}_{init}$ and each projected vertical vanishing point $\hat{\mathbf{v}}_k$,

$$\mathbf{G}(\mathbf{x}) = \sum_{k=0}^{N} \sum_{i=0}^{n_k} \mathbf{PC}(\tilde{\mathbf{x}}_{i,k}) \cdot \mathrm{d}\left(\tilde{\mathbf{x}}_{i,k}, (\hat{\mathbf{v}}_k - \mathbf{x})\right), \quad (1)$$

where n_k is the number of foreground samples based on each direction k, and we use $\mathbf{PC}(\tilde{\mathbf{x}}_{i,k})$, the probability of being foreground, as the weight for each of the foreground sample points.

As shown in Fig. 4, the evaluation based on $\mathbf{G}(\mathbf{x})$ is performed over all directions simultaneously (in this case $N = 3$). The optimal ground-level position of the player is $\mathbf{x}_{opt} = \arg\min_{\mathbf{x} \in \mathbf{W}_{init}} \mathbf{G}(\mathbf{x})$.

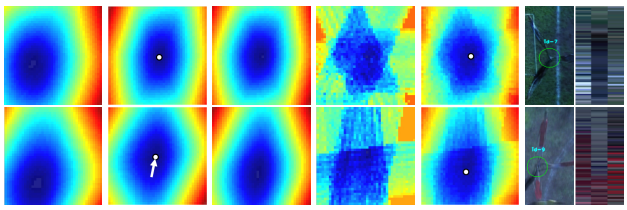
Note that the set of sampling points $\tilde{\mathbf{x}}_{i,k}$ for each $\mathbf{x} \in \mathbf{W_{init}}$ are organized along the line axis $(\hat{\mathbf{v}}_k - \mathbf{x})$. The sampling range is calculated by finding the average height of players using vanishing points [8]. If the summation of all weights $\mathbf{PC}(\tilde{\mathbf{x}}_{i,k})$ for all views is too small (Fig. 4(a)), this is interpreted as a wrong initialization or a false-positive detection of the player and discarded.

To find the corresponding position $\mathbf{x}_{opt}^{t-a}$ of the player in the previous frame $t-a$ we establish a search window $\mathbf{W}_{opt}$ centered around $\mathbf{x}_{opt}$. We use a combination of the geometric constraints $\mathbf{G}(\mathbf{x})^{t-a}$ on the previous top-view frame $\mathbf{I}_{t-a}^{top}$ using Eq. 2, and a color proximity measure $\mathbf{C}(\mathbf{x})^{t-a}$:

$$\mathbf{x}_{opt}^{t-a} = \underset{\mathbf{x}(x,y) \in \mathrm{W}_{opt}}{\arg\min} \left(\mathbf{G}(\mathbf{x})^{t-a} + \beta \mathbf{C}(\mathbf{x})^{t-a}\right) \quad (2)$$

$\mathbf{C}(\mathbf{x})^{t-a}$ is a normalized Bhattacharyya distance of the color (HSV) histogram between the two sets of foreground samples used for $\mathbf{x}_{opt}^{t}$ and $\mathbf{x}^{t-a} \in \mathbf{W}_{opt}^{t-a}$ respectively (Fig. 5). The weighting factor β is usually very small (0.1). The use of color similarity reduces the chance that we are matching a different player. Once $\mathbf{x}_{opt}^{t-a}$ is found, we can define the motion (velocity vector) at $\mathbf{x}_{opt}$ as:

$$[u, v]^T = \frac{\partial \mathbf{x}(x,y)}{\partial t} \cong \left(\mathbf{x}_{opt}^{t} - \mathbf{x}_{opt}^{t-a}\right)/a \quad (3)$$



(a) (b) (c) (d) (e) (f)

Figure 5. Visualizing the components for evaluation of player location at time t and $t-a$. The player in the upper row remains in same location, and the one in bottom row moves. Colors range from blue (low) to red (high). In each row: (a) Values of $\mathbf{G}(\mathbf{x})$ in $\mathbf{W}_{init}$, (b) Values of $\mathbf{G}(\mathbf{x})^t$ in $\mathbf{W}_{opt}$ at current frame t, (c) Values of $\mathbf{G}(\mathbf{x})^{t-a}$ in $\mathbf{W}_{opt}$ at previous frame $t-a$, (d) Distances of the color histogram between frame t and $t-a$ within $\mathbf{W}_{opt}$, (e) Weighted sum of $\mathbf{G}(\mathbf{x})^{t-a}$ and $\mathbf{C}(\mathbf{x})^{t-a}$, and (f) A player of the evaluation and sampled colors. We extract velocity vectors between t and $t-a$ by subtraction of minima between (b) and (e). This vectors are shown in (b) as white arrow.

