Appears in: *Proceedings of the Fifth International Conference on Computer Vision*, pp. 672-678, June 1995

Closed-World Tracking

Stephen S. Intille Aaron F. Bobick

Perceptual Computing Group
MIT Media Lab
Cambridge, MA 02139
(intille | bobick@media.mit.edu)

Abstract

A new approach to tracking weakly modeled objects in a semantically rich domain is presented. We define a closed-world *as a space-time region of an image sequence in which the complete taxonomy of objects is known, and in which each pixel should be explained as belonging to one of those objects. Given contextual object information,* context-specific *features can be dynamically selected as the basis for tracking. A context-specific feature is one that has been chosen based upon the context to maximize the chance of successful tracking between frames.*

Our work is motivated by the goal of video annotation – the semi-automatic generation of symbolic descriptions of action taking place in a contextually-rich dynamic scene. We describe how contextual knowledge in the "football domain" can be applied to closed-world football player tracking and present the details of our implementation. We include tracking results based on hundreds of images that demonstrate the wide range of tracking situations the algorithm will successfully handle as well as a few examples of where the algorithm fails.

1 The problem

Video annotation is the task of generating descriptions of video sequences that can be used for indexing, retrieval, and summarization. It is different than conventional image understanding in that one is primarily interested in what is *happening* in a scene, as opposed to what is in the scene.

Many video annotation domains require documenting the interactions between people and other non-rigid objects against non-static backgrounds and in unconstrained motion. In this paper we describe a technique that incorporates contextual information into low-level tracking to successfully recover the trajectories of such objects.

The method can be used to track objects found in annotation tasks such as describing city street intersections, sporting events, air traffic, pedestrian mall traffic, cell movements from quantitative fluorescence microscopy, groups of animals and meteorological objects. One real-world annotation task which is currently performed manually by professional sporting teams is play labeling. With an eye on this problem, therefore, our test domain is football player tracking.

Figure 1 illustrates the large amount of pan and zoom present in a typical game film of a football play, where the cameraman must keep as many players as possible in the field of view.[1] The pan rate is such that it is not uncommon for the image to shift about five pixels between two frames sampled at thirty frames per second, and the wide-angle focal length induces a fairly substantial barrel distortion when the camera is zoomed out.

A brief analysis of the domain and the imagery reveal why this is such a challenging problem. A typical play lasts about ten seconds yielding 300 frames of deinterlaced, 700x240 video. Once the play has been digitized and deinterlaced, players range in size from about 20 by 20 pixels to about 10 by 10 pixels, depending upon the setting of the camera. A sampling of various players at different times during a play is shown in Figure 2. The players move rapidly and change direction unpredictably, violating the smooth motion assumption of many tracking algorithms. Additionally, accurate motion estimates are difficult to obtain because they are compounded with camera motion and it is hard to define a reference point on a non-rigid, blob-like object from which to compute velocity. Finally, football players frequently collide. Color data, discussed in [7] provides more information that can be used for tracking, but it does not fundamentally solve the tracking problem.

In this paper, we begin by presenting some previous approaches to tracking and we argue that they are severely inadequate for the low-resolution, amorphous, multiple-object tracking required. We next define a "closed-world" as a region of space and time in which all the objects present in that region are known; what is unknown and needs to be estimated is the state and position of each of the objects. The advantage of a closed-world is that knowledge of which

[1] The entire sequence, results described in this paper, reference [7], and more current work can be viewed at: http://www-white.media.mit.edu/vismod/demos/football/football.html.

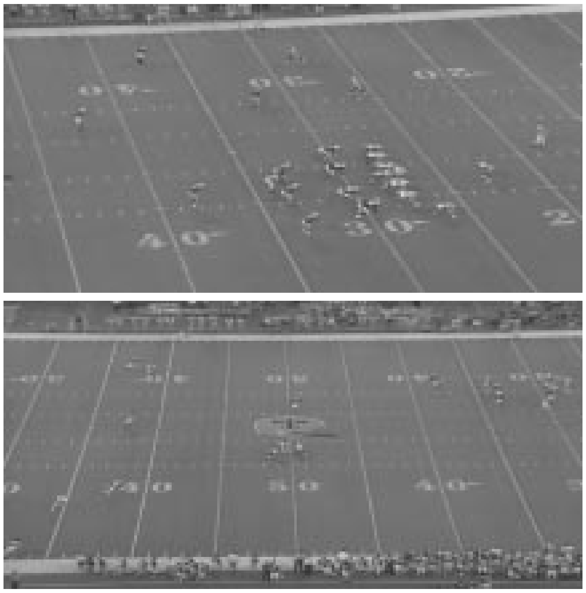

Figure 1: Two 700x240 deinterlaced frames of a typical pass play from pan/zoom video of a college football game.

objects are present can be used to select what information is most powerful for tracking each of the objects of interest. We describe how a closed-world interpretation can be used for tracking in dynamic scenes and the components of the theory are presented using examples drawn from the football domain. Finally we present results of tracking football players that demonstrate the application of the technique on examples consisting of hundreds of frames.

