Detecting Regions of Interest in Dynamic Scenes with Camera Motions

Kihwan Kim[1,2] Dongryeol Lee[1] Irfan Essa[1]
[1]{kihwan23, dongryel, irfan}@cc.gatech.edu [2]kihwank@nvidia.com
[1]Georgia Institute of Technology, Atlanta, GA, USA [2]NVIDIA Research, Santa Clara, CA, USA
http://www.cc.gatech.edu/cpl/projects/roi/

Abstract

We present a method to detect the regions of interests in moving camera views of dynamic scenes with multiple moving objects. We start by extracting a global motion tendency that reflects the scene context by tracking movements of objects in the scene. We then use Gaussian process regression to represent the extracted motion tendency as a stochastic vector field. The generated stochastic field is robust to noise and can handle a video from an uncalibrated moving camera. We use the stochastic field for predicting important future regions of interest as the scene evolves dynamically.

We evaluate our approach on a variety of videos of team sports and compare the detected regions of interest to the camera motion generated by actual camera operators. Our experimental results demonstrate that our approach is computationally efficient and provides better predictions than previously proposed RBF-based approaches.

1. Introduction

Analysis of videos of dynamic scenes with multiple moving objects requires detection of regions of interest in the scene based on the motions of the objects. Automatic extraction of such regions of interest at any point in time over a video sequence is crucial for dynamic scene understanding. This is especially important for scenes captured by moving cameras as the coverage of such dynamic scenes changes with the movement of the camera. For example, in sports videos, expert camera operators actively control the pan-tilt-zoom (PTZ) of their cameras to best capture the dynamics of the play as it unfolds. In surveillance videos, operators move their cameras based on their interpretation of events and activities in a crowded scene.

Automation of where to move the camera in such a scene, which is driven by activities and events in the scene, requires (1) knowledge of how the objects move in the scene, (2) affordances related to how a camera can move to best capture these movements, and most importantly (3) ability to predict from motions of the camera and the dynamics of the scene where regions of interest are in the scene. Such automation can support the capture of the most relevant (interesting) video footage, with smooth motions and transitions. To achieve such automation, we leverage the observation that the global motion field of the scene best encodes the context of the dynamic scene and helps predict future regions of importance. To that end, we present an approach to extract such predictions from real data, which can be used for planning camera motion with smooth transitions.

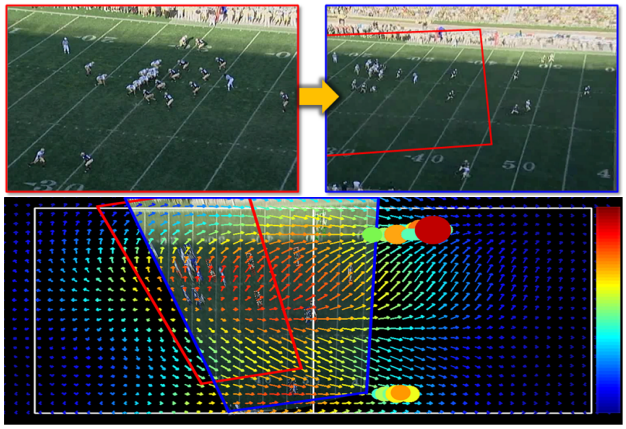

Figure 1. **Overview**: **Top:** An example of the pan-tilt-zoom performed by a camera operator. The field of view changes from the region bounded by red lines to another bounded by blue lines, **Bottom (overhead view):** Arrows indicates a motion field generated only from the ground-motion of players with Gaussian process regression. The certainty level of the velocity vector at each location is represented by different colors. Circles indicate the predicted future locations of interest which can be interpreted as the location where the field of view of the camera will move. Color bar represents the level of values (Red denotes higher values).

We propose a method for constructing a stochastic motion field from a set of sparse motions from sports footage, and provide a method to identify the regions that adequately capture the field of view for the activities in the scene. Figure 1 shows an example of the change of view over time as the camera operator pans and widens the camera view during a football play. The top figures show the two frames

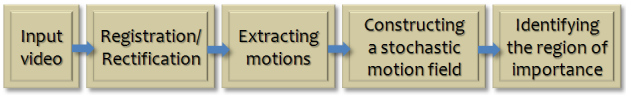

Figure 2. **Overview of our approach:** From input video to the detection of future important locations.

with an overlapping region from left to right. The bottom figure lays out the image on the football field and shows a stochastic motion field on the ground generated using Gaussian process regression, which allows for computing how the play is unfolding, as the players are tracked and followed over the field to determine regions of interest.

Our approach shares similar goals as [6], where motion tracks are used to predict how groups of people move in a scene. However, our approach does not depend on well-configured static-multi-view videos for the precise motion samples on the ground. We support more general and challenging configurations in which the input video (possibly with noise) is from a single moving view, which is a more practical scenario where a PTZ camera is able to both analyze and move itself. In order to model such configurations, we propose a method to predict the regions of interest by constructing a vector field with Gaussian process regression. The vector field is built on the regression model and covariance function that are modeled with residual terms, and all the motion vectors at any location can be represented by a set of means and variances. The means collectively reflect the motion tendency, whereas the variances quantify the level of confidence of the motion tendency as the motions may be sparse and noisy.

