Exploiting Foveal Sensor Geometry and Environmental Constraints for Automatic Sorting of Recyclable Garbage[*]

Johan M. Lammens and Paolo Dario

Scuola Superiore S. Anna
Advanced Robotics Technology and Systems (ARTS) Laboratory
via Carducci 40, 57127 Pisa, Italy
e-mail {lammens|dario}@arts.sssup.it
URL http://www.cs.buffalo.edu/ lammens

Submitted to IEEE workshop on context-based vision

Abstract

We present our ongoing research into the use of foveal vision systems for industrial and other applications, in particular for the automatic sorting of recyclable garbage. After analyzing the constraints inherent in the problem we present a control architecture for the system, and discuss how a combination of sensor geometry and task constraints can lead to simple and fast, yet powerful, solutions. We illustrate this claim for two particular sub-problems: tracking objects, and categorizing them.

1 Introduction

This paper presents some work in progress in our lab, concerning the automatic sorting of garbage for recycling purposes in an industrial setting. The general scenario has a conveyor belt transporting garbage past a vision system and a manipulator, which together must sort the stream of garbage into categories like paper "tetrapacks", plastic PET bottles, etc. (Figure 1).

Some general assumptions constrain the solution in an important way:

1. Accuracy: the sorting must be highly accurate, since one "foreign" object in a bin contaminates the entire bin, from a recycling point of view (we might call this the *rotten apple* principle of recycling). It is better to have a small reject rate (object of category c_i not recognized as such and rejected) with 100% correct classification for the accepted objects than to have no rejects but wrong classifications (object of category c_i classified as belonging to c_j, $i \neq j$).

2. Speed of processing: the complete perception-action cycle must be fast, say ≥ 5 Hz, to be useful in an industrial setting. At the same time, the required hardware should be limited to industry standard platforms (PC) as much as possible.

3. We are using a vision system based on a log-polar organized retina-like CCD sensor [10, 7], as schematized in Figure 2.

[*] The research reported in this paper was supported by an HCM fellowship grant to the first author from the European Union, under the FOVEA program.

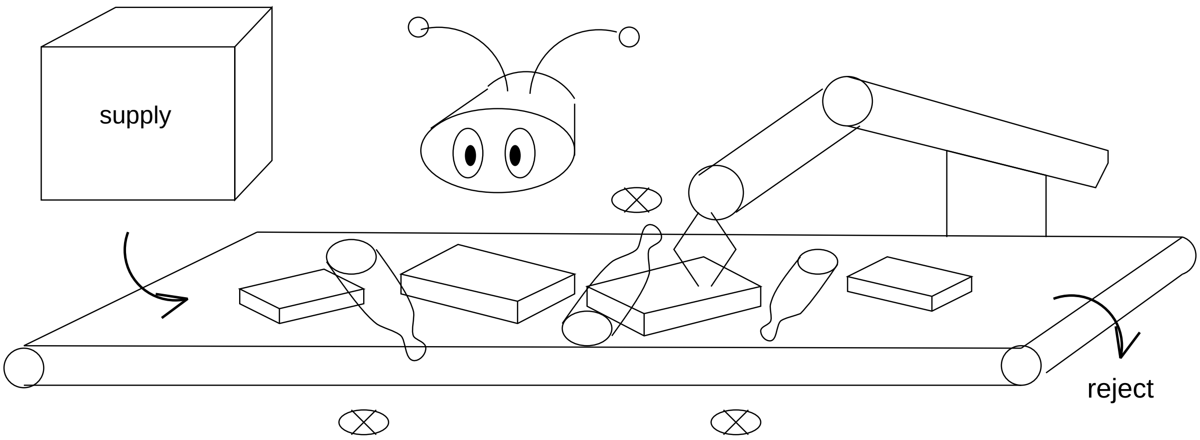

Figure 1: The general scenario for automatic sorting of recyclable garbage.

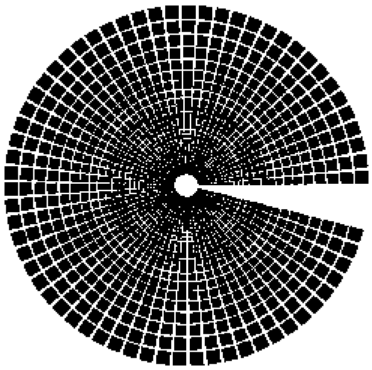

Figure 2: The geometry of the foveal CCD sensor. The very center (fovea) has a rectangular uniform resolution tesselation (not shown), the periphery consists of 30 concentric rings of sensor elements of increasing size. Each concentric ring has 64 sensor elements.

