MODEL-BASED ATTENTION FIXATION USING LOG-POLAR IMAGES[1]

Alexandre Bernardino[1], José Santos–Victor[1] and Giulio Sandini[2]

[1]Instituto de Sistemas e Robótica
Instituto Superior Técnico
Portugal
[2]DIST
University of Genoa
Italy

INTRODUCTION

Foveated active visual systems are widely present in animal life. Despite having limited visual and processing resources, biological systems are capable of very complex visual behaviors with extraordinary performances. A representation of the environment with high-resolution and a wide field of view are capabilities provided by the contribution of the space-variant ocular geometry and the ability to move the eyes.

The use of foveated images can also provide important benefits in artificial systems. The most common space-variant image representation is the **log-polar mapping**, introduced in [1], due to its similarity to the retinal resolution and organization on the visual cortex of primates[2, 3]. Its application to artificial vision was first motivated by its perceptually based data compression capabilities. When compared to the usual cartesian images, log-polar images allow faster sampling rates without reducing the size of the field of view and the resolution on the central part of the retina (fovea) [4]. In the last years, however, it has been noticed that the log-polar geometry also provides important algorithmic benefits such as rotation and scale invariance [5], easy computation of time-to-contact [6], increased stereo resolution on verging systems [4], better sensitivity for vergence control [7, 8], etc. For instance in [8], it is shown that the use of log-polar images extends the range of object sizes that can be tracked using a simple translation model. In this work we increase the "order" of the transformation towards the planar model and show that these advantages can still be observed. To accomplish this goal we designed a tracking system composed by a camera with pan and tilt degrees of freedom, capable of keeping attention fixed on slowly moving objects.

Visual attention mechanisms are commonly defined as the ability to direct visual resources to certain objects in field of view. In biological systems, attention mechanisms are very complex processes that involve high-level and low-level neuronal systems and are influenced by different kinds of visual stimuli. In this work we do not try to address attention mechanisms to the full extent, but to make some analogies between biological evidence and the computer vision perspective, in what is related to visual tracking applications. In terms of visual activity a common classification distinguishes between **overt** and **covert** attention shifts. The first one involves eye movements to center the object of interest in the fovea, directing physical resources (retinal photo-receptors) to inspect the object. The latter do not require ocular motion and allocates "brain" resources to inspect a visual entity even if it is located in the periphery of the visual field. We believe that this classification also makes sense in the "engineering" of an artificial visual tracking application. We design our system based on the cooperation of two mechanisms: (i) the **implicit–tracking / covert attention** mechanism, which is implemented by a novel image registration

[1] This work was partially funded by projects ESPRIT-LTR 30185, NARVAL and VIRGO-TMR FMRX-CT96-0049.

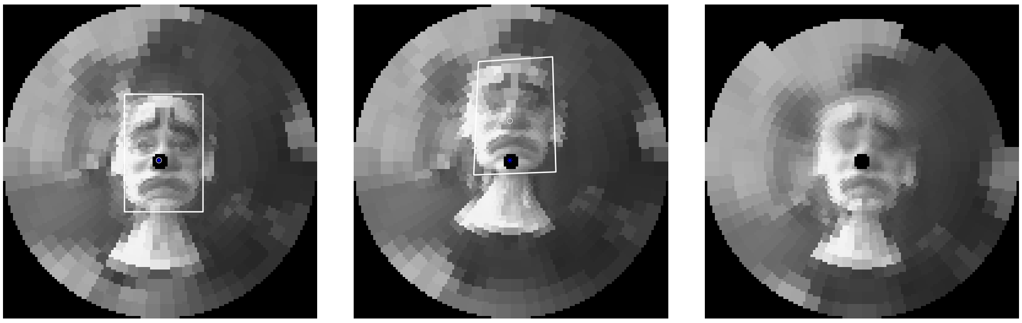

Figure 1: **(left)** The original foveal image, $\mathrm{fov}_{\mathrm{o}}(\mathbf{x})$, defines a **target template** representing the appearance of the object of interest. **(middle)** At time t the target moves to the periphery and is observed at the fovea image, $\mathrm{fov}_{\mathrm{t}}(\mathbf{x})$. **(right)** A **"virtual saccade"** displaces the focus attention to the predicted location of the target. The target is "pulled" to the center of a **"virtual fovea"**, $\mathrm{vf}_{\mathrm{t}}(\mathbf{x})$,which augments the number of "pixels" containing target–related information.