2 Previous approaches for tracking

Visual tracking usually employs one or some combination of the following methods: (1) correlation or adaptive template correlation[19] that can create "drifting templates," (2) energy-based deformable models[2, 21] that require good support from data and slowly changing objects (3) differential motion estimators[15, 6, 3] that use smooth or planar motion models, (4) feature-based edge or blob trackers[5, 14, 8] that require meaningful boundary extraction, and (5) model-based tracking techniques[10, 8, 20, 9, 17] that use 2d and 3d geometrical models of rigid objects combined with recursive smooth-motion estimators. The difficulties of applying these methods to the football tracking problem are discussed in more detail in [7].

The use of high-level domain knowledge is not widespread in the tracking literature. Toal's work touches on the idea that non-geometric information can be used to improve vehicle tracking[18]. Vehicles are constrained in different ways depending upon their environment, and Toal has suggested that this information might be used in a video understanding system. Allen's bird counting system [1] illustrates that recognizing the same type of object in two different contexts (grounded and flying birds) may require two entirely different vision methods. Fu's shopper system[4] and Prokopowicz' active vision tracker[13] also select features based upon the context in a dynamic situation. Finally, Rosin's outdoor security system is unusual because he has specifically invoked non-geometric contextually-dependent information about an outdoor scene to improve the system's tracking and recognition capabilities[14].

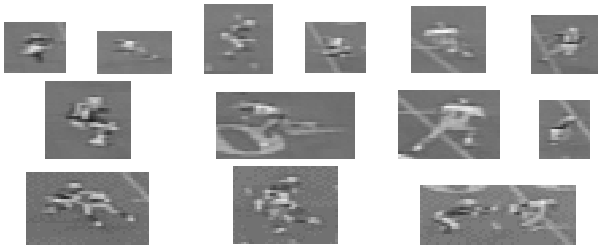

Figure 2: Offensive and defensive players and officials clipped from the football imagery. The objects are difficult to model, especially when colliding) due to the inherent complexity of the object shapes and the poor spatial resolution.

3 Closed-worlds

This section defines "closed-worlds" and describes how they can be used to select and limit the contextual knowledge appropriate for a given tracking situation.

3.1 Context and closed-worlds

The task of tracking objects in a complicated domain such as football requires using some type of knowledge about the world. Limiting the tracking system to a particular domain establishes which body of knowledge is relevant; for tracking football players all knowledge about the field, the rules, the strategy, and the tendencies may reduce the uncertainty inherent in the tracking problem. However, that raises the problem of deciding which information is important at each time instant.

Context is one way of addressing the knowledge-selection problem. For the work we present here we consider the context of a tracking problem to be a boundary in the space of knowledge — a boundary outside of which knowledge is not helpful in solving the tracking problem. Continuing the football example, a context would be something like "a region of the field near the upper hashmark on the 50 yard line that contains two players, one offensive and one defensive." This context is quite specific and is likely to determine the way that vision processing tools are selected and the scene is analyzed.

To use context effectively, we propose using a *closed-world* assumption. A closed-world is a region of space and time in which the specific context is adequate to determine all possible objects present in that region. For the above example, the closed-world contains the two players, the positioned hash-marks and yard-line, and grass. The internal state of the closed-world — e.g. the positions of the players — however is unknown and must be computed from the incoming visual data. Visual routines for computing the internal state can be selected using the context-restricted domain knowledge and any information that has already been learned about the state within the world from previous processing. Closed-worlds circumscribe the knowledge relevant to tracking and therefore reduce the complexity of the tracking problem.

We note that a few other authors developed systems that use contextual information and approaches similar to our closed-worlds. Nagel[12] has hinted at using a closed-world assumption when building systems that extract conceptual descriptions from image sequences. He speculates that one way to improve motion recovery is to exhaustively model all types of motion expected within the given domain. Further, he suggests that a description of a scene will require describing the intentions of the objects in the world.

The Condor system designed by Strat[16] uses the output of many simple vision processes and local context in the scene for recognition of outdoor imagery. The Condor system "treats objects as component parts of larger contexts from which they cannot be separated;" objects have "no independent existence." Strat notes that it is easier to design visual routines that work within some specified context than to construct general purpose algorithms.

Finally, Mundy's MORSE system[11] will operate using a closed-world assumption that all data in a modelboard scene should be consistent with all the rules and objects known to exist in the domain. MORSE assumes a simple explanation for the closed space and then gradually works up to the most complicated examples. Mundy suggests that strong evidence of occlusion cannot be found in an open-world.