2.2. Dense Motion Field Construction

Using our method for tracking player motion, we obtain a sparse set of motions on the ground plane. To generate a dense ground-level flow field we combine these sparse motions using radial basis function interpolation [2]. We also temporally interpolate the flow using a weighted set of motions over time.

As described in Section 2.1, the motion at a location $\mathbf{x}(x, y) \in \mathbf{I}^{top}$ is defined by a velocity vector $[\frac{\partial x}{\partial t} \ \frac{\partial y}{\partial t}]^T = [u \ \ v]^T$. If we detect N_k individual players at a given frame k, then the set of the positions is denoted as $\{\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots, \mathbf{x}_{N_k}^k\}$, and the corresponding sets of velocities for each direction are denoted as $\{u_1^k, u_2^k, \ldots, u_{N_k}^k\}$, and $\{v_1^k, v_2^k, \ldots, v_{N_k}^k\}$ for x and y directions respectively.

We define a temporal kernel of size p, using a half Gaussian function. By applying the kernel to each entry of velocity over time, we can construct two $n \times 1$ vectors which are temporally smoothed versions of u_i^k to u_i^{k-p+1} and v_i^k to v_i^{k-p+1} respectively: $\mathbf{U} = [U_1, U_2, \ldots, U_{N_k}, \ldots, U_n]^T$ and $\mathbf{V} = [V_1, V_2, \ldots, V_{N_k}, \ldots, V_n]^T$, where U_i and V_i $(1 \leq i \leq n)$ are scalar velocities for each direction. The matching for each entry over time is set deterministically [18] (e.g. minimum distance and orientation). Note that commonly $n = N_k$ when the number of detected players does not vary over time. However, when there are less number of entries in a given frame k, compared to previous frames, n becomes larger than N_k.

Now, our problem may be stated as follows: given a collection of n scattered 2D points $\{\mathbf{x_1}, \mathbf{x_2}, \ldots \mathbf{x_n}\}$ on the ground plane, with associated scalar velocity values $\{U_1, \ldots, U_n\}$ and $\{V_1, \ldots, V_n\}$, construct a smooth velocity field that matches each of these velocities at the given locations. Consider the scalar-valued functions $f(\mathbf{x})$ and $g(\mathbf{x})$ so that $f(\mathbf{x_i}) = U_i$, and $g(\mathbf{x_i}) = V_i$ respectively, for

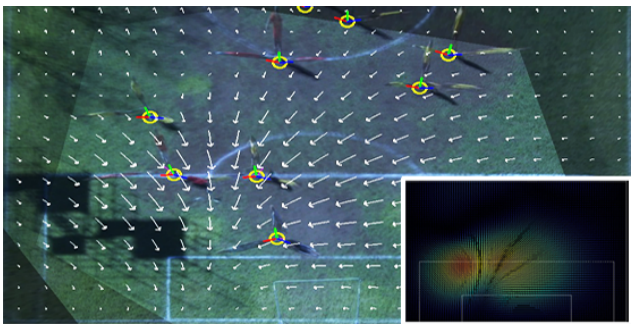


Figure 6. **Motion Field** Φ: White arrows represent the dense motion field generated from a sparse set of motions of players movements. Note that for visualization purposes the dense field is displayed sparsely by averaging the flow at each block.

$1 \leq i \leq n$. For the case of interpolating velocity in the x-direction, we can express the interpolation function as:

$$f(\mathbf{x}) = c(\mathbf{x}) + \sum_{i=0}^{n} \lambda_i \phi(\|\mathbf{x} - \mathbf{x}_i\|), \qquad (4)$$

where $c(\mathbf{x})$ is a first order polynomial that accounts for the linear and constant portions of f, λ_i is a weight for each constraint, and $\mathbf{x}_i$ are the locations of the scattered points (*nodes*). Specifically, the radial function ϕ was chosen as the *thin-plate spline*, $\phi(r) = r^2 \log r$, as it gives us C^1 continuity for smooth interpolation of the velocity field[1].

To solve for the set of weights λ_i so that the interpolation satisfies the constraints $f(\mathbf{x}_i) = U_i$, we solve the equation by evaluating each node at Eq. 4 (e.g. $U_i = c(\mathbf{x}_i) + \sum_{j=0}^{n} \lambda_j \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$).