algorithm to keep track of object motion in a passive manner, i.e. without moving the cameras; (ii) the **explicit–tracking / overt attention** behaviour, which is implemented by controlling the camera pan and tilt angles to keep the object in the center of the fovea, where more information can be obtained for motion estimation.

The paper organization is the following. Section 2 describes the implicit–tracking module that implements the covert-attention behaviour, focusing on the problem of target position estimation using model-based appearance prototypes and comparing the proposed method with classical ones. In section 3 we overview the explicit–tracking module, responsible by the overt-attention behaviour, and describe a system simulator used to obtain results with ground truth data. Section 4 presents the adopted foveated image representation (log-polar) and discuss the advantage of this geometry over the conventional one (cartesian), mainly in what concerns to tracking applications. Results are shown in section 5 and illustrate the main contributions of this work: the advantages of using a foveated image representation; the real-time (25 Hz) performance of the cooperating behaviours; and the robustness to deviations from the assumed geometric model.

IMPLICIT–TRACKING AND THE COVERT ATTENTION MECHANISM

Covert attention is related to the ability to attend to objects without moving the eyes. In biological systems this is achieved by directing neuronal resources to the location of the object of interest, similarly to performing a **"virtual saccade"** towards the object. In reality, true saccades can not perform certain types of transformations on the observed images (e.g. scaling), so we introduce the term "virtual saccade" to express, in an intuitive manner, more general image transformations. The process of allocating neuronal resources to certain regions of the visual field involves complex mechanisms and, depending on the subject motivational state, objects can be detected by color, shape, motion, texture, etc. In our computational framework we define one object of interest by its appearance (texture) on the first acquired image and call it **target template**. In the next time instances, the system should be able to locate the object in the visual field and keep track of its varying position (see Fig. 1). This is a very hard problem since an exhaustive search of the object appearance on the visual field is computationally prohibitive and classical local optimization algorithms have a very limited search range. We propose an algorithm that represents a compromise between local and global optimization methods and exhibits good search range and convergence properties with controlled computational cost. In analogy with biological evidence, the system will direct the computational resources to the estimated object position, implementing a target centered **"virtual fovea"**, to maximize the amount of object related information extracted from the image.

Computational Framework

Computing target locations is, in general, equivalent to solve the "correspondence problem", which is one of the most difficult problems of computer vision. Many research efforts are still directed to develop strategies to find corresponding features between two or more images. This problem arises in many

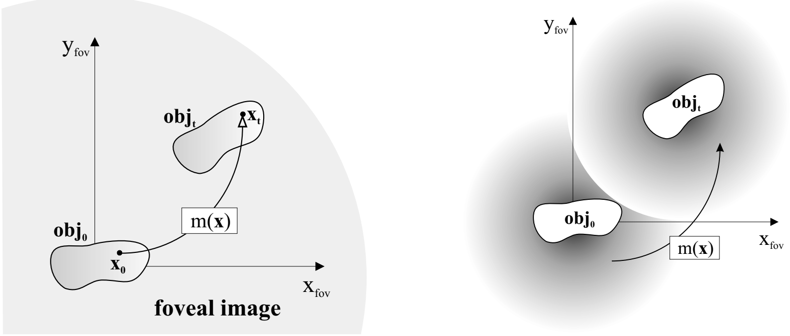

Figure 2: **(left)** The correspondence map m transforms point coordinates from the initial to the final target positions. **(right)** For an ideal allocation of resources, the spotlight of attention (virtual fovea) must be "shifted" according to the same map m.