3.2 Entities in a closed-world

Two types of entities exist in a closed-world, *objects* and *image regions*. Objects are the physical things in the real world scene that the system must monitor in order to develop a useful interpretation. The knowledge of the domain dictates how objects can interact and is independent of how the scene is captured for vision analysis. Image regions are the image data, or the objects projected onto the image plane.

3.2.1 Objects

From a computer vision standpoint, not all objects are created equal. Some objects, typically man-made structures, are well-defined by geometrical measurements. Others, like most biological matter, are blob-like and more difficult to describe precisely. The degree to which an object can be precisely specified helps determine the type of visual processing appropriate for that object. In the football domain, we group the objects found in closed-words into three categories: precise, approximate, and amorphous.

Precise models can be modeled analytically and are common in the computer vision literature. Examples from the football domain are geometric field objects like lines and hashmarks. *Approximate* models are those whose specifications are not geometrically precise; such models are far more common in many applications. While the position and size of the lines and numbers on a football field are exactly specified, the actual font for the numbers is not fixed. Other field markings (such as the mid-field decal) are completely arbitrary. While there may be no simple way to obtain an exact geometrical specification of these objects, it is possible from the football video to reconstruct approximate pixel-based models for the given field (details available in [7]). Finally, *amorphous* objects are those that cannot be well-defined visually. In the football domain, the players are the amorphous objects since they change rapidly over time in complex ways that are hard to model, especially given the low-resolution, discretized data. What can be known about these objects is not their visual geometric structure but their characteristics, such as light-colored jerseys. This type of knowledge can be incorporated into tracking algorithms if the algorithms know which information to consider when deciding how to track an object. The closed-worlds define which knowledge is relevant.

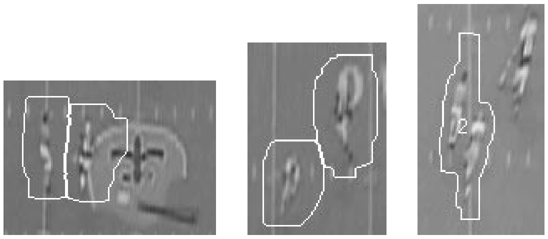

Figure 3: Five different closed-world regions. The rightmost closed-world contains two players and line and hashmark objects.

3.2.2 Image regions

Objects in a closed-world are projected onto the image plane into a *closed-world image region.* Because image projection preserves object identity, the objects present in the closed-world region are known. However, their distribution among the pixels of the image region is unknown and needs to be computed if the objects are to be tracked to the next frame of an image sequence.

Several closed-world image regions are shown in Figure 3. The regions contain turf, a line, hashmarks, one or two players, and/or part of a logo object. Ideally, every pixel in the image region should be *explained* by some closed-world object, and thus used to track the object to the next frame. Computing this explanation requires the use of visual processing, and the selection of processing routines should be based upon the context of the closed-world.

3.3 Isolating closed-worlds in a dynamic scene

If a player is running down the field, isolated from any other players, then the only knowledge a tracking system would need to know is related to the player himself and any nearby "field objects." Together they form an appropriate closed-world. However, when that player moves close to another player, the closed-worlds of each player must be merged into one closed-world; the action of one player may affect the other or the spatial distance between the players may be too close for vision algorithms to interpret without additional domain knowledge.

The above example is designed to motivate the use of *independence* in determining the boundaries of closed-worlds. When local movements and visual interpretations of an object are independent of all other objects, that first object can be analyzed within its own closed-world. When two objects are interacting, however, a single closed-world must contain them both. Without considering all interacting objects simultaneously, the vision system cannot properly determine which types of processes are best suited for analyzing the closed-world events.

For the local tracking analysis in the football domain, object proximity can be used to identify independent closed-world boundaries. We can assume that if two objects are not physically near each other they will not influence each other in any way that a tracker must consider when only tracking an object from one image sequence frame to the next. In section 4.3 we describe one of two mechanisms we have implemented for finding closed-worlds in the football domain. Both are based upon defining image regions known to contain as few objects as possible while ensuring that each tracked object is wholly contained by some closed-world.

3.4 Selecting context-specific features

For robust tracking in a complex scene, a tracker should understand the context of the current situation at a particular time well enough to know which *features* of an object can be tracked from frame to frame and which features cannot. By feature we mean some image-based descriptor that can be localized in a given image. One example is a template containing all of an object's pixels; another is a motion vector template for part of the object. In a dynamic scene, the features of an object that can be reliably tracked are likely to change as the object's interactions change. Feature trackers, therefore, should be selected according to the context in force during the time they are being used.

