Intelligent Global Vision for Teams of Mobile Robots

Jacky Baltes and John Anderson
Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, R3T 2N2 Canada
Email: jacky,andersj@cs.umanitoba.ca

1 Introduction: Global Vision

Vision is the richest of all human senses: we acquire most of our perceptual information through vision, and perform much of our own vision processing with little conscious effort. In contrast, dealing intelligently with the enormous volume of data that vision produces is one of the biggest challenges to robotics. Identifying an object of interest in a random camera image is a difficult problem, even in domains where the number of possible objects is constrained, such as robotic soccer. This difficulty increases in magnitude when attributes of interest involve change such as movement, and thus require both state information and examining change in visual images over time. Visual analysis also involves many subtle problems, from very low-level issues such as identifying colours under changing lighting conditions, to higher-level problems such as tracing the path of an object under conditions of partial occlusion, or distinguishing two objects that are next to one another but appear as one larger object.

Humans deal with vision through very specialized massively-parallel hardware, coupled with a broad range of common-sense knowledge. Neither of these is currently feasible to apply to a mobile robot platform. While mobile processors are becoming more powerful, we are still far below what is required to process vision at a human level, and common-sense knowledge has always been one of the most difficult problems associated with artificial intelligence. Vision on mobile robots is thus largely about producing heuristic solutions that are adequate for the problem domain and allow the available hardware to process vision at the frame rate required.

Vision in robots may be divided into two types. The first of these, *local vision*, involves providing a first-person perspective using a camera attached to the robot itself. Local vision shares many of the same problems that humans deal with in visual analysis, such as determining whether motion in a visual stream is due to the motion of objects captured by the camera, or the motion of the agent itself (ego motion). In a local vision setting, each member of a team of robots receives its own unique camera feed and be responsible for analyzing and responding to that feed.

There are a number of reasons why local vision may not be preferable in a given application, the foremost of which is the heavy requirement for computational resources. When each robot must perform its own visual processing, it must be able to carry enough on-board processing to do so, which may not be possible in smaller robots or applications with competing resource needs. The fact that each robot is entirely responsible for its own vision will also mean that there will be significant redundancy in processing across a team of robots in many applications as well. Local vision may also be undesirable in applications where a large number of very simple robots may be able to do the job of a few complex robots, in environments where shared vision is amenable (that is, where a unique perspective for each individual is unnecessary), and in educational environments where it is desirable to separate the problems of computer vision from the rest of robotics. In these domains, the second form of vision, *global vision*, is often preferred. Global vision provides a single third-party view to all members of a robot team, analogous to the view of a commentator in a soccer game.

Global vision shares many of the problems associated with local vision. Objects of interest must be identified and tracked, which requires dealing with changes in appearance due to lighting variation and perspective. Since objects may not be identifiable in every frame, tracking objects across different frames is often necessary even if the objects are not mobile. The problem of identifying objects that are juxtaposed being as one larger object rather than several distinct objects, and other problems related to the placement and motion of objects in the environment, are also common.

In domains such as robotic soccer, where pragmatic real-time global vision is large part of the application, many of the more difficult problems associated with global vision have been dealt with through the introduction of artificial assumptions that greatly simplify the situation. The cost of such assumptions is that of generality: such systems can only operate where the assumptions they rely upon can be made. For example, global vision systems for robotic soccer (e.g. (Bruce and Veloso, 2003;

Browning et al., 2002; Simon et al., 2001; Ball et al., 2004)) generally require a camera to be mounted perfectly overhead in order to provide a simple geometric perspective (and thus ensure that any object is the same size in the image no matter where in the field of view it appears), simplify tracking, and eliminate complex problems such as occlusion between agents. If a camera cannot be placed perfectly overhead, these systems cannot be used. Such systems also typically recognize individuals by arrangements of coloured patches, where the colours (for the patches and other items such as the ball) must be pre-defined, necessitating constant camera recalibration as lighting changes. Such systems can thus only operate in environments where lighting remains relatively consistent.

While such systems will always be applicable in narrow domains where these assumptions can be made to hold, the generality lost in continuing to adhere to these assumptions serves to limit the applicability of these approaches to harder problems. Moreover, these systems bear little resemblance to human vision: children playing with remote-controlled devices, for example, do not have to climb to the ceiling and look down from overhead. Similarly, human vision does not require significant restrictions lighting consistency, nor any specialized markings on objects to be tracked. In order to advance the state of the art in robotics and artificial intelligence, we must begin to make such systems more generally intelligent. The most obvious first steps in this direction are considering the assumptions necessary to make a global vision system operate, and then to find ways of removing these.

Our approach to real time computer vision arises from a desire to remove these assumptions and produce a more intelligent approach to global vision for teams of robots, not only for the sake of technological advancement, but from a pragmatic standpoint as well. For example, a system that does not assume that a camera has a perfect overhead mount is not only more generally useful, but requires less set-up time in that a perfect overhead mount does not need to be made. Similarly, an approach that can function in a wide range of lighting conditions saves the time and expense of providing specialized lighting for a robotic domain. Over the past six years, we have developed a series of real-time global vision systems that, while designed for the robotic soccer domain, are also generally useful anywhere global vision can be used. These systems have been used in RoboCup and FIRA robotic soccer competitions by ourselves and other teams, and have also been employed in such applications as robotic education and imitation learning. All are open source, and can be easily obtained by the reader for use or as a basis for further research work (Baltes and Anderson, 2006).

Each of the systems we have developed deals with some of the assumptions normally associated with global vision systems, and thus produces a more generally intelligent approach. This Chapter overviews the work necessary to deal with these assumptions, and outlines challenges that remain. We begin by examining the steps necessary to deal with a more general camera position, how objects can be tracked when the camera is not perfectly overhead, and how an overhead view can be reconstructed from an oblique camera capture. This necessitates dealing with objects that are occluded temporarily as robots move around on the field, and also requires dealing with three dimensions rather than two (since the height of an object is significant when the view is not a perfect overhead one). We then turn to dealing with assumptions about the objects being tracked, in order to minimize the need for recalibration over time, and to make global vision less vulnerable to problems of lighting variability. We examine the possibility of tracking objects using only the appearance of the object itself, rather than specialized markers, and discuss the use of machine learning to teach a global vision system about the objects it should be tracking. Finally, we examine removing the assumption that specific colours can be calibrated and tracked at all, in order to produce a vision system that does not rely on perfect colour calibration to recognize objects.

2 Doraemon: Real-Time Object Tracking without an Overhead Camera

DORAEMON (Anderson and Baltes, 2002; Baltes, 2002) is a global vision system that allows objects to be tracked from an oblique camera angle as well as from an overhead view. The system acts as a server, taking frames from a camera, and producing a description of the objects tracked in frames at regular intervals, sending these over a network to clients (agents controlling robots, for example) subscribing to this information stream. Figure 1 is a sample visual frame used as input to DORAEMON to illustrate the problems involved in interpreting visual images without using a perfect overhead viewpoint. The image is disproportionate in height because it is one raw field from the interlaced video stream provided by the camera. It is easy to see that colour features are hard to extract, in part because the shape of coloured patches are elongated by the visual perspective, and in part because colour is not consistent across the entire image.

In order to be able to track images from an oblique angle, a calibration must be provided that allows an appropriate translation from a particular pixel in a visual frame to a coordinate system in the real world. The calibration process used by DORAEMON, described in detail in (Anderson and Baltes, 2002), utilizes the well-established Tsai camera calibration (Tsai,

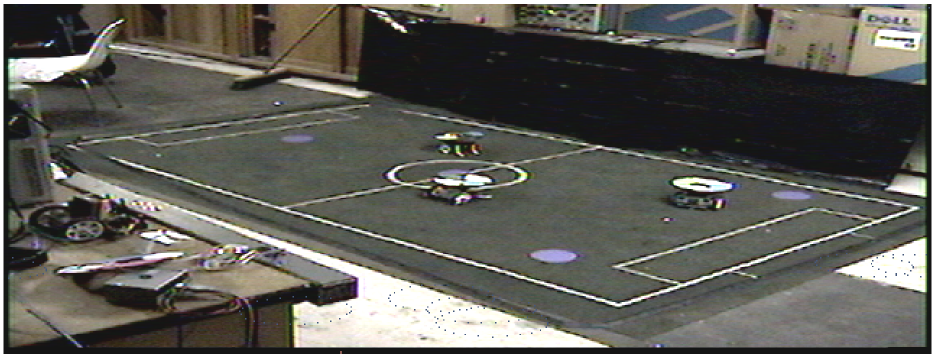


Figure 1: A sample visual frame taken from an oblique angle.