These assumptions are not entirely independent, as one of the main advantages expected from the development of the retina-like CCD sensor is the high speed of vision processing [8]. The sensors we use are mounted in the ARTS head: a 4 DOF stereo camera head using foveal CCD sensor chips. Mechanically, it features a 1,000 deg/s maximum angular velocity, and 20,000 deg/s maximum acceleration.

In developing vision algorithms for the sorting system, we make maximal use of the particular sensor geometry and the environmental (semantic) constraints inherent in the task. We present some initial results below. In the final version of the paper we hope to be able to provide further results, and a working prototype.

2 Decomposition into Tasks

To simplify our analysis and given the assumptions mentioned above, we can break the problem down into a number of tasks:

1. *Select objects.*

In due course, all objects must be selected, i.e. examined by the vision system and sorted into one of the bins. There may be a *reject bin* or a second belt for objects that can either not be reliably classified or that do not belong to any of the "interesting" categories, i.e. the ones to be recycled. The objects that were rejected because of insufficient reliability of classification could be processed by a second cascaded automatic sorting system (perhaps slower but more accurate), or dealt with by human operators. They could even be fed back onto the beginning of the belt, as things may work better the second time around (some Zen is in order).

2. *Foveate and track objects.*

   Once an object has been selected, it has to be kept centered on the sensor because of the sensor geometry and the vision algorithms used, as explained below. If the belt moves continuously, this implies tracking. If the belt moves in a start/stop fashion (one batch at a time processing), no tracking is necessary, but the start/stop action of the belt needs to be controlled based on visual or other feedback. In principle it might be possible to pick the objects out of a large bin or container, which would not require object tracking, but the vision processing and manipulator control would probably become too complicated in that case.

3. *Categorize the foveated object.*

   I.e. assign it to one of the interesting categories c_i, $1 \leq i \leq N$, the "uncertain" category c_u, $u = N+1$, or the "reject" category c_r, $r = N+2$. Categorization in this case comes down to *discrimination* among the a priori categories c_i, which is an easier problem than unconstrained *identification.* Since the categories are known a priori, it is possible to use a rather simple (and fast) template matching approach [6], insofar as shape discrimination alone is sufficient. For a feature based approach it suffices to find the simplest possible set of discriminating features (where *simple* is defined in terms of processing speed).

4. *Report the position and trajectory of the foveated object.*

   Since a manipulator is used to sort the objects, their position and trajectory needs to be known. These can easily be derived from the control parameters of the tracking behavior. If absolute (world) coordinates are required, visual scene markers could be used to avoid having to (re)calibrate the vision system. Depending on the type of manipulator, this position/trajectory feedback may not have to be very accurate. An alternative is visual feedback during manipulation (hand-eye coordination) which can be done without calibration, but this may take too much time. A second vision system could be mounted on the arm for this purpose (an eye-in-hand system).

5. *Pick up and deposit the foveated object into the appropriate bin.*

   This makes use of both the categorization and position/trajectory information. The details depend on the particular mode of manipulation implemented. Due to the high speed requirements this is a complicated problem in its own right, but we will not discuss it any further. One possible type of manipulator is depicted in Figure 3.

3 Control Architecture

We now discuss a preliminary design for the vision system control architecture. The basic design is a behavior-based subsumption-style one [3], with each task from Section 2 conceptualized as a parallel and independent process. The behaviors are as much as possible coupled through the environment, rather than through internal means, which simplifies incremental

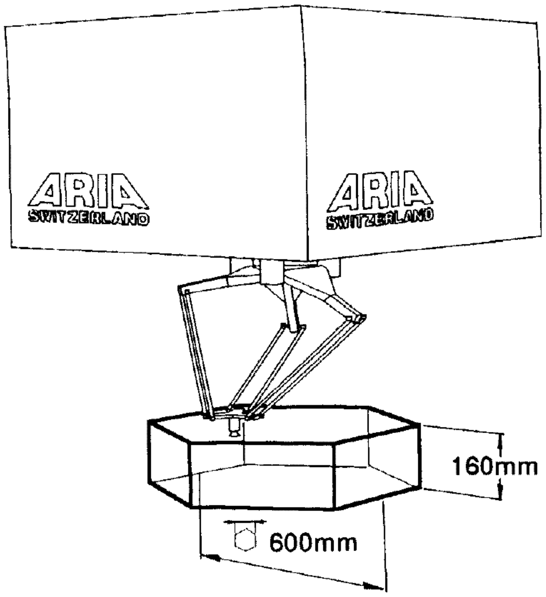

Figure 3: The Aria C1000 very high speed manipulator. Payload $1kg$, max. speed $4.5m/s$, max. acceleration $45m/s^2$. The C300 is even faster ($6.0m/s$, $75m/s^2$), but has a smaller payload ($0.3kg$).