Our contributions in this paper are: (1) A method to generate a stochastic motion field that represents a global motion tendency using Gaussian process regression (GPR). (2) Techniques for predicting important future locations from mean and variance fields computed from the stochastic vector field. (3) An evaluation method for measuring the *goodness* of predicted important regions. We utilize the Jaccard coefficient [14] computed from the fields of view covered by our approach and *actual* camera operators (the base-line comparison). The Jaccard coefficient is an intuitive similarity measure, the size of the intersection divided by the size of the union of two sets. Based on our criterion, we demonstrate that our approach can predict regions of importance quite accurately. We demonstrate the validity of our approach over a very complex data set, which will be made available with the paper.

2. Related Work

There has been some work on automated video capturing/directing that is relevant and worth mentioning. Pinhanez [11] introduced an automatic broadcasting system for a TV cooking show. Since the show was recorded in a studio in which only few panels move with a pre-defined script, there was no need for either the analysis of multiple moving objects or the prediction of regions of importance. Ariki et al. [2] proposed a system that automatically broadcasts soccer games by cropping the field of view of a virtual camera from static wide angle view scenes. Their approach relied mostly on the tracks of the foreground objects, and applied them to the set of rules (*i*.e., each labeled event belongs to a specific camera rule) for smooth transition of virtual cameras.

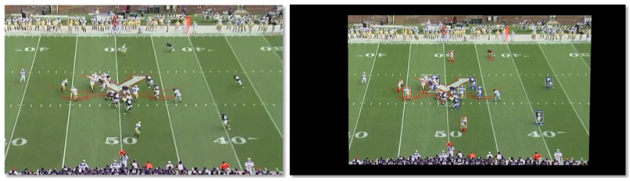

Figure 3. **Registration of each frame onto the reference view for a player tracking**: **Left:** Original view, **Right:** A rectified view used for tracking players.

Recently, Kim et al [6] introduced an approach to measure global tendencies from sparse set of motion with radial basis function interpolation. This approach is used to predict the regions of importance in the scene of the soccer videos captured from multiple-static cameras. This effort significantly motivates our work and we have even tried some analysis on data used in this effort. The differences lie in the fact that this work requires multiple static views for the stable acquisition of motions on the ground, and does not seem to be adequate for the scenario in which a PTZ camera both analyzes the scene and adapts its field of view, which is the case considered in this paper. We compare to this work, specifically the use of Radial Basis Functions (RBF) as compared to GPR flow in this paper.

In this work, we use Gaussian process regression to generate stochastic vector fields from a sparse set of motion tendency vectors. Gaussian process [12] has been widely used in many data regression problems such as modeling motion trajectories and tracking objects [7, 4], and these approaches motivate our work in generating stochastic flow fields.

3. Detecting Regions of Interest

To automatically capture the dynamic sports scenes by identifying where the global motion tendency moves, we first extract motions on the ground plane in the scene. We then generate a stochastic motion field for representing the global motion tendency. Finally, we detect the locations where the motion field converges. Figure 2 shows the overall framework.

3.1. Computing Motion on the Ground

We first register video frames into the known field coordinates using successive local features appearing in each video frame [5]. Then we rectify video frames into a reference frame with successive homographies extracted from the registration. We chose the first frame of each video clip from the data sets as a reference for the registration (Figure 3 (Right)). This rectified video frame is used for extracting the ground motion of each player by applying particle-filter based tracking [8]. We then approximate the motion vector on the ground as the vector between the center of bottom edge of tracked blobs in each consecutive frame. In order to construct the motion field from the extracted motions on the ground, we project all the motions into the overhead-view of the ground field as shown earlier in Figure 1 (Bottom).

3.2. Stochastic Motion Field

Our task is to generate a dense motion field representing global tendency with sparse motion extracted from the scene. Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \Re^d$ be a set of locations of extracted motion, in which d is the dimension of the input motion that we want to model. Each location $\mathbf{x}$ has a set of noisy observed velocity vector components: y_u (the velocity component in the u-axis), y_v (the velocity component in the v-axis), (and optionally y_t for modeling the component in the time-axis). We assume that each velocity component at the location $x \in \Re^d$ follows the regression model $\hat{y} = f(\mathbf{x}) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, *i*.e., Normal distribution.

Gaussian Process Regression. We propose using the Gaussian process regression model, where $f(\mathbf{x})$ is a zero-mean Gaussian process with covariance function $K(\mathbf{x}, \mathbf{x}'') : \Re^d \times \Re^d \rightarrow \Re$. A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution [12]. It is completely specified by a mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ (typically assumed to be $\mathbf{0}$) and a covariance function $K(\mathbf{x}, \mathbf{x}'') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}'') - m(\mathbf{x}''))] = \mathbb{E}[f(\mathbf{x})f(\mathbf{x}'')]$.