Closed-world analysis provides a complete description of closed-world image regions. By knowing which objects are present in a closed-world, a tracking system can select features which are most likely to be reliable in separately tracking each of them. Knowledge of the player types (e.g. one is offensive, one is an official) and object types (numbers, grass) can be used to select maximally discriminating pixels. Examples of such context-specific assignment will be given in the next two sections.

4 Tracking football players

In this section we describe the application of the closed-world analysis to the problem of tracking football players. The basic algorithm is:

1. The positions of the players and of the field objects are initialized.
2. Closed-world image regions around the players are computed for the current frame.
3. Each pixel within each closed-world image region is assigned to one of the objects within its closed-world.
4. Context-specific features are used to construct templates for tracking each player in the closed-world.
5. Players are tracked to the next frame using the templates. Go to 2.

In this section we present the details of each of the above steps when tracking single players. In Section 5 we extend the method to multi-player tracking where steps 3 and 4 are more complicated and context-specific feature selection is more important.

4.1 Non-player objects

All non-player objects in the football domain are "field objects." We have modeled these objects using bitmaps and intensity histograms. Each type of object is represented slightly differently, depending upon how well the object's bitmap and histogram could be estimated or recovered. The helmet logo, for example, is modeled with a rough intensity bitmap and an allowable intensity variation range but the turf is modeled only with an intensity histogram.

In order to know the spatial relationship between these field objects and the players as the players move about the field, we converted the original image sequence into a rectified one in which the square grid lines of the field appear as squares in the image, as shown in Figure 4-a. This is simple to do using a homographic transform (perspective four-point transform). This rectification was achieved automatically by tracking many line intersections and computing a least squares solution. Details can be found in [7]. Once this is accomplished, at any player position we know which field objects are nearby.

4.2 Initialization

Before tracking can begin, the position of each player in the original imagery must be identified so that closed-worlds can be localized and so that features can be chosen for tracking. For the results shown here, the closed-world initialization (first frame only) is performed manually by marking one point on the center of each player's torso and identifying the type of player. That point is used as the initial position of the template.

4.3 Isolating closed-worlds

Closed-world boundaries for tracking can be defined using independence. For football, local independence can be identified using spatial proximity and isolation; objects that are near one another should be considered simultaneously when tracking. We have implemented two techniques for determining such regions, both of which have been used for closed-world tracking. The first, not presented, is based upon variance in image intensities and exploits the uniformity of intensity is most regions of the field. The second, described here, uses motion differencing on the rectified imagery.

Significant lens distortion and error in camera-motion removal prevents simple background subtraction. Instead, spatio-temporal operators are used to compute a smoothed temporal derivative over the image sequence. Morphological erosion and dilation operations are then used on each frame, producing contiguous blobs. Unfortunately, the morphological operations also magnify errors resulting from rectification jitter, as shown in Figure 4-b.

To ensure that motion blobs completely contain player objects, all blobs that are identified as containing players are simultaneously "grown" outward from the edge of the motion blob contour. The expanded regions can touch but not overlap, as shown by the regions around the center players in Figure 4-c.

When players are locally spatially independent, their motion difference blobs will not merge and can be be used to define the closed-world region around a player. If no motion blob exists in the region around player, the player

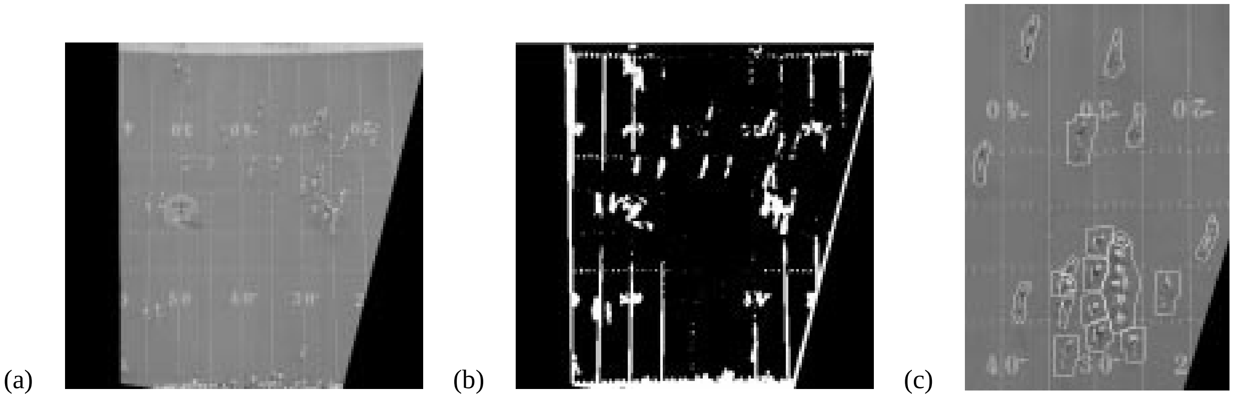

Figure 4: (a) A rectified image frame. (b) Initial object position and closed-world boundaries. (c) Stabilizing the imagery is difficult due to lens-distortion and line-finding errors, as is highlighted by motion blob detection using the rectified imagery.

is assumed to be static and the player's previous closed-world is used at the current frame. For the details of the motion-based closed-world region finding algorithm, see [7].