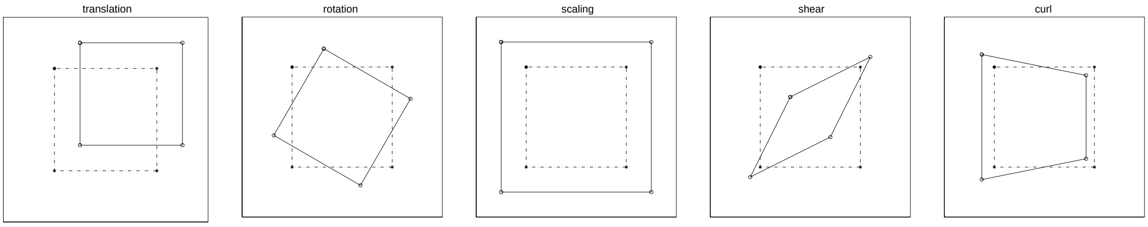

Figure 3: 2D planar transformations express possible 2D deformations of the target. Are composed by translation, rotation, scaling, shear and curl.

different forms, and most applications in computer vision must address it one way or another. Optic Flow and Stereo are two methods that depend extensively on matching issues.

In a purely geometric framework, let us consider the target support region represented as a set of 2D image points $\mathbf{obj}_o = \{\mathbf{x}_0^1, \cdots, \mathbf{x}_0^n\}$. Given a new image, these points are displaced to different coordinates $\mathbf{obj}_t = \{\mathbf{x}_t^1, \cdots, \mathbf{x}_t^n\}$. This "disparity" can be represented by a map m that establishes the correspondence between point coordinates on the initial and final target positions (see Fig. 2):

$$\mathbf{obj}_t = m(\mathbf{obj}_o) \tag{1}$$

The target localization problem consists in computing an estimate $\hat{m}$ of this correspondence map. In a general setting, this problem is computationally hard and ill–posed (e.g. the aperture problem [9]). In order to deal with these difficulties, some assumptions are commonly made:

1. Information at each corresponding point is the same in the initial and final images (eg. the Brightness Constancy Assumption – BCA [9]), or changes according to some appropriate model. Assuming BCA means that all changes in brightness between the initial and final foveal images are completely described by the true correspondence map m^*:

$$\mathrm{fov_o}(\mathbf{x}) = \mathrm{fov_t}(m^*(\mathbf{x})) \tag{2}$$

2. The mapping function follows some model, i.e. it is not completely arbitrary. We model the mapping function using a planar model, with the motivation that planar surfaces can be found in many human made environments and represent good approximations for other kind of surfaces. The planar projective transformations can be represented by an 8 parameter vector μ, and the mapping function is rewritten as $m(\mathbf{x};\mu)$. In a cartesian image plane, the motion field of a moving 3D plane is given by the following equation:

$$\mathbf{x}_t = \mathbf{m}\left(\mathbf{x}_0;\mu\right) = \left(\frac{\mu_1 \cdot x_0 + \mu_2 \cdot y_0 + \mu_3}{\mu_7 \cdot x_0 + \mu_8 \cdot y_0 + 1}, \frac{\mu_4 \cdot x_0 + \mu_5 \cdot y_0 + \mu_6}{\mu_7 \cdot x_0 + \mu_8 \cdot y_0 + 1}\right)^T \tag{3}$$

Fig. 3 illustrates some common planar transformations.