Since the equation is linear in the unknowns, it can be formulated as a linear system:

$$\begin{bmatrix} \mathbf{A} & \mathbf{Q} \\ \mathbf{Q}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{U} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 1 & x_1 & y_1 \\ \vdots & \vdots & \vdots \\ 1 & x_n & y_n \end{bmatrix},$$

where $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]^T$, $\mathbf{c} = [c_1\ c_2\ c_3]^T$ and the $n \times n$ matrix $\mathbf{A} = (a_{ij}) = \phi(||\mathbf{x}_i - \mathbf{x}_j||)$.

Once we solve the system, the interpolated velocity of the x-direction at any location $\mathbf{x}_a(x_a, y_a) \in I^{top}$ can be evaluated as: $u_a = c_1 + c_2 x_a + c_3 y_a + \sum_{i=1}^{n} \lambda_i \phi(\|\mathbf{x}_a - \mathbf{x}_i\|)$. The velocity of y-direction is interpolated similarly. To generate temporally smoother transitions the flow is finally smoothed using a 1×5 box filter. We refer to this flow as the *motion field* on the ground, and denote it as $\Phi(\mathbf{x}) = f(\mathbf{x})\mathbf{i} + g(\mathbf{x})\mathbf{j} = u\mathbf{i} + v\mathbf{j}$. See Fig. 6.

2.3. Detecting Points of Convergence

The motion field reflects the local and global player motion representing the *play* (i.e. the strategy or intention of the players). We now define a *point of convergence* (POC) as the spatial location that play evolution is proceeding toward in the near future. In this section, we provide a method to detect POCs of the game by finding locations where the motion field merges.

[1]Thin-plate spline minimizes the energy function : $E = \int_{\Re} (\frac{\partial^2 f}{\partial x^2})^2 + 2(\frac{\partial^2 f}{\partial x \partial y})^2 + (\frac{\partial^2 f}{\partial y^2})^2 dxdy$ over all interpolants [3, 17].

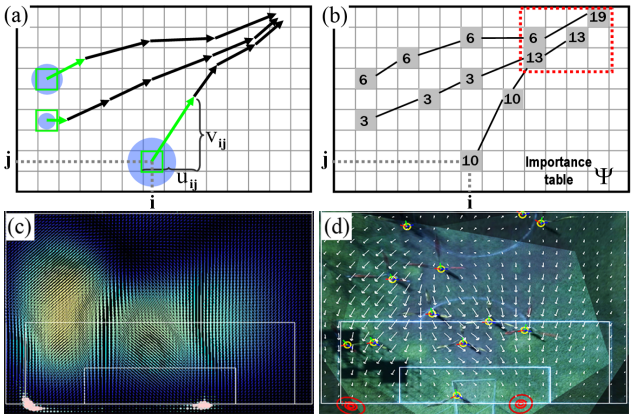


Figure 7. **Points of Convergence detection**: (a) Starting from the position $\mathbf{x}_{ij}$ and another point having a different magnitude of motion vector as an example. The magnitude of ρ_{ij} is propagated through the point $(i + u_{ij}, j + u_{ij})$. (b) An importance table Ψ is updated by adding propagated confidence along Φ. (c) Pink circles at bottom are the location where the accumulated importance is high enough (larger is higher confidence). (d) Meanshift clustering and Guassian mixture modeling detects two POCs in this case.

Point of convergence detection is implemented in two steps. First, the motion field on the ground Φ, is used to propagate a confidence measure forward to calculate an importance table Ψ whose size is the same as $\mathbf{I}^{top}$. Then, the accumulated confidence in Ψ are clustered and a Gaussian Mixture Model is used to detect POC clusters. Fig. 7 shows an example of how POCs can be automatically detected from a motion field Φ.

We introduce a confidence value, defined as the local magnitude of velocity at any location on the ground. In the first step, we propagate (copy) this value at a fixed time t from each starting location through Φ. Then, we accumulate these values along the trace in an *importance table* Ψ. Given a location $\mathbf{x}(i, j) \in \mathbf{I}_t^{top}$, Ψ is calculated by performing a forward propagation recursively based on the motion field Φ. The magnitude of the velocity $\rho_{ij}^2 = u_{ij}^2 + v_{ij}^2$ is propagated by updating Ψ as follows: $\Psi(i + u_{ij}, j + v_{ij}) = \Psi(i + u_{ij}, j + v_{ij}) + \rho_{ij}$. We continue this forward propagation along the motion field until the attenuation which is proportional to $\rho_{i,j}$ is smaller than ϵ. Consequently, locations having a large ρ in Φ can have a large influence on far away locations as long as the motion field moves in that direction.

