Real-Time Vision-Based Pattern Tracking Without Predefined Colors

Paul Furgale, John Anderson, and Jacky Baltes
Department of Computer Science, University of Manitoba, Winnipeg, Canada R3T2N2
ptf,andersj,jacky@cs.umanitoba.ca

Abstract

Global vision in the robotic soccer domain provides an excellent vehicle for developing pattern tracking algorithms, since many fast-moving objects must be differentiated and individually tracked in real time. We present a system and novel pattern design that may be used to track robots from a camera mounted at an oblique angle and without the use of predefined colors. Avoiding the need for color calibration results in more robust vision that can be deployed across a wider range of problems and environmental conditions, since issues such as the appearance of color under various lighting conditions can be avoided.

1 Introduction

Global vision in the robotic soccer domain is an extremely interesting problem: a significant number of objects must be differentiated and accurately tracked in real time while maintaining a high frame rate. Our interest in the problem of global vision for a team of robots is twofold. First, we wish to provide accurate vision over as wide a range of conditions, with as few underlying assumptions as possible. Second, from the standpoint of artificial intelligence and computer vision research, we wish to provide better solutions to this and related problems by focussing on intelligent, adaptive approaches rather than relying on specialized hardware.

One key theme in our global vision research that contributes to both these goals is doing away with the common assumption of a perfect overhead camera view [2, 3]. While a perfect overhead camera view simplifies global vision computationally (e.g. calibration, geometry), such a system is restricted to only situations where a perfect overhead view can be guaranteed. A more broadly-applicable solution, and one that bears more similarity to human vision, can be obtained if we assume an oblique camera angle and force the vision system to take this into account. While this results in more work for a vision system, the solution is more broadly applicable to problems other than robotic soccer. It also results in an approach that is easier to work with in practice in a robotic soccer domain as well: we need only set up a camera in any space from which a field view can be taken, and require no overhead infrastructure to mount a camera.

A critical weakness in many global vision systems for robotic soccer, including our own work thus far, is a continued reliance on predefined colors. In the RoboCup small-sized league, colors are used to identify a robot's team, to distinguish individual robots within a team, and to identify the ball. The typical process used by these systems (e.g. [1]) involves defining acceptable ranges for the colors used, classifying all pixels in each image capture by these colors, joining these into regions and searching these for patterns that describe robots or the ball, and mapping the coordinates of these in the image to the real world. While this method certainly works well for many teams in the small-sized league, its success relies on some basic assumptions. First, with the camera mounted overhead, the system designers may assume that segmented regions will not be of significantly different size at any place in the image. Because of this, vision systems may color threshold and segment the raw image, only correcting for perspective and lens distortion after individual objects have been found and identified. Such a system would have trouble processing a side mounted camera image where regions distant from the camera are heavily foreshortened. Second, the reliance on color thresholding demands that team colors must be broadly separated along the color spectrum from each other and that, for the systems to work well, they must utilize expensive digital cameras which return accurate and consistent color over a range of lighting conditions.

This paper describes a new global vision system that attempts to reduce reduce reliance on these assumptions. Like our previous work, our new tracking system operates from a side angle, interpolating captured images to construct an overhead view. The work described here introduces an approach based on the use of motion detection for object identification and noise reduction that, in conjunction with the use of a novel tracking pattern, alleviates the need for predefined colors. Following this, we describe how this is employed to identify and track robots in real time, and then present some initial performance results and show the operation of our system under RoboCup conditions. We begin with a brief examination of the interpolated images, in order that difficulty of object recognition under these conditions may be fully appreciated.

2 Image Interpolation

Due to real-time constraints in the robotic soccer domain, we use the raw fields from the video capture device rather than a full interlaced image. The field is then interpolated using a method based on the well established Tsai camera calibration [7] and previously described in [3] to construct an overhead view from the oblique captures of the camera. Interpolation represents a significant computational expense, and to maintain real-time performance we work with an interpolated image of reduced resolution. This results in an increase in noise in the image and obvious aliasing.

Figure 1 illustrates a sample de-interlaced field and the corresponding low-resolution interpolated image generated from this field. The interpolated image is representative of the type of image we wish to process, and nicely illustrates the difficulty faced here. To capture the field in the image with a reasonable border, a pixel dimension of 25mm was required (i.e. each pixel represents an area of 625mm^2), resulting in a low-resolution interpolated image 125x76 pixels. A 50mm diameter ball on the field is easily seen in the original image, but occupies between 1 and 4 pixels in the low-resolution interpolated image. This is easily small enough to be considered noise by many algorithms.