3. An initial guess $\bar{\mu}$ can be obtained at any time instant, reducing the search space to a neighborhood of the initial solution. This is a realistic assumption since the displacement of physical objects is

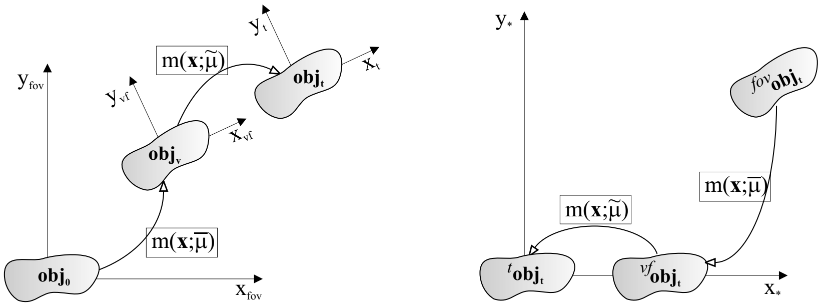

Figure 4: The full motion transformation $m(\mathbf{x};\mu)$ is obtained by first applying a known predicted transformation **("virtual saccade")** $m(\mathbf{x};\bar{\mu})$ and then computing the remaining unknown residual transformation $m(\mathbf{x};\tilde{\mu})$. **(left – fixed reference frame)** $\mathbf{obj_o}$, $\mathbf{obj_v}$ and $\mathbf{obj_t}$ represent the original, predicted and current positions of the target, viewed from a fixed reference frame (the fovea). **(right – moving reference frame)** The evolution of the current target appearance when the reference frame changes from the original fovea (fov) to the virtual fovea (vf) and to the true target location (t).

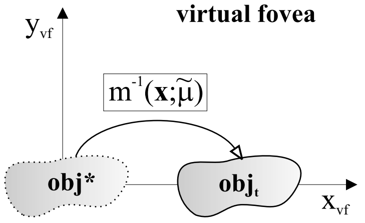

Figure 5: The computation of the residual motion $m(\mathbf{x};\tilde{\mu})$ is estimated in the "virtual fovea" reference system. The target template is "inverse transformed" by $m(\mathbf{x};\tilde{\mu})$ until it best matches the appearance of the current target.

constrained by inertial physical laws. It can be obtained simply as the estimate of the target location parameters in the previous time instant $\mu(t-1)$ or by a suitable prediction based on past time information and/or the motion model, like in a Kalman Filter [10]. With this predicted location, we can redirect our **"virtual fovea"** to the vicinity of the expected target position:

$$\mathrm{vf_t}(\mathbf{x}) = \mathrm{fov_t}(m(\mathbf{x})) \tag{4}$$

Thus, the application of the correspondence map $\mathbf{m}$ expresses a **"virtual saccade"** in image coordinates. In a latter section, we show that virtual saccades can be efficiently emulated by software in both cartesian and log-polar coordinates.

In general, the initial guess $\bar{\mu}$ does not coincide with the true location and a residual error $\tilde{\mu}$ must be computed by image processing algorithms. Using the two components (prediction $\bar{\mu}$ and innovation $\tilde{\mu}$), we define the composition rule that generates the full motion field (see Fig. 4):

$$m(\mathbf{x};\mu) = m(m(\mathbf{x};\bar{\mu});\tilde{\mu}) \tag{5}$$

With these assumptions we can recast the tracking problem as one of determining the "innovation term" $\tilde{\mu}$. This can be obtained by minimizing a least squares objective function:

$$O(\tilde{\mu}) = \sum_{\mathbf{x}} \left[\mathrm{vf_t}(\mathbf{x}) - \mathrm{fov_o}(\mathbf{m}^{-1}(\mathbf{x};\tilde{\mu}))\right] \tag{6}$$

The image $\mathrm{vf_t}(\mathbf{x})$ (virtual fovea) is obtained by transforming the current image $\mathrm{fov_t}(\mathbf{x})$ with the predicted location parameters $\bar{\mu}$ (see Eq. 4). It is equivalent to what one would see after performing a "virtual saccade" to the predicted location of the target. The image $\mathrm{fov_o}(\mathbf{m}^{-1}(\mathbf{x};\tilde{\mu}))$ is called the *deformed template* and represents the transformation on the target template required to match the target appearance in the **"virtual fovea"** (see Fig. 5).

Classical Optimization Methods To minimize this error function, several strategies can be used, ranging from exhaustive search methods (global) to gradient based techniques (local).