development and enhances robustness. For an actual PC-based implementation the HARTIK real time programming system seems to be well suited [4]. The control diagram presented in Figure 4 assumes a totally uncalibrated setup. The control for simpler scenarios is easily derivable from this by disabling a select number of behaviors. For instance, stereo vergence is not needed if the normal distance to the belt is known a priori, and where behaviors now receive inputs from both eyes' position encoders, only one would suffice. In that case the two eyes could be used independently, effectively giving us two cooperating vision systems (some coordinating policy would have to be worked out). In general we can use environmental constraints to simplify all the processes involved [5].

We now briefly discuss the behaviors individually, going from top to bottom in Figure 4.

3.1 Categorize object in fovea

- *Inputs:* left and right eye position encoders, left retinal image, "reset" signal.
- *Outputs:* object category (including "dont' know" value), "ready" signal.
- *Periodicity:* background.
- *Description:* The most important module from an external functional point of view. on reset, puts a "don't know" value on the object category output and starts the recognition process for the object in the fovea, using 2D cortical template matching (peripheral area of the sensor only, see below), possibly also foveal texture matching. Only the left retinal image is used (the right eye is a "slave" eye for vergence only, see below). The eye position feedback can be used to undo perspective distortions, within limits. The cortical image is already scaling and rotation invariant (up to a linear shift [6]), so only aspect distortion needs to be dealt with. As soon as a satisfactory match is found, puts the category and a