If we have training data $\mathbf{D} = \begin{bmatrix} \mathbf{x_1}, & \cdots, & \mathbf{x_N} \\ y_1, & \cdots, & y_N \end{bmatrix}$, the $N \times N$ covariance matrix $\mathbf{K}$ is now defined as $[\mathbf{K}]_{jk} = K(\mathbf{x_j}, \mathbf{x_k})$. We then define the observation vector $\mathbf{y} = [y_1, \ldots, y_N]^T$; $\mathbf{y}$ can be shown as a zero mean multivariate Gaussian process with a covariance matrix $\mathbf{K}^* = \mathbf{K} + \sigma^2\mathbf{I}$. The posterior density for a test point $\mathbf{x}^*$, $p(y^*|\mathbf{x}^*, \mathbf{D})$ is a univariate normal distribution with the mean $\bar{y}^*$ and the variance $\text{var}(y^*)$:

$$\bar{y}^* = \mathbf{k}(\mathbf{x}^*)^T(\mathbf{K}^*)^{-1}\mathbf{y}$$
$$\text{var}(y^*) = K(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*)^T(\mathbf{K}^*)^{-1}\mathbf{k}(\mathbf{x}^*)$$

where $\mathbf{k}(\mathbf{x}^*) = [K(\mathbf{x}^*, \mathbf{x_1}), \ldots, K(\mathbf{x}^*, \mathbf{x_n})]^T$. We use Gaussian Automatic Relevance Determination (ARD) kernel [3] as the covariance function. We select and optimize its hyper-parameters using the limited memory Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimizer [10] by maximizing the marginal log-likelihood of the training data.

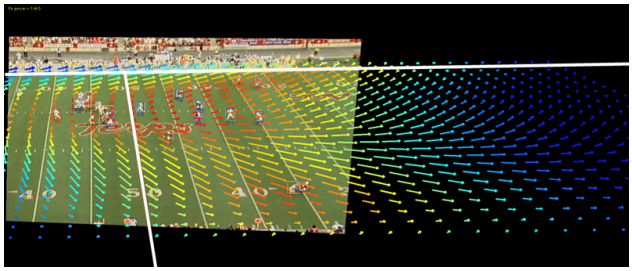

Figure 4. **Stochastic motion field and its certainty field generated from GPR:** The arrows indicate the vectors in the field generated from the motions of players. The colors of the arrows represent the level of certainty. Red arrows have larger certainty level (narrower confidence band) and blue ones have lower certainty level. Therefore extrapolated vectors are more likely to be blue. Bold white lines indicate the references for the ground plane. Note that all calculations were done in the overhead projection.

We can then express the mean flow as a vector field for two dimensional motions, $\mathbf{\Phi}(\mathbf{x}) = \bar{y}^*_u(\mathbf{x})\mathbf{i} + \bar{y}^*_v(\mathbf{x})\mathbf{j} \in \Re^2$ (u, v represent a spatial domain), and for three dimensional motions as $\mathbf{\Phi}(\mathbf{x}) = \bar{y}^*_u(\mathbf{x})\mathbf{i} + \bar{y}^*_v(\mathbf{x})\mathbf{j} + \bar{y}^*_t(\mathbf{x})\mathbf{k} \in \Re^3$ (t indicates a temporal domain) with a variance for each velocity component $\text{var}(y^*_u(\mathbf{x})), \text{var}(y^*_v(\mathbf{x})), \text{var}(y^*_t(\mathbf{x}))$ respectively. Figure 4 shows the generated stochastic motion field using mean field $\mathbf{\Phi}(\mathbf{x})$. In the figure, the certainty level is shown as color values indicating a 95% of distribution (confidence band) with the mean and variances in each velocity of the motion.

Comparison with the Deterministic Motion Field. We note that the mean of the posterior density $p(y^*|\mathbf{x}^*, \mathbf{D})$ can be interpreted as a linear combination of observations $\mathbf{y}$, *i*.e., a *linear predictor*: $\bar{y}^* = \sum_{y_i} \alpha_i y_i$ where $\alpha = (\mathbf{K}^*)^{-1}\mathbf{k}(\mathbf{x}^*)$. At the same time, it can also be written as: $\bar{y}^* = \lambda^T\mathbf{k}(\mathbf{x}^*) = \sum_{\mathbf{x_i}} \lambda_i K(\mathbf{x}^*, \mathbf{x_i})$ where $\lambda = (\mathbf{K}^*)^{-1}\mathbf{y}$, *i*.e., a weighted kernel sum that appears in the RBF approach shown in [6]. The GPR approach is similar to the RBF, except for one key difference. RBF is used primarily for interpolation of known observation values, whereas GPR can operate on noisy observation values. By specifying the mean function $m(\mathbf{x})$ and the covariance function $\mathbf{K}(\mathbf{x}, \mathbf{x}'')$, we can construct confidence bands around each predicted value to quantify the certainty level. Readers can refer to [1] for more details on some key similarities and differences between the RBF and GPR techniques. In our experiments, we validate the effectiveness of confidence bands for identifying convergence locations of mo-

tion trends.

3.3. Detecting Locations of Convergence

Detecting convergence locations of the motion field can guide the movement of the PTZ camera as shown in [6]. The approach propagates and updates a magnitude of a velocity along a motion field. However, this approach often suffers from two problems. First, propagating every vector (regardless of its similarity to actual global motion tendencies) in the field has been shown to be computationally intensive. Secondly, extrapolated velocity vectors with large magnitudes can seriously bias accumulation and yield an unstable localization of converging points.

We make several modifications to address the issues listed above. First, instead of propagating a magnitude of velocity, we *transport* certainty levels computed from GPR. Second, we transport the certainties only for the locations with high certainty levels. We note that confidence bands predicted from GPR are wide for locations that are far away from actual input motion vectors; these locations with extrapolated velocity vectors are otherwise unnoticed under the RBF computation [6], but can be accounted by our new approach. Third, transporting the certainty level requires updating only the last destination point and is computationally more efficient.