The most straightforward and precise method is **exhaustive search**, but is also the most computationally demanding. This method consists in computing the error between the current image and the *deformed template* for "all" possible deformations $\tilde{\mu}$. In practice, we must test a dense set of discrete hypothesis $\tilde{\mu}_\mathrm{i}$ and choose the one that globally minimizes the error.

Contrasting with the previous technique, the class of **gradient methods** use only information in a local neighborhood of the error function. Despite existing several methods of this class, they are all based on the fact that a local minimum of the error function may be achieved by iteratively moving the solution in the opposite direction of the local gradient. They assume that the error function is smooth enough to avoid local minima and the initial condition is close to the real solution. Usually these methods are less computationally expensive but have a much smaller convergence interval and require an appropriate control of the iterative procedure to avoid divergence and oscillations.

A very simple iterative implementation of the algorithm can be described as follows – start with an initial estimate $\tilde{\mu}_0$ and update it by subtracting a fraction η of the error function until convergence is achieved:

$$\tilde{\mu}^{(k+1)} = \tilde{\mu}^{(k)} + 2\eta \sum_{\mathbf{x}} \frac{\partial \mathrm{fov_o}(m^{-1}(\mathbf{x};\tilde{\mu}))}{\partial \tilde{\mu}} \left[\mathrm{vf_t}(\mathbf{x}) - \mathrm{fov_o}(m^{-1}(\mathbf{x};\tilde{\mu}^{(k)}))\right] \tag{7}$$

where the partial derivatives are computed at $\tilde{\mu} = 0$.

The previous two algorithms represent extreme cases of the use of global and local information, showing also extreme performances in terms of the convergence intervals and computational complexity. In the next section we present an algorithm that pretends to extend the convergence interval relative to local methods but with a controlled computational complexity. A simple experiment will illustrate its performance in comparison with the classical methods.

Appearance Prototypes

We propose an algorithm that balances computational complexity and range of convergence. It is not as demanding as exhaustive search because it is based on a sparse sampling instead of dense sampling. Also, it is not as local as usual gradient descent methods since we can represent our data with samples (prototypes) that cover a wide search range.

Considering the estimation of a planar transformation, we define a set of samples $\mathcal{V} = \{\tilde{\mu}_i, i \in (1 \cdots m)\}$ of the 8 dimensional parameter space. This set must form a basis and cover a suitable interval of the search space. Once a target template $\mathrm{fov_o}$ is selected, we can transform it according to such deformations and build a set of *appearance prototypes*:

$$\mathcal{R} = \left\{R_i = \mathrm{fov_o}\left(m^{-1}(\mathbf{x};\tilde{\mu}_i)\right), i \in (1 \cdots m)\right\} \tag{8}$$

Each appearance prototype represents the expected appearance of the **"virtual fovea"** $\mathrm{vf}(\mathbf{x})$ if the true residual transformation is $\tilde{\mu}_i$. Fig. 6 shows an example of a prototype transformation. *A priori* knowledge can be used in the choice of the sampling vectors. For instance, they should sample more densely the expected image deformations and range — if our expected deformation model consists basically on translations with a uniform distribution on the range -6 to 6 pixels, we may define sample vectors spaced uniformly within that range.

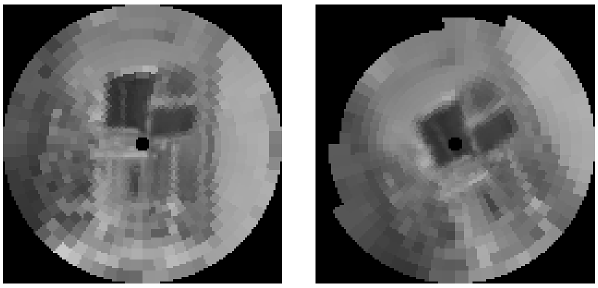

Figure 6: The "appearance prototype" **(right)** is obtained by transforming the reference image **(left)** according to a prototype transformation (the one shown includes translation and rotation)