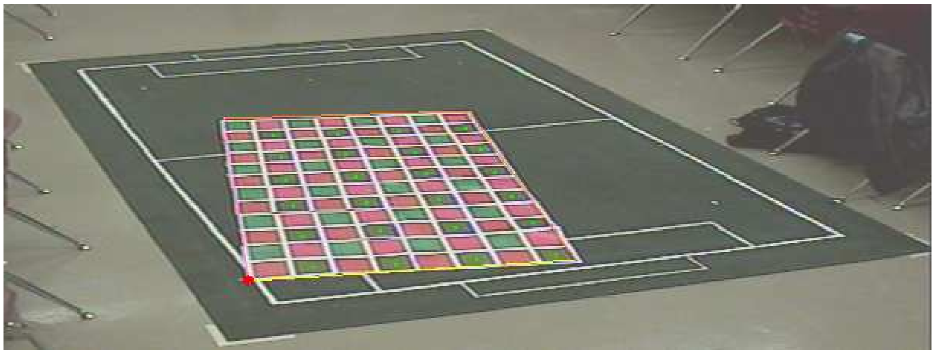


Figure 2: Tsai Camera Calibration used in Doraemon

1986), which can compute a camera calibration from a single image. This method computes six external parameters (the X, Y, and Z coordinates of the camera position, and angles of roll, pitch and yaw) and six internal parameters using a set of calibration points from an image with known world coordinates. This requires a set of coordinates to be imposed on the world via a sample visual image. Since Tsai calibration normally requires at least 15 calibration points (i.e. points with known X,Y coordinates), a calibration carpet with a repetitive grid pattern is used to easily provide a significant number of points. The known grid size is input to the system, and the coloured squares can be then be selected by the user and the calibration points obtained from the square centres (Figure 2). Even using an oblique view of the playing field, the calibration results in object errors of less than 1 cm. To make calibration more flexible, we also define a rotation matrix on the field that allows the calibration to be adjusted (for example if the camera shifts during play) without having to recalibrate using the carpet.

Objects in DORAEMON are identified by the size and arrangement of coloured patches. The simplest objects may be simply a single coloured area of a given size - e.g. a ball might be described as an orange item 5cm in diameter. More sophisticated items (e.g. individual robots) are identified using unique arrangement of coloured patches on the top surface, as shown in Figure 1 (e.g. a blue patch for the front of all robots on one team, with an arrangement of other colours uniquely identifying each team member). The system is thus heavily dependent on accurate colour models. DORAEMON uses a sophisticated 12 parameter color model that is based on red (R), green (G), and blue (B) channels as well as the difference channels red-green (R-G), red-blue (R-B), and green-blue (G-B). The channel differences are less sensitive to lighting variations than the raw channels, and allow more robust colour recognition than the raw channels alone. While there are other models that are less sensitive to brightness, (for example, HSI), this approach attempts to balance sensitivity with computational resources. The channel differences are similar to the hue values used in HSI, for example, while this model is less computationally expensive.

Colours of interest are defined using a colour calibration procedure, during which areas of the visual image intended to be matched to a particular named colour are selected. This reliance on colour, like that of other global and local vision systems, limits generality and forces recalibration to be performed when lighting conditions change. Moving beyond this dependence

```
7 6188 0.000290976 ; #defined objects, frame#, time diff. from last frame
1605.82 -708.394 1321.44 ; x, y, z coordinates of camera
2 spot1 Found 1232.5 416.374 0 0 0 0 ;information about each defined object
2 spot2 Found 1559.22 417.359 0 0 0 0
2 spot3 Found 1260.55 812.189 0 0 0 0
2 spot4 Found 902.726 1002.43 0 0 0 0
2 spot5 Found 746.045 735.631 0 0 0 0
1 ball1 Found 1677.99 1205.55 50 0 -2.75769 1.19908
0 car54 Found 1783.53 873.531 100 2.63944 1.47684 -6.49056
```

Figure 3: A sample message from Doraemon

on colour will be described in Sections 3 and 4. Once colours are defined, camera images can be colour thresholded and particular colour patches can be recognized in an image.

The size of any patch of color can be determined by its position on the field, since the perspective of the field is known through calibration. This still requires a model describing the arrangements of the coloured patches marking objects as well as their heights above the playing field, since, for example, an object that is 50cm tall will have markers of the same height appear differently in the camera image than that of an object that is flush with the playing field surface. The descriptions of the size, colour, arrangement, and height of objects to be recognized are described in a configuration file.

Each frame is colour thresholded and the recognized patches are matched against the size and configuration information provided. Not every object will be recognized in every frame, since lighting fluctuations, for example, may make some spots difficult to recognize across the entire field area. To compensate for this, the locations of recognized objects in previous frames are used both to infer likely positions in future frames and to calculate the speed and orientation of motion of tracked objects.

Occlusion in robotic soccer is normally not an issue for tracking robots, even with an oblique camera, since the markers are on top of the robots and are thus the highest points on the field. Occlusion certainly happens when tracking the ball, however, and is also possible in any tracking scenario where obstacles on the field could be taller than robots. There is also the possibility that robots may abut one another, presenting a display of coloured patches that is similar to a different robot altogether, or presented in such a way that no one robot is easily recognizable. These situations are dealt with by tracking objects over time as well - an object may be lost temporarily as it passes behind an obstacle, or may be more momentarily unrecognized due to abutting other tracked objects - because objects are intended to be in motion, such losses will be momentary as new information allows them to be disambiguated.

Doraemon transmits information about tracked objects (position, orientation, velocity) in ASCII over ethernet to any client interested in receiving it. A sample message is shown in Figure 3.

The first line of each message contains the number of objects that video server is configured to track, followed by the video frame number and time difference in seconds between this message and the previous one. The next line contains the x, y, and z coordinates of the camera, and following this is a line for each object being tracked. Each of those lines consists of a numeric object class (e.g. a ball, robot, etc.), the unique defined identifier for the object, whether the object was located in the current frame or not, the x, y, and z coordinates of the object, the orientation of the object in radians, and the velocity of the object in mm/second in the x and y dimensions.

Doraemon was later extended (Baltes and Anderson, 2005) to provide its own reconstructed overhead view through interpolation, which allows the perspective distortion created by the oblique camera angle to be corrected, allowing objects to be more accurately tracked. While this interpolation does slow down the vision process, it provides a remarkable improvement in image quality. As an example, Figure 4 shows Doraemonś reconstruction of the oblique view shown in Figure 1.

Doraemon takes several steps beyond global vision systems that maintain a fixed overhead camera in terms of being able to deal with the real world. It is quick to calibrate and simple to recalibrate when this is necessary (e.g. due to camera shift or changing lighting during use). However, there are still significant assumptions about the domain that affect the system's generality. Doraemon is heavily dependent on good colour models, something that is not easy to maintain consistently over time in real-world domains without recalibration, and relies on a fairly naive model for dealing with occlusion. Dealing with these assumptions is the subject of the remaining sections in this Chapter.

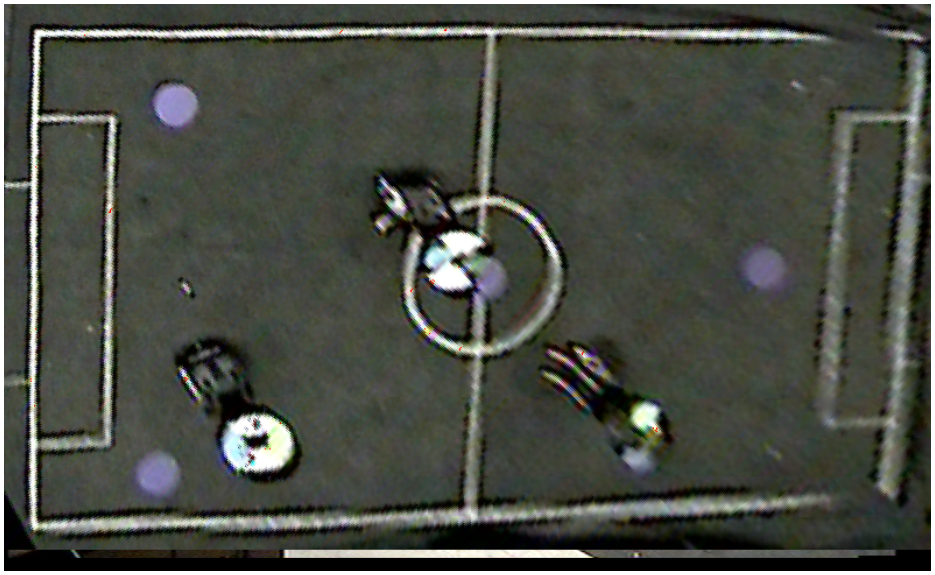


Figure 4: Doraemon's overhead reconstruction (using average gradient interpolation) of the camera image shown in Figure 1.

