M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 294
November 1994. Abbreviated version appears in ICCV '95.

Visual Tracking Using Closed-Worlds

Stephen S. Intille and Aaron F. Bobick
Perceptual Computing Group
MIT Media Laboratory
20 Ames St.
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 refer to our method as *closed-world* tracking. A closed-world is a space-time region of an image sequence in which the complete taxonomy of objects is known and in which all pixels can be explained as belonging to one of those objects. Given object information (and additional contextual information if available) *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 dynamic scene. Common tracking techniques such as correlation, adaptive-templates, and deformable models typically fail in situations where it is difficult to syntactically isolate an object (e.g. background subtraction), where the shape of an object may change dramatically from frame to frame, and where the behavior of an object is not smooth or predictable.

We test our algorithm in the "football domain," a real video-annotation domain currently being annotated manually. We describe how closed-world analysis and context-specific tracking can be applied to tracking football players 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.

Category: Object-tracking, video-annotation, context-sensitive vision

1 The problem

In preparation for the playoff game next week, the head coach asks the Video Athletic Coordinator (VAC) for some compilation tapes. The tapes should contain every offensive play over the last three years in which the ball is within the red zone, it is 2nd or 3rd down, there is more than 5 yards to go, and the team executed a either a sweep or a trap running play. The VAC types the necessary information into the computer, and a short while later hands the coach the separate computer-edited tapes he asked for.

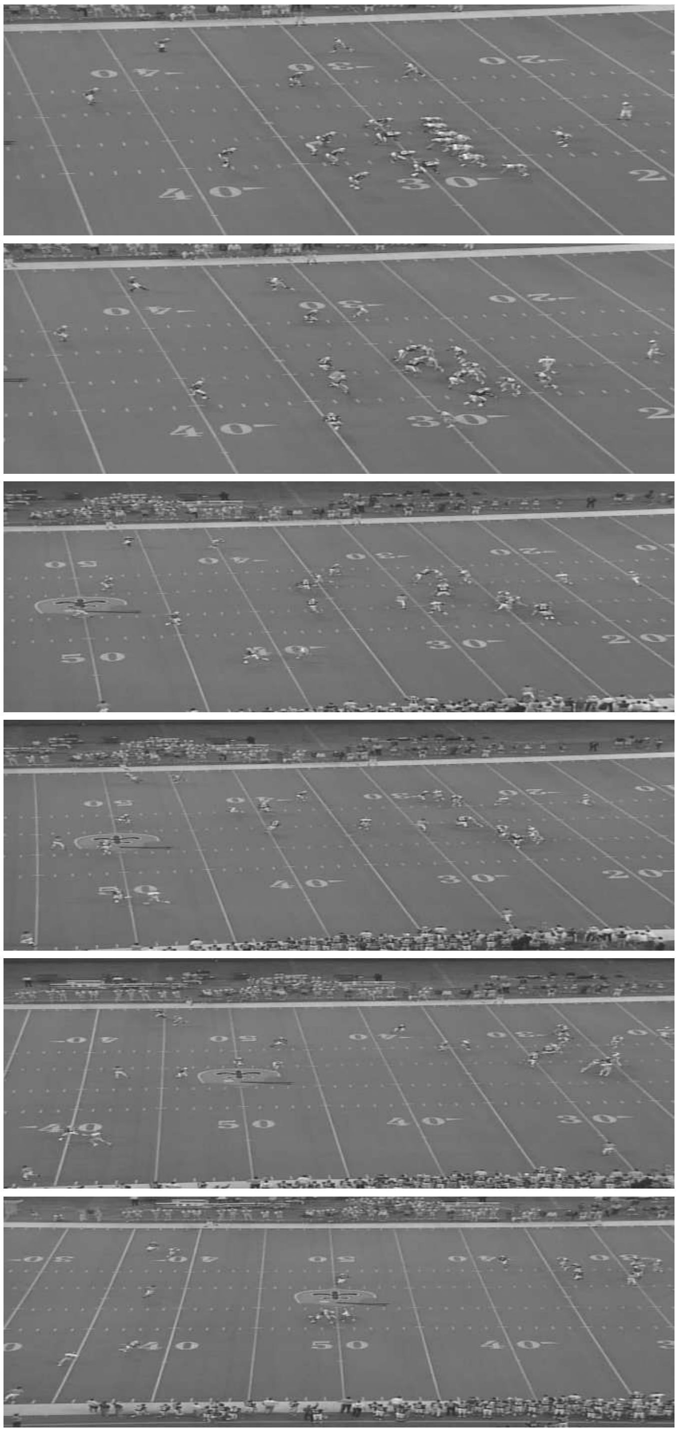
Video annotation is the task of generating descriptions of video sequences that can be used for indexing, retrieval, and summarization. It is different than conventional image understanding in that one is primarily interested in what is *happening* in a scene, as opposed to what is in the scene. Though the scenario above takes place today at universities with major football programs, it does so because of laborious manual effort. While not the prototypical computer vision problem, automatic or semi-automatic video annotation will become increasingly important as access to video databases increases.

Unfortunately, many video annotation domains involve documenting the interactions between people and other non-rigid objects against non-static backgrounds and in unconstrained motion. In these situations even the early vision process of tracking individuals requires contextual information. In this paper we develop a technique that lets us incorporate this information in the tracking of complex interacting objects. Object trajectories are needed for many useful video 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. Our test domain is football player tracking.

Figure 1 illustrates the magnitude of the tracking problem. The figure displays several frames of a single play extracted from the game "films" used by a major university. Because the cameraman is tasked with keeping as many players as possible or necessary in the field of view there is a great deal of panning and zooming present in the sequence. 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. Further, the wide-angle focal length is large enough to induce a fairly substantial barrel distortion when the camera is zoomed out. Our task is to track as many players as possible throughout the entire play. 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 600 frames of 60Hz, 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. Intensity features such as lightly colored helmets are visible in some frames, but they do not persist for the entire image sequence. A sampling of various players at different times during a play is shown in Figure 2.

While the movement of the camera is distracting, the



Frame 20

Frame 60

Frame 100

Frame 140

Frame 180

Frame 220

Figure 1: Several 700x240 frames of a typical pass play from Betacam video of a college football game taken with a panning and zooming camera. The video was sampled at thirty frames per second, digitized, and deinterlaced.

movement of the players is the motion we are trying to capture. Football players strive to move rapidly, change direction unpredictably, and collide with other players at high velocity. In the process, they violate 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 blob-like player from which to compute velocity.

Furthermore, football players are highly non-rigid objects, especially since they flail arms and legs as they run. Their erratic movement, combined with their non-rigidity, the camera motion, and the low-resolution video make the players look more like moving blobs than moving people. A sequence of a single player running is shown in Figure 3; notice the shape of the white player changes significantly between frames five and six, violating any smoothly varying shape assumptions.

Finally, football players frequently collide, often in groups of more than two players. A player adjacent to another player or an official[1] can create a partial occlusion. Figure 4 shows examples where two players or a player and an official run near each other. Given the other tracking difficulties listed in this section, the collision and occlusion problems are particularly difficult.

Color data provides more information that can be used for tracking, but it does not fundamentally solve the tracking problem.[2] Even with color information, both offensive and defensive football player uniforms can blend with background objects. Using color does not solve the tracking problems caused by non-rigid, erratically moving objects, and the tracking would still require methods for dealing with approximate models. Finally, the color distinctions between objects like offensive players and officials are often subtle, especially given noisy color data. Our tracking method should perform better using color data, but we use grayscale imagery in this work.

