Space-variant image sampling for foveation

Paul Fitzpatrick
Machine Vision (6.866)
Term Paper
MIT ID 984985527
paulfitz@ai.mit.edu

Abstract

In active vision applications, there can be a conflicting need for both high acuity and a wide field of view. One possible trade-off in such cases is to provide high acuity in a small central area, and to use lower resolution information in a wider field of view to control where that central area is placed. This arrangement is akin to the foveated retina of the human eye. For binocular vision under these circumstances, it is important that both cameras center the same object. In humans, this is achieved by vergence eye movements. One simple, fast method for implementing vergence is to use correlation of a log-polar sampling of the images from the cameras. I will examine a vergence algorithm based on log-polar sampling, and present an analysis and implementation of a modified version of this algorithm.

1 Introduction

Visual tasks in robotics, particularly humanoid robotics, can have a conflicting requirement for high acuity and a wide field of view. High acuity is needed for recognition tasks and for controlling precise visually-guided motor movements. A wide field of view is needed for search tasks, for tracking fast or multiple objects, compensating for involuntary ego-motion, etc. A common trade-off is to sample part of the visual field at a high enough resolution to support the first set of tasks, and to sample the rest of the field at an adequate level to support the second set. This is the same trade-off used in animals with foveate vision, such as humans, where the density of (color) photoreceptors is highest at the center and falls off dramatically towards the periphery. It can be implemented by using multiple cameras with different fields of view [16], or specially designed hardware [15].

For a binocular vision system, it is useful to have some way to ensure that both cameras foveate, or center, the same object. This is called vergence in human vision, and is the only eye movement where the eyes are moved towards and away from each other rather than moving in lock-step. Humans use many cues for performing vergence, including motion and shading, but the main cue is disparity. This is the also the main cue that machine vision implementations use for applications in unconstrained environments. Computing disparity involves establishing a correspondence between the images from the two cameras, which is a problem that has been studied extensively in stereo vision. The main barrier to bringing all this work to bear on vergence is the severe restrictions on the amount of computation that can be performed, since the vergence algorithm must be fast enough to control the cameras in real-time. For contemporary hardware, this generally restricts vergence algorithms to simple, limited, techniques such as correlation of patches of the images, or correlation in the frequency domain. An example of the latter is the cepstral filter [8], a type of filter originally developed to analyze signals containing echoes.

Another way to improve correlation performance without moving to the frequency domain is to work with the log-polar transformation of the image. The log-polar transform gives a space-variant sampling of the image

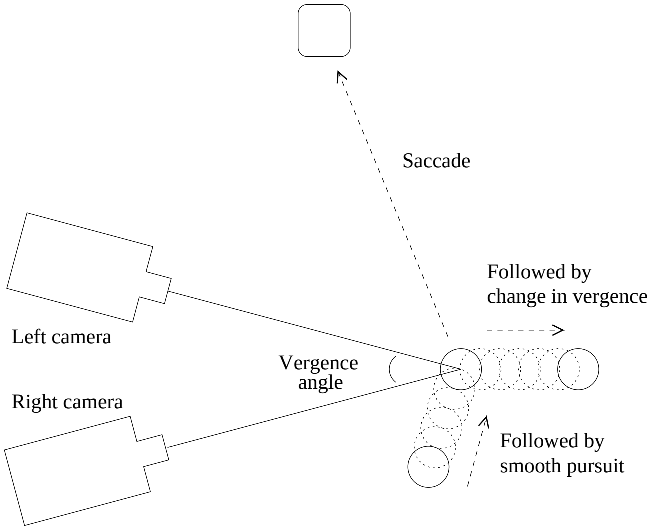

Figure 1: This figure shows a simplified decomposition of eye movements in humans. Vergence movements allow the eyes to center objects at varying depths in the highest resolution area of the retina. Smooth pursuit tracks objects across the visual field, and interacts with vergence. Saccades move the eyes rapidly to a new part of the visual field.

that is most dense around the center and least dense towards the periphery, so that the center of the image is represented with a higher resolution than the rest of the image. Correlation of such images with each other amounts to little more than a weighted correlation of the original untransformed images, but it nevertheless gives surprisingly good results, and is faster than current methods that operate in the frequency domain. The nature of the log-polar transform is described in Section 3, and details of an implementation of vergence using the transform by Santos-Victor and Bernardino [4] are given in Section 4. The algorithm is analyzed in Section 5. I implemented a variant of their algorithm on an active vision head, the details of which are given in Section 7, with results in Section 8.

The active vision head I worked with was from the Cog humanoid robot project at the MIT AI Lab [16]. The head is binocular, with each 'eye' having both a wide angle camera and a 'foveal' camera with a narrow field of view. An object can therefore be imaged with greatest detail when information from the wide field of view cameras is used to bring the object into the field of view of the foveal cameras. Relevant details of the head are given in Section 7.1, when I describe the details of my implementation.

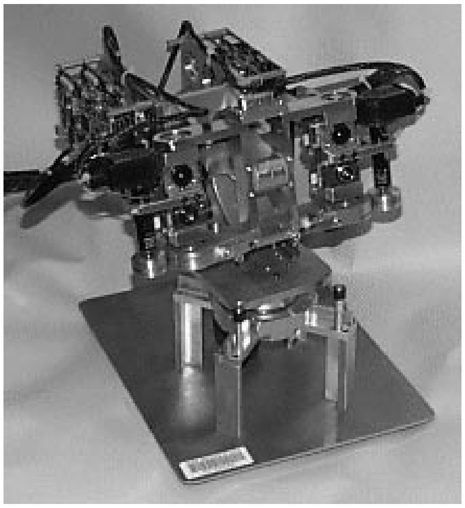

Figure 2: The active vision head

2 Approaches to vergence

Vergence seems a simpler problem than general stereo, since it is only necessary to identify a single point of correspondence between the camera images. Unfortunately, this is not an easy task, and typical methods for establishing correspondences rely on the optimization of a global error measure for robustness. In other words, to establish a single correspondence reliably it is necessary to solve the total stereo vision problem, which is computationally expensive. Since vergence has to be done in real-time, there are severe practical limits on the computation that can be devoted to identifying the correspondence.

