A Global Vision System for Robotics Courses

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

Abstract

This paper describes our work on practical global vision systems (DORAEMON and ERGO). These vision systems have formed the basis of several undergraduate and graduate courses since 1998 and have consistently been improved to perform accurately and robustly over a wide range of applications. DORAEMON uses a sophisticated camera calibration method and colour model to remove the need for an overhead view of the world. ERGO minimized the use of colour information to provide more robust object recognition under varying lighting scenarios. Most recently, these video servers have been used to control robots in a shared virtual/physical world.

Introduction: Global Vision in an Educational Environment

Perception is the most difficult element to present realistically when educating students in hands-on robotics. While elements of mechanics and control can be nicely scaled down by using simplified robotic software, limiting perception severely limits the applications that can be developed. It is certainly possible to develop interesting robotics projects using simple perceptual devices: a single sonar, for example, can be used to avoid obstacles directly in front of the robot, while a light sensor can be used to give a basic goal for a robot, the sophistication of resulting applications will always be limited without vision. Vision is the richest of all human senses, but generates an enormous amount of data, and requires sophisticated algorithms to deal with issues such recognizing basic objects, let alone the sophisticated processing that humans do judge distances, deal with noise, and track objects over time. This presents two significant challenges to those wishing to use it as a basis for an undergraduate or high-school class dealing with robotics. First, making use of such a large volume of information and such sophisticated processing on inexpensive robots, and second, ensuring that vision can be employed by students without overwhelming them in complexity. Another set of challenges also arises from the standpoint of managing such an educational program, namely the setup that an ongoing vision system must undergo, together with ongoing maintenance and the effort required to adapt it to different problem-solving environments.

This paper presents some of our ideas on using vision in educational robotics, together with recent work on a system that can be used for undergraduate and high-school robotics classes, as well as for advanced research. Our approach begins by accepting that sophisticated visual processing is beyond the local capabilities of lower-level students using inexpensive robots. Currently available inexpensive platforms (e.g. Commercial PDAs) achieve only a very low frame rate when visual processing is run locally, so many popular applications such as robotic soccer would be out of the question, while common student lab equipment such as Lego MindStorms are too weak to do any local visual processing. From a student standpoint, the sophisticated algorithms for local vision are out of the question, and frameworks for local vision are not simple to adapt to new environments.

Like many robotic soccer leagues (e.g. the RoboCup F-180), we advocate the simplicity of using a global vision approach (where a single third-party view is provided to all members of a robot team, analogous to the view of a commentator in a soccer game). While most robotics leagues take this route to remove the local processing requirements, we find that using global vision allows us to introduce the ideas involved in computer vision, and allows students to see some of the issues involved in employing such systems in the real world, while drastically lowering the information load that would be required for local vision. The end result is that we can have systems that employ vision, using tools that are simple enough that students can eventually calibrate them and modify them for new domains themselves. Students can learn the rudiments of computer vision and benefit from having more interesting robotic domains to work in.

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 & Veloso 2003; Browning *et al.* 2002; Simon, Behnke, & Rojas 2001; Ball, Wyeth, & Nuske 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 & Anderson 2006).

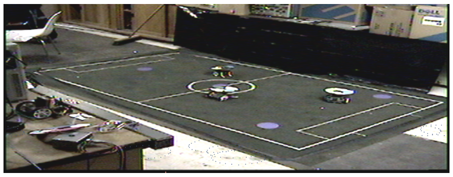


Figure 1: A sample visual frame taken from an oblique angle.

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.

Doraemon: Real-Time Object Tracking without an Overhead Camera

DORAEMON (Anderson & 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 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 appropri-

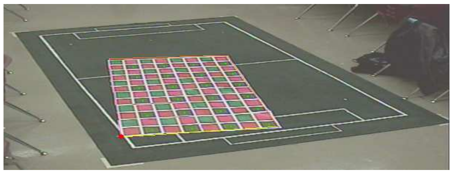


Figure 2: Tsai Camera Calibration used in Doraemon

ate 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 & Baltes 2002), utilizes the well-established Tsai camera calibration (Tsai 1986), which can compute a camera calibration from a single image. We have used the Tsai calibration since 1998 with good results on mono-plane calibrations. 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. Even using an oblique view of the playing field, the calibration results in object errors of less than 1 cm.

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.