In this paper we develop a technique to support tracking in as difficult but as knowledge rich a domain as football. We begin by presenting some alternative 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 exact state and position of each of the objects. The advantage of a closed-world is that knowledge of which objects are present can be used to select what information is most powerful for tracking each of the objects of interest. We describe how 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.

[1] Officials, of course, must be tracked as well so they can be ignored.

[2] This is similar to the object recognition problem in computer vision; color can be useful but in most cases it does not make the underlying recognition problem easy to solve.

2 Previous approaches

Here we consider only previous approaches to tracking objects in video sequences. We postpone mentioning work that incorporates contextual knowledge in image interpretation until the next section.

Correlation template tracking is a popular method for tracking objects[32], but template tracking fails when the tracked feature deforms or changes over time. Simple auto-adaptive templates can adjust as a non-rigid object changes, but they tend to "drift" off the region of interest without some powerful underlying model of allowable change. As we show in section 4.6, a simple adaptive tracker fails for the football player tracking task.

Energy-based deformable models have been used to improve adaptive tracking [5, 35] by requiring the template to vary slowly or smoothly. Typically using such models requires good support from the data, necessitating high resolution. Also such models do not help to reason about occlusion, a non-smooth shape varying process. Deformable trackers that have tracked small images of people require accurate object segmentation based upon intensity[4] and optical flow[20], which is difficult to obtain in the football tracking problem.

Differential motion estimators use smooth or planar motion models [26, 11, 6, 9, 3], but in the football domain objects move erratically and noisy image data and spatial and temporal subsampling can make motion and certainly acceleration estimation difficult. General motion based tracking usually assumes a reasonable boundary can be drawn around the objects of interest [33, 27]. In fact, as we will show, it is often difficult to separate the tracking from the boundary estimation problem.

The amount of shape change between frames also impairs the use of edges for tracking[25, 10]. Motion difference blob size and shape characteristics are another feature used for tracking[23, 14, 24], but as we demonstrate later such features cannot always be obtained reliably. Other features that have been used for tracking include relative pixel intensities, corner detection, region detection, and color histogramming. Given that players collide, there is usually no single feature on a football player in the football imagery that remains trackable throughout an entire football play. Space-time tracking methods require strong camera or object motion models that are not available in our current domain[19].

Finally, there are model-based tracking techniques that use geometrical models of rigid objects combined with a recursive motion estimators[16, 14, 34, 15, 30]. Clearly, tasks like tracking football players, or, say, cells in a microscopic image do not easily fit within these 3D matching frameworks.

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[31]. 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 objects (birds) or the same type of object two different context (grounded and flying auklets) may require two entirely different vision methods. Fu's shopper system[8] and

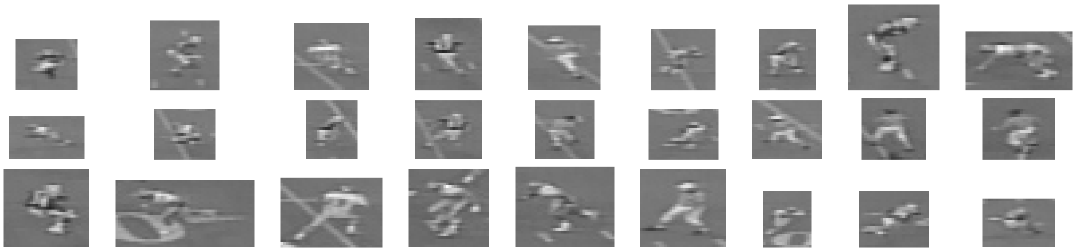


Figure 2: Offensive players, defensive players, and officials clipped from imagery of a football game. The objects are difficult to model due to poor spatial resolution and the inherent complexity of the object shapes.

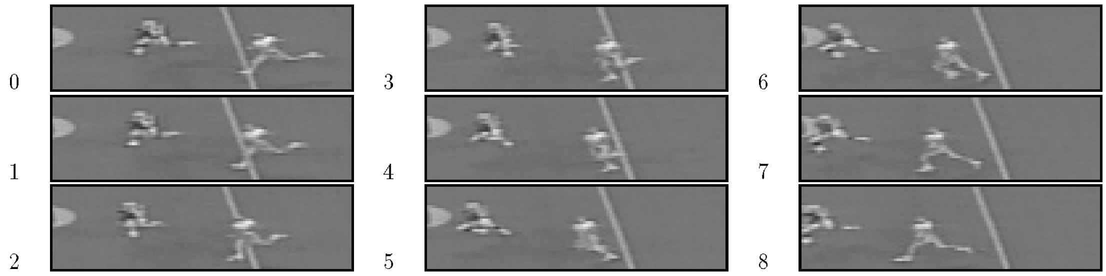


Figure 3: The shape of a single, non-rigid player changes rapidly over just a few frames. Even in a thirtieth of a second, the objects change significantly, as illustrated by the change in the offensive (white) player from frame five to frame six.

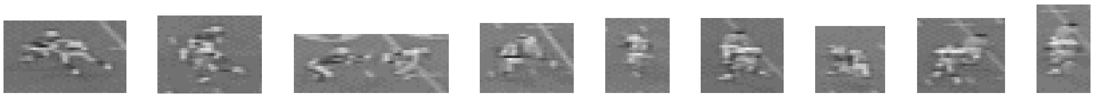


Figure 4: Several examples of two players or a player and an official nearby each other during a football play.

Prokopowicz' active vision tracker[21] also select feature finders 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[23].

Finally, a few systems that have been designed are particularly relevant to the football player tracking problem undertaken here. The coach COACH system uses plan creation to analyze football plays and generate new ones[7]. The SOCCER system generates incremental linguistic reports of short sections of a soccer game given player trajectories and states (eg. "tackle" and "have ball") as input. [2]. The REPLAI system uses the events recovered by SOCCER to estimate the intention of the player actions[22]. All three of these systems use manually generated trajectory input. Histograms have been used to analyze the group behavior of soccer players [13], but that system does not track individual players.

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. The theory is developed and examples are drawn from the football domain. We postpone the details of implementation until sections 4 and 5.

3.1 Closed-worlds

The underlying argument of the previous sections is that the task of tracking objects in a complicated domain such as football requires using some type of knowledge. 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.

Besides using closed-worlds to circumscribe the knowledge relevant to tracking, we also exploit them to reduce the complexity of the problem. Within a closed-world we need to generate consistent interpretations as to player positions and spatial extents. However, we require that overall global consistency can be achieved by independently generating the description within each of the closed-worlds. For example, if the left-corner-back is covering the left-wide-receiver, we often may need to consider the image region that contains both of the players. However, we do not need to consider that interpretation problem while separating and tracking the quarterback and referee behind him. The closed-worlds yield independent interpretation problems greatly reducing the order of interaction that needs to be considered.

We note that a few other authors developed systems that use contextual information and approaches similar to our closed-worlds. Nagel[18] 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[28] 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[17] 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.

Objects

From a computer vision standpoint, all objects are not created equal. The degree to to which the geometric and photometric properties of an object will vary depends greatly on the type of object. 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. The exact dimensions of these objects can be obtained from official football rule books, as shown in Figure 5, and can be used to generate exact models of the objects that can be matched to image data. If the camera's projective geometry and lens distortion is corrected, the image and the model should correspond.