We first define an evaluating function $\mathcal{E}(n, \mathbf{x}, \Phi)$, where Φ is a motion field (mean field), $\mathbf{x}$ is a starting location, and n is a number of the iteration. Starting from $\mathbf{x}$, $\mathcal{E}(n, \mathbf{x}, \Phi)$ follows the flow of Φ by integrating predicted velocity vectors at each iteration. For example, if we denote the location of a motion as $\mathbf{x}_i = [u_i \; v_i] \in \Re^2 \; (0 \leq i \leq n)$, the evaluating function $\mathcal{E}(n, \mathbf{x}, \Phi)$ iterates the locations by $u_{k+1} = u_k + \bar{y}^*_u(u_k)$, where $y^*_u(u_k)$ is computed from $\Phi(\mathbf{x}_k)$. We denote the final location returned by the function after n iterations as $\mathbf{x}_n$ (See Figure 5). Through this function, we transport the certainty value ρ (95% of confidence, $1.96\sigma^2$) from $\mathbf{x}$ to $\mathbf{x}_n$ along the field Φ. Therefore, we only add the certainty values at the destination. We evaluate $\mathcal{E}$ at each position in the field with velocity vectors with sufficiently low variance. Figure 5 shows the resulting accumulated distribution of certainties Ψ from the original field Φ. In the following sections, we denote the locations with the values accumulated more than 50% [1] of maximum accumulated values in the field as *merging points*.

3.4. Measuring the Similarity with Field of View

We use the region of the field of view controlled by real camera operators as the baseline comparison (see Figure 6). For each approach to be compared against the baseline, we first construct a convex hull formed by the locations of players and convergence locations detected by the approach. The similarity metric between the constructed convex hull region of an approach and the baseline field of view is chosen to be Jaccard coefficient. We measure the similarity between the field of view decided by camera operators and the region of the convex hulls in the evaluating frame. We repeat this evaluation for each successive pair of the baseline field of view and the convex hull region for an approach as shown in Figure 7 (Top).

There are several reasons for this evaluation; first, we want to validate our hypothesis that the predicted global motion tendency and its merging points (in addition to the locations of moving objects) can be used to adjust the field of view. Second, we want to verify whether the prediction

[1] This threshold is used for outlining the accumulation of lower confidences, and is chosen empirically. Based on our test, more precise threshold from a specific data does not provide a dramatic improvement as the difference between low and high accumulation is usually high.

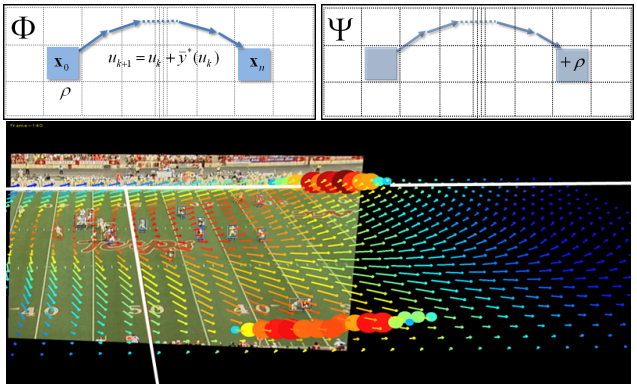

Figure 5. **Certainty transfer through stochastic motion field and merging points:** **Top-left:** The certainty level ρ at a location $\mathbf{x}_0$ is transferred to the location $\mathbf{x}_n$ through the stochastic motion field Φ. **Top-right:** In a separate grid Ψ, the value ρ is accumulated at the location of $\mathbf{x}_n$. Accumulated certainties in Ψ will be used to predict locations of future importance. **Bottom:** Colored circles indicate accumulated certainties from the motion field shown in Figure 4 (red circles with larger accumulations and blue ones with smaller accumulations). Note that we visualized the only locations that have more than 80% of maximum accumulation.

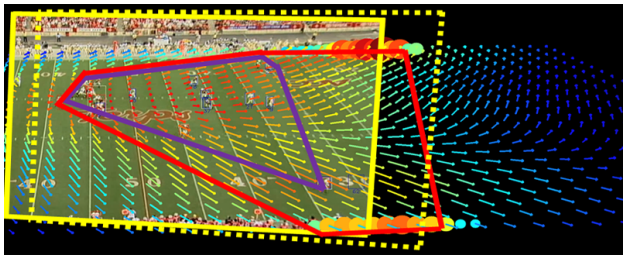

Figure 6. **Evaluation for the comparison of actual camera operator's field of view:** The region with solid yellow lines denotes the camera operator's field of view, whereas the region with dotted yellow lines represent the field view 10 frames later. The convex hull of only the locations of players is shown with purple lines. The region with red lines is decided by the locations of players and the merging points computed by GPR.

based on motion tendency is similar to the field of view adjusted by actual camera operators. We note that the evaluation of the second criterion is not possible under multiple static-view camera approaches. Finally, we want to verify whether the predicted regions of importance (represented as player locations or merging points) computed from each method can be readily deployed *without* additional post-processing methods. The examples of such post-processing methods include a bounding rectangle with margins [2] and linear camera motions [6], which are not suitable for automated *live* application.