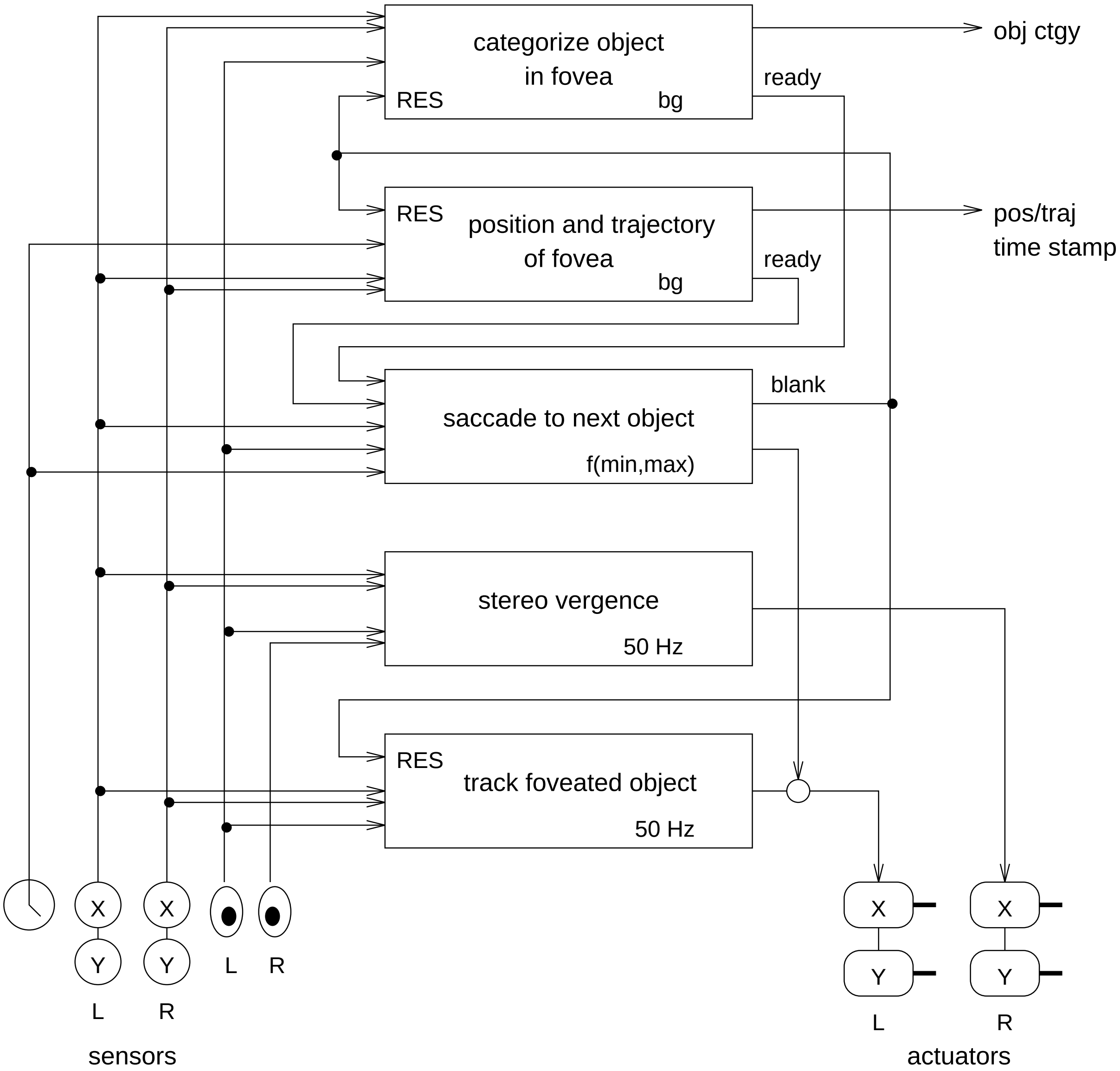

Figure 4: Subsumption style control diagram for the vision system. Each box represents an independent behavior, operating in parallel with all the others. Sensors (system clock, eye position sensors, CCDs) and inputs to behaviors are on the left, actuators (eye motors) and outputs of behaviors are on the right. The periodicity of each behavior is indicated in the lower right of the box.

"ready" signal on the corresponding outputs. Depending on the recognition algorithms used, a second recognition cycle could then be started (for the same object), slower but more accurate. That way the overall system performance will degrade gracefully as the cycle time is shortened. A pyramidal [1] or "polygonal" [2] multi-resolution recognition algorithm also offers interesting possibilities in this respect.

3.2 Position and trajectory of fovea

- *Inputs:* left and right eye position encoders, system clock, "reset" signal.
- *Outputs:* position and trajectory of fovea (in world coordinates) with time stamp, "ready" signal.
- *Periodicity:* background
- *Description:* On reset puts a "dont' know" value on the position/trajectory output and starts computing the trajectory of the left fovea, using only the eye position encoders. As soon as a reasonable estimate is reached this is put on the output, together with a "ready" signal. Analogous to the the categorization module, it can then incrementally improve its estimate. Position and trajectory are expressed in world coordinates; all objects move in a fronto-parallel plane with respect to the camera, and we can safely assume uniform linear motion on that plane, to be derived from the angular velocities and distance of the camera to the object.

3.3 Saccade to next object

- *Inputs:* system clock, left retinal image, left eye position encoders, "ready" signals from categorization and trajectory modules.
- *Outputs:* "blank" signal, left eye motor control (subsuming the tracking module's motor outputs).
- *Periodicity:* some function of minimum and maximum time spent per object.
- *Description:* The most important module from an internal coordination point of view. Some minimum and maximum time spent per foveated object must be defined, to give the manipulator enough time to deal with the object on the one hand, and to guarantee a minimum total throughput on the other hand. Looks for nearby objects in the peripheral FOV, and chooses the nearest one in the opposite direction of tracking, with vertical deflection allowed (saccading mainly right-to-left across the belt, from the observer's point of view as rendered in Figure 1). After the set maximum time has elapsed, or the minimum time has elapsed and both categorization and trajectory modules are ready, or the camera reaches its position limit during tracking, initiates a saccade to the chosen object (possibly two-stage: course and fine). The "blank" signal during the saccade signifies garbage visual data to other modules.[1] The conveyor belt background must be easily segmentable in the camera images.

[1]Perhaps it would be "purer" subsumption style to avoid inter-behavioral "wires", and to blank the actual visual input to other modules in stead. However, that would require those modules to monitor and interpret that condition as a reset signal, which is considerably less efficient than the proposed solution. One might also put some flag value onto the position encoder lines, but that is an overloading of these wires with different functions, which should be avoided. Similarly for the "ready" output signals of some modules, one might monitor the actual outputs to deduce the ready state, but this would also be less efficient.