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 is 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 [12]). Examples are shown in Figure 6. But such models contain errors such as the flat bottom and top of the zero, the exact shape of the numbers, and the poor helmet resolution. Another approximate model is the field turf. Using histogramming, a turf intensity range can be estimated. However, the model of turf is still weak, since some parts of players will look like turf and not all turf will fall into the selected intensity range.

Finally, *amorphous* objects are those that cannot be well defined visually. In the football domain, the players are the amorphous objects. As Figure 2 illustrates, players 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 the visual appearance 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.

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 7. 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

Having defined closed-worlds, we now must prescribe how such bounded regions of space-time are found. We motivate our approach by considering the tracking and interpretation problem from the football domain. 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 another object, that 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. Again consider the case of an isolated player. Since the background can be modeled better than the player, a robust strategy is to select features that can be used to assign pixels to the background. Any pixels not so assigned can be assumed to be the player.

Suppose, however, that two players are in contact (as

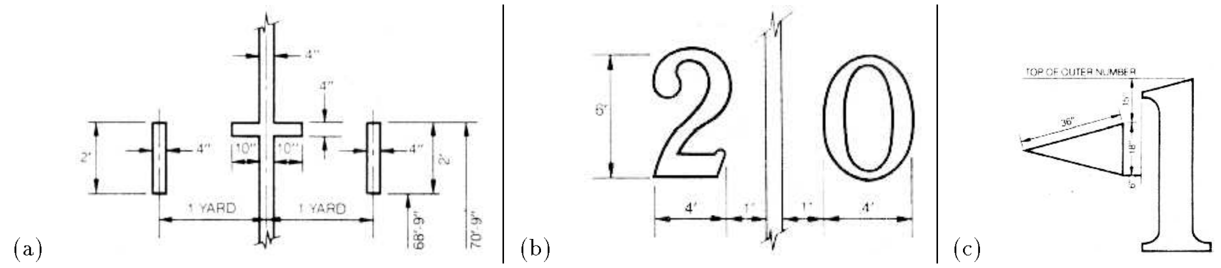


Figure 5: The official NFL geometric specification[29] for field hashmarks (a) can can be used to create *precise*, geometric models of the field lines for any standard football field. However, for some field objects, — shapes of numbers (b) or the position of arrows (c) (which depends upon the number shape) — only *approximate* models are available.

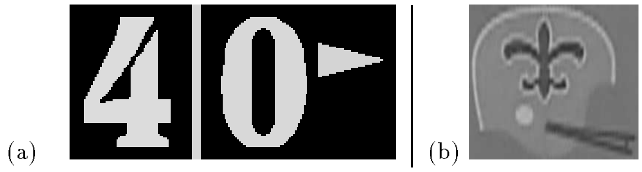


Figure 6: Approximate pixel-map representation of (a) a field number and (b) the field logo. Geometrical specifications on the numbers and logo shapes are unavailable, so the tracking algorithm must be able to exploit these imprecise representations.

Figure 7: Five different closed-world regions. Objects that can be within the closed world are precise (lines and hashmarks), approximate (numbers and logos) and amorphous (players). One closed-world contains two players and line and hashmark objects.

happens in football!). Now background features can only be used to assign background pixels. Other features must be selected to determine which pixels correspond to each player. Here knowledge of the player types (e.g. one is offensive, one is an official) can be used to select 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.
6. Go to 2.

In this section we present the details of each of the above steps when tracking single players — isolated players against the background. We present results which demonstrate both the power and potential weakness of the technique. In Section 5 we develop the multi-player tracking where steps 3 and 4 are more complicated and context-specific feature selection is more important.

4.1 Background knowledge

The basic background knowledge for the football tracking problem is an understanding of the field. We have constructed a detailed field that represents the compilation of the known field objects described in section 3.2; the model is shown in Figure 8. The field model was obtained by combining the precise geometrical objects like lines and hashmarks that were obtained from the official rule book of the NFL[29] and modified for a collegiate field with bitmap objects (e.g. numbers, field logos, directional arrows) extracted from rectified imagery. Black areas in the image are known to be turf which is modeled by an intensity range.

In order to apply this background knowledge to the tracking problem 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. This is simple to do using a homographic transform (perspective four-point transform) as illustrated in Figure 9. This rectification was achieved automatically by tracking many line intersections and computing a least squares solution. Details can be found in [12].

One result of the rectification is that camera motion is removed, giving the appearance of a stationary camera and background. When viewing such imagery one is inclined to believe that it is possible to perform simple background subtraction as a means of isolating the players. Such approaches have been suggested in other domains[23, 31, 14, 4]. In reality, lens distortion (especially with the zooming cameras) and line-finding errors make such stabilization and thus subtraction unreliable. Figure 10-a illustrates the results when applying local spatio-temporal filters to detect rapidly moving objects; even the local derivative operators reveal the inadequacies of the stabilization.

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. That point is used as the initial position of the template. From these positions, the first closed-world region is generated by one of the methods described later. An example of the initial delineation of the closed-worlds is shown in Figure 10-b. The starting formations for a football game are highly structured, and it is not unreasonable to assume that closed-world initialization could be automated in the future using a database of starting offensive and defensive positions and some simple texture or color histogram detectors.

4.3 Isolating closed-worlds

As described in section 3, closed-world boundaries for tracking can be defined using independence. For tracking football players, 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.

The motion-based process to find closed-world image regions is illustrated in Figure 11. 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, as in Figure 11-b. Morphological erosion and dilation operations are then used on each frame, producing contiguous blobs like those in Figure 11-c. Unfortunately, the morphological operations also magnify errors resulting from rectification jitter, as shown by Figure 11-d.

To ensure that motion blobs completely contain player objects, once blobs are identified that contain players, they 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 the previous Figure 10-b.

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 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 [12].

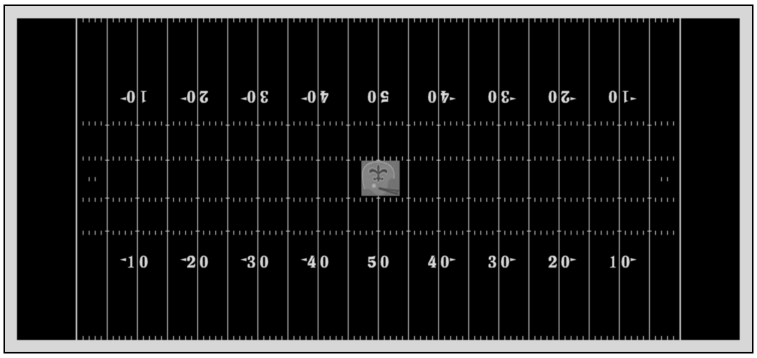


Figure 8: The model of the football field used to remove camera motion and recover the global context of the scene. The geometrical objects (i.e. lines) are modeled precisely. The turf (shown in black here), numbers, and field logos are modeled using approximate pixel-map representations extracted from the image sequence.