Relying heavily on color in a situation like this is extremely error prone. We thus developed an approach that would allow us to work with images such as the one depicted above, under varying lighting conditions, that would replace the use of pre-defined colors as the primary means of identification. There are two major elements we have added to support this: first, the use of motion detection against a static background as a replacement for color matching in each frame; and second, a method for defining robot markers that will allow an efficient search to be performed in an interpolated image. The following sections define each of these in turn.

Figure 1: Captured field and corresponding low-resolution interpolated image

3 Motion Detection

Because the ball and the color patches typically used to mark robots are represented by so few pixels in an image such as that of Figure 1, we forgo color thresholding after interpolation, and employ motion detection as our primary means of identifying regions of interest. Indeed, because global vision in a soccer situation involves a large static background, motion detection is a logical basis for this task. Moreover, if we can separate a static background, the portions remaining represent a much smaller dataset from which to perform further processing (e.g. proper, high-resolution interpolation of select areas, if necessary).

In designing a motion detection technique, we attempted a number of methods for thresholding using pixel intensity, as well as distance in color space, including a global threshold and multiple local thresholds modifying a global threshold. None of these were satisfactory, partly because of the effects of field lines in conjunction with the noise and aliasing in the image. In addition, we could not comfortably identify a thresholding mechanism that would not lose information from darker parts of the image before the thresholding mechanism removed noise in the lighter portions.

We alleviated these difficulties by adapting $\Sigma\Delta$

background estimation [6], a computationally inexpensive method of recursively estimating the mean color and variance of each pixel.

After collecting the background, we detect motion in the image by scaling a global threshold by the variance for each pixel and comparing the sum-squared error of the incoming image to this threshold. Under these conditions, we found that increasing the global threshold enough to suppress all noise had the undesirable side-effect of suppressing the ball (1-4 pixels) when it was in darker or more aliased parts of the image.

To solve this problem, we pre-compute the threshold for each pixel by applying a convolution to the variances using the following kernel:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

and scale the result by a global threshold. Now, in order to discern motion in an incoming image we calculate the sum-squared error of each pixel, apply the above convolution kernel to the result and compare each of these values to the pre-computed threshold. This has the effect of blending motion in the nine pixel neighbourhood together, working against single pixel noise while drawing out the ball by adding the weight of subtler changes in neighbouring pixels.

After removing the background, we apply the fast region growing algorithm of [4] to generate a list of regions of interest to feed into our object identification algorithm. Figure 2 shows the result of background removal.

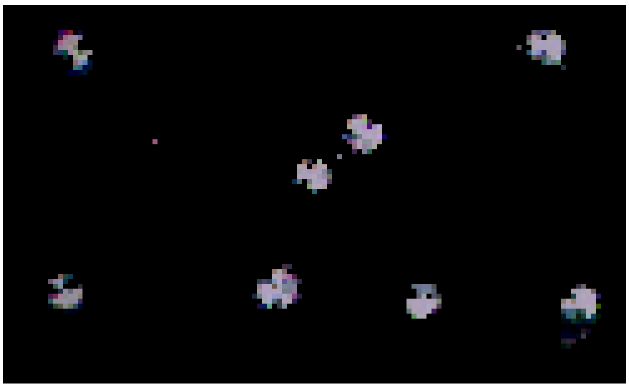


Figure 2: Separation of objects from the background.

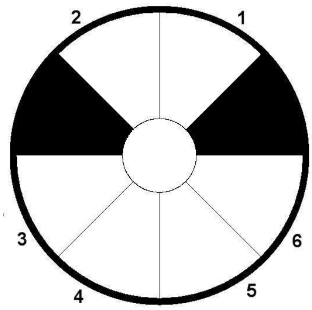


Figure 3: A new hat design: fixed black values allow an orientation to be recognized, while white and non-white values in locations 1-6 allow identity to be determined

4 Pragmatic Object Identification

The motion detection described in the previous section allows us to efficiently determine, for any frame, which pixels do not form part of the static background, and separate these from the image. We must now determine the identities, orientations, and positions of the objects in the captured image, in a manner that uses a minimum of computational resources.