3.4 Stereo vergence

- *Inputs:* left and right retinal images, left and right eye position encoders.
- *Outputs:* right eye motor controls
- *Periodicity:* 50 Hz (frame rate)
- *Description:* Vergence uses a master-slave configuration, the left eye being the master and the right eye the slave. The slave position is grossly constrained by the master position, and fine control of vergence is done e.g. by cross-correlating the left and right cortical images [9]. The eyes can be put further apart for more accurate measurements of distance. For our application, vergence can be omitted if the normal distance from cameras to belt is known and fixed.

3.5 Track foveated object

- *Inputs:* left retinal image, left and right eye position encoders, "reset" signal.
- *Outputs:* left eye motor controls (subsumed by the saccade module's motor outputs).
- *Periodicity:* 50 Hz (frame rate)
- *Description:* On reset starts to track the foveated object using only the left retinal image, and position encoders from both eyes (for depth estimates), using as simple as possible image metrics (preferably avoiding optical flow computation). Some foveal tracking algorithms are discussed below.

4 Foveal Object Tracking

We now discuss two different algorithms for object tracking using 2D foveal images, taking as much advantage as possible of the constraints inherent in the task and the particular sensor geometry.

4.1 Using temporal difference images

Assuming that initially the Center of Mass (CoM) of a convex solid object is foveated (centered on the fovea of the sensor), a small displacement of the object in any direction results in a temporal difference image (TDI) in which the old and new object contours cross each other twice (Figure 5).[2] Perhaps the assumptions of convexity and solidity (containing no holes) could be relaxed, but they are justified by the environmental (task) constraints. Because of the increasing size of sensor elements towards the periphery, the area (number of pixels) of the TDI enclosed between the old and new object contours is larger closer to the fovea, and smaller away from the fovea. Hence a vector from the fovea to the point of maximum distance between the old and new contour gives the direction and orientation of motion of the object. The maximum distance between the contours gives a direct estimate of the displacement of the object. Hence the sensor can be repositioned to again foveate the CoM of the object, and the cycle can start again.

The image metrics needed for this algorithm are directly derivable from the sensor image, due to its geometry:

[2] All images shown in this paper are simulated foveal images created with a drawing package that have had random noise added to them. For technical reasons it was not possible to use real foveal images.