We compute the accumulated distribution of confidence Ψ (Fig. 7(c)), by computing confidence propagation for any location in $\mathbf{I}^{top}$. To determine the location and the number of POCs at a given frame k, we apply mean-shift clustering [5] to find an optimal number of clusters. Based on the

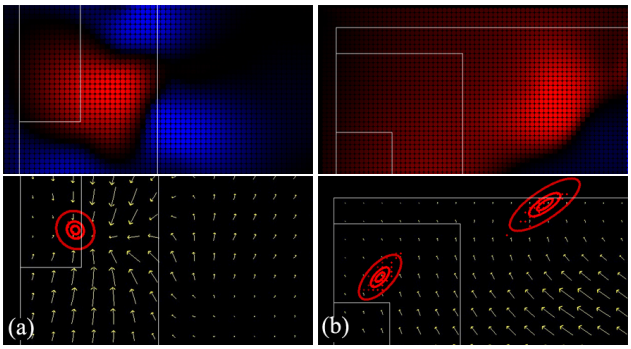


Figure 8. **Divergence and Points of Convergence**: Upper row: Red regions denote the regions where $\nabla\Phi < 0$, and blue regions denote regions where $\nabla\Phi > 0$. Lower row: Φ in (a) has specific singular sink, while Φ in (b) has no specific singular region. In both cases our approach detects the POCs (red elipses).

initial mean and the number of clusters (modes), we fit a Gaussian Mixture model to the distribution of those regions using Expectation Maximization (EM) (Fig. 7(d)).

Note that our POC detection is different than classical singular (critical) points detection. Primarily, POC is a global measurement of the flow, while the critical point is a local extremum of the velocity potential and the stream functions [7].

The divergence of the motion field $\Phi(\mathbf{x})$ at the location $\mathbf{x} \in \mathbf{I}^{top}$ is defined as $\mathrm{div}(\Phi(\mathbf{x})) = \nabla\Phi = \frac{\partial \mathbf{u}}{\partial x} + \frac{\partial \mathbf{v}}{\partial y}$. If $\nabla\Phi$ is negative, the flux of the motion field across the boundary of the region is inward, while positive is outward. Thus, if the motion field flows to the boundary of a specific region, the divergence of that region becomes negative (see Fig. 8).

In practice, many of the detected POCs exist in regions where the local measurement of divergence becomes negative because the POC proceeds in the direction of the motion field flow. Therefore, in many cases, a POC exists where there is a significant singular sink point (Fig. 8(a)). However, if the majority of flows in a specific scene are not regular enough to construct an apparent local extremum, determining a POC by detection of singularities will fail (Fig. 8(b)). In such cases, our forward-propagation method can still locate regions of major global flows which signify positions where the play evolution may proceed to.

3. Related and Motivating Past Work

Our approach described in the previous sections, is motivated by previous efforts in tracking, flow estimation, modeling from data and motion field analysis. In this section we briefly highlight some of this relevant work.

The multi-view tracking element of our algorithm is similar to Kim and Davis's [11] person tracker that uses multiple views. In this approach, they first detect a person as a blob, and use the vertical axis of the blob to aid in seperated occluded people. Thus, they explore the data fusion problem from measurements of each view, while we look for an optimal solution at the merged top-view. Eshel et al. [9] introduced homography-based head tracking using multiple views to track a dense crowd. Khan and Shah [10] propose a method that is similar to our initialization step. They refer to this as a "synergy map" which corresponds to our position confidence map. However they do not provide an optimizing step, so their approach works best when the data is sparse and there are no significant shadows.

In our work on motion field analysis, our task is similar identifying critical points in a flow field. Previous work has considered detecting critical points in optical flow extracted from satellite image sequences [6] and used topological structure for identifying the points in global orientation field extracted from images [19]. Rao and Jain [14] make use of a symbolic descriptor of orientation texture for detecting critical points for finger print recognition. While these methods work well for their specific applications, their local criterion are not robust to noise and they are not ideal for our task of analyzing multi player motions and flow.

Corpetti et al. [7] introduced a robust method to detect singular points from synthetic and real dense motion fields generated from optical flow of satellite cloud images. They decompose irrotational and solenoidal components from dense flow and extract velocity potentials to find a local extrema for finding critical points. Tong et al. [16] also proposed a robust method to detect critical points by decomposing a vector field into curl free, divergence free, and harmonic fields with Hodge Helmholtz decomposition. Methods using vector field decomposition report very stable results from image and video data, yet are still computationally expensive, and the output only has *local measurement*. None of their methods detect the target location of harmonic motion, such as linear movements or curved motion which occur often in sports scenes, of particular interest here.