To do this, we require a simple and effective form of representation to be able to distinguish the objects (robots and the ball) in which we are interested. For each of these, a single image can provide the orientation and identity. The difference across multiple images can determine the velocity, and multiple images can be used to correct errors when an object is not successfully identified in every image.

Previously, most teams in the RoboCup small-sized league and elsewhere have used colored markings on hats sitting on top of the robot. The small-sized league rules state that one colored spot must be used to identify the team to which the robot belongs, and a number of other colors are normally employed to provide a series of color combinations that will both identify individuals and indicate their orientation.

It has been shown previously [5] that that an invariant, asymmetrical pattern on a robot hat allows one to identify more robots with fewer colors. In order to recognize both robot identity and orientation, we employ a novel hat design, as shown in Figure 3, and use motion detection to prime a multi-pass tracking system.

The hat pattern shown in Figure 3 contains two black wedges that are identical on all robots, allowing a scanning algorithm to determine the orientation from these wedges alone. The remaining six wedges can be identified based on their relative positions to the two black segments. By marking just one of these segments with some color (neither black nor white; Our tracking system is indifferent to the color) we can identify up to six different robots uniquely. By marking two of these segments, up to twenty-one individuals can be determined. For the purposes of RoboCup the centre of the hat is left for the team color.

The motion detection described above gives us a list of regions of interest. To identify a robot, we begin with the center of each region and employ a restricted high-resolution interpolation. We interpolate three concentric circular strips of pixels around the assumed centre point, storing only the intensity of each pixel. The strips have a radius 50%, 56% and 62% of the hat radius respectively, with each strip 64 pixels long. We then build a representative strip by taking the median intensity pixel for each of the 64 cells and running this representative strip through our robot identification algorithm.

To maintain indifference to color, we must determine two boundaries within the representative strip: the boundary between black and *other* (the marker which is neither black nor white), and the boundary between other and white. We use a histogram of intensity values gathered while interpolating the strip. The hat design is such that any point near the centre will yield approximately 25% black pixels in the strip, so we employ the black-other $(b - o)$ threshold value that approximates this the closest.. For the other-white threshold, we may employ the same method if we know what robot we are looking for (as this determines the number of colored wedges). Otherwise, we use an iterative method where we initialize the other-white $(o-w)$ boundary as the centre between $b-o$ and the top of the histogram's range, and then replace the boundary with the average of the weighted sum of the histogram counts below $o - w$ and those above $o - w$. Two iterations of this provides us with a reasonable threshold the majority of the time.

Once these thresholds are determined, the circular interpolation is scanned for the two black regions. The orientation of the robot is determined by taking the average of the centre of the two black regions. This allows us to determine orientation with a granularity of 1.4 degrees. Additionally, the two black wedges form a basis to the plane on which the hat lies. From this, we assess the widths of the two black regions and adjust the position of the robot accordingly. If the width of a black wedge is too big, it pushes the robot's centre away from the wedge. If it is too small, it pushes the robot's centre toward the wedge. This allows us to recursively refine our estimate of the robot's position. After determining centre and orientation, the remaining parts of the representative strip are partitioned relative to the black wedges and the identification number of the robot is determined by simply counting the number of white and colored pixels in each wedge.

5 Tracking

We employ the above method for identifying a robot as the core of a complete tracking system that operates in real time. While in theory one could look for any area that could approximate a robot, hypothesize a centre, and match from there, there are simply too many potential areas to accomplish this in real time in a data-directed manner. Moreover, regions representing robots may be grouped, in that robots can abut one another and form a larger region in the image, whose centre will never allow a robot to be matched.

To deal with this, we employ tracking of objects across images. Before any motion detection takes place, each robot found in the previous frame predicts its position in the new image (based on previous position and velocity). This will allow some robots to locate themselves in the image quickly, since they have a basis for a centre point that will allow recognition using the process described above. If a robot is able to locate itself in this pass, it applies a mask to the motion detection, which prevents the pixels it occupies from being flagged as motion. This assists in recognizing other robots, in that a large clump of robots may be broken into clearer regions once those recognized by prediction are masked.

For the remaining unidentified robots, we find the region in the image associated with the predicted position of each. This can be accomplished through a binary search of the run-length encoded pixels and their associated regions. If the lookup is successful, and the region is of an appropriate size, the centre of mass of the region is passed to the robot identification algorithm.