4.4 Pixel assignment and template formation for isolated players

The goal is to construct a template of the player using information about other objects in the closed-world. We do so by assigning pixels to either field objects or to the player where the method by which we identify player pixels changes based upon what other objects are nearby. The pixels identified as player pixels are used in a template that is matched to the next frame.

We construct the player templates as follows: Each non-player object known to be in the closed-world from the global rectification process is projected onto the closed-world image region. At each pixel in the closed-world image region, the algorithm checks if there is any type of field object within a small spatial region. If so, the pixel intensity is checked to see if it falls within an allowable range for each of the candidate field objects. If it does, then the pixel in the closed-world is marked as "don't care." This processing is used for all field objects – turf, lines, hashmarks, numbers, arrows, and logos.

A closed-world is shown in Figure 5-a, and bitmap representations of the objects in that closed-world are shown in Figure 5-b. The actual models used for all these objects are slightly different (i.e. the histogram ranges and variances are all different). The objects are used to mark "don't care" pixels. The final "player pixel" template is shown in Figure 5-c. This template is the context-specific feature used to track the player.

We note that each pixel removal decision is made independently. There is no restriction stipulating that a given field object can only cause a certain number of pixels to be removed. Any pixel that could reasonably be part of a field object based on its spatial location and the model of the field object is removed and not used for tracking. This algorithm is simple but powerful, since it does not require that our models of objects like numbers and logos be exact, and objects in the closed-world can all be represented differently. Further, it allows for some error in the rectification process. As long as the positions of the objects in the closed-world are approximately known (either from previous tracking or from some global information like the field model), the majority of pixels that remain after the non-player object removal will belong to the tracked player.

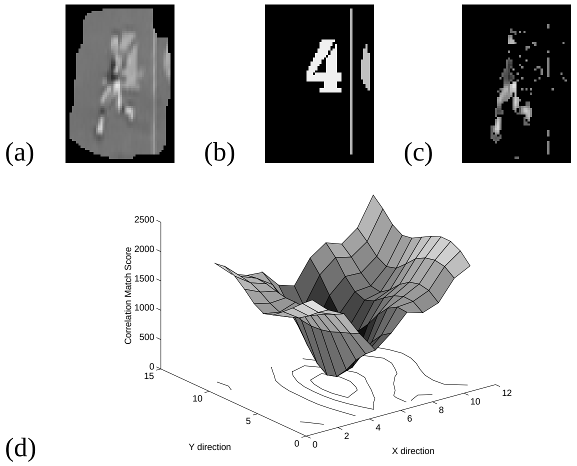

Figure 5: (a) Closed-world image region. (b) Objects in closed-world. (c) Remaining template after pixels are removed by being "possibly-explained" by the objects. (d) A typical correlation match score for the next frame.

4.5 Template tracking

Once a template has been constructed using the closed-world, it is matched to the next frame using correlation. Matching occurs over a small region in the next frame centered around the old player position. The rectified imagery is used for matching since camera motion has been removed. Figure 5-d shows a typical matching score for a template in the next frame. There is a clear correlation peak, despite the small number of pixels in the template. Since features that might be incorrectly matched due to non-player objects have been removed, the template is matching only "player features" and a few erroneous pixels. As long as the majority of the pixels in the template are truly player pixels, the template will not drift off the player and onto field objects.

4.6 Single-player results

The algorithm described has been tested on one nine second football play which consists of 270 frames; for this section we use subsequences that have isolated players running over field objects. There are fourteen such test sequences with an average length of 113 frames and a maximum length of 240 frames. The test sequence, some frames of which are shown in Figure 1, has significant camera motion and zoom.

A simple adaptive template tracker, described in [7], will tend to drift from the player as the player moves over field objects, as shown in Figure 6-a. In comparison, Figure 6-b and Figure 6-c illustrate 200 frames of successful tracking of the same player using the closed-world technique. Because the template does not contain many contaminating pixels from the background objects, the field markings do not come to control the behavior of the template. Finally, in Figure 6-d the tracker is quickly pulled off the offensive player around frame 215 because of interference by the defensive player in the closed-world. In section 5 we will address this type of interaction.