Our work has direct impact on generating automatic broadcasts from a series of videos. Similar efforts have been explored in the context of an Intelligent Studio for cooking shows using scripts and simple vision techniques [13], and for sports using player location for rule-based framing, Araki et al. [1]. The Pajama channel [4] explores a similar concept by measuring horizontal flow of the game from a mosaic video. None of these applications consider the detection of future events, which can aid in planning and controlling camera moves, as shown here.

4. Results and Evaluation

We collected a data set by recording a soccer match between a local college team and a local amateur team using three synchronized HD cameras. Each camera was mounted on top of an approximately 12m high scaffold. Most of our results are also presented in the accompanying video, which is perhaps a more compelling way to view the output.

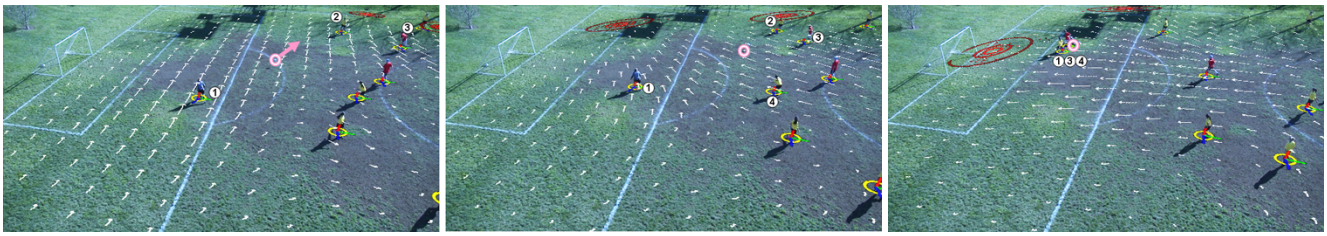


Figure 9. **Example 1 of play evolution.** In all figures, the the pink circles are the location of the ball, the arrows denote the initial direction of the movement of the ball, white arrows denote the motion field on the ground, and red contours are the distribution of POC where play evolution aims. In this example - interception & goal keeping. **(Left)** Goalkeeper 1 passes a ball to the last defender 2. A POC appears near the defender, **(Middle)** While the ball is still on the way to the defender 2, another POC appears at different location as offender 3 approaches and the goalkeeper and defender 4 decide to defend. **(Right)** The ball is intercepted by offender 3 and the POC event in the left region actually occurs. Finally, the ball was saved by the goalkeeper. Note that two POCs appear in both possible directions.

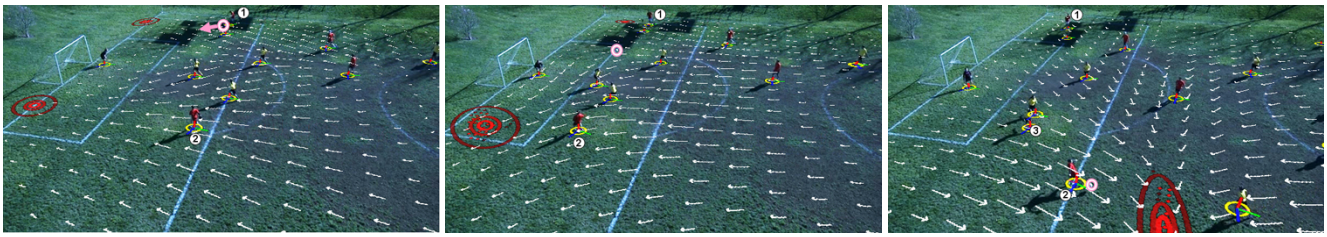


Figure 10. **Example 2 of play evolution - center pass**: **(Left)** Offender 1 dribbles towards the upper corner while offender 2 runs toward the other corner (two POCs). **(Middle)** Offender 1 kicks a center pass. While the ball is travelling the POC near offender 2 becomes larger. **(Right)** Defender 3 intercepts and the ball changes its direction. Offender 2 and another defender approach to the ball.