In constrained environments, vergence can be quite easy to implement – for example, when there is a high contrast between the object and the background, or when the object is moving and is the only part of the environment doing so, or when the object is a light bulb [19]. But for arbitrary stationary objects in a cluttered environment, the correspondence problem is much harder.

It is possible to make the correspondence problem considerably easier by simply changing the hardware. For example, the correspondence problem becomes increasingly straightforward as the distance between the cameras is reduced, since the images will differ less through occlusions, foreshortening and shading effects. However, the most accurate depth measurements can be deduced from correspondences when the cameras are far apart. So the cameras could simply be moved very close together, making correspondences simple to establish, and then moved apart at a rate that allows the correspondences to be updated iteratively, with the search at each iteration being highly constrained by the results of the previous iteration. This is the approach used by FOVEA [12], a foveated vergent active stereo vision system using an expanding baseline. While the goal of that work was to create dense depth maps, the hardware base would make vergence an easier task. This is not an option for the Cog project because of a commitment to human-like vision, for reasons that will be discussed in Section 9. Other engineering approaches include using multiple (>2) cameras with multiple baselines (for example, [11]), or illuminating the scene with carefully engineered light patterns (for example, [13]).

If the hardware is taken as a given, then the problem becomes finding a method for establishing correspondences that is a good balance between accuracy and speed. The simplest way to search for correspondences is to correlate a small patch of the image from one of the cameras with small patches of the image from the other camera, within some area or along an epipole. This is very straightforward to implement, although it has well known failings, such as coping poorly with differences in foreshortening effects [10]. Vergence modules along these lines have been implemented on the Cog active vision head by a number of people including Yamato [21]. As a baseline for comparison, I implemented such a system too, and found it to be very unreliable. The next step generally taken after this approach is abandoned is to switch to doing correlation in the frequency domain [8]. This requires using the FFT, which takes considerably more computation then correlation on the raw image. A less computationally intensive alternative that has arisen from considerations of primate vision makes use of a log-

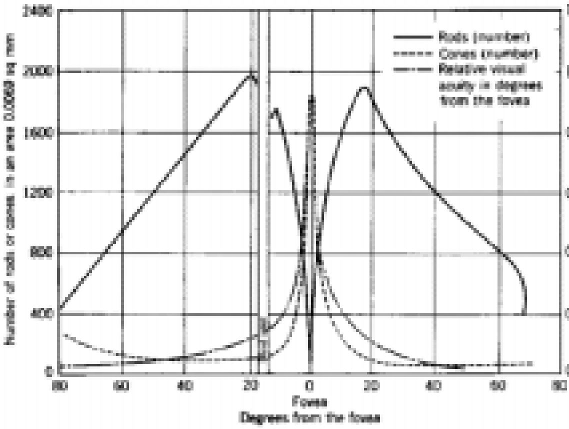

Figure 3: Photoreceptor density across the retina. Visual acuity is greatest within a region of about 0.3°, and then falls off very rapidly. (Taken from [9])

polar sampling of the image, the nature of which will now be described.

3 The log-polar transform

In the retina, the density of photoreceptors grows rapidly towards the center of the visual field (the fovea). This effect can be approximated using a log-polar transformation. This maps the image on to a polar coordinate system where the distance coordinate is logarithmic:

$$\begin{aligned}(x, y) &\Longrightarrow (\rho, \theta)\\ \rho &= \log_b r\\ &= \log_b \sqrt{x^2 + y^2}\\ \theta &= \tan^{-1} \frac{y}{x}\end{aligned}$$

The parameter b controls how quickly the density falls off with distance from the center. Clearly, there is a singularity at the center of the coordinate system. This is often handled by introducing another parameter ρ_{min} to control the minimum ρ value that will be used. Another choice would be to use $\rho = log_b(r + a)$ where a is some constant to avoid the singularity when $r = 0$. This is the choice used in models of the actual retinal density [5], and for this application it has the advantage of not cutting out information from the one part of the image you care most about.

Figure 4 shows a regular grid in (ρ, θ) space mapped onto an image in Cartesian space. Notice how the grid size decreases towards the center, showing that the central part of the image is sampled at the highest resolution. Figure 5 shows an example of a transformed image.

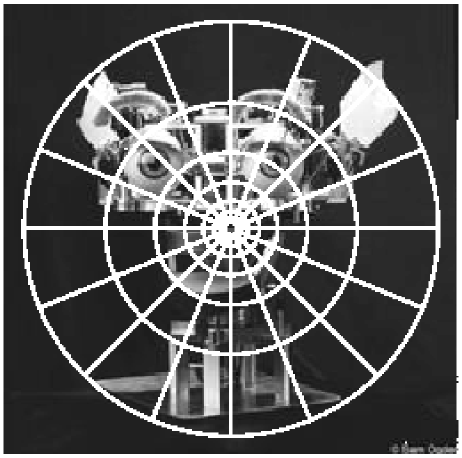
© Sam Ogden

Figure 4: Log-polar image sampling. The grid size increases exponentially with distance from the center. There is a singularity at the center where the grid size goes to zero, so the grid must be truncated at some minimum radius.

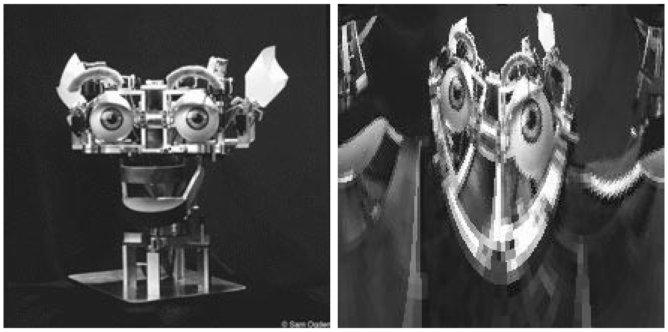
© Sam Ogden

Figure 5: An example of a transformed image. The image has been shifted in θ to facilitate comparison with the original. The figure on the left shows the original image, in the (x, y) coordinate system. The figure on the right shows the transformed version of this image in the (ρ, θ) coordinate system. The grid size is, of course, much finer than shown in Figure 4. Notice, for example, that the ears become smaller than the eyes, since they are further from the center of the original image.