3 Ergo: Removing Dependence on Predefined Colours

The reliance on colour thresholding by both DORAEMON and related systems places some severe restrictions on the applicability of a global vision system. Not only are lighting variations a problem, but the colours themselves must be chosen so that there is enough separation between them to allow them to be distinguished across the entire field of play, and the quality of the camera used is also a major issue. In practice, even with the extra colour channels employed by DORAEMON tracking is practically limited to around 6 different colours by these restrictions.

To increase the applicability of global vision to a broader array of real-world tasks, as well as to increase the robustness of the system in robotic soccer, we focussed on two major changes in approach: the use of motion detection to focus on areas of interest in the field, and different methods of marking objects that deemphasize the use of colour. These and other extensions resulted in the next generation of our global vision system, known as ERGO (Furgale et al., 2005).

One additional pragmatic step was also necessary in ERGO in order to attain a comparable frame rate as that employed in the original DORAEMON: the resolution of interpolated images was decreased, in order that interpolation did not inordinately slow down visual analysis. The result of this introduced an additional challenge, in that a typical 5cm soccer ball would now occupy only a 1-4 pixel range in the reduced resolution, allowing a ball to easily be interpreted as noise (Figure 5).

Rather than performing direct colour thresholding of camera images, ERGO thresholds for motion across pixels in each frame compared to a background image. A number of common thresholding techniques (using pixel intensity and distance in colour space, with global and local thresholds) were experimented with under poor lighting conditions and with common domain elements such as the presence of field lines and aliasing between camera frames. None of the common approaches were adequate in avoiding losing information from dark parts of the image while removing noise from lighter portions. In the end, an adaptation of $\Sigma\Delta$ background estimation (Manzanera and Richefeu, 2004) was employed, which provides a computationally inexpensive means of recursively estimating the average color and variance of each pixel in a camera image.

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

Detecting motion involves setting a threshold above which variation across pixels will be considered to be motion. In experimenting with this, it was found that increasing a global threshold enough that all noise would be eliminated also had the effect of eliminating any object of the size of a typical robotic soccer ball, since the size of such an object in the image (¡=4

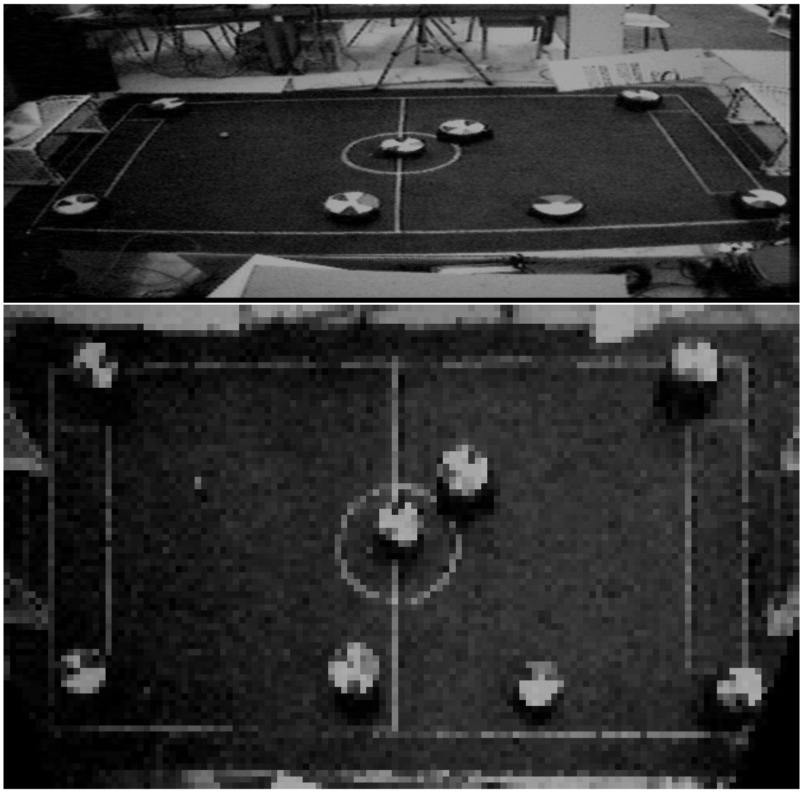


Figure 5: Captured field and corresponding low-resolution interpolated image in Ergo. Note that the ball is easily visible in the former image, but blends with noise on the field lines in the latter.

pixels) is easily interpreted as noise. To deal with this, a means was required to consider variation more locally and eliminate noise, while still being able to pick up the motion of small objects, and so a combination of local and global thresholding was employed. A threshold is set for each pixel by examining the variance for each pixel in the background image, then apply a convolution (1) in order to consider a pixel's variance across its 9-pixel neighbourhood. This local threshold is then scaled by a global threshold. To detect motion, each incoming image has its sum-squared error calculated across all pixels against the background image, the same convolution is applied to the result, and each value is compared to its corresponding pre-computed threshold. The use of the convolution has the effect of blending motion in small areas to eliminate noise, while making the movement of small objects such as the ball more obvious by also considering small changes in neighbouring pixels.

This thresholding process causes motion to be separated from the background, after which the region growing algorithm of Bruce et al. (2000) is employed to generate a list of regions to match against the descriptions of objects that Ergo is tracking.

Since Doraemon relied on patterns of coloured blobs to identify moving objects such as robots, a change in pattern representation was necessary in Ergo in order to remove the dependence on predefined colours. The two basic requirements of a representation are the determination of identity and orientation (since the remaining item of interest, velocity, can be obtained through knowing these over time). Previous research (Bruce and Veloso, 2003) has shown that asymmetrical patterns can be used to allow a range of objects can be identified with fewer colours, and these ideas were extended to develop a representation and associated matching mechanism for tracking objects while minimizing the need for predefined colours.

The marking approach designed for Ergo divides the marker for a robot (or similar moving object) into a circular series of wedges (Figure 6). Two black wedges are the same on all robots, allowing a tracking algorithm to determine the labeled object's orientation. The remaining six wedges are marked with white and non-white (i.e. *any* colour other than white or black) to allow the determination of identity. Marking only two of these segments would allow up to twenty-one individuals

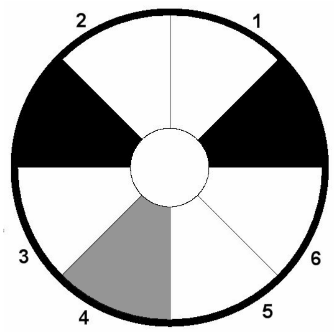


Figure 6: A new approach to labeling objects for tracking (Furgale et al., 2005): fixed black areas allow orientation to be recognized, while white and non-white values in locations 1-6 represent identity

to be identified uniquely (the centre is left open for a possible team identifier if desired).

An associated algorithm for identifying objects assumes that such a marking system is in use, and begins with a set of hypotheses of objects of interest, based on the regions of the camera image that have been flagged as motion. The original image is reinterpolated with a higher resolution in (only) three concentric circular strips of pixels (each 64 pixels long) around the centre of each region of motion. This allows enough high-resolution interpolated area to more accurately determine the marking pattern without the computational demands of large-scale interpolation. The mean is taken across these to reduce noise and error, resulting in a single array of 64 elements, providing an encoding for that region of motion that can be matched against the labeled pattern described above. To be able to match the pattern in this strip, two boundaries must be determined in this strip: the boundary between black and the marker that is neither black nor white, and the boundary between that and white. These boundaries are determined using a histogram of intensity values produced as part of the reinterpolation. The black-other threshold can be approximated based on the fact that any point near the centre will be 25% black. The other-white boundary is arrived at by starting a marker at the top of the range of the histogram, and then iteratively replacing that with that average of the weighted sum of the histogram counts above other-white and those below other-white. It is possible to avoid this process based on the pattern if a known pattern is being searched for, so it is not required in all cases.