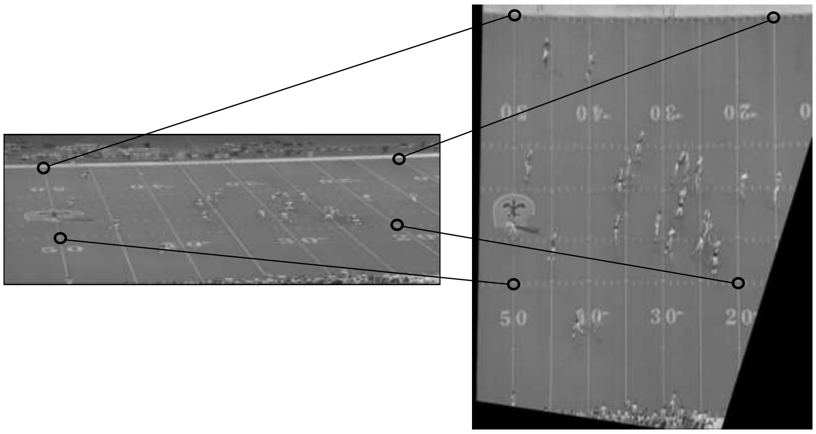


Figure 9: An example of an original and rectified frame. Four of the line intersection points used to perform a planar homographic transformation have been marked.

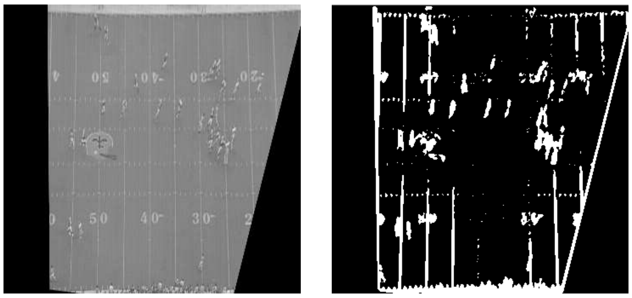
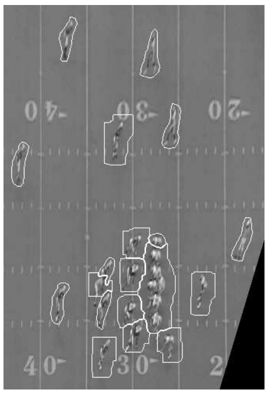
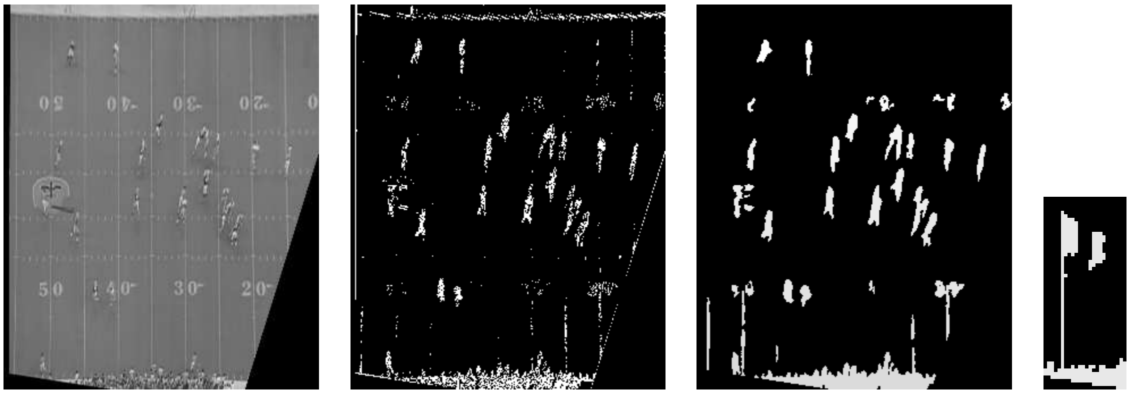
(a)  (b) 

Figure 10: (a) Stabilizing the imagery is difficult due to lens-distortion and line-finding errors, as is highlighted by motion blob detection using the rectified imagery. (b) Initial manual specification of object positions results in these closed-world regions.



(a) (b) (c) (d)

Figure 11: This figure shows the stages involved when obtaining the difference blobs: (a) a frame of the rectified imagery, (b) the frame after space-time smoothing and differencing with the previous frame, and (c) the final motion differencing blobs after morphological processing. When the camera is zoomed out, minor errors in the rectification can occasionally cause the motion blobs to grow quite large, as shown in (d). Size thresholds based upon player positions are used to prevent closed-worlds from becoming too big.

4.4 Pixel assignment and template formation for isolated players

As mentioned, when tracking an isolated player the most robust strategy is to use the field model to determine which pixels belong to the background. Remaining pixels can be assigned to a player template. For isolated players, using context-specific features reduces to using non-background pixels to define the tracker for the player. Of course, for each new frame, a new template must be constructed for each player being tracked.

Recall that the model of the background is approximate and that the rectification of the imagery is also imperfect. Therefore we cannot "subtract" the background. Instead we use a "possible-explanation" strategy. An example is shown in figure Figure 12. Figure 12-a shows an example of a closed-world containing a single player and some background area. Figure 12-b shows the background objects known to be in the same region of the imagery; these objects are found by overlaying the closed-world region of the original imagery, Figure 12-a, onto the background field model. The black regions are modeled as turf.

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. Any pixels not marked "don't care" are used to form the player templates. The final template is shown in Figure 12-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 global contextual knowledge about the location of the model 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. Further, it allows for some error in the rectification process. As long as the closed-world is close to being consistent with the global model, the majority of pixels that remain after the background removal will belong to the tracked player.

4.5 Template tracking

Once a template has been adapted using the closed-world, it is matched to the next frame. 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. The template is matched over an 11 x 11 region centered over the previous position. Figure 13-a shows part of an image with the approximate location (the white rectangle) of the previously tracked template shown. Figure 13-b shows the closed-world region around the player, which was recovered using the motion blobs. Figure 13-c shows the model objects in the closed-world region (excluding the turf), and Figure 13-d shows the context-specific template. Figure 13-e shows the matching score for that template in the next frame. There is a clear peak when the template is matched within a window of the following frame, despite the small number of pixels in the template. Since features that might be incorrectly matched due to the contextual position of the player 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.

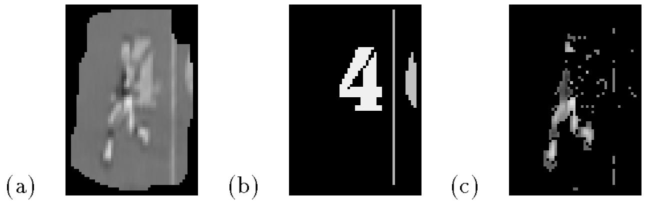


Figure 12: (a) Closed-world image region. (b) Closed-world model of same image region. (c) Remaining template after pixels in (a) are removed by being "possibly-explained" by those in (b).

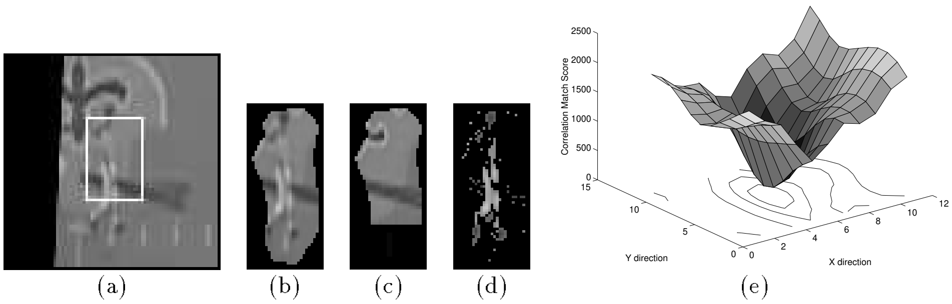


Figure 13: Template creation and matching: (a) The image patch with the template overlayed. The player is over the helmet field logo. (b) The closed world region. (c) The model in the closed-world region around the player. (d) The closed-world image region where all potentially contextually-variant pixels have been removed. Black regions are "don't care" values. (e) The correlation matching scores (low values are good matches).