The log-polar transform has some properties that make it potentially useful for vision. One property that stimulated early interest is that scaling and rotation of the image correspond to simple translation in (ρ, θ) coordinates. Of course, the down side of this is that translation of the image in Cartesian coordinates corresponds to a complicated warping in (ρ, θ) coordinates. But foveation centers an object in a stereotyped way, so in theory at least this could provide a basis for scale and rotation invariant object recognition. Another reason for interest in log-polar sampling is that it can give a dramatic reduction in image size, and hence increase the speed of processing the image. This applies only to tasks for which the center of the image is all-important and the periphery is less so.

4 Using the transform

Here I give details of how the log-polar transformation has been used to implement vergence. The particular algorithm I describe is due to Bernardino and Santos-Victor [2, 4].

First, the log-polar transform is applied to the images from the two cameras. This reduces the size of the image significantly (a factor of eight for Bernardino and Santos-Victor), since the periphery is sampled at a much reduced resolution. Then a correlation measure is applied to the two transformed images, over their full area. As the cameras move, the sign of the change in the correlation measure is used to control whether the cameras should stay moving in the same direction, or switch directions. The speed at which the cameras should move is determined from the magnitude of the correlation measure.

The computation required for this scheme is very minimal, so vergence can be extremely rapid. Unfortunately the cameras can get captured at locations that correspond to local maxima of the correlation function, and the speed of convergence is strongly dependent on fairly arbitrary properties of the scene.

Bernardino and Santos-Victor also describe a more robust vergence strategy called “preprogrammed vergence control”, modeled after a theory of vergence in humans. Here they directly estimate the disparity, at the price of more computation. The disparity estimate is made by performing the correlation for various assumed disparities between the images, and finding the correlation peak. This makes the system less susceptible to local minima of the correlation measure. Performing the correlations requires a complicated warping of the images to reflect translation in Cartesian space, maps for which are calculated off-line. The disparities tested for are sampled with highest density around zero and decreasing density for larger disparities.

The correlation measure used by Santos-Victor and Bernardino in [2] is:

$$\frac{\sum_{\rho,\theta}(I_l(\rho,\theta)-\mu_{(I_l)})(I_r(\rho,\theta)-\mu_{(I_r)})}{\sqrt{\sum_{\rho,\theta}(I_l(\rho,\theta)-\mu_{(I_l)})^2\sum_{\rho,\theta}(I_r(\rho,\theta)-\mu_{(I_r)})^2}}$$

This first subtracts the mean value from the images, and then treats them as vectors and calculates the cosine of the angle between them, using this as a measure of similarity. Since the measure is invariant to bias and scaling applied to the images, there is less need to carefully calibrate the cameras or monitor changes in illumination conditions.

If I_l and I_r are treated as random variables, of which the images represent $N \times N$ pairs of samples, then this measure is exactly the correlation coefficient of the two variables.

$$c = \frac{\sigma_{(I_l I_r)}}{\sigma_{(I_l)}\sigma_{(I_r)}}$$

where

$$\begin{aligned}\sigma_{(I_l I_r)} &= \mathcal{E}[(I_l-\mu_{(I_l)})(I_r-\mu_{(I_r)})]\\ \sigma^2_{(I_l)} &= \mathcal{E}[(I_l-\mu_{(I_l)})^2]\\ \sigma^2_{(I_r)} &= \mathcal{E}[(I_r-\mu_{(I_r)})^2]\end{aligned}$$

with $\mathcal{E}(x)$ being the expected value of the random variable x, empirically estimated by the mean.

This correlation is carried out on the image after it has been transformed into its log-polar equivalent, so it is weighted towards measuring the statistics of the center. It is of course possible to fold the transformation into the correlation measure, and simply apply the combined measure to the untransformed images. I give the relevant analysis in the next section, since it clarifies the nature of the correlation.

5 Analysis

Santos-Victor and Bernardino in [2] compare the properties of a particular similarity measure in Cartesian coordinates and log-polar coordinates when the disparity is low. For some reason, presumably simplicity, the similarity measure is not the one they actually use in their algorithm. I give here an analysis for the correlation metric used in the paper, and show what the metric is expressed in terms of the original Cartesian image.

Consider the contininuous analog of the correlation measure given earlier:

$$c = \frac{\iint(I_l-\mu_{(I_l)})(I_r-\mu_{(I_r)})d\rho d\theta}{\sqrt{\iint(I_l-\mu_{(I_l)})^2 d\rho d\theta\iint(I_r-\mu_{(I_r)})^2 d\rho d\theta}}$$

Writing (x, y) in terms of (ρ, θ) gives:

$$\begin{aligned}x &= b^\rho\cos\theta\\ y &= b^\rho\sin\theta\end{aligned}$$

The transformation between the coordinate systems is then:

$$dA = dxdy = \begin{vmatrix}\frac{\partial x}{\partial\rho} & \frac{\partial x}{\partial\theta}\\ \frac{\partial y}{\partial\rho} & \frac{\partial y}{\partial\theta}\end{vmatrix} d\rho d\theta$$

where the determinant of the Jacobian is:

$$\begin{aligned}\begin{vmatrix}\frac{\partial x}{\partial\rho} & \frac{\partial x}{\partial\theta}\\ \frac{\partial y}{\partial\rho} & \frac{\partial y}{\partial\theta}\end{vmatrix} &= \begin{vmatrix}b^\rho\cos\theta\log b & -b^\rho\sin\theta\\ b^\rho\sin\theta\log b & b^\rho\cos\theta\end{vmatrix}\\ &= b^{2\rho}\log b = r^2\log b\end{aligned}$$