4. Evaluation and Results

For our experiments, we have worked with two different data sets. First, we use American football video data taken by real camera operators; the videos are taken from moving cameras and have various PTZ changes. The data set consists of 8 different video clips (cv1 - cv8) from US College football games. This data set is used for the comparison of existing algorithms to actual camera operators' view-adjustment. In addition, this evaluation will measure the similarity between the predicted camera movement by each approach and the actual camera operator's decisions. Secondly, we also use video data sets from static-multi-view videos for comparing our approach with existing methods used for static videos. The data sets consist of several videos from a soccer game used by [6]. We will contact the owners of these datasets for releasing the data sets publicly along with our results.

4.1. Similarity with Camera Operator's View

Graphs in Figure 7 show the average similarity using the Jaccard metric between camera operator's field of view and the region decided by each method (1.0 is ideal). As expected, methods using the motion-field-based prediction outperformed the method using only tracked results on every data set. Among the methods, the 2D GPR-based prediction was better than RBF-based approach, and was even slightly better than the 3D GPR-based approach in most cases. While the 3D GPR approach was shown to be effective in representing 2nd order movement of motion and to be useful for motion recognition [7], the temporal derivatives do not seem to play an important role in discovering global motion tendency to identify the *location* of importance. The projected 2D tendency has sufficient representation for the task.

The GPR-based approach generally outperformed the RBF method. Unlike the RBF interpolation-based approach, the GPR-based approach provides confidence bands at each velocity vector. Using these confidence bands, we can selectively propagate certainty values, whereas RBF interpolation requires iterating over every velocity vector. As a result, the GPR-based approach is less affected by (1) extrapolated velocity vectors than the RBF-based approach, and (2) noise from registration and tracking of the original video data.

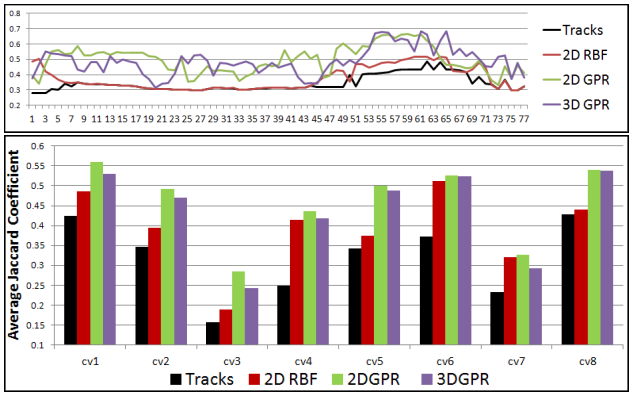

Figure 7. **Quantitative evaluation for the comparison of actual camera operator's field of view:** The values in vertical (y)- axis in both graphs indicate a Jaccard coefficient between a camera operator's region and each computed region, which uses the location of players (black), the 2D RBF (red), the 2D GPR (green), and the 3D GPR (violet) respectively. A graph in the top shows the evaluation over all frames from one sample from our data set (cv5). The bottom graph shows the average of Jaccard coefficient for all the football data sets (cv1 to cv8).

Figure 8 describes a more reasonable comparison in which we compare the region computed from each method with future regions adjusted by camera operators by differing frame offsets from 10 to 40. This evaluation provides a notion of how each method *foresees* the important regions in the scene. 2D GPR-based approach outperforms the other approaches including the method using only tracking information. In addition, because the tracking-based approach uses only the current locations of moving objects, its effectiveness in directing the camera movement (shown via the Jaccard coefficient) drops markedly as the frame offset is increased from 10 to 40.

4.2. Computational Expense

Figure 9 shows the computational expense to perform the RBF-based and GPR-based methods. Both RBF-based and 2D GPR-based methods require the inversion of n by n kernel matrix (generally $\mathcal{O}(n^3)$) and the weighted kernel summations ($\mathcal{O}(n^2)$)[2]. One key difference between the two methods is the additional $\mathcal{O}(n^2)$ computation incurred in evaluating confidence values for the GPR-based method. However, we argue that this additional computation actually helps reduce the overall evaluation time compared to the RBF-based method. While the RBF-based method propagates and updates all the vectors followed by the motion field, the GPR-based method transfers and updates only

[2]Note that there are fast methods for speeding up the matrix inversion and the evaluation [13, 9]. Exploring this further remains our future work.