4.6 Single-player results

The algorithm described has been tested on a 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.

We have selected the following method for presenting our results in hardcopy. Since the player paths are difficult to interpret when superimposed with camera motion, the paths are overlayed onto the first rectified frame of the tracked sequence. To show the state of the template at different points during the tracking, a few frame numbers have been drawn on the path. Below the path image, "focus" images of the player and the template are shown for the marked frames.

As part of our testing, a simple adaptive template tracking system has been implemented. A template is placed down on a player in the rectified football imagery. At each step it is moved to the best match in the next image within a small window. A typical result of using this adapting template is shown in Figure 14-b. The template will track isolated players for a few frames when the player was in the homogeneous turf regions of the field, but the template will drift from the player as the player runs over field objects. Without using the rectified imagery, the template performance degrades even further. Median filtering to recover the field background and background subtraction might be used to improve these results, somewhat. However, camera distortion and inaccuracies in the rectification are likely to leave artifacts at the edges of removed field-object features that will still cause the adaptive template to drift.

In comparison, Figure 14-c illustrates 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, the field markings do not come to control the behavior of the template. Finally, in Figure 14-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 15 shows the result of tracking the right outside linebacker for 230 frames. The player starts standing still, accelerates to a sprint while running over several field numbers, and then stops and changes direction. The context-specific template succeeds by only tracking the parts of the player that are distinguishable from the objects the player occludes. Further, since no assumptions have been made about smooth velocity, the template can capture the player's sharp change in movement.

The most difficult field object for the tracker is the helmet field logo. Figure 16 shows the results of tracking an official who runs directly over the helmet. As demonstrated

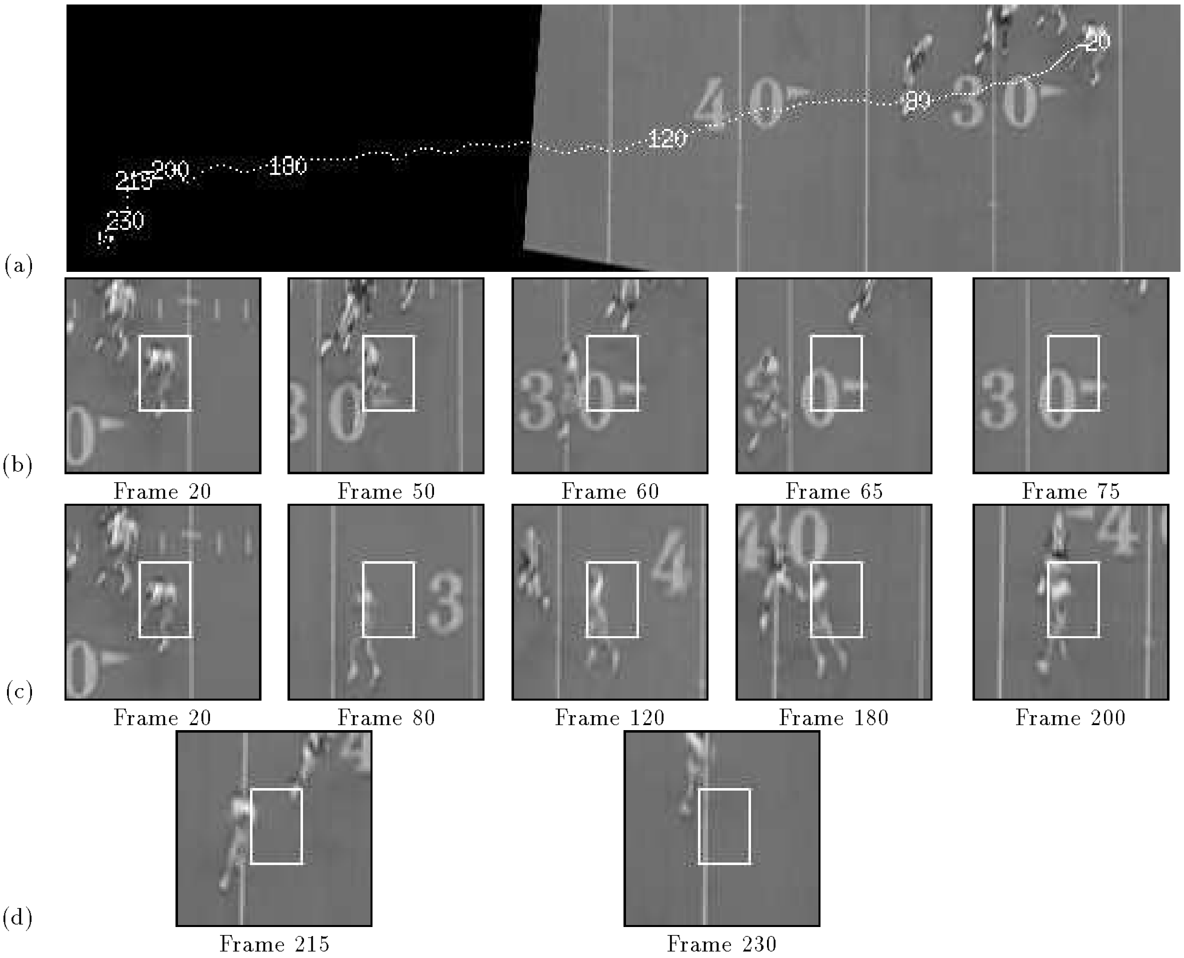


Figure 14: (a) Path run by offensive player, shown by trace on the first rectified image of the sequence. (b)The results of using a simple adaptive template tracker that does not use any contextual information. By frame 60 the tracker has stuck to a field marking. (c) Closed-world tracker correctly tracks the player for 200 frames. (d) Tracker loses the player because another player runs close by.

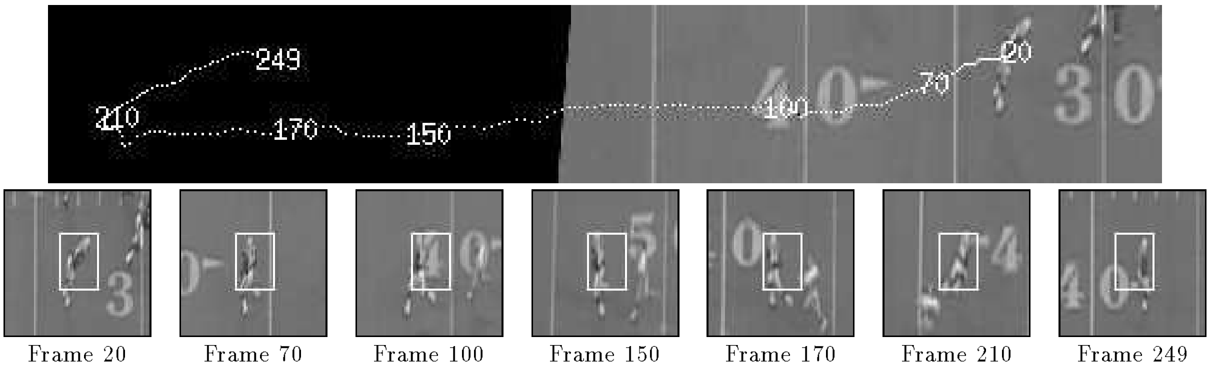


Figure 15: Results for tracking the ROLB, who starts from a stationary position, accelerates to a sprint while running over field numbers, and then stops and changes direction.

by the focus images for frames 80 and 120, the player intensities are very similar to the pattern in the helmet. That similarity makes careful selection of features to track critical. By frame 150, very little of the official is actually in the template, but the pixels that remain in the template are context-specific, and the tracker correctly follows the player and pulls away from the field object.