Where $r = \sqrt{x^2+y^2}$ as usual. So the correlation measure can be rewritten as:

$$c = \frac{\iint\frac{I_l-\mu_{(I_l)}}{r}\frac{I_r-\mu_{(I_r)}}{r}dxdy}{\sqrt{\iint(\frac{I_l-\mu_{(I_l)}}{r})^2 dxdy\iint(\frac{I_r-\mu_{(I_r)}}{r})^2 dxdy}}$$

Ignoring the subtraction of averages, this is just the correlation of the images after they have been individually weighted by a factor inversely proportional to the distance from the center. Bernardino makes an equivalent point for another similarity measure (a Euclidean distance metric). I think it is important to stress that this does not imply that the overall correlation is weighted by $\frac{1}{r}$. That would be a rather weak weighting, since the total weight along lines of equal distance from the center would remain constant. This is undesirable because there is more area in the periphery of the image than in the center, so if the weighting doesn't fall fast enough the total weight of the periphery can be higher than that of the center. The actual nature of the weighting is clarified by expanding the above out fully as:

$$\frac{\frac{\iint \frac{I_l I_r}{r^2} dxdy}{\iint \frac{1}{r^2} dxdy} - \frac{\iint \frac{I_l}{r^2} dxdy}{\iint \frac{1}{r^2} dxdy} \frac{\iint \frac{I_r}{r^2} dxdy}{\iint \frac{1}{r^2} dxdy}}{\sqrt{\left[\frac{\iint \frac{I_l^2}{r^2} dxdy}{\iint \frac{1}{r^2} dxdy} - (\frac{\iint \frac{I_l}{r^2} dxdy}{\iint \frac{1}{r^2} dxdy})^2\right]\left[\frac{\iint \frac{I_r^2}{r^2} dxdy}{\iint \frac{1}{r^2} dxdy} - (\frac{\iint \frac{I_r}{r^2} dxdy}{\iint \frac{1}{r^2} dxdy})^2\right]}}$$

This is exactly the formula for correlation in Cartesian space, but with an extra weighting of $\frac{1}{r^2}$ appearing everywhere.

Everything in the above equation is straightforward to calculate without making the log-polar transformation. Notice that the b parameter of the transformation does not appear. The minimum radius parameter ρ_{min} translates to leaving a 'hole' in the middle of the area over which the integral is performed.

Clearly, this integral could have been derived by a direct attempt to weight the correlation measure. It is interesting to now work backwards, and see what transformations different weightings would correspond to.

Consider a radially symmetric weight $w(r)$, which for the above case was $\frac{1}{r^2}$. We want to find a coordinate system (ϕ, θ) for which normal correlation will give the same result as a weighted correlation in the Cartesian coordinate system. We can constrain the coordinate system to be polar:

$$\begin{aligned} x &= r(\phi)\cos\theta \\ y &= r(\phi)\sin\theta \end{aligned}$$

The variable r has its usual meaning, with $r = \sqrt{x^2 + y^2}$. The determinant of the Jacobian of this transformation is $r\frac{dr}{d\phi}$. As shown earlier, all the integrals in the correlation end up divided by this quantity, so it must be is the reciprocal of the desired weighting function $w(r)$. So:

$$\begin{aligned} r\frac{dr}{d\phi} &= \frac{1}{w(r)} \\ \int rw(r)dr &= \int 1d\phi \\ \phi &= \int rw(r)dr + C \end{aligned}$$

For a weighting of $\frac{1}{r^3}$, for instance, $\phi = \frac{1}{r}$ is a solution (ignoring scale factors, which don't affect the correlation). This means that the image is turned inside out, with points approaching the center of the image in Cartesian coordinates appearing out towards infinity in the transformed image. Higher powers of r are similar. In fact the weighting $\frac{1}{r^2}$ is a special case since it leads to the integral of $\frac{1}{r}$, which is logarithmic instead of a power of r. This suggests that the log-polar transformation is not easily "tweaked" to give stronger weight to the center – it is tied strongly to the weighting of $\frac{1}{r^2}$.

6 Comments

Researchers who use the log-polar transform for vergence stress its speed advantages. This is indeed a crucial factor for vergence, since it sits inside a control loop trying to keep the cameras foveated on an object, and the less latency there is in the loop the more tightly the object can be kept foveated and the more stable the overall system is.

Another advantage cited for the log-polar transform is its close match with the nature of foveated tasks, where the central area of the image is all-important and the periphery is less so. It seems to be taken for granted that this match also applies to the vergence task, since it too is concerned with the center of the image. But I don't think this is as obvious as it might seem, and contains an implicit assumption about the type of vergence being implemented.

By definition, when vergence has any work to do, it must be the case that the cameras are not centered on the same object. To move the cameras so that they are centered on the same object, the important information will in general lie away from the center of the images from the cameras. Therefore it does not seem sensible to throw away information from the periphery of the images while sampling the center of the image densely.

Vergence can be divided into two phases: tracking and acquiring. When an object is moving slowly, the vergence angle needed to fixate it will change smoothly. If the frame rate is high enough, the correct fixation point will be close to the center of the images, and so the log-polar sampling scheme retains the important information. But this form of vergence is quite simple to implement using any one of a number of schemes, including simple correlation over small patches. The problem of initially acquiring correct vergence on a new object of interest or an object that has just appeared is a more difficult part of the vergence problem. In this case, the center of the images from the current view of the camera are no more likely to be important than any other horizontally offset position.

In these cases then, vergence does not immediately appear to be a good candidate for a foveated approach. One argument to dispute this conclusion would be to make an appeal to biology, since humans can clearly verge robustly working from an approximately log-polar sampled image. More convincingly, it could be argued that the larger the disparity, the less accurately that disparity needs to be known, since the vergence control is part of a closed loop system – and this is exactly what the log-polar transform will give, since it has high resolution at the center and lower resolution towards the periphery. But it seems likely that the lower resolution at the periphery can only reduce the set of circumstances under which the qualitatively correct vergence angle will be detected.