Once these thresholds are available, the identification algorithm begins by looking at for the two black regions, and the average of the centre between these is the orientation. These wedges also provide the plane on which the pattern, and based on that plane the recorded centre of the object is refined. The remaining parts of the interpolated strip are then partitioned relative to the black wedges and the identification pattern can then be determined by counting the number of white wedges and the number of wedges that are neither white nor black.

This identification algorithm is very effective and computationally minimal, but is complicated in application by two factors. First, the list of regions of motion may be significantly larger than the number of objects to be tracked (due to extraneous movement by other objects, for example): large enough that this algorithm cannot process them all in real time in the data directed manner that would be ideal. Second, successful identification of an object relies on an accurate centre point. If two or more moving objects appear in close proximity to one another (or even partly occlude one another), motion analysis will view this as one large region of motion, with a centre that will not be helpful in identifying anything. This algorithm thus needs to be applied in a more goal-directed manner, and have some means of dealing with clumps of objects.

ERGO deals with these problems by tracking objects across images, which provides for a goal directed application of this algorithm. Prior to motion analysis, every object found in the previous frame predicts its position in the next image based on velocity and time difference. Some objects may thus be found very quickly, since their centre point will be predicted and can easily be confirmed using the identification algorithm. The area in the image occupied by object recognized during this phase is masked during motion analysis. This masking serves two purposes: it produces no hypothesis, since the object has already been dealt with, but it also may serve to remove one of a group of objects that may appear together in a moving region. Masking the area will then leave a smaller region and a smaller number of grouped objects (possibly only one, which can then be handled as any other object would).

For the remaining unrecognized objects, the region most closely predicted by each is selected. If an appropriate-sized region is found near the predicted location, it is passed to the identification algorithm, along with the hypothesized identity to speed up the identification process. This step, along with those detailed above, turns the identification process into a largely goal-directed one, and the vast majority of objects in any frame are recognized in this manner. Data-directed processing is still required, however, to deal with any objects that remain unidentified at this point.

There are realistically two possibilities for the remaining objects: a region of motion is outside the predicted area for the object, or it is part of a clump of objects occupying a larger region. To deal with the former, ERGO examines the sizes of all unexplained regions of motion, and if it is a size that could suitably match an object of interest, it is passed to the identification algorithm. In the case of multiple objects occupying the same space, the regions of interest will be those that are too large for any one object. If any of these regions were to contain more than one object, at least one recognizable object will be touching the edge of the region, and so the edge is where recognition efforts are focussed.

To analyze regions that could be multiple robots, extra samples are taken one object-radius in from the region's edge and obtain a set of encodings that should cross the centre of at least one object if multiple objects are in the region. From this, those that are at least one object diameter long are chosen, and the identification algorithm above is run on each of these using each pixel as the potential centre of the object. If any object is identified, it is masked from the region in the next frame, allowing further objects to be distinguished in subsequent frames. This could be repeated in the analysis of the same frame to distinguish further objects, but since ERGO can determine likely positions of unrecognized objects just as DORAEMON could in frames where some objects were unrecognized, this strikes a balance toward computational efficiency.

Not every object is large enough to be labeled using the scheme shown in Figure 6, nor do all objects need an encoding to uniquely identify them. In robotic soccer, for example, the ball is physically unique, and its nature does not require a pattern for identification. The use of motion tracking to distinguish an element as small as the ball has already been described. In frames where this motion tracking does not allow the ball to be found, the ball's location is predicted from the previous frame, and an area eight times the ball's size is scanned for regions of the correct size and dimension after colour thresholding. Colour thresholding here is simply used to distinguish regions at all given that motion detection has failed, and no predefined colours are employed.

These techniques allow ERGO to perform well under very challenging conditions. Figure 7 illustrates a screenshot from an extreme example, with lighting positioned across the viewing area, causing a wide disparity in brightness, and significant shadowing. Motion tracking is shown in the upper right, and the system output in the bottom of the image. All robots are identified except for one completely hidden in shadow, and the other in complete glare from the lighting source.

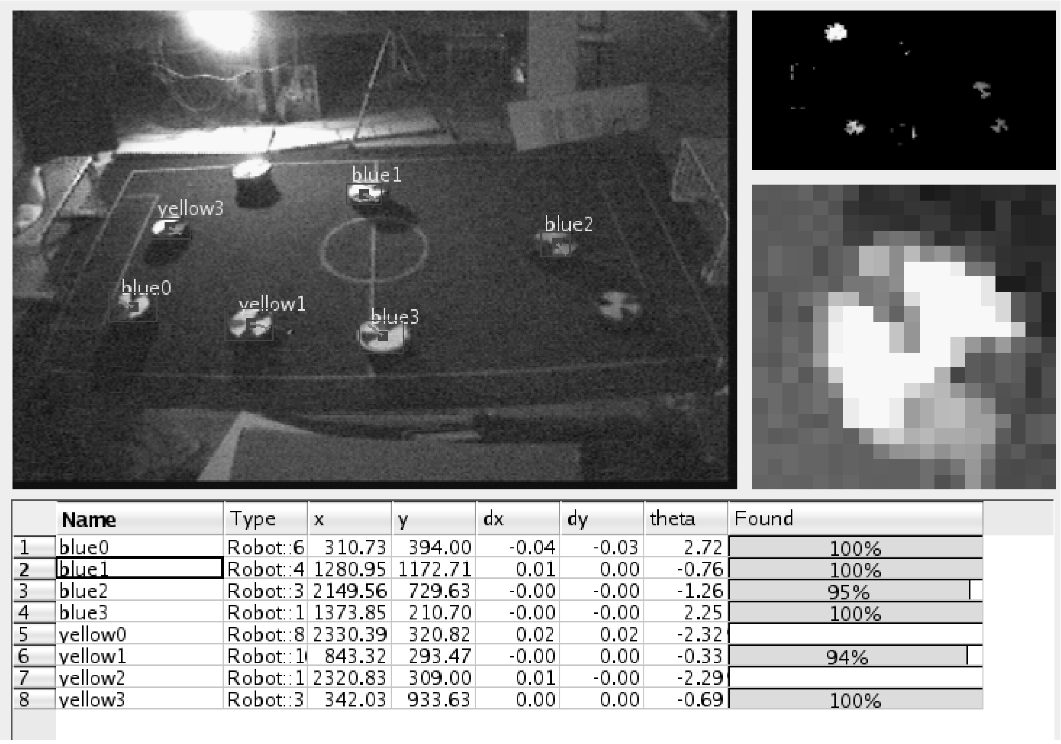
ERGO has gone a long way in making a global vision system more applicable to real-world situations, in that it has both removed the need for a fixed overhead camera as well as any predefined colours, and thus can operate across a much broader range of condition s than previous systems. There are still assumptions it operates under, the largest being that a pattern can be used to consistently identify objects that need to be tracked. In the remainder of this Chapter, we will explore the possibility of tracking objects without such patterns.

4 Removing Dependence on Predefined Patterns

The ability to move beyond from predefined colours or patterns for identifying objects is important in vision, for a number of reasons. From an immediate practical standpoint, scalability is always an issue. Even when using patterns without colour, there is a very finite amount of variation that can fit on a small pattern and be recognized reliably at a distance. While there are alternative approaches, such as just as arranging objects in predefined patterns before any movement begins and then tracking motion, such approaches can only operate for a short time before robots are misidentified as they move about. Once misidentified, there is no easy way to re-establish identity without stopping to do this manually.

The issue of generality is much more significant in the long term than scalability, however. While patterns are employed by humans during visual tracking (e.g. in soccer, teams wear structured uniforms involving both colour and pattern to distinguish themselves visually for the benefit of players and spectators), such patterns do not have to be pre-programmed. We need only watch an ongoing soccer game for few seconds to understand the pattern and be able to track it without any significant conscious effort. Humans can also switch between activities quickly, while equally quickly adapting to the details necessary for visual tracking in the new domain.