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 (e.g., because of fluctuations in lighting). 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.

```
7 6188 0.000290976 ; #defined objects, frame#, time diff
1605.82 -708.394 1321.44 ; x, y, z coordinates of camera
2 spot1 Found 1232.5 416.374 0 0 0 0 ; object information
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

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 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.

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, Anderson, & Baltes 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 4).

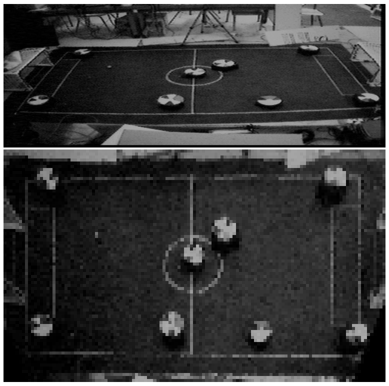


Figure 4: Captured field and corresponding low-resolution interpolated image in Ergo. Note that the ball is easily visible in the former image, but blends with noise on the field lines in the latter.

Rather than performing direct colour thresholding of camera images, Ergo thresholds for motion across pixels in each frame compared to a background image. An adaptation of $\Sigma\Delta$ background estimation (Manzanera & Richefeu 2004) is used, which provides a computationally inexpensive means of recursively estimating the average color and variance of each pixel in a camera image.

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 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 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. The individual motion pixels are then merged together into regions.

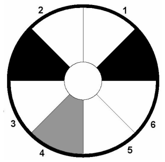


Figure 5: A new approach to labeling objects for tracking (Furgale, Anderson, & Baltes 2005): fixed black areas allow orientation to be recognized, while white and non-white values in locations 1-6 represent identity

Ergoalso introduced a new pattern representation. 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 & Veloso 2003) has shown that asymmetrical patterns can be used to allow a range of objects to 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.

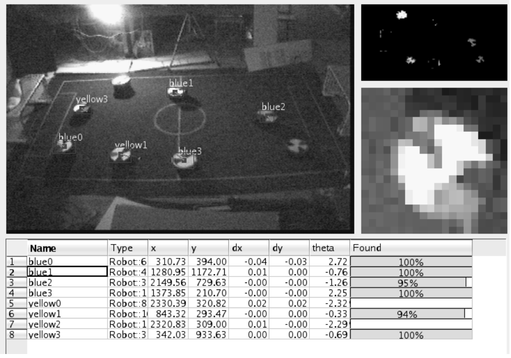
The marking approach designed for Ergo divides the marker for a robot (or similar moving object) into a circular series of wedges (Figure 5). Two black wedges are the same on all robots, allowing a tracking algorithm to determine the labeled object's orientation. The remaining six wedges are marked with 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 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.

Once these thresholds are available, the identification algorithm begins by looking 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).



	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 6: Using Ergo under very poor lighting conditions (Furgale, Anderson, & Baltes 2005)

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.

Not every object is large enough to be labeled using the scheme shown in Figure 5, 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 6 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.

Conclusion

This paper 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).

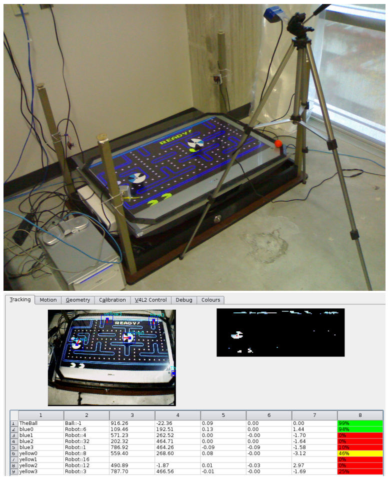
If readers are interested in using the work described here in their own future work, open-source code for DORAEMON, ERGO, and other systems is available (Baltes & Anderson 2006).

We are currently extending our environment to a mixed real virtual environment. A large TV mounted on the side is used as the playing field for small robots based on remote controlled toy tanks. The display of the TV is controlled by a world server, that can modify parts of the environment. For example, students are currently working on implementing a physical Pac-Man game, where physical robots perform as Pac-Man and ghosts and walls form the labyrinth, but pills, power ups and fruits are created virtually. An image of the new setup using ergo is shown below.

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).



	1	2	3	4	5	6	7	8
1	TheBall	Ball::-1	916.26	-22.36	0.09	0.00	0.00	99%
2	blue0	Robot::6	109.46	192.51	0.13	0.00	1.44	94%
3	blue1	Robot::4	571.23	262.52	0.00	-0.00	-1.70	0%
4	blue2	Robot::32	202.32	464.71	0.00	0.00	-1.64	0%
5	blue3	Robot::1	786.92	464.26	-0.09	-0.09	-1.58	10%
6	yellow0	Robot::8	559.40	268.60	0.08	-0.00	-3.12	46%
7	yellow1	Robot::16						0%
8	yellow2	Robot::12	490.89	-1.87	0.01	-0.03	2.97	0%
9	yellow3	Robot::3	787.70	466.56	-0.01	-0.00	-1.69	25%

Figure 7: A picture of our new shared virtual physical world setup and a fiew of the Pac Man game from Ergo

References

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

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

Ball, D.; Wyeth, G.; and Nuske, S. 2004. A global vision system for a robot soccer team. In *Proceedings of the 2004 Australasian Conference on Robotics and Automation (ACRA)*.

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

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

Browning, B.; Bowling, M.; Bruce, J.; Balasubramanian, R.; and Veloso, M. 2002. Cm-dragons01 - vision-based motion tracking and heterogenous robots. In Birk, A.; Coradeschi, S.; and Tadokoro, S., eds., *RoboCup-2001: Robot Soccer World Cup V*. Berlin: Springer-Verlag. 567–570.

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

Furgale, P.; Anderson, J.; and Baltes, J. 2005. Real-time vision-based pattern tracking without predefined colors. In *Proceedings of the Third International Conference on Computational Intelligence, Robotics, and Autonomous Systems (CIRAS)*.

Manzanera, A., and Richefeu, J. 2004. 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*, 46–51.

Simon, M.; Behnke, S.; and Rojas, R. 2001. Robust real time color tracking. In Stone, P.; Balch, T.; and Kraetszchmar, G., eds., *RoboCup-2000: Robot Soccer World Cup IV*. Berlin: Springer Verlag. 239–248.

Tsai, R. Y. 1986. An efficient and accurate camera calibration technique for 3d machine vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern*

Recognition, 364–374. Miami Beach, FL: IEEE Computer Society Press.