In my opinion, there are two separable ideas in Bernardino and Santos-Victor's work that have been somewhat conflated. One is the use of the log-polar transform to bias correlation towards measuring the statistics of the center of the image. Another is the use of the log-polar transform to reduce the size of the image and speed processing. In particular, it is possible to use the transform as a weighting scheme only, and to relinquish its use in compressing the image (and also its biological plausibility).

For example, instead of warping already-transformed images to produce the versions needed for various assumed disparity values, this could be done from the original Cartesian images. This means there is no longer a bias towards small disparities. Doing this sacrifices some speed, since there are more operations over the larger Cartesian images. I considered it a worthwhile trade-off, so this is the algorithm I implemented.

7 Implementation

I implemented a variant of Santos-Victor and Bernardino's algorithm for performing vergence. The nature of this algorithm, and the variation I introduced were described in the previous section. This section briefly describes the properties of the active vision head on which this work was done, and outlines some important implementation details that have not yet been touched on.

7.1 Characteristics of the head

The active vision head I used for this work has two pairs of color cameras. One pair has 3 mm lenses, giving wide fields of view (about 120° along the horizontal). The other pair has 11 mm lenses, giving narrow fields of view (about 25° along the horizontal). I only made use of the cameras with wide fields of view, and converted the color images to greyscale so that the vergence system could be used on other active vision heads in the lab which currently have monochrome cameras.

The head has three degrees of freedom: independent pan for the left cameras and the right cameras, and a shared tilt of the entire head. For this work, only the pans were used. The cameras can pan very rapidly – fast enough for the limiting factor to be the computation to decide when to stop them. The baseline distance between the left and right cameras is about 13cm (human average inter-ocular distance is about 6.5cm).

The cameras are controlled by a set of TMS320C40 digital signal processors. Code for these is written in Parallel C from 3L.

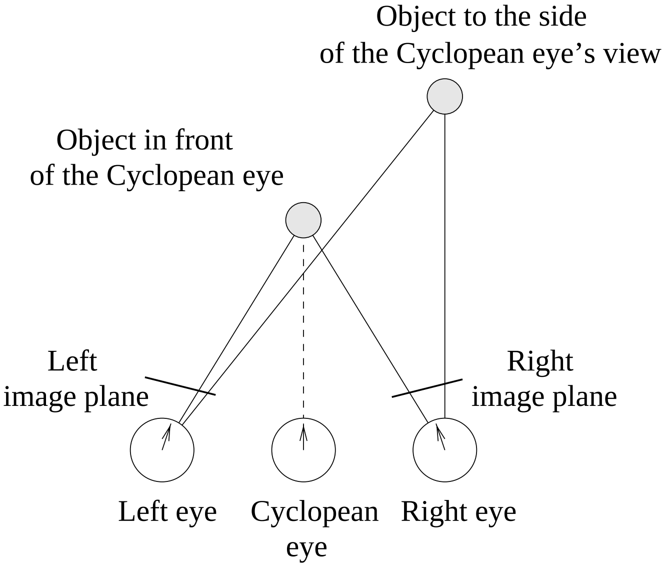

Figure 6: This shows how the target for vergence is chosen. Objects in front of the Cyclopean eye project onto mirror locations in the images from the left and right cameras, while other objects do not.

7.2 Choosing the target

One issue that has not been mentioned yet in this paper is how to chose the target on which the cameras are to verge. In the absence of any higher-level goals, one common answer is to make one of the cameras 'dominant' and verge on whatever appears in the center of the image from that camera. Since moving the dominant camera will change the target and could cause looping behavior, this means that, in practice, this camera needs to be held stationary while the other converges. This is very unnatural in appearance.

I chose to make the cameras verge on anything that appeared directly in front of the active vision head. Specifically, I imagined a virtual 'Cyclopean' eye between the cameras, facing in a direction intermediate to the two cameras, and defined whatever appeared at the center of its view to be the target for vergence. When the Cyclopean eye is facing forwards, anything in front of it will be projected to an equal distance from the inner side of each image of the left and right cameras, as shown in Figure 6. When the Cyclopean eye faces off-axis, there is a similar relationship. This constrains the number of

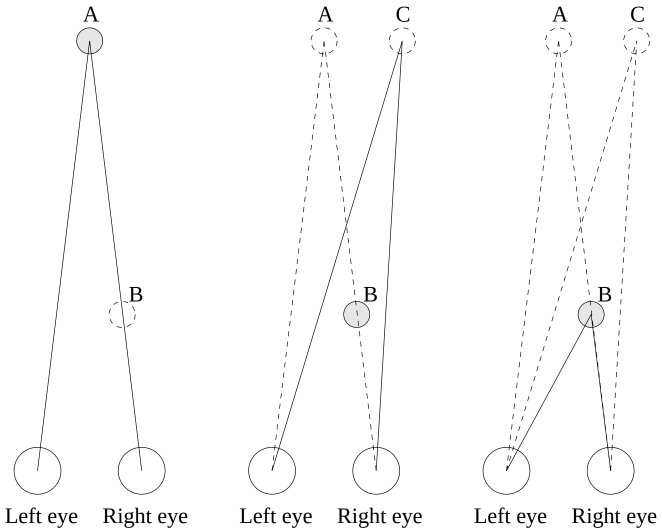

Figure 7: Sequence of movements in changing fixation from A to B. Eyes move to center the new target, and then to verge on it (After [20]). These actions are somewhat superimposed.

comparisons that need to be made. There can be more than one match if there is more than one object directly in front of the Cyclopean eye. The match whose vergence angle shows that it is closest to the head should be chosen. In my implementation, I don't do this. I simply choose the one with the largest correlation. If the closest object is not too small, it will occlude, at least partially, the camera's views of objects behind it, and so it likely to have the highest correlation value.

I kept the Cyclopean eye facing directly forwards so that only objects directly in front of the head would be used as targets for vergence. In other words, I drove the position of the cameras so that they always rotated by equal amounts in opposite directions.