In order to make a computer vision system truly intelligent, we must work toward this level of generality by removing assumptions of predefined patterns and demonstrating similar adaptability to that observed in human vision. In order for this to be achieved in a vision system, one of two things must happen: either additional sources of information must be exploited to make up for that provided by assumed patterns, or the system must be able to adapt to patterns itself over time. Both of



	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 7: Using Ergo under very poor lighting conditions (Furgale et al., 2005)

these are significantly beyond the level of production vision systems at the present time, and represent some of the core ideas for improving this technology in future. In the following subsections, we present recent work in both of these areas, and then conclude by summarizing some of the issues yet remaining.

4.1 Object Tracking Based on Control Information

There are numerous techniques used by humans that can be exploited in an intelligent system for visually tracking objects. One of the most powerful can be seen any time a group of children operate remote-controlled vehicles. If the vehicles all look alike, a child quickly realizes that the one he or she is controlling can be identified by its response to the control commands being sent. While vision alone can be used to track the vehicle under control after it has been identified, when it is lost (e.g. has crossed paths with several other identical vehicles), this control information can be used to re-identify the vehicle. Such information can also be used to further confirm identity throughout the control process.

In recent years we have been working toward extending the abilities of our global vision systems by intelligently applying such control information. The original versions of DORAEMON and ERGO both maintain the identity and velocity of objects being tracked, in the form of a hypothesis with an associated measure of likelihood. As already discussed in Sections 2 and 3, this information is used to predict the future positions of moving objects in subsequent frames, to allow a more goal-directed tracking process and to account for objects when they cannot be recognized in every frame. If objects are no longer visually distinct, in that there is no predefined identification pattern, there may also no longer be any way to obtain orientation information visually in a single frame (depending on whether markings are present to provide such information). However, the addition of control information affords a better predictor of future object locations, because control commands are presumably the single most important factor in future movement. This same control information can also indirectly supply orientation information if it is not otherwise available, since the orientation is also obtainable based on the object's response to commands. Control information is still imperfect, since commands may not be carried out correctly or even be received properly by a mobile device, and unpredictable events (collisions, outside forces) can affect objects' future positions as well. However, such information should provide enough information to reliably support object tracking even where objects are not

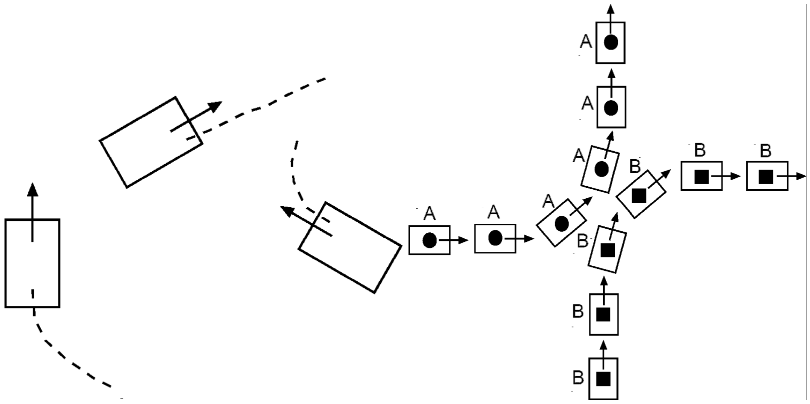


Figure 8: Situations easily resolved using control information. Left: identification is simple given control command (dotted line) and velocity (arrow). Right: Two robots appear to cross paths unless control information is known.

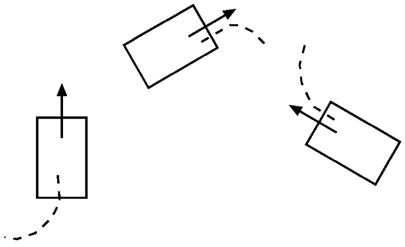


Figure 9: A situation where control information alone cannot resolve object identities.

otherwise visually distinct, much as it does for humans operating remote controlled vehicles.

In some situations, making use of control information is straightforward. Consider the situation depicted in the left side of Figure 8. Here, three robots are depicted with current orientation indicated with an arrow, and the motion that would result from the current command shown by a dotted line. Assuming they are identical, they can easily be differentiated in subsequent frames, since the motion each will exhibit is very distinct. If a trail is maintained over time, there are equally straightforward situations that would not be obvious using vision alone. In the right side of Figure 8, for example, two robots turn near one another, leading to the appearance of crossed paths. Using vision alone, misidentification during and after the turn is very likely, but this is easily resolved if the intended turns are known.

In other situations, this is not the case. Figure 9, for example, shows the same robots as the left side of Figure 8, but with different intended movements. Here, the two robots on the right cannot be differentiated based on control information, since they will both be moving similarly. The robot on the left in the same image can only be identified if orientation is known, since all robots are turning clockwise.

It can be seen from the latter example that current control information alone is not enough for reliable recognition. Even in a simple situation such as the first case, intended motion still may not be seen in future frames, because of the external factors. An obstacle might prevent movement, robots might collide, a command might not be received, or the robot might not even be perceived in the next frame. Control commands are a valuable source of information, but we must deal with the fact that they are uncertain. If the result of each individual command is viewed as a separate piece of evidence supporting identity, situations such as that shown in Figure 9 can be resolved using accumulated evidence over time (that is, an accumulated trace or trail supplied by an object's ongoing movement).

To extend our global vision systems to deal with the uncertainty involved with using command information, we experimented with moving from the *ad hoc* approach used in our production systems to a Bayesian approach (Baltes and Anderson, 2003a). This approach accumulates evidence in the form of traces of robot movement over time, and reasons probabilistically about the identities of robots given the traces of movement seen. The system uses evidence accumulated over a window of 100 frames, and computes an ongoing maximum likelihood hypothesis (h_{ML}) for each tracked object. The trace of the motion consists of the position, orientation, and state of the object, where state is one of turning clockwise, turning counter-clockwise, moving straight, or stopped.

$$P(h|D) = \frac{P(D|h) \cdot P(h)}{P(D)} \tag{2}$$

Bayes' formula ((2), where $P(h|D)$ is the posterior probability of hypothesis h given the observation D, $P(D|h)$ is the prior probability of observing data D given hypothesis h, and $P(h)$ is the prior probability of hypothesis h) is used to compute the maximum posterior hypothesis for each robot's identity given prior observations.

In this application, the hypotheses are the form *Commands for robot 1 are observed as trace 1* ,*Commands for robot 2 are observed as trace 2*, etc., where a trace is a set of positions and states over time. All traces are assumed to be equally likely, and so the prior probability $P(h)$ can be assumed to be uniform and thus ignored.

The system determines the maximum likelihood (ML) assignment of identities to robots that maximizes the posterior probability:

$$h_{ML} = \text{robot1} = (\text{trace1}, \text{command1}), ... = \text{argmax}_{h \in H} P(D|h) \tag{3}$$

To apply this calculation, $P(D|h)$ was determined through empirical observation to be approximately 0.7. In other words, the system detects and labels a stationary robot as stopped in 70% of the cases. To further simplify the calculation of the probabilities of the match of a command sequence and a motion trace, we assume that the probabilities of observing any feature are statistically independent.

The system computes the likelihood of all possible command traces to all observed traces and chooses the identity assignment that maximizes the likelihood for all robots. This approach was shown to work well in the soccer domain with a window of 100 frames and a small team of robots. The probability calculation as it stands is computationally expensive, which will result in limitations if scaled to teams of significant size, and so a significant element of future work will entail making such an approach efficient enough to use for large teams in real time. It is also possible that other applications might not require such a large evidential window, and thus may be less computationally demanding.

This is only one example of exploiting information to substitute for a pre-defined pattern. It is possible that other equally valuable sources remain to be discovered, and that this approach could be used in conjunction with other specific means of exploiting information as well. In terms of generality, however, it is still reasonably specific, in that it assumes there is a manageable range of discrete commands whose effects could be understood and recognized by vision system. While object tracking in human vision for specific domains such as soccer does appear to include information analogous to that exploited here (in the form of what a player would be expected to do given the object and rules of the game, for example), a more general approach to object tracking would be able to move beyond this knowledge as well.

4.2 Inferring Orientation Without Prior Knowledge

Completely removing any type of predefined marker results in an object recognition problem that is an order of magnitude more difficult than those described thus far, since features for recognition must be discovered as part of the recognition process itself. These features may at times be shadowed or otherwise occluded, making the discovery of such features more difficult than the recognition process itself. However, a system that has the ability to discover useful patterns will be more generally applicable than any current system, and may also be more robust, in that subtle patterns can be used by such a system that would be very difficult to even attempt to represent in an object description.