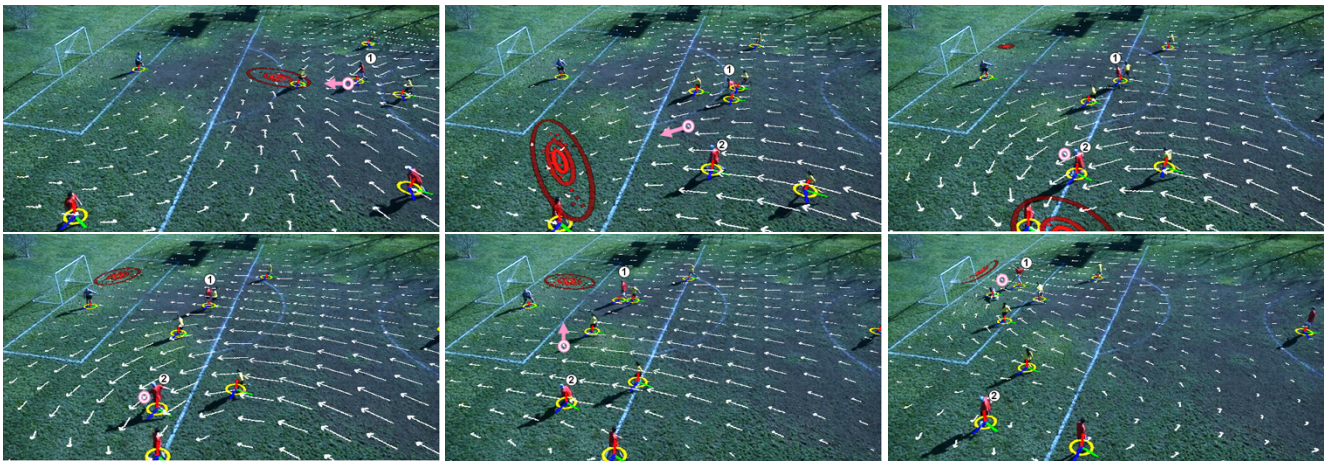


Figure 11. **Example 3 of play evolution - back-door and through pass**: **Upper row: (Left)** Offender 1 dribbles forward (a POC is in front of the player). **(Middle)** Offender 1 passes the ball to offender 2 as part of a back-door strategy. A POC appears near offender 2. **(Right)** While the offender 2 waits to receive the ball, offender 1 and the defenders are running toward the goal-post. Another POC appears near goal. **Lower row: (Left)** Before offender 2 kicks the ball, the POC near the goalpost becomes larger. **(Middle)** Offender 2 through-passes to offender 1. **(Right)** Offender 1 attempts to score.

To evaluate the ground-level motion detection we back-project our automatic detection results onto each view and manually tracked them to find discrepancies (Fig. 13). We evaluated 14,814 individual players in 2,000 frames from one of our data sets. As we only track the players covered by at least two views, players seen in only one camera were not evaluated. Table. 1 summarizes our results. Note that the false-negatives for two views are three times larger than using three views. This is due to the reduced number of constraints (vanishing directions) and fewer sample points that often results in a solution which is less confident. This tendency is also reported in other multi-view approaches [11, 10].

Qualitative evaluation of the dense motion field generation and POC detection is demonstrated in Figs 9, 10 and 11. The distributions of POC clusters are shown as red contours, and the motion field on the ground as white arrows. These figures, and more examples in the accompanying video, il-

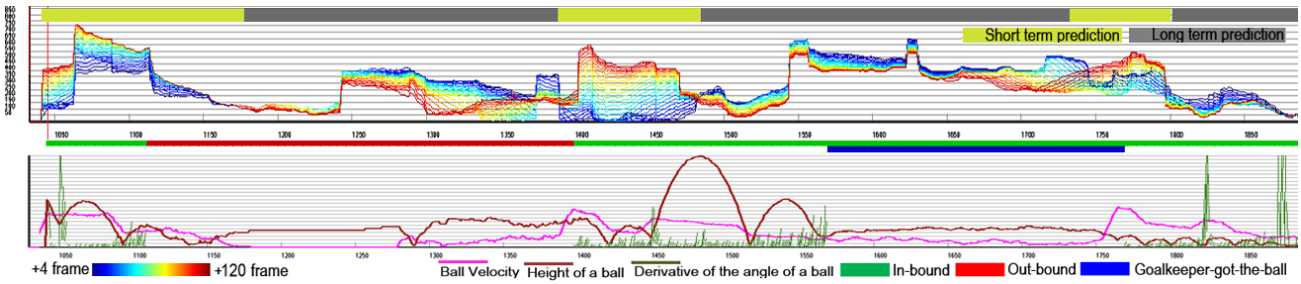


Figure 12. **Distances between the location of the detected POC in the current frame and the location of the ball in future frames**: For both graphs, x-axis is the frame number, and y-axis is a pixel level distance (100 pixels is approximately 4 meters). The measurement varies from blue (4 frames later) through red (120 frames later). In the upper graph, we partitioned the measurement into two types of the prediction: short term and long term. In the bottom graph, more information extracted from the game scene is shown.