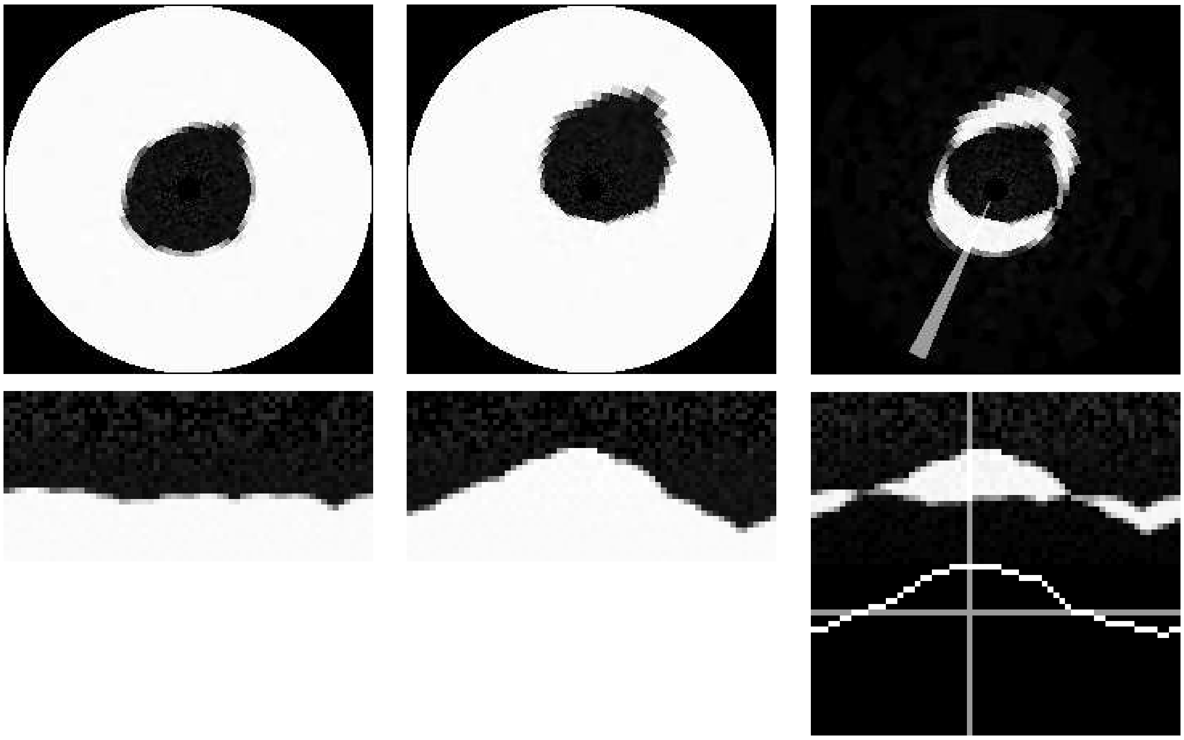

Figure 5: Tracking solid convex objects using foveal temporal difference images. Top: remapped retinal images, bottom: cortical images. The rightmost image also shows the distance between the contours (bottom), and the location of the maximum (top and bottom).

- To find the object contour, a one-dimensional edge detector can be used that follows the radii of pixels outward from the center, i.e. the columns of the "cortical" image (bottom of Figure 5), if necessary followed by a 1D smoothing of the edge locations found.
- Finding the maximum difference between old and new contours amounts to a simple 1D operation, since both contours are represented as vectors whose elements are distances (number of pixels) to the center.
- The angular direction of displacement can be retrieved by a simple table lookup indexed by the number of the radius (column) where the maximum occurs. Similarly, the angular magnitude of the displacement is a simple function of the number of pixels that the maximum is expressed as and the displacement from the center, which is also amenable to table lookup due to the small size of the foveal images (about 2K pixels).

There are some drawbacks to this algorithm, however:

- The displacement of the object must be small enough that the image contour does not move past the fovea. Given that the maximum speed of the objects is known, this constraint determines the minimum sampling frequency for the algorithm.
- There is a cumulative error in sensor placement due to inherent inaccuracies of the computation (in part because of the space-variant resolution of the sensor) as well as the mechanical control of the head. This can be alleviated by doing a complete CoM computation from time to time, and repositioning the sensor accordingly. Given that object

motion is approximately uniform and linear, the error can be kept within reasonable bounds this way.

In fact the algorithm can be further simplified, since the objects always move in the same direction. It therefore suffices to examine a small number of sensor radii (cortical columns) along that direction.

A crucial assumption is of course that initially the CoM of the object is foveated, so we turn to that problem next.

4.2 Using Center of Mass

As mentioned in the previous section, a CoM computation is necessary at least from time to time. We can also implement the usual type of object tracking using CoM computations with foveal images, centering the sensor over the CoM which is newly computed at each frame, or after a fixed number of frames. We need to use retinal coordinates for this purpose, however (top of Figure 5), since the cortical coordinates cannot be meaningfully interpreted for CoM computations. This coordinate conversion step may seem like a drawback of using foveal images, but due to their small size one can easily do the conversion using table-lookup (including compensation for varying pixel size), and the resulting computation is very fast. Including a simple 1D edge detection and smoothing as mentioned above, the computation typically takes about 1 ms on a Sun 4/75 workstation.

5 Foveal Pattern Matching

As we mentioned before, the problem is amenable to a simple pattern matching strategy, at least in as far as the object categories are discriminable by 2D shape alone. Perhaps other dimensions (like color) will have to be used in a full scale implementation, but we will disregard that for now.

As already described in [6], foveal images offer several advantages for this simple type of pattern matching based on binarized 2D images: they are rotation and size invariant, up to a linear shift in the cortical image (Figure 6). To convert a cortical image to a template, the image is binarized via thresholding (possibly adaptive), the minimum significant eccentricity is computed, and a normalized orientation is computed via the Principle Axis of Inertia (again using table-lookup for coordinate conversions). To match a cortical image to a template, the image is also binarized and orientation-normalized, and convolved with the template over a small horizontal and vertical neighborhood [6].

However, if the pattern matching is based on object contours only, as is the case with binarized images of solid objects, the matching actually reduces to a 1D problem in stead of a 2D one, since object contours can be represented as vectors of distances (in pixels) to the center of the image (the fovea of the sensor). The procedure typically takes less than 1 ms on a Sun 4/75 (for most shapes), including 1D edge detection and smoothing. Figure 7 illustrates the procedure.

The matching procedure displays some interesting properties:

- It is also in practice size and rotation invariant to a large degree, as expected from theoretical considerations.

- Even partial images like the third one from the left in Figure 7 is recognized well, which is a consequence of the size invariance property.

- Non-matching objects score considerably lower, as desired. The last image in Figure 7 is a cubic spline interpolation of the vertices of the first, for instance.