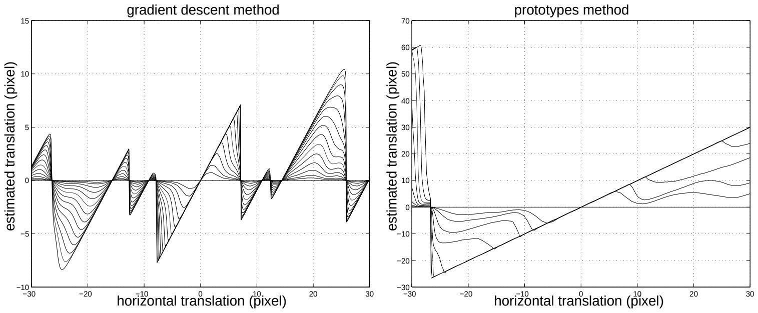

Figure 7: Translation estimation with different optimization methods: **(left)** Local gradient method. Notice the limited convergence range (about ±8 pixels). Different lines correspond to different number of iterations $(10, 20, \cdots, 150)$. **(right)** Prototype sampling method. Different lines correspond to different number of iterations $(2, 4, \cdots, 14)$.

The basic assumption on the proposed method is that the "differential" information observed in the **"virtual fovea"** ($D = \mathrm{vf_t} - \mathrm{fov_o}$) can be represented as a linear combination of the "differentials" of the appearance prototypes ($R'_i = R_i - \mathrm{fov_o}$), with coefficients $\mathbf{k} = (k_1, \cdots, k_m)^T$. The objective function to minimize becomes:

$$O(\mathbf{k}) \approx \sum (D - \sum_{i=1}^{m} k_i \cdot R'_i)^2 \tag{9}$$

A closed-form least–squares method can be used to compute the coefficients $\mathbf{k}$ and the location parameters are obtained by linearly combining the prototype vectors [11]:

$$\tilde{\mu} = \sum_i k_i \cdot \tilde{\mu}_i \tag{10}$$

One of the advantages of this method over standard local gradient descent methods is the ability to customize the set of sample vectors according to the kind and range of expected image deformations. Also with the increase of computational power, we can easily add new sample vectors to improve the estimation results. The algorithm can be customized in order to estimate the larger and more constrained motions in the first iterations and the finer and more generic transformations in the last iterations, which improve its robustness.

Experiment Let us consider one simple example and compare the performance of three algorithms: *gradient descent*, *appearance prototypes* and *exhaustive sampling*. The experiment consists in defining a reference template as in Fig. 1 and simulating a target translation from −30 to 30 pixels in small steps (0.1 pixels). For each translation we apply the different algorithms and compare the solutions.

Results are presented in Fig. 7. The first plot is relative to translation estimation by the **gradient descent** method. Several curves are presented, corresponding to the estimated translation with different number of iterations $(10, 20, \cdots 150)$. We can observe that the convergence interval of this algorithm is limited to about ±8 pixels. Another aspect of concern is the convergence speed of this kind of algorithms. The number of iterations depends on the required precision – if translation is small, good estimates can be obtained with a few iterations, but more iterations are required in the limits of the convergence interval. The second plots shows the performance of the **appearance prototypes** method. Again several curves are presented for the evolution of the estimation process with different number of iterations $(2, 4, \cdots 14)$. We can observe that the convergence interval is much larger in comparison with the gradient descent method. Also, the convergence rate is higher – the algorithm reaches a stable solution in about 10 iterations. The results that correspond to the **exhaustive sampling** method are not shown because the estimated value was always identical to the real one. However, though the precision and robustness of this method is the best, its computation is too demanding for high dimension spaces.