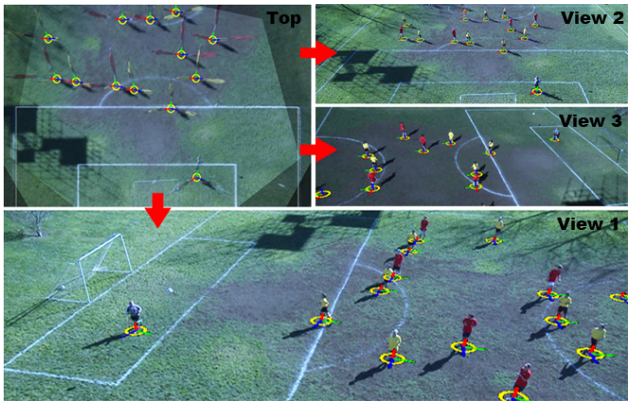


Figure 13. **Evaluation of automatic player location on the ground:** We back-project the position onto each view for evaluation. Each R, G and B line with yellow circle denotes the direction of the vertical vanishing points for each view (geometric constraints).

Views	FP (E1)	FN (E2)	Total
3	0.7162%	0.5649%	1.2811% (128/9913)
2	1.1000%	1.8120%	2.9120% (143/4901)

Table 1. **Detection error:** False Positive (FP) and False Negative (FN) for each test. Two-views have larger FN error due to fewer constraints.

lustrate how our motion fields and POC detection can reveal interesting game events and foresee important future positions on the field.

Quantitative evaluation of our approach is not an easy task since, (1) the position of important regions in the game can be subjective, and (2) there is no ground truth for defining these regions. To give some quantitative measure, and since ball location is one of the important regions (especially in soccer), we evaluate the results of trying to predict the location of the ball using POCs. We assume that the current POC will be a good indication to where the ball will be in future frames, and then compare the two by collecting ground truth of the ball position manually. To investigate how well our estimated region reflects the future, we vary the temporal offset for measuring the ball location from 4 frames to 120 frames from the current frame.

Fig. 12 (upper) shows the distance between the current POC and the ground truth future position of the ball (determined manually in video) over several frames. We observe that sometimes the prediction difference is smaller in temporally closer frames than more distant frames, while at other times this trend is reversed. We denote the former as "*short-term prediction*" and the later as "*long-term prediction*".

To investigate when short-term or long-term prediction occur, we track additional information from the video. The plots in Fig. 12 (lower) show the height, velocity and derivative of the angle of direction of the ball. Additionally, we add high-level game states such as "in-bound", "out-bound" and "goalkeeper-got-the-ball (GGB)" as horizontal color bars in Fig. 12 (lower). Based on these information, we observe that, (1) short-term prediction happens during in-bound states, (2) it is more likely to have long-term prediction in out-bound and GGB states, and (3) during in-bound states, if the ball position is high and speed of the ball is slowing down, long-term prediction is more likely.

These data-derived observations appear physically plausible. For example, during in-bound states, the game speed is faster, and players do not have time to estimate longer durations. If the ball position is high in the air, players have more time to evaluate where the ball will fall in the long-term. In summary, if we take the minimum distance over all offsets, the average distance between estimated POCs and the corresponding position of the ball is 6.6 meters on average over all time offset, and 4.5 meters excluding GGB states when all players just go back in the field.

The computational time for each part of our approach are given in Table 2. Note that the computation is highly dependent on the resolution of the field ($\mathbf{I}^{top}$). Our current implementation is not real-time, but it could be optimized.

5. Applications

In the accompanying video, we demonstrate two possible applications for our work:

Field Resolution	Motion	Motion Fields	POC
670×500	166.33 ms	15.5 ms	65.10 ms
945×700	325.5 ms	36.5 ms	254.4 ms
1350×990	871.2 ms	93.5 ms	568.24 ms

Table 2. **Computational time with various sizes of top-view:** Test results using an Intel Core i7-965, 3.2GHz with 3GB RAM.

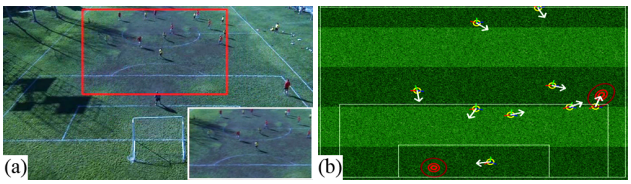


Figure 14. **Applications:** (a) Simulation of automated broadcasting: Red rectangle indicates the field of view of main camera, and an image at bottom-right shows the output image (virtual crop). (b) Top-view Visualization. More application results can be found in the accompanying video.