The fact that what is being offered to the system is a set of subtle features spread across noisy images points to the potential for using a more decentralized approach that can consider recognition across the image as a whole, as opposed to representing specific features. The natural choice for such a task is a neural-net based approach: the robustness this approach in the face of noisy and uncertain data is well-known (Mitchell, 1997).

Neural nets have been used extensively for image processing tasks in the presence of noise, from close-range applications such as face recognition (Mohamed, 2002) to remote sensing applications such as oil spill detection or land classification (Kubat et al., 1998; Schaale and Furrer, 1995). It is important to consider such prior work in the context of applications such as that described in this Chapter. Recognizing a face, detecting an oil spill, or classifying vegetation from single image, for example, is a much simpler recognition problem than dealing with the subtler issues of ongoing tracking over time that

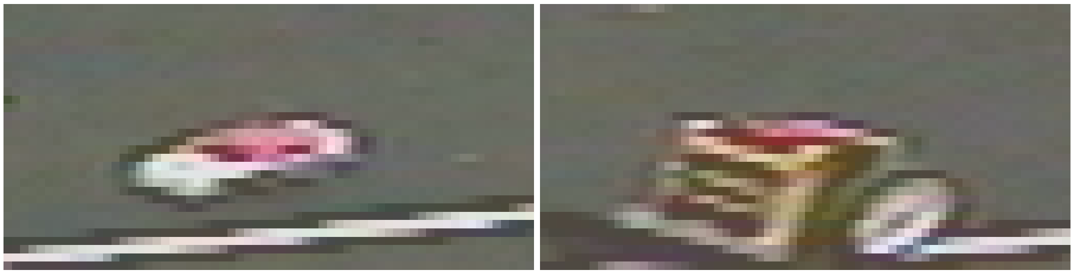


Figure 10: Sample images taken and annotated by Doraemon. The left image is a remote controlled toy car, while the right is a robot built from a Lego MindStorms kit.

have been presented in the previous sections. Neural networks have also been applied, though less extensively, to tracking information over time (e.g. (Cote and Tatnall, 1997)), and this also supports their use as a viable choice in real-time robot tracking.

At this point in time, there are no systems that can recognize and track moving objects in real time (i.e. in the same fashion as Doraemonand Ergo) adaptively and with no prior knowledge. In working toward this goal, however, we have been working with a subset of the general object recognition problem – recognizing orientation alone, using only images of robots as opposed to pre-defined markings – as a means to gauge the applicability of neural nets to this particular task and as a foothold for more general future work (Baltes and Anderson, 2003b).

Rather than using an artificial set of high-resolution images to train a network, we used actual data that was obtained by Doraemon. The original Doraemonwas modified (as a preliminary test for larger-scale motion detection in the development of Ergo) to examine the difference between frames to note likely locations for the coloured patterns the system tracks. Where a strong difference is noted, the sub-area of the image (64 x 32 pixels) is stored and the direction of change noted as a basis for the matching described in Section 2. These sub-images can be viewed on the interface in an enlarged fashion for colour calibration purposes, but in this case, this internal portion of the system serves to be able to gather a set of close-up images of robots over time. A set of training data annotated with estimated orientation can be thus be recorded if it can be assumed that the robot is always facing forward when moving. Examples of these annotated images from training datasets are shown in Figure 10. Note that lines on the field, as well as the fact that the images are quite small, both serve to promote noise. A training set of 200 images of each of two robot types was created (each network was ultimately trained to recognize orientation in only one type of robot).

The network employed to track robot orientations without patterns is a 3-layered feed-forward backpropagation network, as shown in Figure 11. Since the robot-sized images from Doraemon will be ultimately used as input for visual recognition of orientation, the input layer must ultimately receive these images. However, an RGB image of 64 x 32 pixels results in 6144 individual color-separated pixels. A neural net constructed with this many input nodes was attempted, but performance was found to be poor and training extremely slow, necessitating sub-sampling of the original image to allow fewer input nodes. This was done by averaging over each 4 x 4 pixel neighbourhood, resulting in 384 input nodes. The hidden layer is 32 nodes, or approximately 10% of the input layer. The number of hidden nodes was arrived at by experimentation (Baltes and Anderson, 2003b): using a learning rate of 0.3 and a momentum term of 0.2, a network with a 32 node hidden layer was the first that could learn 100% of a sample dataset after 2000 epochs (compared to 88% after 5000 epochs for the next best performing network topology, with 24 hidden nodes). The output layer is an encoding of the orientation angle discretized into 5-degree steps, resulting in 72 output nodes. The highest-strength output node is taken as the orientation classification.

For a test data set, we used one robot as a test subject and caused it to drive at random across the same field on which the training data were gathered. This introduced the same lines and other noise that were present in the training images. In addition, we placed stationary robots on the field so that the system was exposed to more varied examples of robot images, and to avoid overfitting.

To evaluate learning performance, we employed mean squared error (MSE), a common measure of error employed with neural nets. Mean squared error is the sum of the squared errors (SSE) over the output units, over the entire set of training examples (i.e. over one epoch), divided by the number of training patterns in the epoch. That is, MSE is the mean error for a given pattern.

One interesting element in object recognition using a sub-symbolic approach such as a neural network is the relative utility

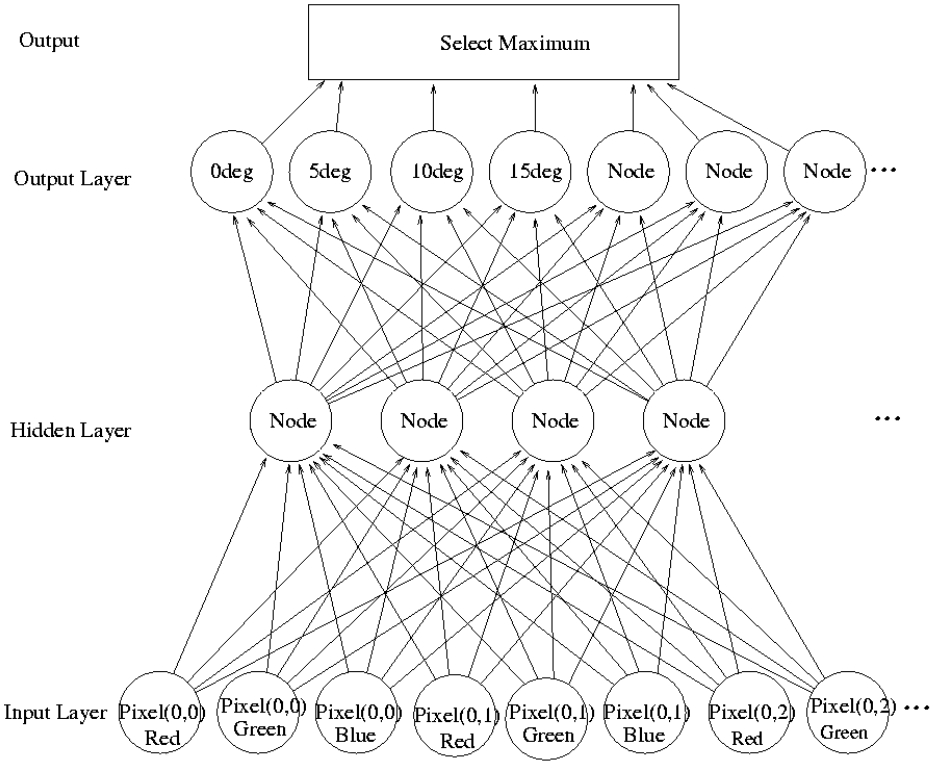


Figure 11: The Neural Network Architecture

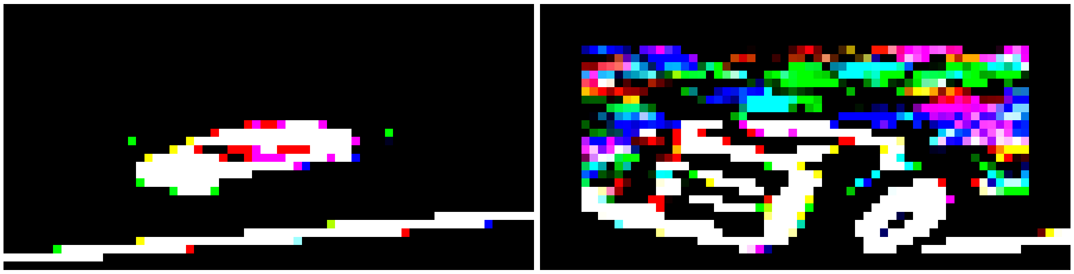


Figure 12: Training images from Figure 10 after 2 x 2 Sobel edge detection.

of the designer attempting to emphasize or likely useful information in images beforehand, as opposed to simply allowing the approach to operate completely unbiased. Since the edges in an image of a robot contain much information that is useful for orientation classification, we attempted to contrast the recognition process using the images already described, with one employing sub-sampled images that had 2 x 2 Sobel edge detection performed on them. Figure 12 illustrates the edge maps created by performing edge detection on the images in Figure 10. Since edge detection also removes colour information from the original image, fewer input nodes were needed in this case.