Human vision behaves in a way reminiscent of this. Consider the situation shown in Figure 7. When a target disappears from position A and appears at position B, both eyes move to compensate. It might appear that only one eye need move while the other could stay stationary. This is true, but is not what occurs. In human vision both eyes turn to center the object on a line drawn from the midpoint of the baseline (where the Cyclopean eye would be), and then vergence brings the object back onto the fovea. The right eye ends up back where it started, but moved through a complicated path to get there.

7.3 *Control strategy*

Since vergence is performed in a feedback loop with latencies, care must be taken to ensure that the controller will be stable. I chose a very simple control strategy, where the velocity of the camera motors was made proportional to the disparity between the camera images.

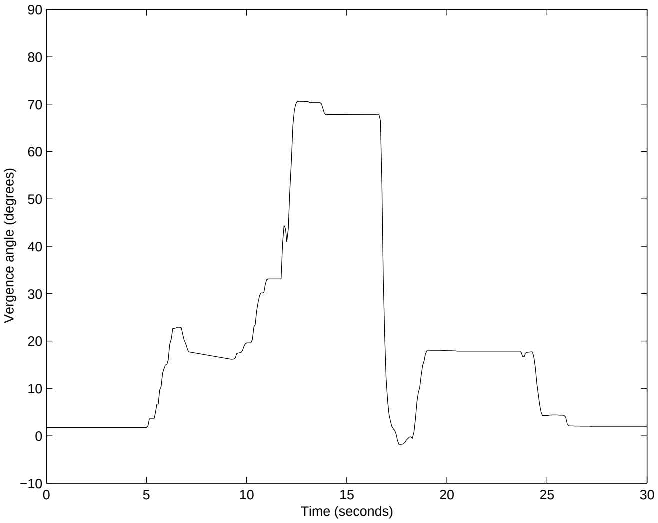

Figure 8: A record of the camera movements corresponding to the sequence of events in Figure 10. The vergence angle is estimated from shaft encoder ticks, so the scale factor of the graph is approximate but its shape is accurate. Between 0 and 5 seconds, the cameras are verged on the background. A human figure enters the foveal region at 5 seconds. The human extends his hand at 10 seconds, and removes it at 17 seconds. This triggers a reflex return to vergence on infinity. Vergence is regained on the human figure at about 20 seconds. At 14 seconds the human figure withdrew, and the cameras returned to verging on the background. Note that the sharp reflexive loss of vergence does not occur here.

There are many ways this could be improved, but the frame rate was high enough (15Hz) and the inertia of the cameras was low enough for this strategy to be stable and still allow fast camera movement.

The vergence algorithm made no attempt to monitor the positions of the cameras. Position measurements were used only to monitor for drastic failures of vergence – for example, the cameras diverging beyond the point where they are parallel, or converging so far as to approach the hard limits of motion of the camera.

One detail was that when the cameras were verged on a very close object, and that object was removed very rapidly, too little of the frontal view remained for there to be a high enough correlation to attract the cameras back to facing forward. So when the best correlation measure fell below a threshold, the cameras were made to "reflexively" return to verging on infinity. Usually that threshold was only exceeded when there was no object in view that could be verged on.

8 Results

Figure 8 shows how the vergence angle of the cameras changes over time as the object being verged on changes.

The sequence of events is as follows (shown in Figure 10). The cameras verge first on the background. Then a human figure enters the scene. The person extends their hand very close to the camera. The hand is then removed, and the person leaves. The graph shows that the vergence system tends to overshoot slightly, due to the simplicity of the controller used. It also shows that when the eyes are highly verged, loss of a target triggers a 'reflex' that brings the cameras back to verging on infinity before control is returned to the vergence system. This was necessary because the images in the camera are often too different in this circumstance for the system to regain correct vergence.

Figure 10 also shows correlation measurements under the different circumstances. Since these apply to verged conditions, the peak is always at zero disparity. The peak is sharpest when verged on the background, and smallest when verged on a very close object.

Figure 11 shows correlation values in verged and non-verged circumstances. In the non-verged case, the peak is off center, and indicates how much the right image should be shifted left, and the left image shifted right, for the best match at the center. While the highest peak is the correct disparity, there is a smaller peak corresponding to the disparity of the background.

Figure 9 shows a time sequence of the cameras verging on a smoothly moving object.

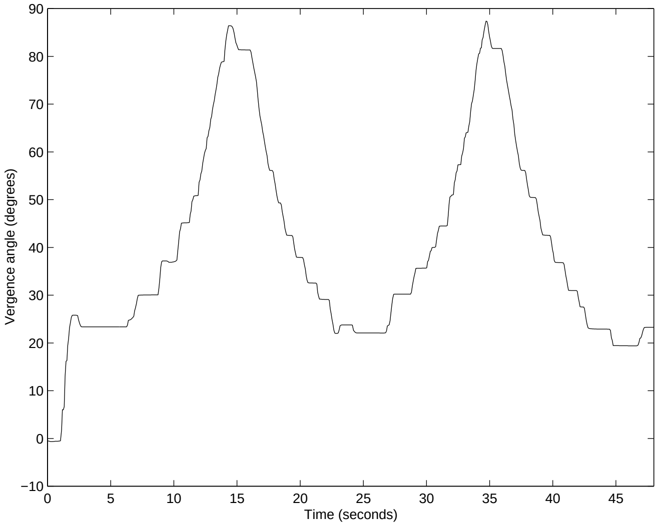

Figure 9: This time series shows the cameras verging on an object moving towards, then away from the active vision head. Notice the jumps in the graph. This is due to the low resolution of the images used. The smallest change in disparity that can be detected is one pixel relative to the virtual Cyclopean eye, or two pixels total, which corresponds to a larger distance the further the object is from the camera. This is why the graph gets rougher towards the lower values of the vergence angle, when the object is furthest away.