Automatic Camera Control: Estimating where the important events will happen without accessing future video frames is very important for automated live broadcasting. To mimic human camera operators, an automatic broadcasting algorithm needs to control the pan and zoom of cameras and direct them towards regions of importance while maintaining smooth transitions and good field of view (FOV) framing. We use our POC detection to forecast future important events, and demonstrate control of a virtual camera by using cropped windows (Fig. 14(a) and the accompanying video). We assume that the red rectangle represents the field of view of a camera and move it to follow the position of our POC. Note, that the ball is not always centered in the frame providing richer context of the scene, and the movement of the FOV is smoothed based on the play evolution.

Sports Visualization and Analysis: Tracking the location, the number, and the size of POC may provide a good indication for interesting and important events during a game. This can be a useful tool for analyzing games offline by coaches and trainers, or by broadcasters during a live game to show novel views of the game (Fig. 14(b)).

6. Summary and Future Work

We introduce a novel approach for play evolution analysis using multiple views. We detect and track the ground-level motion of players through optimization of geometric constraints. Using these sparse sets of tracks, we generate a dense motion field on the ground-plane and detect points of convergence in the field as possible future interesting locations of play evolution. We evaluate our approach both quantitatively and qualitatively.

In the future we plan to develop more efficient ways to extract the motion fields and reduce the computational cost. We are interested in pursuing robotically controlled cameras to realize our goal of automated broadcasting [15].

Acknowledgements: Thanks to Pixar Animation Studios, and specifically Jay Weiland, for coordinating the data capture session of a soccer games. Thanks also to the Diablo Valley College Woman's Soccer team and Pixar folks being the soccer talent for our data capture sessions.

References

[1] Y. Ariki, S. Kubota, and M. Kumano. Automatic production system of soccer sports video by digital camera work based on situation recognition. In *ISM*, pages 851–860, 2006. 5

[2] M. D. Buhmann and M. J. Ablowitz. *Radial Basis Functions : Theory and Implementations.* Cambridge, 2003. 3

[3] J. C. Carr, W. R. Fright, and R. K. Beatson. Surface interpolation with radial basis functions for medical imaging. *IEEE Transactions on Medical Imaging*, 16:96–107, 1997. 4

[4] P. Channel. "http://www.pajamachannels.com/". 5

[5] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. PAMI*, 17(8):790–799, 1995. 4

[6] I. Cohen and I. Herlin. Optical flow and phase portrait methods for environmental satellite image sequences. In *ECCV*, pages 141–150, 1996. 5

[7] T. Corpetti, E. Mémin, and P. Pérez. Extraction of singular points from dense motion fields: An analytic approach. *J. Math. Imaging Vis.*, 19(3):175–198, 2003. 5

[8] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *IJCV*, 40:123–148, 1999. 3

[9] R. Eshel and Y. Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *CVPR*, pages 1–8, 2008. 5

[10] S. M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *ECCV*, 2006. 2, 5, 6

[11] K. Kim and L. Davis. Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In *ECCV*, pages 98–109, 2006. 5, 6

[12] D.-S. Lee. Effective gaussian mixture learning for video background subtraction. *IEEE PAMI*, 27(5), 2005. 2

[13] C. Pinhanez and A. Bobick. Intelligent studios: Modeling space and action to control tv cameras. In *Applications of Artificial Intelligence*, pages 285–306, 1997. 5

[14] A. R. Rao and R. C. Jain. Computerized flow field analysis: Oriented texture fields. *IEEE. PAMI*, 14:693–709, 1992. 5

[15] F. Szenberg, P. C. P. Carvalho, and M. Gattass. Automatic camera calibration for image sequences of a football match. In *ICAPR*, London, UK, 2001. 2, 8

[16] Y. Tong, S. Lombeyda, A. N. Hirani, and M. Desbrun. Discrete multiscale vector field decomposition. *ACM ToG*, 22(3):445–452, 2003. 5

[17] G. Turk and J. F. O'Brien. Shape transformation using variational implicit functions. In *SIGGRAPH '99*, pages 335–342, New York, NY, USA, 1999. 4

[18] C. J. Veenman, M. J. T. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE PAMI*, 23:54–72, 2001. 3

[19] J. Zhou, F. Chen, and J. Gu. A novel algorithm for detecting singular points from fingerprint images. *IEEE PAMI*, 31(7):1239–1250, 2009. 5