We ran a comparison on this network once the optimal number of hidden units was decided upon, comparing the accuracy and speed of learning using the toy car training data under three different input representations: 4 x 4 sub-sampled colour images described above, 2 x 2 sub-sampled edge-detected images, and 2 x 2 sub-sampled grey scale images. The third representation was chosen in order to see the effect of removing colour information alone. As in preliminary experimentation, a learning rate of 0.3 and a momentum term of 0.2 were used. In all cases, training data was randomly reshuffled after each

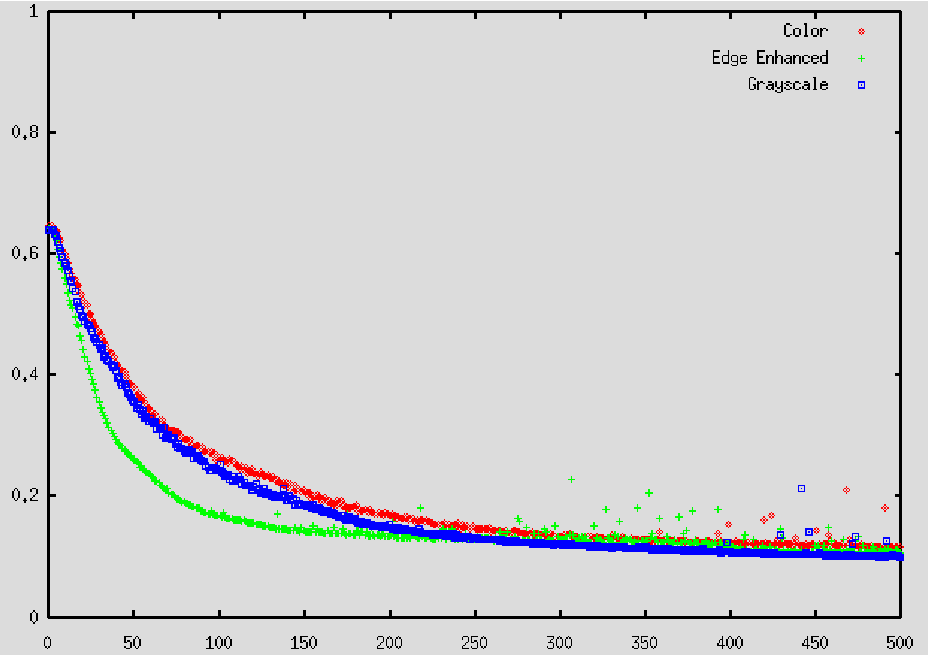


Figure 13: Evolution of the MSE over training epochs under different input representations, using the toy car image training set

epoch to avoid over-fitting the neural network to the specific sequence of training images.

The results of this (Figure 13) showed that the representation made little difference. Although edge detected images allowed faster MSE improvement over the first few hundred epochs ultimate performance was worse than other representations. The differences are not statistically significant, however, and all finish with an MSE of approximately 0.10.

The neural networks were then tested on accuracy of orientation classification. The results were also similar by representation. The network trained using colour images was able to classify 99% of all images within 5 degrees, while grey-scale or edge map trained networks classified 97% of all images correctly. There was no correlation between missed images over representations: that is, different images were misclassified by the three different representations.

Similar testing was done with Lego MindStorms robots, which as can be seen in Figure 10, have more distinct visual features. This lead to networks trained with colour and grey-scale images to finish training much earlier than edge-detected images (Figure 14). These two alternatives both terminated training early with 100% accuracy, while the network trained with edge-detected images still had only 93% accuracy after 5000 epochs.

These results seem to indicate that preprocessing and basic feature selection is not useful for a neural network, and may in fact decrease the performance. While this seems counter-intuitive, in retrospect the training images themselves seem to indicate that orientation is often about noting small pieces of evidence being combined into a consistent view, as opposed to extracting features such as edges. Edge-detection, while emphasizing some elements, loses many small details, especially with subjects such as the Lego robots, where much detail is present. Each pixel provides a small amount of information, but its relationship to other pixels makes this information important, and this relationship is diluted by preprocessing such as edge detection. This was fairly obvious in some cases: some images had shadows along one side, where this shadow was incorporated into the robot shape via edge detection, for example. Artificial neural networks, on the other hand, are especially well-suited to combining large amounts of input into a consistent view to deal with elements such as shadows.

We also examined the ability of these networks to generalize, by extracting 20 images (10%) at random from the set of 200 training images. After training the network on the training set minus these 20 images, the 20 unseen images were tested. These results were not as encouraging: the network as trained was unable to generalize well. Further investigation showed that the generalization ability was limited because there were few training examples for 5-degree turns compared to larger values.

These efforts focus on only one sub-problem of object recognition: orientation. The results could certainly be used in a

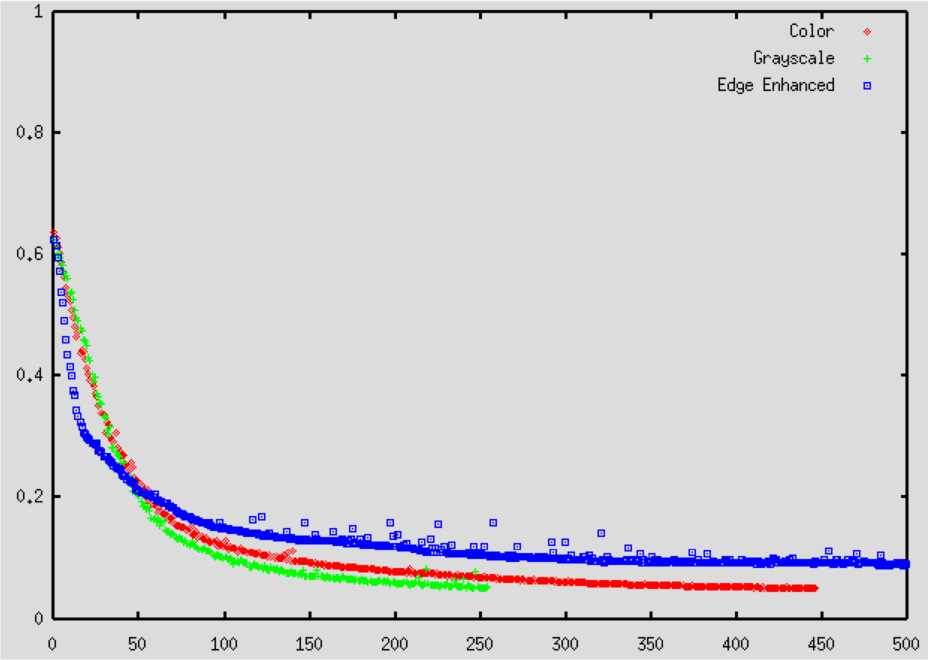


Figure 14: Evolution of the MSE over training epochs under different input representations, using the Lego robot image training set

production vision system, but this is still far from dealing with the larger identification and tracking problem. However, the results presented here do show that artificial neural networks are a promising approach to this problem.

One issue that will require significant work is that of training time. The work here was performed on a dual dual 1900+ MP Athlon system with 1 GB of RAM, and a training run took approximately 30 minutes. It is certainly conceivable to allow 30 minutes of observation before a system is used in some applications, but this would be unacceptable in others. Current neural network technology requires significant training time, and being able to classify images with very little training will ultimately require significant breakthroughs in many areas outside of computer vision. Another concern is the computational requirements of the neural network after training. Once trained, this same system could process a classification in around 0.07 msec, however, which would be fast enough to apply to a 5-on-5 robotic soccer game. Scaling this to larger teams would require additional resources or significant improvements in efficiency. One possibility to improve speed would be to extrapolate a small set of rules that approximate some of the knowledge in the neural net. Applying these in conjunction with a simpler network may be faster than calculating the output of the entire network employed here.