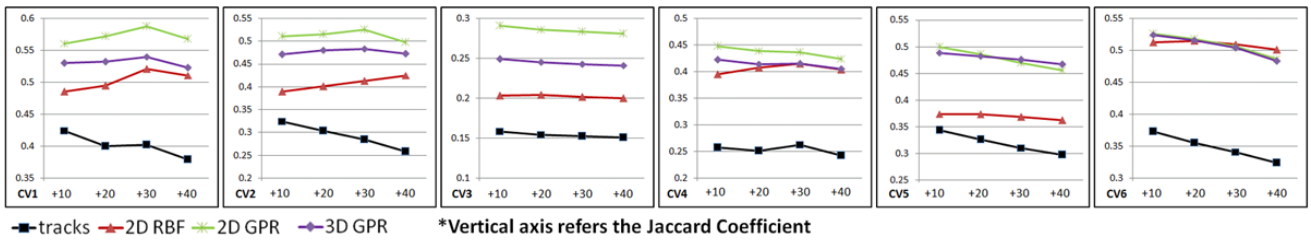

Figure 8. **Evaluation of the future region of camera operator by differing frames:** We evaluated each data set (cv1 to cv6 are shown here) by comparing the field of view of camera operator in future frames with the predicted merging regions at current frame by differing the frame difference from +10 to +40 frames. As shown in each figure, the method using only tracked player locations has lower similarity, and is generally decreasing as it always stay at the current location of players while methods using motion field have larger similarity over different frame offsets. In the results from cv4 and cv6, both GPR-based and RBF-based methods give similar results, while the other data sets show that GPR-based approaches work better. In the two data sets (cv4 and cv6), because the actual region of interests are close to boundary and fewer extrapolated vectors are involved in the prediction, both GPR-based and RBF-based methods give similar results.

the final destination by excluding the extrapolated vectors with low confidence. Therefore, the overall computation needed for the GPR-based method is faster. The 3D GPR-based method requires the formation of kernel matrices of size n^2 by n^2, resulting in $\mathcal{O}(n^6)$ in the matrix inversion and $\mathcal{O}(n^4)$ for evaluating each velocity component along with its confidence value. The 2D GPR-based approach is not only more computationally efficient but also effective in generating qualitative and quantitative results similar to those achieved by the 3D GPR-based approach.

4.3. Qualitative Evaluations

Figure 10 demonstrates qualitative results from both RBF-based and GPR-based approaches in the video from moving cameras. As shown in the figure, our 2D GPR-based approach can predict regions of interest similar to ones implied by the camera movement. On the other hand, the merging points computed from RBF are usually located near boundary regions because of the portion of extrapolated motions involved in the detection.

Figure 11 showcases resulting examples from our data sets (using 2D GPR). The distribution and movement of detected merging points from each example reasonably describe the motion of the actual camera. Note that the distribution of merging points correlates the zoom of the camera because the separate merging points can be interpreted as a larger field of view of camera for covering both regions. We can also measure how relatively far the converging points are from the camera from the homography computed for each frame. From the measurement, we can also predict the tilting of the camera view from the location of merging points. We can also see the merging points move to the direction where camera moves.

To give additional comparisons with existing methods in static-view, we also applied our approach to the data sets captured from multiple-static cameras, which were used in [6]. Figure 12 shows some qualitative results showing

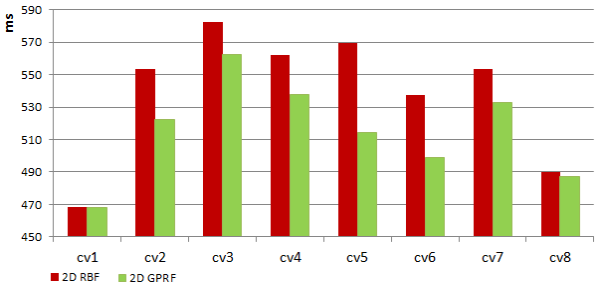

Figure 9. **Computational expense:** Red bars indicate the computational expenses of 2D RBF-based approach. Green bars indicate those of 2D GPR-based approach. The expenses for 3D GPR-based approach vary from 8000 ms to 9500 ms. x axis refers the each data set, and y axis refers the millisecond.

the comparison between our approach and the RBF-based method. As shown in each sequence in the figure, the results from both approaches (RBF and GPR) do not look too different unlike the test using moving camera. First, because the motions on the ground are relatively stable (as they are extracted from static multi-view), the error handling in GPR does not play an important role. Second, as the region covered by multi-view is smaller (covering only a half of the field with well-defined boundary conditions) than the data sets from moving cameras, there are fewer extrapolated vectors in the scene. Therefore, a velocity propagation without filtering the extrapolated velocity may be enough for identifying the merging locations.

5. Conclusion

We have shown that the prediction of the region of interests from stochastic field using Gaussian Process Regression provides robust results even with noisy motions from moving cameras. We demonstrate that the GPR-based approach can model the camera motion performed by actual camera operators more closely. In our future work, we will work on (1) improving the scalability of the code-base by