The method also performs well on more complicated examples where players change direction quickly and run over field numbers. Erratically-moving objects are problematic for Kalman filter based trackers that estimate velocity. Figure 6-e and Figure 6-f show the result of tracking the player for 230 frames, where the player stops and changes direction and runs over field objects. The context-specific template succeeds by only tracking the parts of the player that are distinguishable from the objects he occludes. Further, since no assumptions have been made about smooth velocity, the template can capture the player's sharp change in movement.

A difficult field object for the tracker is the helmet field logo. However, the algorithm can successfully track players running over the helmet despite the similarity between the helmet intensities and the player. See [7] for this example and more results.

The closed-world tracking, as described in this paper, will fail when (simultaneously) models are imprecise, spatial resolution is low, and the player being tracked is unusually close in appearance to some nearby object. One example is shown in Figure 6-g. Here the player briefly turns in such way so that he is almost entirely "white" as he crosses a "white" number on the far side of the field. The tracker mislabels too many of the closed-world region points as belonging to the "zero" object, and the template loses the player. In the example play, two isolated-player paths failed to be tracked successfully.

5 Multi-player closed-worlds

The failure of the single player tracker in Figure 6-d occurred when a second player encroached in the closed-world of the first, violating the single player closed-world assumption. When two players are in a single closed-world, therefore, the algorithm should use information about both players to select and track the features that maximally distinguish the two players from each other and from all other objects.

To be complete, every object type in a closed-world should have a function that defines which features distinguish that entity from every other possible closed-world entity. When the two entities are in the same closed world, the features that are most likely to distinguish the two objects are extracted from the closed-world image region and then used for tracking to the next frame.

For two player closed-world tracking, we use pixel intensities as our discriminating features where we weight pixels based upon the most distinctive intensity ranges of offensive and defensive players and the officials. For details, see [7]. In short, the algorithm works as follows. When players are in separate closed-worlds and closed-world image region partitioning is good, all "player pixels" are used for tracking as described in section 4. When two player closed-worlds merge into one larger closed-world, small but distinctive intensity features are selected based upon the type (offensive, defensive, official) of the player objects in the closed world. Those regions are tracked until the closed-world splits into two, when the tracker reverts back to the more robust single-player tracking method. The two player tracking does not handle full occlusion.

The error of Figure 6-d is avoided when the multi-player technique is employed. Both players are tracked simultaneously and when their closed-worlds merge context-sensitive features are selected and tracked correctly. Figure 7 shows a two-player tracking example where a bounding box has been drawn around the active pixels. In Figures 7-a and 7-e the players are being tracked using the single-player tracker. However, in the intermediate frames only highly distinguishing pixels are automatically selected to be tracked – the light back of the offensive player and the dark hat of the official.

6 Summary

In this work we have defined *context* as a boundary in the space of knowledge, and we have used the notion of a closed-world to contextually restrict the type of knowledge relevant for locally tracking an object. Our closed-world tracking algorithm performs well tracking complex objects, even when object motions are not smooth, small, or rigid and when multiple objects of different types are interacting. The algorithm was tested with real video taken from a panning and zooming camera. Extensions to this work will include more quantitative testing of the algorithm's performance and improvement of the multi-player tracking. In addition, we plan to apply the technique to other domains like hand and body tracking.

In Figure 8 we have overlayed some of the recovered player paths on a background image of the field obtained using median filtering over the entire image sequence. These paths are the type of input that will be used by a play understanding video annotation system we are developing.

References

[1] P.E. Allen and C.E. Thorpe. Some approaches to finding birds in video imagery. Robotics Institute Technical Report 91-34, Carnegie Mellon University, Dec. 1991.