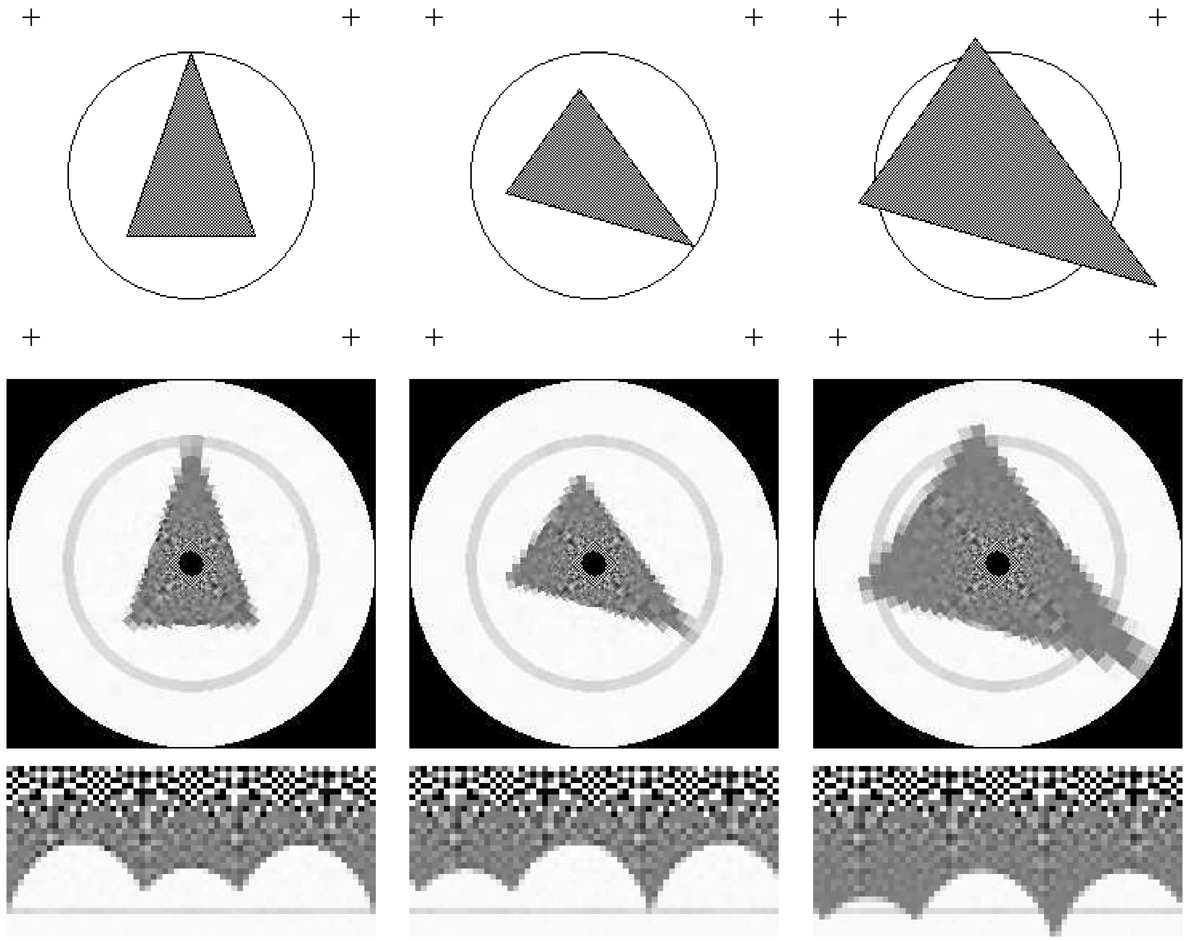

Figure 6: Rotation and size invariance of foveal images, up to a linear shift in the cortical image. Top: scene, middle: remapped retinal images, bottom: cortical images. The bounding circles are for position reference only. The middle and right cortical images are horizontal and/or vertical shifts of the left one (the cortical images should be thought of as being cylindrical, or "wrapping around" the left and right edges).

The features of the algorithm are largely due to the small size and geometric properties of the images (sensor), which combined with the environmental constraints allows the reduction to a 1D problem, and allows a simple representation of the objects which is very close to the images, hence easy and fast to compute.