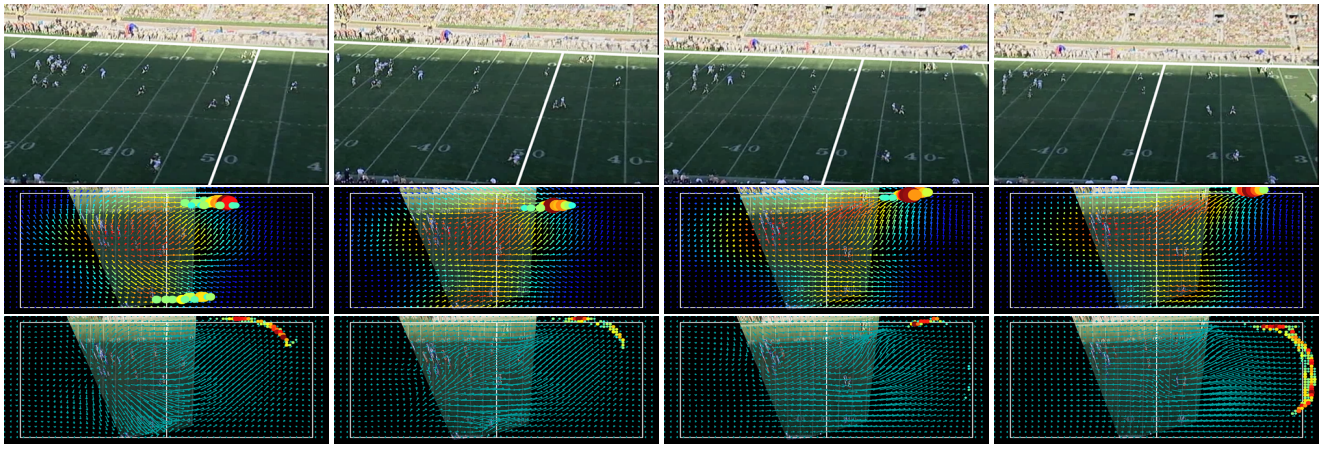

Figure 10. **Qualitative evaluation between RBF method and GPR method: Top row** shows the transition (PTZ) of the original views adjusted by the camera operator. To give a better understanding of how the original view moves, we added white lines to represent the 50 yard line and the upper boundary of the ground field. The view is being panned to the right direction, and zoomed out. **Middle row** shows the registered over-head projection of the stochastic motion field, and merging points computed from the 2D GPR method. **Bottom row** represents the result from the RBF based approach. Note that the merging points in RBF method are often concentrated near the boundary of the field because the computation of the merging points are highly affected by the extrapolated vectors in RBF (see the last example of the third row).

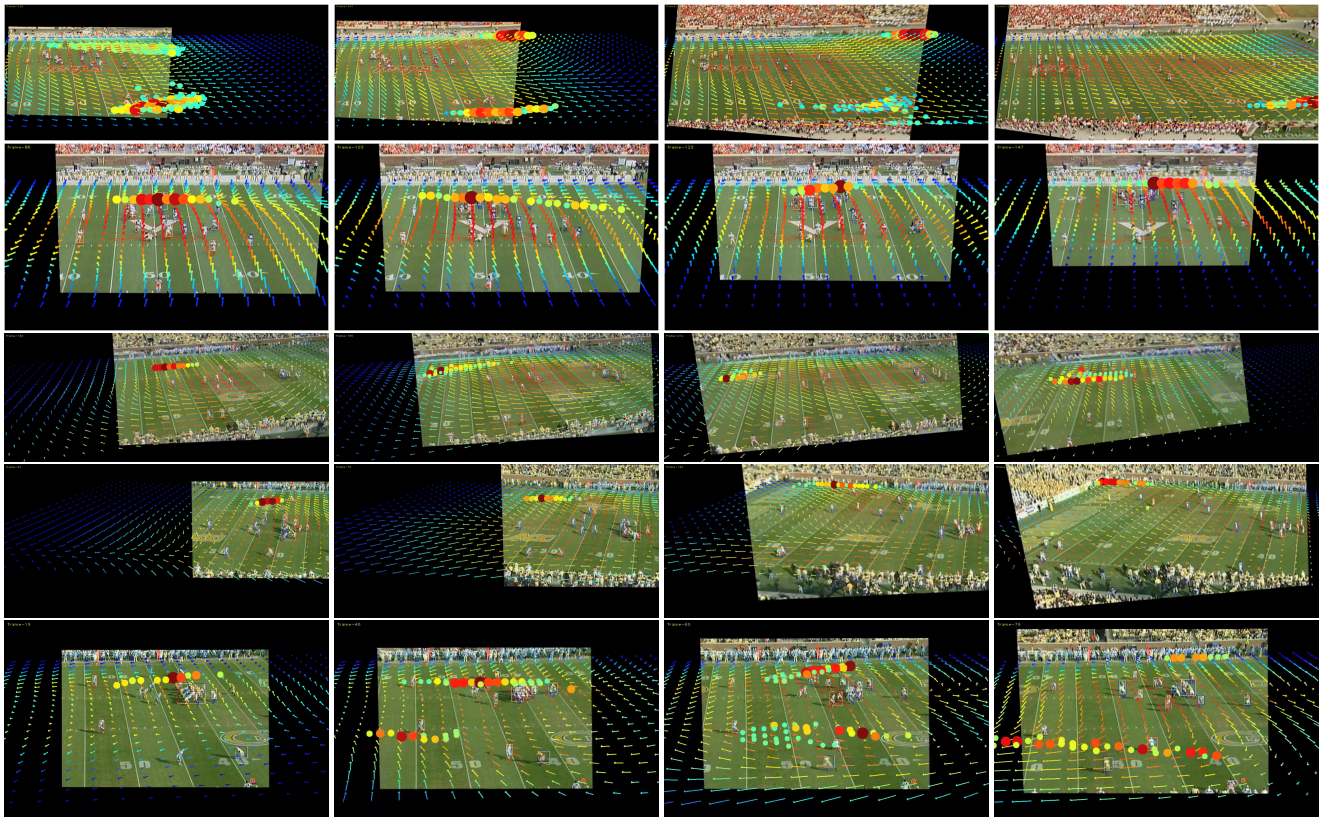

Figure 11. **Additional comparison of our results (with GPR method) and actual camera motion:** Sequences in **1st (top) row** show the example of pan (left to right) and zooming out motion. Merging points detected both in top and bottom regions of the field (zooming out), and move to right. **2nd row** shows the sequences of the camera motion with zooming in and tilting up. **3rd and 4th rows** demonstrate panning sequences (right to left). **5th row** shows the sequences of zooming out, and detected merging points are shown in top and bottom regions.