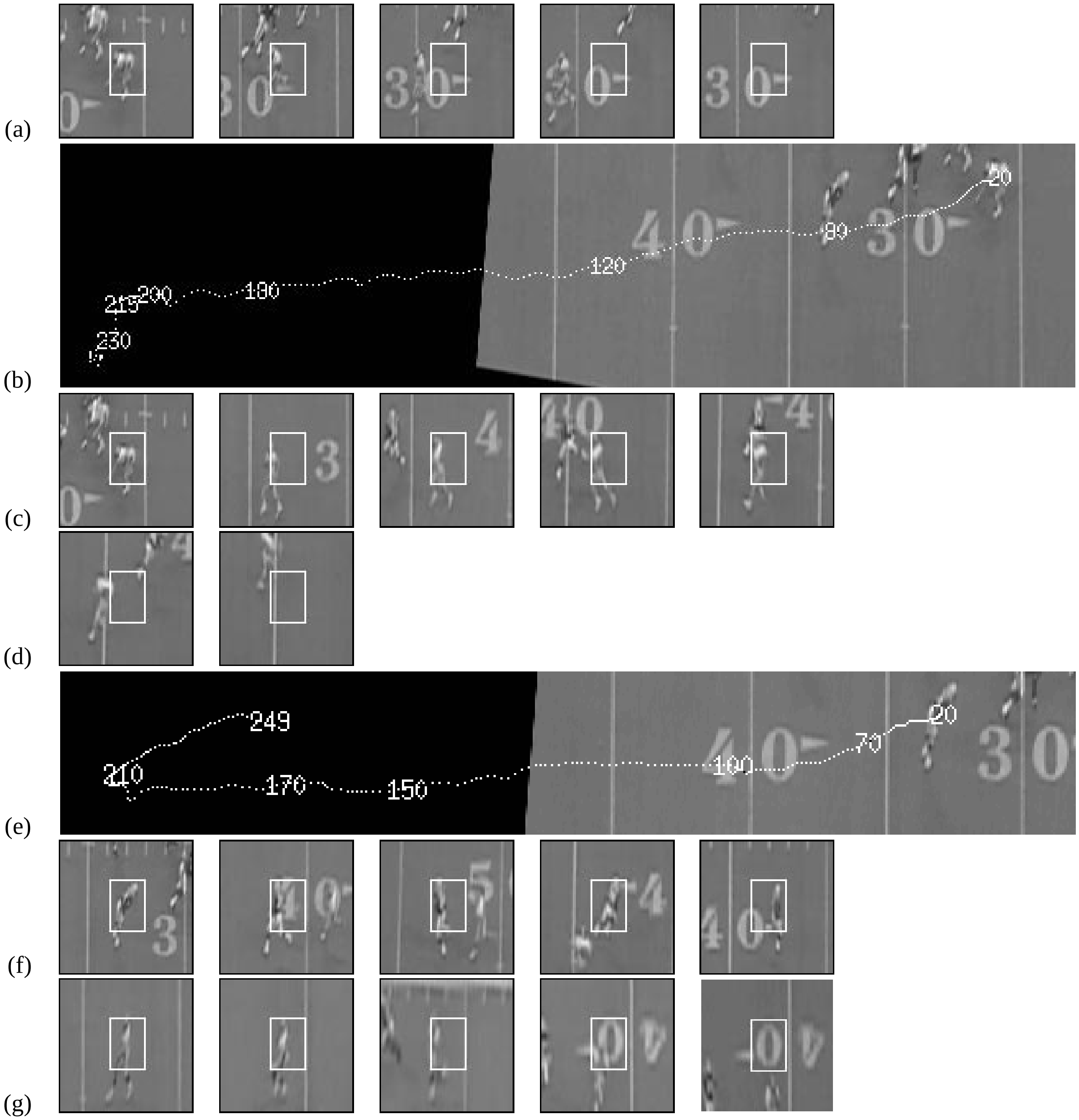

Figure 6: Examples of some tracking successes and failures, as described in the text. (a) Failure of a simple correlation tracker due to background objects. (b,c) Successful closed-world tracking through frame 200, despite path over field objects. (d) Closed-world tracking failure (after frame 215) due to second player which is corrected by the multi-player closed-world algorithm. (e,f) Successful closed-world tracking or rapid change in motion without using motion estimation. (g) Failure of closed-world tracker due to several innacuracies occuring simultaneously.

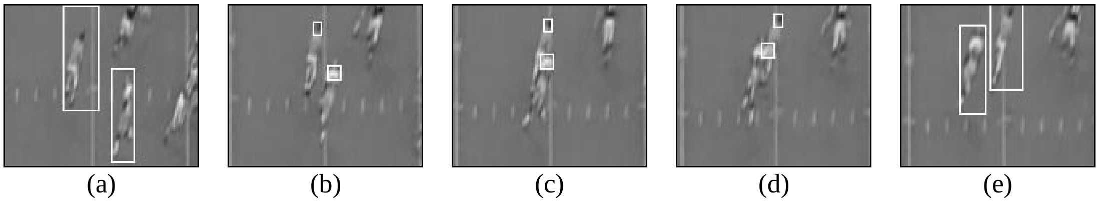

Figure 7: The offensive player and the official are tracked successfully as they move past each other. The white bounding box indicates the region from which pixels were selected to construct the context-sensitive template.