The described implementation of context-specific features is relatively simple, and the method will occasionally fail when models are imprecise, spatial resolution is low, and the object being tracked is very close in appearance to some nearby object. One example is shown in Figure 17. Here the player is turned in such a direction 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 background, and the template cannot find the player in the next image. In the example play of 270 frames, two isolated-player paths fail to be tracked successfully.

5 Multi-player closed-worlds

Failure of the single player tracker occured when a second player encroached in the closed-world of the first, violating the single-player closed-worlds generated. Here we modify the algorithm of the previous section to accommodate closed-worlds that contain more than one player. The two components that need to be revisited are the pixel assignment and the context-specific template formation.

5.1 Context-specific pixel assignment

The partitioning of the closed-world image region is determined by knowledge about the characteristics of the various objects in the closed-world. In the case of two football players, the partitioning of each pixel can be based upon the following factors:

- Agreement of the pixel with modeled object knowledge (e.g. pixel intensity) of objects (both field and players).
- Proximity of the pixel to each of the other objects in the closed-world, as determined from the previous frame.
- Proximity of the pixel to other pixels in the current frame that have been assigned to each object.
- How well the pixel's assignment preserves object contiguity.
- Whether the pixel belongs to one object and occludes another object.

These and other information sources could be combined into an energy-minimizing partitioning function to be solved using some iterative technique. However, we have found that even simple partition algorithms are adequate for constructing context-specific templates. For the work presented here, we compute the closed-world partitioning based upon the relative agreement of a pixel with the model of each object in the closed-world, the proximity of the pixel to other pixels currently assigned to an object, and the model of the background (same as used for single player tracking).

Of course, the relative agreement of a pixel with the model of each closed-world object depends upon the functions used to assess the consistency of some pixel with a particular object. This is where we exploit the idea of context-specific features. The basic approach is to select some properties of the pixels that can be used to discriminate between the objects within the closed-world. While the simplest property is that of intensity, any other pointwise function (e.g. variance or motion-vector) can be used. 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.[3] 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 the examples shown here, 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.[4] Using these discrimination functions, the pixels in the two-player closed-worlds are assigned to objects using the following heuristic: Within a closed-world image region, all background pixels are eliminated using the strategy described in section 4, effectively leaving only player pixels. These pixels are then assigned weights based upon the intensity weighting functions determined by the types of players in the closed-world. For instance, if a closed-world contains an offensive player and a defensive player, each non-background pixel will receive two weights. The first weight indicates pixels that are good at discriminating the offensive player from the defensive player. The second weight marks pixels that are good at discriminating the defensive player from the offensive player. The resulting weighting functions are then smoothed using a Gaussian, creating peaks around small spatially coherent regions of highly discriminative pixels.

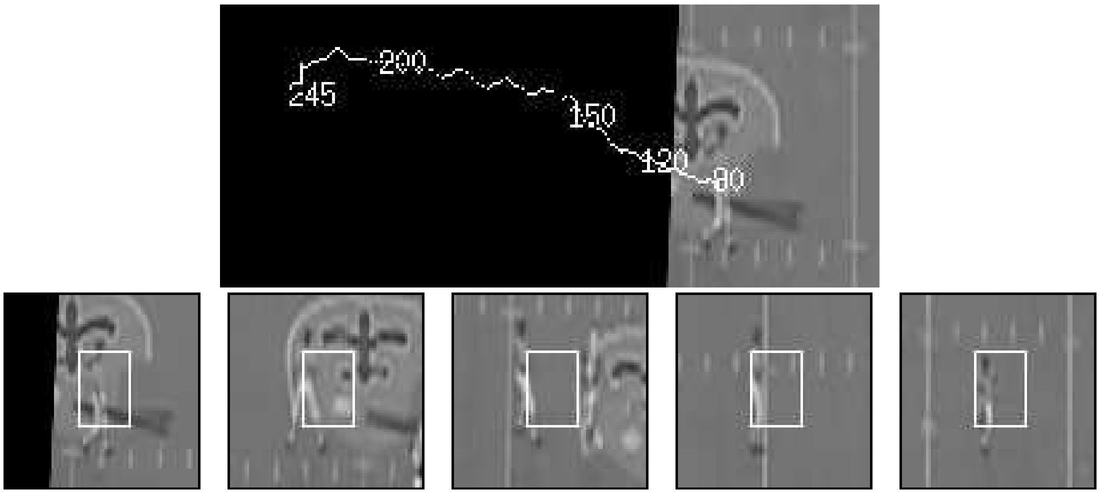
5.2 Context-specific template formation

Ideally, the closed-world region will be partitioned, or "explained," well-enough so that good context-specific features can be reliably selected to use for tracking. For example, if the partitioning algorithm assigns a large percentage of the closed-world pixels as highly probably belonging to some object, the partitioning is assumed to be well-determined. This is the case in the single-player method described above.

Because of our weak models, however, the closed-world partitioning will often have a high uncertainty in pixel assignment when two players are interacting. Therefore, creating new templates from all of the player-assigned pixels might be inappropriate. Rather some context-specific "feature" — here a subset of the pixels that are highly diagnostic for the object within the context — would be better. In the extreme case of uncertainty in closed-world partitioning, it is better not to update the template at all. Here the context-specific aspect of template formation is knowing *not* to create a new template based on current pixels.

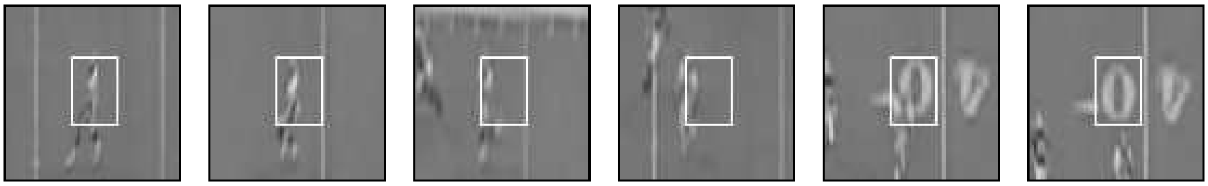
[3] Ideally, we would like to have the probabilities that various features will distinguish one object from all other possible objects *and all combinations of all other objects.*

[4] These soft weighting functions were estimated by manually comparing histograms of isolated players.



Frame 80 Frame 120 Frame 150 Frame 200 Frame 245

Figure 16: Result demonstrating the context-specific features can track a player (in this case the field judge) running over the field helmet logo, even when the player intensities are similar to the helmet intensities, as in frames 80 and 120.