9 Discussion

Vergence using log-polar sampling of the images proved considerably more robust than any previous implementation of vergence on the Cog active vision platform. It is by no means perfect, and suffers from the predictable defects of a weighted correlation over the entire image: it won't work on small objects, it won't work with objects that contrast poorly with the background, and sometimes it just won't work. My implementation is also not particularly precise, as Figure 9 showed, but this is not intrinsic to the algorithm. My priority was to make vergence robust to sudden changes in disparity, since tracking small changes in disparity is a straightforward task. With a relatively coarse but robust version of vergence working, the views from the foveal cameras will be of approximately the same part of the scene, so it should be possible to now take the views from the foveal cameras and use them for precise vergence.

Log-polar sampling is such a simple technique that I am almost embarrassed to submit a term paper on the topic, but it is currently an attractive method for performing vergence. And vergence is an attractive visual behavior to implement, because if it can be done robustly without object recognition, it helps to solve the figure/ground separation problem. Once the cameras have verged, the object they have verged on can be extracted from the background using, for example, a zero disparity filter [8]. This can be used to implement a robust object tracking behavior [4]. Vergence also facilitates stereo fusion around the foveation point, since stereo algorithms that will only operate with small disparity values can be applied once the cameras have verged [7]. Both these properties facilitate object recognition. Other advantages are that the vergence angle gives a measure of the distance to the object, and that vergence is a step towards having an object-centered coordinate system.

All these advantages are relevant for the Cog project. There is also another more specific and less technical reason for vergence being important for the type of research being pursued for the Cog project. Part of the project focuses on social interaction as scaffolding for learning [6]. Mechanisms of joint attention, such as pointing and gaze direction, are a key element of this work. It is important that Cog can infer what a human is interested in from their gaze direction. But the dynamic of social interaction makes it equally important that a human can infer the location to which Cog is paying attention. It is helpful to make Cog's gaze as human-like as possible. Currently Cog keeps its cameras parallel when it turns towards an object of interest. Adding vergence will make its locus of interest easier for a human to deduce, by adding information about distance. This is particularly so because Cog's cameras are about twice the distance apart as human eyes, exaggerating the difference between

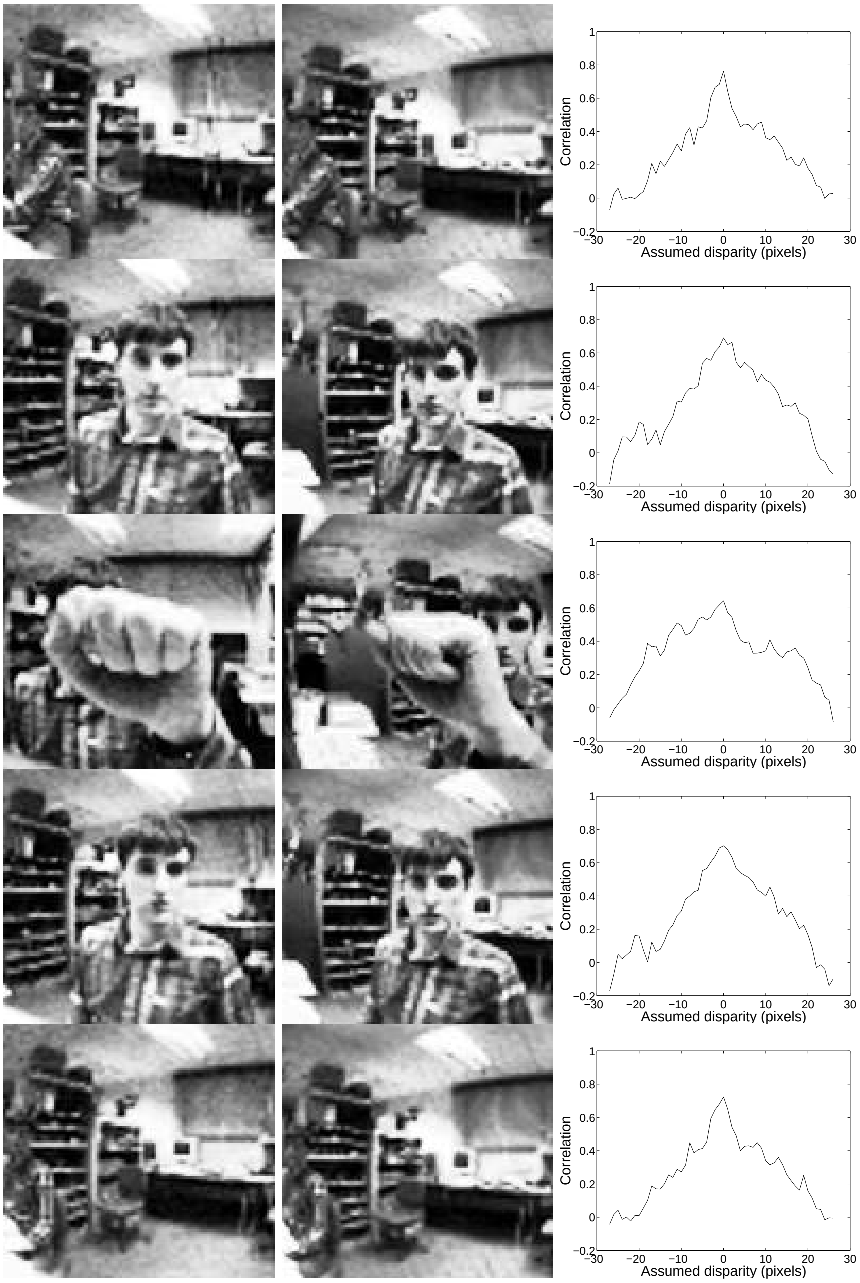

Figure 10: The above images are views from the two cameras taken at five second intervals. Views from the left camera are on the left, views from the right camera are on the right. The cameras are designated left and right from the point of view of a homunculus behind the cameras. The first pair of images show the cameras in near-parallel orientation, verging on the distant background. The correlation measure, given beside the pair of images, is sharply peaked. The next pair of images show the cameras verged on a face. Objects in the background now appear shifted. The correlation measure is less sharply peaked. The next pair of images show the cameras verged on a hand. The human figure now appears shifted outwards. Notice that the appearance of the hand differs significantly between the images. The correlation measure is quite erratic. The following two pairs of images show the cameras reverting to verging on the face when the hand is removed, and then to the background when the face is removed. The images shown here are scaled versions of the 64×64 pixel images actually used by the vergence system.