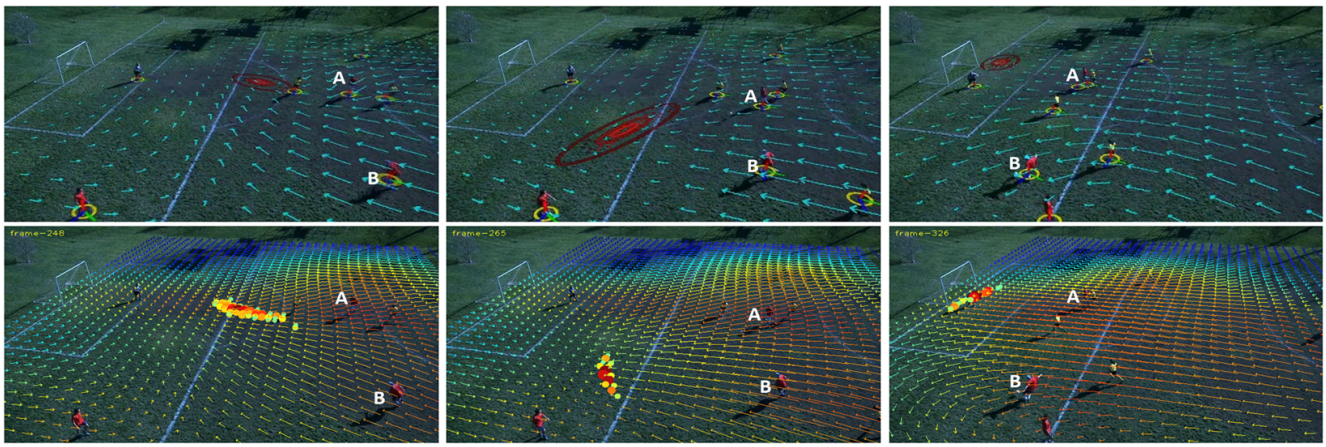

Figure 12. **Qualitative comparison between RBF method and 2D GPR method:** The sequence of scenes in **top row** show the result from RBF method (the images are captured from the demonstration video of [6]; the red contour indicates the location where the motion field merges. The scenes in **bottom row** show the result of our approach using GPR. The location of where the motion field merges is shown with circles in which the colors represents the amount of accumulated transferred certainties. For each row, first scene describes the merging location lies in front of player **A** who dribbles the ball. In the second scene, merging location describes the location where the other offender **B** will receive the ball. In the last scene, results shows the location for the other pass.

utilizing GPGPU-based acceleration since most computations consist of matrix-matrix products (an embarrassingly parallelizable primitive); (2) applying our approach for controlling actual robotic cameras in real-time.

References

[1] K. Anjyo and J. Lewis. RBF interpolation and gaussian process regression through an RKHS formulation. *Journal of Math for Industry*, 3(6):63–71, 2011. 3

[2] 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. 2, 5

[3] W. Chu and Z. Ghahramani. Preference learning with gaussian processes. In *Proc' of the ICML*, pages 137–144, 2005. 3

[4] D. Ellis, E. Sommerlade, and I. Reid. Modelling pedestrian trajectories with gaussian processes. In *International Workshop on Visual Surveillance*, pages 1229–1234, 2009. 2

[5] R. Hess and A. Fern. Improved video registration using non-distinctive local image features. In *2007 IEEE CVPR*, pages 18–23. IEEE Computer Society, 2007. 2

[6] K. Kim, M. Grundmann, A. Shamir, I. Matthews, J. Hodgins, and I. Essa. Motion field to predict play evolution in dynamic sport scenes. In *IEEE CVPR*, pages 840–847, 2010. 2, 3, 4, 5, 6, 8

[7] K. Kim, D. Lee, and I. Essa. Gaussian process regression flow for analysis of motion trajectories. In *Proceedings of IEEE ICCV*. IEEE Computer Society, November 2011. 2, 5

[8] E. Koller-Meier and F. Ade. Tracking multiple objects using the condensation algorithm. *Robotics and Autonomous Systems*, 34(2-3):93–105, 2001. 3

[9] D. Lee, R. Vuduc, and A. G. Gray. A distributed kernel summation framework for general-dimension machine learning. In *SIAM International Conference on Data Mining 2012*, 2012. 5

[10] J. Nocedal and S. Wright. Numerical Optimization, Series in Operations Research and Financial Engineering, 2006. 3

[11] C. Pinhanez and A. Bobick. Intelligent studios: Modeling space and action to control tv cameras. In *Applications of Artificial Intelligence*, pages 285–306, 1997. 2

[12] C. Rasmussen. Gaussian processes for machine learning. In *Book, MIT-Press*. MIT-Press, 2006. 2, 3

[13] A. Smola and P. Bartlett. Sparse greedy gaussian process regression. In *Advances in Neural Information Processing Systems 13*. Citeseer, 2001. 5

[14] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005. 2