Frame 20 Frame 80 Frame 150 Frame 200 Frame 249 Frame 259

Figure 17: This figure shows an example where the tracker failed. Here the low-resolution player is turned in such a direction 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 background because three difficult problems occur simultaneously: the camera is zoomed out causing low spatial resolution, the model of the object being crossed is an approximation, and the player crossing the object happens to appear in a "featureless" view where few non-white pixels are visible.

Rather, older, less current but more diagnostic template information is preferred.

Therefore, 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. Two isolated players, the defensive LOLB player and the offensive SB, and their context-specific pixels (shown in black) are shown in Figure 18-a. When the player closed-worlds merge into one larger closed-world, distinctive features are selected based upon the objects in the closed-world. In this case the partitioning of the closed-world in Figure 18-b has large uncertainty. Therefore, the distinctive template pixels are selected just before the closed-worlds merge when the objects are visually distinct and the partitioning is reliable. When the players move close together, the distinctive tracking features are no longer adaptively selected from the closed-world descriptions since the closed-world descriptions are unreliable. Instead, they are "frozen." The features are not ideal, since the players will change shape as they move past one another and one feature could potentially become occluded. However, as shown by Figures 18-b-g, even very few well-selected pixels can track the objects successfully. Once the LOLB is spatially separated from the SB (and is in its own closed-world), the tracker reverts back to the more robust single-player tracking method.

Figure 18 also shows an example where the tracker fails. As the LOLB and SB are being tracked together, a third player, the defensive LILB, approaches in Figure 18-f. This player is tracked independently, until Figure 18-g, when all three closed-worlds merge. At this point, the LILB begins to completely occlude the SB. By Figure 18-l, the SB tracker has mistakenly locked onto the LILB, even though the tracker is using features selected to maximally distinguish offensive and defensive players. Our model currently does not handle full occlusion, which is relatively rare in the football video.

In section 4.6 we showed that the single-player tracker can get confused when other players run nearby (see previous Figure 14-d). Using the multi-player tracker, that error is avoided. Both players are tracked simultaneously and when they move near one another context-sensitive features are selected and tracked correctly.

A final example of the successful application of the two-player tracking is shown in Figure 19, where a bounding box has been drawn around the active pixels. In Figures 19-a and 19-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.

5.3 Complexity

A word about complexity: The algorithm described here requires a large amount of processing that makes evaluating computational performance difficult.[5] Real-time implementation of the current tracker will require systems capable of fast computation, specialized real-time image-processing hardware, *and* hundreds of megabytes of fast memory.

[5] One nine second play is over forty-five megabytes of input data. Each frame of that sequence must have lines detected and field features tracked so that a new rectified sequence can be generated. The rectified frames are 700 by 679 pixels, so that sequence alone is another 128 megabytes of data. The difference blob detection requires more processing on the rectified frames, including morphological operations and filtering in space and time. At each tracking step, a large number of correlations must be performed. Finally, the algorithm usually requires running multiple trackers at once so that each tracker must know about all the players nearby. However, the current system is parallelizable and could be designed to run in near-real-time with some specialized hardware and hundreds of megabytes of fast memory.

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 performed 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.

In Figure 20 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 going to be used for development of a play understanding video annotation system. In future work, we hope to use closed-worlds to place bounds upon higher-level domain knowledge that might be used for object tracking and video annotation.

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, December 1991.

[2] E. Andre, G. Gerzog, and T. Rist. On the simultaneous intepretation of real world image sequences and their natural language descriptions: the system soccer. In *ECAI*, pages 449–454, August 1988.

[3] S. Ayer, P. Schroeter, and J. Bigun. Segmentation of moving objects by robust motion parameter estimation over multiple frames. In *ECCV*, volume 2, pages 316–327, Stockholm, Sweden, May 1994.

[4] A. Baumberg and D. Hogg. Learning flexible models from image sequences. In *ECCV*, volume 1, pages 299–308, Stockholm, Sweden, May 1994.

[5] A. Blake, R. Curwen, and A. Zisserman. Affine-invariant contour tracking with automatic control of spatiotemporal scale. In *ICCV*, pages 66–75, Berlin, Germany, May 1993.

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

[7] G. Collins. Plan creation: using strategies as blueprints. Computer Science Dept., Technical Report RR 599, Yale University Department of Computer Science, May 1987.

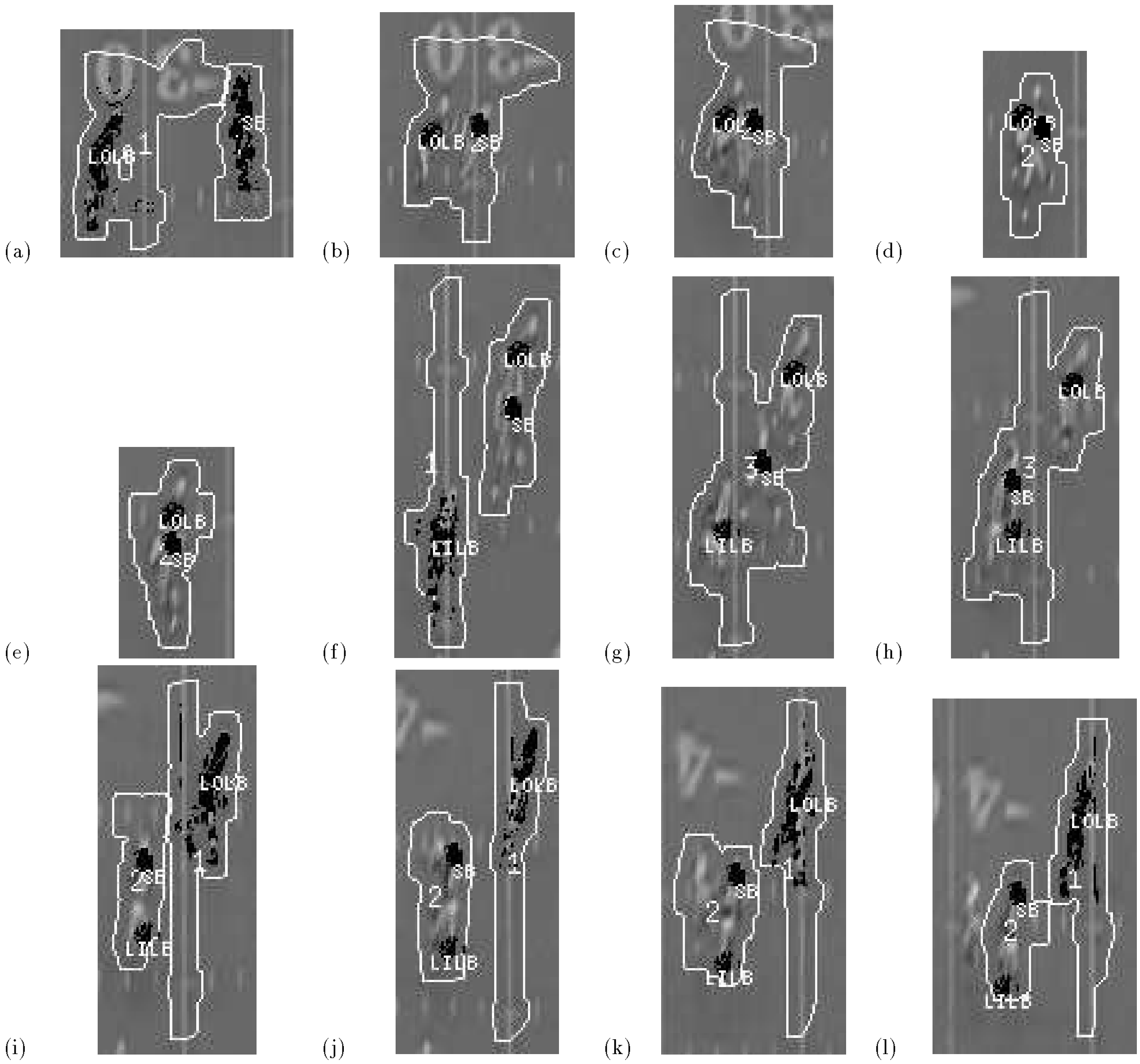


Figure 18: A multi-player tracking example. The defensive LOLB and offensive SB are tracked successfully. The black pixels are those used to construct the context-sensitive template features, which changes based upon the objects in the closed-world. The tracker fails when full occlusion occurs and the SB passes the defensive LILB and the SB tracker locks onto the LILB.