5 Conclusion

This chapter has reviewed some of the issues involved in creating pragmatic global vision systems. We have discussed the assumptions on which traditional systems are based, pointed out how these differ with the observed abilities of human vision, and described how these assumptions limit the applicability and generality of existing systems. We then described techniques that allow some of these assumptions to be discarded, and the embodiment of these techniques in our production global vision systems, DORAEMONand ERGO.

Both DORAEMONand ERGOare used in a number of ways. DORAEMONhas been in use every year by a number of teams from around the world in the F-180 (small-size) league at RoboCup. ERGOis the current global vision system in use in our own laboratories, and is currently being employed in a number of projects, such as imitation learning in groups of robots (Allen, 2007).

We have also described some of our recent work toward creating much more general global vision systems that take advantage of additional knowledge or adaptability in order to avoid the need for any type of predefined markings on objects. The latter work is very preliminary, but shows the potential for improved techniques to eventually be the basis for more general vision systems.

In working toward such generality today, there are a number of very important areas of immediate future work. Existing approaches to global vision are well-understood and immediately deployable. The fact that they rely heavily on elements such as the ability to recognize colour patches, for example, means that anything that can be done to improve these abilities will serve to improve existing systems. While systems such as Doraemonare already exploiting much in terms of maximizing flexibility while still assuming colours can be defined and matched, future work may still improve this further.

Any small steps that can be performed to wean existing systems away from their traditional assumptions will serve as a backbone for further future work. While Ergois a significant improvement over the abilities of Doraemon, for example, it still conforms to some traditional assumptions in terms of relying on predefined patterns, and instead exploits different mechanisms to be more flexible and offer a better performance in a wider array of domains. There will be many similar steps as we move to more general vision systems.

Any single tracking or identification technique has its limitations, and just as neither Ergonor Doraemonuse a single mechanism to identify and track objects, future systems will require a synergy of techniques. Attempting to leverage the strengths of techniques off of one another will always be an important part of future work in this area. In our own work, we are currently attempting to employ the addition of control knowledge to the sub-symbolic orientation recognition described in Section 4. For example, if we are uncertain of a robot's location and orientation at the current time, we can start with the robot's last known location/orientation at previous time, and constrain the potential solution set by the likely outcome of the most recent command sent to the robot.

The iterative steps taken in improving global vision are in turn a useful source of future work in improving application areas as well. For example, the work on recognizing orientation without markers described in Section 4 was undertaken as convenient sub-problem of the overall vision task useful in robotic soccer, in order to track a team's own players for control purposes. The ability to infer robots' orientation without prior knowledge, however, also allows a team to infer the orientation and identity of the opponent's robots. This in turn can allow for more sophisticated tactical decision making than would otherwise be possible. For example, robots that have a strong kicking device can be extremely dangerous. If an opponent's robot is oriented away from the current locus of activity, the situation is not as dangerous.

In both current and ongoing work, there is also a great need for improvements to computational efficiency. While computer power is always improving, the demands of more sophisticated techniques will always exceed this. While we have attempted in Ergo, for example, to have as much of the matching be done in a goal-directed fashion, data-directed processing is still required, and so there is still ample opportunity for improving the frame-rate in ergo through improvements in pattern-matching efficiency. In using control information to anticipate future movement, techniques that do not require the calculation of all possible robot assignments to all traces would be an enormous improvement.

Finally, it should be noted that despite the fact that we have emphasized global vision in this chapter, the techniques employed in object tracking and identification by Doraemon, Ergo, and the other work described here are all equally applicable to local vision. If I have a local vision robot playing a soccer game, the robot still must be able to track its teammates and opponents across its field of vision, and update an internal model of the state of play in order to make intelligent decisions. Thus advancement in technology in one area is immediately applicable to the other. Although it does not compare to the limitations of human vision, omni-vision (that is, vision based on a 360 image, usually done with a camera and a parabolic mirror) has become largely a standard in some local vision robotic soccer leagues, most notably the RoboCup middle-sized league. Such vision ultimately allows a reconstruction on a local basis that bears a strong analogy to global vision, especially once a camera is not placed overhead and issues such as occlusion and complex geometry come into play.

If readers are interested in using the work described here in their own future work, open-source code for Doraemon, Ergo, and other systems is available (Baltes and Anderson, 2006).

Acknowledgement The authors would like to thank all of the students in our laboratories in Manitoba and Auckland that have worked on global vision software in the past, including Paul Furgale and Ben Vosseteig, as well as all of the students in our robotics courses that stress-test this software and continue to drive ongoing development.

This work was funded in part by Natural Science and Engineering Research Council of Canada (2002 onward), and the University of Auckland in New Zealand (prior to 2002).

References

Jeff Allen. Imitation learning from multiple demonstrators using global vision. Master's thesis, Department of Computer Science, University of Manitoba, Winnipeg, Canada, 2007. (forthcoming).

John Anderson and Jacky Baltes. Doraemon user's manual. http://robocup-video.sourceforge.net, 2002.

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.

Jacky Baltes. Doraemon: Object orientation and id without additional markers. In *2nd IFAC Conference on Mechatronic Systems*, pages 845–850, Berkeley, CA, December 2002. American Automatic Control Council.

Jacky Baltes and John Anderson. Identifying robots through behavioral analysis. In *Proceedings of the Second International Conference on Computational Intelligence, Robotics, and Autonomous Systems*, Singapore, December 2003a.

Jacky Baltes and John Anderson. Learning orientation information using neural nets. In *Proceedings of the 2003 FIRA Congress*, Vienna, Austria, October 2003b.

Jacky Baltes and John Anderson. Interpolation methods for global vision systems. In Daniele Nardi, Martin Riedmiller, , and Claude Sammut, editors, *The Seventh RoboCup Competitions and Conferences*, Berlin, 2005. Springer Verlag.

Jacky Baltes and John Anderson. Doraemon, Ergo, and related global vision systems. http://robocup-video.sourceforge.net, 2006.

Brett Browning, Michael Bowling, James Bruce, Ravi Balasubramanian, and Manuela Veloso. Cm-dragons01 - vision-based motion tracking and heteregenous robots. In Andreas Birk, Silvia Coradeschi, and Satoshi Tadokoro, editors, *RoboCup-2001: Robot Soccer World Cup V*, pages 567–570. Springer-Verlag, Berlin, 2002.

James 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, Takamatsu, Japan, October 2000.

James Bruce and Manuela Veloso. Fast and accurate vision-based pattern detection and identification. In *Proceedings of Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-03)*, pages 567–570, Taipei, Taiwan, May 2003.

S. Cote and A. R. L. Tatnall. The hopfield neural network as a tool for feature tracking and recognition from satellite sensor images. *International Journal of Remote Sensing*, 18(4):871–885, 1997.

Paul Furgale, John Anderson, and Jacky Baltes. Real-time vision-based pattern tracking without predefined colors. In *Proceedings of the Third International Conference on Computational Intelligence, Robotics, and Autonomous Systems (CIRAS)*, Singapore, December 2005. URL `http://avocet.cs.umanitoba.ca/ andersj/Publications/pdf/ergoCIRAS.pdf`.

Miroslav Kubat, Robert C. Holte, and Stan Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30(2-3):195–215, 1998.

Antoine Manzanera and Julien Richefeu. A robust and computationally efficient motion detection algorithm based on sigma-delta background estimation. In *Proceedings of the 4th Indian Conference on Computer Vision, Graphics and Image Processing*, pages 46–51, Kolkata, India, December 2004.

Tom Mitchell. *Machine Learning*. McGraw-Hill, 1997.

Rein-Lien Hsu Mohamed. Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):696–706, 2002.

M. Schaale and R. Furrer. Land surface classification by neural networks. *International Journal of Remote Sensing*, 16(16): 3003–3031, 1995.

Mirk Simon, Sven Behnke, and Raul Rojas. Robust real time color tracking. In Peter Stone, Tucker Balch, and Gerhard Kraetszchmar, editors, *RoboCup-2000: Robot Soccer World Cup IV*, pages 239–248. Springer Verlag, Berlin, 2001.

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, Miami Beach, FL, June 1986. IEEE Computer Society Press.