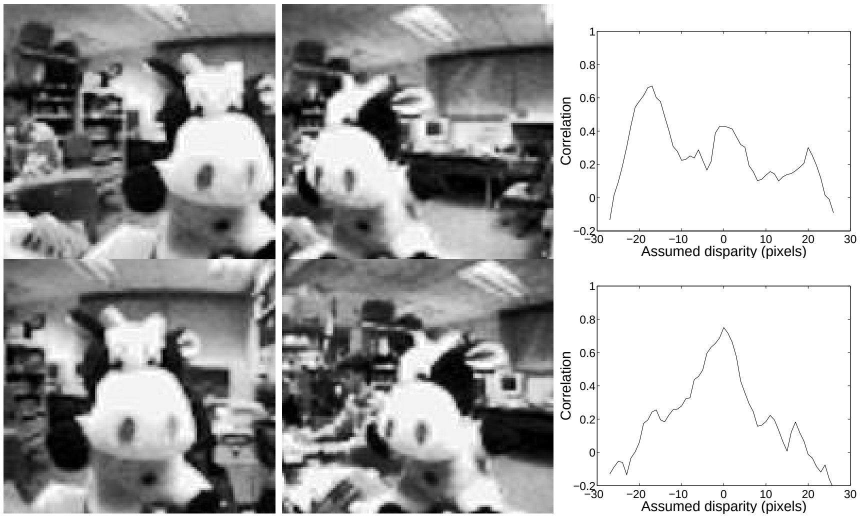

Figure 11: The upper pair of images show a toy in front of the cameras with the vergence system disabled. The correlation graph on the top right shows that there is a disparity of 20 pixels in the convergent direction. This disparity is with respect to a virtual 'cyclopean' eye, so the total disparity between the two images is 40 pixels (image width is 64 pixels). There is a secondary peak in correlation for a disparity close to zero, corresponding to the background. The lower pair of images show that when the vergence system is enabled, the cameras move to center the dominant disparity.

parallel and convergent camera angles.

One difficulty with the log-polar transform for vergence is that is no obvious way to improve its performance. Details such as accounting for the significant radial distortion of the wide angle cameras used might have some small benefit. Using color images might help somewhat as well. But there is no real theoretical meat to build on. Therefore, for this application, the log-polar transformation may be just a stop-gap approach until the price of computation falls enough to allow the use of more rigorously developed techniques.

References

[1] D. Ballard. Animate vision. *Artificial Intelligence*, 48(1):1–27, February 1991.

[2] A. Bernardino and J. Santos-Victor. Correlation based vergence control using log-polar images. *International Symposium on Intelligent Robotic Systems*, 1996.

[3] A. Bernardino and J. Santos-Victor. Sensor geometry for dynamic vergence: characterization and performance analysis. *European Conference on Computer Vision*, 1996.

[4] A. Bernardino and J. Santos-Victor. Visual behaviours for binocular tracking. *Robotics and Autonomous Systems*, to appear, 1998.

[5] G. Bonmassar and E. Schwartz. Lie groups, space-variant fourier analysis and the exponential chirp transform. *Computer Vision and Pattern Recognition*, 3:229–237, 1996.

[6] R. A. Brooks, C. Breazeal, R. Irie, C. C. Kemp, M. Marjanovic, B. Scassellati, and M. Williamson. Alternate essences of intelligence. *AAAI*, 1998.

[7] C. Capurro, F. Panerai, and G. Sandini. Dynamic vergence using log-polar images. *International Journal of Computer Vision*, 24(1):79–94, August 1997.

[8] D. Coombs. *Real-time gaze holding in binocular robot vision.* PhD thesis, University of Rochester, 1992.

[9] C. Graham. *Vision and visual perception.* John Wiley and Sons, Inc.

[10] B. K. P. Horn. *Robot Vision.* MIT Press, Cambridge, Massachusetts, 1986.

[11] T. Kanade, S.B. Kang, J.A. Webb, and C.L. Zitnick. An active multibaseline stereo vision system with real-time image capture. In *CMU-CS-TR*, 1994.

[12] W. N. Klarquist and A. C. Bovik. Fovea: A foveated vergent active stereo vision system for dynamic three-dimensional scene recovery. *IEEE Transactions on Robotics and Automation*, 15:755–770, 1998.

[13] A. Koschan, V. Rodehorst, and K. Spiller. Color stereo vision using hierarchical block matching and active color illumination. *International Conference on Pattern Recognition*, A:835–839.

[14] T.J. Olson and D. Coombs. Real-time vergence control for binocular robots. *International Journal of Computer Vision*, 7(1):67–89, November 1991.

[15] S. Rougeaux and Y. Kuniyoshi. Robust real-time tracking on an active vision head. *Proc. Int. Conf. on Intelligent Robots and Systems*, 1997.

[16] B. Scassellati. A binocular, foveated active vision system. *Technical Report 1628, MIT Artificial Intelligence Lab Memo.*

[17] E. Schwartz, D. Greve, and G. Bonmassar. Space-variant active vision: Definition, overview and examples. *Neural Networks*, 8(7/8):1297–1308, 1995.

[18] C. Silva and J. Santos-Victor. Egomotion estimation using log-polar images. *International Conference of Computer Vision*, 1998.

[19] A. Takanishi, T. Matsuno, and I. Kato. Development of an anthropomorphic head-eye robot with two eyes-coordinated

head-eye motion and pursuing motion in the depth direction. In *International Conference on Intelligent Robot and Systems*, volume 3, pages V27–28, 1997.

[20] G. Westheimer. Oculomotor control: the vergence system. In R. A. Monty and J. W. Senders, editors, *Eye movements and psychological processes*. John Wiley and Sons, Inc.

[21] J. Yamato. Tracking moving object by stereo vision head with vergence for humanoid robot. Master's thesis, MIT Department of Electrical Engineering and Computer Science, 1998.