These three steps allow the vast majority of the recognition process to be done from a goal-directed perspective. For any robots that remain unidentified, we must heuristically attempt to match likely regions in the image. We begin by assessing each unused region of motion, and if it is of a size that could contain a robot, pass its centre of mass to the robot-identification algorithm.

Regions of motion that are too large to be a robot could still be a group of robots in close proximity. To deal with this case, we observe that in the absence of noise, a region of motion containing two or more robots will have at least one robot touching each edge of the region's bounding box. As the regions are run-length encoded, we may move the equivalent of one robot radius in from the region's edge, perpendicular to the axis of encoding, and retrieve a list of runs that should cross the centre of at least one robot. This method is applied to both the top and the bottom of each remaining region. From the list of runs we choose of those that are at least one robot diameter long. The robot-identification algorithm is then applied along these runs, using each pixel's location as the potential centre of a robot. If a robot is found in this pass, it should mask itself out in the next frame and allow regions containing several robots to be broken up recursively over the course of several frames. While this could be done using only one image capture, this puts a cap on the amount of processing applied to a single frame.

Tracking is similarly employed when identifying the ball. The difficulty in locating the ball in the interpolated images has already been described in Section 2. In practice, we have found it less error prone to verify the location of the ball through traditional region-matching after it has been found through motion detection, or in situations where no ball can be identified (e.g. when a robot is next to the ball, the 1-4 pixels occupied by the ball easily disappears into the robot's region). Where motion detection alone does not provide an obvious ball, a high-resolution interpolation is performed on an area eight times the ball's size in its predicted location, this area is color thresholded and segmented, and scanned for regions of the correct size and dimension. This does require a predefined color, but serves only to be used when motion detection fails to provide a strong enough match.

In order to support the longer-term autonomous operation of the system, every 180 frames the remaining unused runs are used to update the background image. In practice this means that shifts in lighting and slight changes in camera position are accounted for over time. The infrequency of updates quells persistent noise while allowing moving objects to avoid becoming part of the background.

6 Evaluation

We have implemented the algorithms described in this paper in the global vision system *Ergo*. Ergo is currently used in our laboratory for robotic soccer, and also other field-based domains for both teaching and research and is available for download at http://avocet.cs.umanitoba.ca. Ergo has been used extensively under what would be considered normally variable lighting conditions in a laboratory (shadows from passers-by, flickering fluorescent lighting, reflections from nearby objects). In all cases, frame rates of 28-35fps can be maintained while tracking eight robots and the ball using the techniques described above.

We have also examined the performance of this system under more extreme lighting conditions, such as the directional, uneven lighting shown in Figure 4. Even under conditions such as these, which would never be considered acceptable for a robotic soccer game, the approach described here proves robust enough to identify the majority of the objects in the system. Indeed, the only two it fails to identify are the robot whose hat is almost entirely wiped out by glare, and the robot that remains in the dimmest corner of the screen. Even in this situation, the vision server ran at 29 fps.