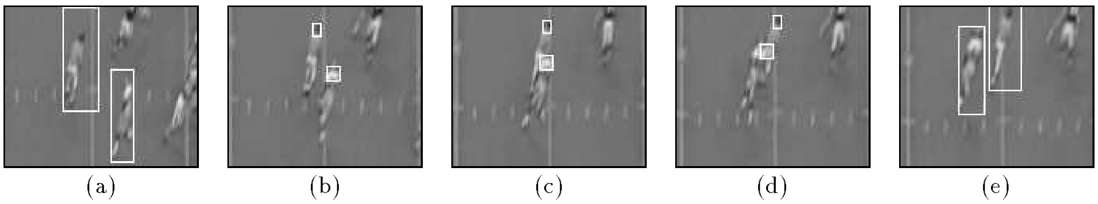


Figure 19: The offensive player and the official are tracked successfully as they move past each other. The white bounding box indicates the region from which pixels were selected to construct the context-sensitive template.

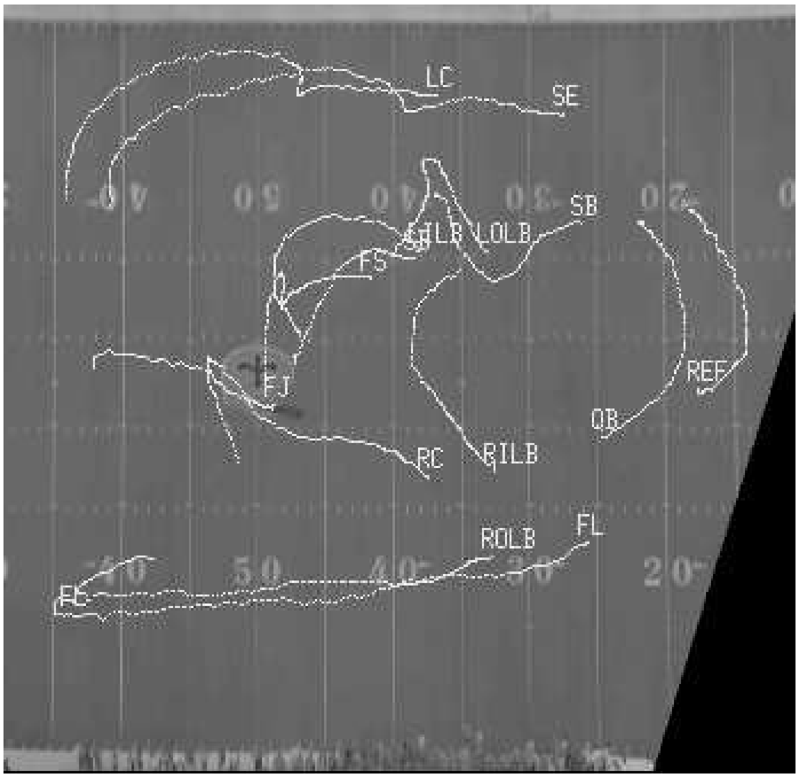


Figure 20: All recovered paths overlayed on an image of the field. These paths will be used as input for a video annotation play understanding system.

[8] 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 *W. Vis. Behaviors*, pages 20–26, Seattle, June 1994.

[9] C. Huang and C. Wu. Dynamic scene analysis using path and shape coherence. *PR*, 25(5):245–461, 1992.

[10] D. Huttenlocher, P. Noh, J. Jae, and J. William. Tracking non-rigid objects in complex scenes. In *ICCV*, Berlin, Germany, September 1999.

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

[12] S.S. Intille and A.F. Bobick. Tracking using a local closed-world assumption: Tracking in the football domain. MIT Media Lab Perceptual Computing Group Technical Report No. 296, Massachusetts Institute of Technology, August 1994.

[13] T. Kawashima, K. Yoshino, and Y. Aoki. Qualititative image analysis of group behavior. In *CVPR*, pages 690–693, June 1994.

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

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

[16] R.F. Marslin, G.D. Sullivan, and K.D. Baker. Kalman filters in constrained model-based tracking. In *BMVC*, pages 371–374, Glasgow, UK, September 1991.

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

[18] H.-H. Nagel. From image sequences towards conceptual descriptions. *IVC*, 6(2):59–74, 1988.

[19] T. Nakanishi and K. Ishii. Automatic vehicle image extraction based on spatio-temporal image analysis. In *ICPR*, volume A, pages 500–504, 1992.

[20] R. Polana and R. Nelson. Low level recognition of human motion. In *Work. Non-Rigid Motion*, pages 77–82, November 1994.

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

[22] G. Retz-Schmidt. Recognizing intentions in the domain of soccer games. In *ECAI*, pages 455–457, August 1988.

[23] P.L. Rosin and T. Ellis. Detecting and classifying intruders in image sequences. In *BMVC*, pages 24–26, September 1991.

[24] A. Sato, K. Mase, A. Tomono, and K. Ishii. Pedestrian counting system robust against illumination changes. NTT HIL tr, NTT Human Interface Laboratories, 1994.

[25] H. Sawhney and A. Hansen. Trackability as a cue for potential obstacle identification and 3-d description. *IJCV*, 11(1):237–265, 1993.

[26] I.K. Sethi and R. Jain. Finding trajectories of feature points in a monocular image sequence. *PAMI*, 2:574–581, 1987.

[27] A. Shio and J. Sklansky. Segmentation of people in motion. In *W. Vis. Motion*, pages 325–332, 1991.

[28] T.M. Strat and M.A. Fischler. Context-based vision: recognizing objects using information from both 2D and 3D imagery. *PAMI*, 13(10):1050–1065, 1991.

[29] P. Tagliabue. *Official playing rules of the National Football League*. Triumph books, Chicago, IL, 1993.

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

[31] A.F. Toal and H. Buxton. Spatio-temporal reasoning with a traffic surveillance system. In *ECCV*, pages 884–892, S. Margherita Ligure, Italy, May 1992.

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

[33] J. Woodfill and R. Zabih. An algorithm for real-time tracking of non-rigid objects. In *Nat. Conf. on AI*, pages 718–723, July 1991.

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

[35] A.L. Yuille, D.S. Cohen, and P.W. Hallinan. Feature extraction from faces using deformable templates. In *CVPR*, pages 104–109, June 1989.