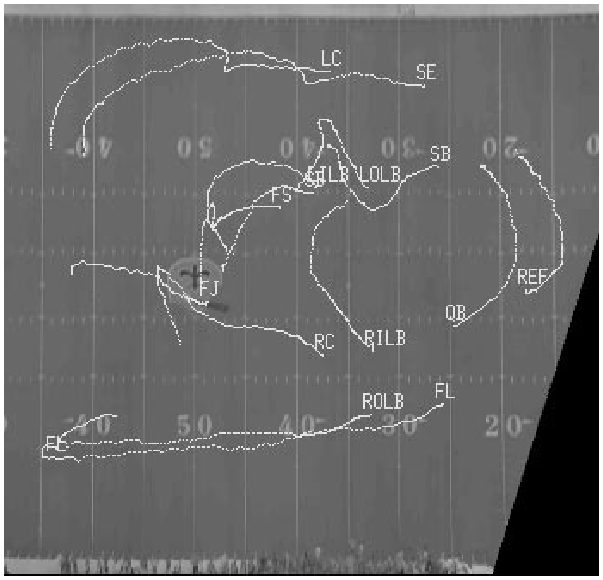

Figure 8: All recovered paths overlayed on an image of the field. These paths will be used as input for a video annotation play understanding system.

[2] A. Blake, R. Curwen, and A. Zisserman. Affine-invariant contour tracking with automatic control of spatiotemporal scale. In *Proc. Int. Conf. Comp. Vis.*, pages 66–75, Berlin, Germany, May 1993.

[3] S.D. Blostein and T.S. Huang. Detecting small, moving objects in image sequences using sequential hypothesis testing. *IEEE Trans. Signal Proc.*, 39(7):1611–1629, 1991.

[4] D.D. Fu, K.J. Hammond, and M.J. Swain. Vision and navigation in man-made environments: looking for syrup in all the right places. In *Proc. Work. Visual Behaviors*, pages 20–26, Seattle, June 1994.

[5] D.P. Huttenlocher, J.J. Noh, and W.J. Rucklidge. Tracking non-rigid objects in complex scenes. In *Proc. Int. Conf. Comp. Vis.*, pages 93–101, Berlin, Germany, May 1993.

[6] V.S.S. Hwang. Tracking feature points in time-varying images using an opportunistic selection approach. *Pattern Recognition*, 22(3):247–256, 1989.

[7] S.S. Intille and A.F. Bobick. Visual tracking using closed-worlds. MIT Media Lab Perceptual Computing Group Technical Report No. 294, Massachusetts Institute of Technology, Nov. 1994.

[8] D. Koller, K. Daniilidis, and H.-H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *Int. J. of Comp. Vis.*, 10(3):257–281, 1993.

[9] H. Kollnig, H.-H. Nagel, and M. Otte. Association of motion verbs with vehicle movements extracted from dense optical flow fields. In *Proc. European Conf. Comp. Vis.*, volume 2, pages 338–347, Stockholm, Sweden, May 1994.

[10] R.F. Marslin, G.D. Sullivan, and K.D. Baker. Kalman filters in constrained model-based tracking. In *Proc. British Mach. Vis. Conf.*, pages 371–374, Glasgow, UK, Sep. 1991.

[11] J. Mundy. Draft document on MORSE. Technical report, General Electric Company Research and Development Center, Feb. 1994.

[12] H.-H. Nagel. From image sequences towards conceptual descriptions. *Image and Vision Comp.*, 6(2):59–74, 1988.

[13] P.N. Prokopowicz, M.J. Swain, and R.E. Kahn. Task and environment-sensitive tracking. In *Proc. Work. Visual Behaviors*, pages 73–78, Seattle, June 1994.

[14] P.L. Rosin and T. Ellis. Detecting and classifying intruders in image sequences. In *Proc. British Mach. Vis. Conf.*, pages 24–26, Sep. 1991.

[15] I.K. Sethi and R. Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 2:574–581, 1987.

[16] T.M. Strat and M.A. Fischler. Context-based vision: recognizing objects using information from both 2D and 3D imagery. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 13(10):1050–1065, 1991.

[17] T.N. Tan, G.D. Sullivan, and K.D. Baker. Pose determination and recognition of vehicles in traffic scenes. In *Proc. European Conf. Comp. Vis.*, volume 1, pages 501–506, Stockholm, Sweden, May 1994.

[18] A.F. Toal and H. Buxton. Spatio-temporal reasoning with a traffic surveillance system. In *Proc. European Conf. Comp. Vis.*, pages 884–892, S. Margherita Ligure, Italy, May 1992.

[19] J.A. Webb and J.K. Aggarwal. Visually interpreting the motion of objects in space. *Computer*, 14:40–46, 1981.

[20] A.D. Worrall, G.D. Sullivan, and K.D. Baker. Pose refinement of active models using forces in 3D. In *Proc. European Conf. Comp. Vis.*, volume 1, pages 341–350, Stockholm, Sweden, May 1994.

[21] A.L. Yuille, D.S. Cohen, and P.W. Hallinan. Feature extraction from faces using deformable templates. In *Proc. Comp. Vis. and Pattern Rec.*, pages 104–109, June 1989.