6 Conclusion

We have presented work in progress on a system for the automatic sorting of recyclable garbage, using a stereo foveal vision head and a manipulator. The combination of the small size and geometric properties of the foveal images with the constraints inherent in the task allows us to use simple and fast image understanding algorithms, which are nevertheless effective. We have illustrated this claim for two particular sub-problems involved, viz. the tracking and categorization of objects using foveal images. The solutions presented are tailored to the problem at hand, and hence cannot be used without modifications for other types of problems,

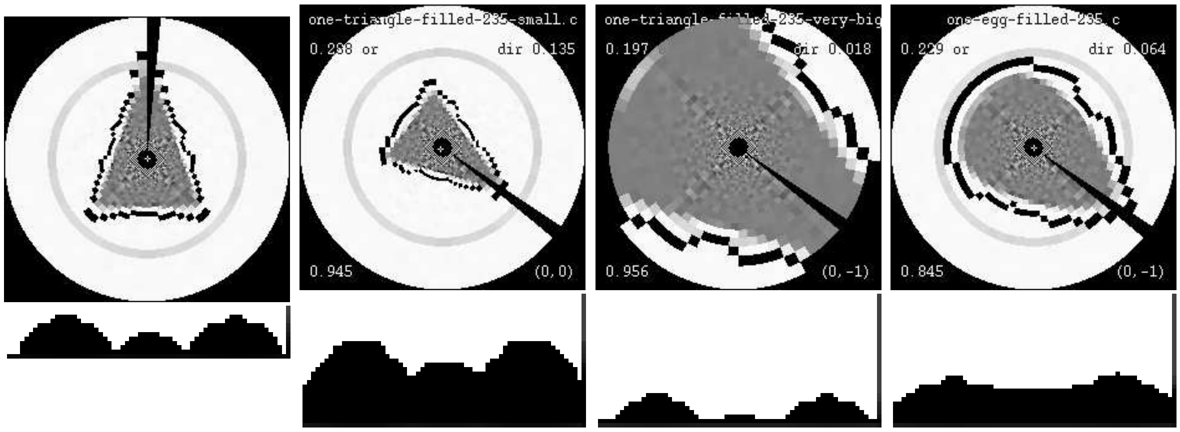

Figure 7: Cortical template matching. Top: remapped retinal images, bottom: cortical images. From left to right: the template, a smaller rotated and a larger rotated instance of the matching class, and an instance of a non-matching class. Matching scores are indicated in the lower left corner of the retinal images. Black pixels in the retinal images represent detected edges, and the black radii represent the principle axes of inertia.

but the effectiveness and small computational cost of the resulting system will be well worth its while, we believe. Nevertheless, we don't think of the presented solutions as completely ad-hoc "hacks", but rather as the result of a consistent attempt to exploit the properties of the vision system used and the environment in which it has to operate.

References

[1] D. H. Ballard and C. M. Brown. *Computer Vision.* Prentice-Hall, Englewood Cliffs NJ, 1982.

[2] C. Bandera and P. Scott. Multiacuity target recognition and tracking. In *Proceedings of the second automatic target recognizer systems and technology conference.* Fort Belvoir Center for Night Vision and Electro-Optics, 1992.

[3] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE-RA*, RA-2(1), March 1986.

[4] G. Butazzo, G. Borlizzi, and M. D. Natale. Il sistema HARTIK, versione 1.2. Technical Report 94-07, ARTS Lab, SSSUP S. Anna, 1994.

[5] I. Horswill. *Specialization of perceptual processes.* PhD thesis, Massachusetts Institute of Technology, 1993.

[6] L. Massone, G. Sandini, , and V. Tagliasco. "form-invariant" topological mapping strategy for 2d shape recognition. *Computer Graphics and Image Processing*, 30(2):169–188, 1985.

[7] G. Sandini and P. Dario. Active vision based on space-variant sensing. In H. Miura and S. Arimoto, editors, *Robotics Research, the fifth international symposium*, pages 75–83. MIT Press, Cambridge MA, 1990.

[8] G. Sandini and V. Tagliasco. An anthropomorphic retina-like structure for scene analysis. *Computer Graphics and Image Processing*, 14:365–372, 1980.

[9] M. Tistarelli and G. Sandini. Dynamic aspects in active vision. *Computer Graphics and Image Processing*, 56(1):108–129, 1992.

[10] J. V. der Spiegel, G. Kreider, C. Claeys, I. Debusschere, G. Sandini, P. Dario, F. Fantini, P. Bellutti, and G. Soncini. A foveated retina-like sensor using CCD technology. In C. Mead and M. Ismail, editors, *Analog VLSI and Neural Network Implementations*. DeKluwer, Boston MA, 1989.