7 Conclusion

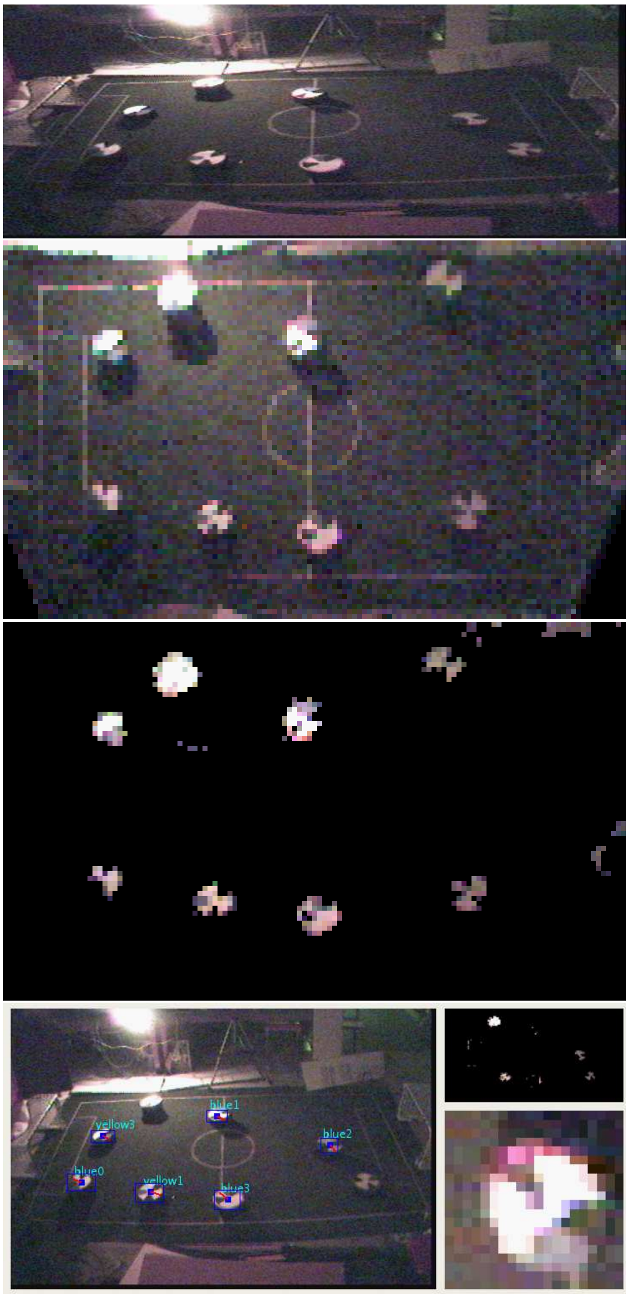
In this paper we have described a vision system that can be used for robotic soccer and which does not require any predefined colors, nor the need for an overhead camera. Colors are employed only to satisfy the RoboCup demand that each team be marked with a colored marker, and to verify the position of a ball under very error prone conditions.

One shortcoming in our system is the current reliance on calculating the centre of a robot based on the width of the black segments. Our current implementation causes the centre to drift based on inaccuracies in this estimation. This drift is corrected by motion detection, but as every object we can identify prior to motion detection decreases the computation required for motion detection itself, it is in the best interests of speed to improve upon this.

References

[1] David Ball, Gordon Wyeth, and Stephen Nuske. A global vision system for a robot soccer team. In *Proceedings of the 2004 Australasian Conference on Robotics and Automation (ACRA)*, 2004.

[2] Jacky Baltes. Doraemon: Object orientation and id without additional markers. In *2nd IFAC Con-*



	Name	Type	x	y	dx	dy	theta	Found
1	blue0	Robot::6	310.73	394.00	-0.04	-0.03	2.72	100%
2	blue1	Robot::4	1280.95	1172.71	0.01	0.00	-0.76	100%
3	blue2	Robot::3	2149.56	729.63	-0.00	-0.00	-1.26	95%
4	blue3	Robot::1	1373.85	210.70	-0.00	-0.00	2.25	100%
5	yellow0	Robot::8	2330.39	320.82	0.02	0.02	-2.32	
6	yellow1	Robot::1	843.32	293.47	-0.00	0.00	-0.33	94%
7	yellow2	Robot::1	2320.83	309.00	0.01	-0.00	-2.29	
8	yellow3	Robot::3	342.03	933.63	0.00	0.00	-0.69	100%

Figure 4: A situation with very poor lighting conditions (top), the interpolated and motion-detected images (middle) and Ergo output (bottom)

ference on Mechatronic Systems. American Automatic Control Council, December 2002.

[3] Jacky Baltes and John Anderson. Interpolation methods for global vision systems. In *Proceedings of RoboCup 2004*, Lisbon, Portugal, July 2004.

[4] J. Bruce, Tucker Balch, and Manuela Veloso. Fast and inexpensive color image segmentation for interactive robots. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00)*, volume 3, pages 2061 – 2066, October 2000.

[5] James Bruce and Manuela Veloso. Fast and accurate vision-based pattern detection and identification. In *Proceedings of ICRA-03, the 2003 IEEE International Conference on Robotics and Automation (ICRA '03)*, May 2003.

[6] Antoine Manzanera and Julien Richefeu. A robust and computationally efficient motion detection algorithm based on sigma-delta background estimation. In *Proceedings Indian Conference on Computer Vision, Graphics and Image Processing*, December 2004.

[7] Roger Y. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–374, Los Alamitos, CA, June 1986. IEEE Computer Society Press.