OVERT ATTENTION AND ACTIVE TRACKING

Overt attention is responsible to place the attended object in the center on the visual field using eye/head motion. The purpose of this behaviour is related to the space-variant resolution of the retina. By placing the object of interest in the fovea the visual system maximizes the amount of information extracted from

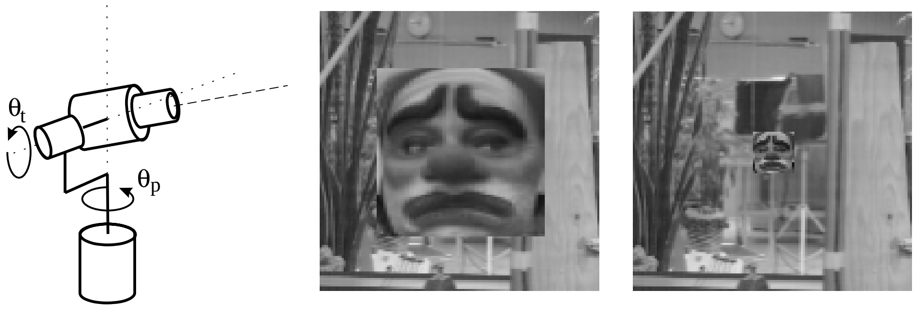

Figure 8: **(left)** Pan/Tilt camera. The axis of rotation are assumed to intersect in the camera optical center. **(right)** Simulated images with targets of scales 36% and 2.25%.

the object. Furthermore, in tracking tasks, this behaviour maximizes the possible target displacements while still keeping the object in the field of view.

In this work we consider a pan/tilt camera, and develop an explicit–tracking module to keep the observed object in the vicinity of the image center. This behaviour increases the amount of object data obtained by the implicit–tracking module. Since the explicit–tracking module is driven by motion measurements computed by the implicit–tracking module, we can say that both modules cooperate on the tracking task.

Camera Kinematics and Control

The camera used in this work has a simple pan/tilt configuration, as shown in Fig. 8. The explicit–tracking goal is to control the pan and tilt angles (θ_p, θ_t) according to the position of the template obtained by the implicit–tracking algorithm. The purpose is to make the optical axis intersect the center of the target template. When that happens, the image error (x, y) is zero. Otherwise, it is related to the angular position of the target relative to the camera optical axis. Although the real kinematic relations between image error and angular error are non linear, we will control the pan and tilt angular velocities of the camera with a linear proportional controller on the image error:

$$\dot{\theta}_p = -k_p x \quad ; \quad \dot{\theta}_t = -k_t y \tag{11}$$

Simulator To evaluate the performance of the algorithms with ground truth data we developed a simulator for the system. We assume a simple first order dynamic model for the velocity of the camera joints with a time constant of 200 msec and the sampling frequency is 10Hz (100 msec), which define a relatively slow dynamics. Therefore the control is not "one step" but instead has a lag that depends on target velocity and the camera model parameters.

We use two planar surfaces to simulate the environment : one is the background located $10m$ away from the camera and the other is the target at $0.5m$. We tested different scales for the target, from 36% to 2.25% of the full image area (see Fig. 8).

THE LOG-POLAR MAPPING

The foveated image representation used in this work is based on a log-polar image sampling. Beside its similarity to the human retina and visual cortex, one of the main characteristics of this representation is data reduction. This is obtained by reducing the resolution at the image periphery but maintaining high resolution in the fovea such that the overall information contained in the image is still perceivable. Just to illustrate this fact, in our particular implementation we map 128x128 cartesian images to 64x32 log-polar images and achieve 8 times increase in efficiency (both storage and speed). The log-polar transformation is defined as a conformal mapping from the points on the *cartesian* plane $\mathbf{x} = (x, y)$ to points in the *cortical* plane $\mathbf{z} = (\xi, \eta)$ [1], as represented in Fig. 9. The mapping is described by :

$$[\xi, \eta]^t = \mathbf{l}(x, y) = \left[\log(\sqrt{x^2 + y^2}), \arctan\frac{y}{x}\right]^t \tag{12}$$

$$[x, y]^t = \mathbf{l}^{-1}(\xi, \eta) = \left[\mathrm{e}^{\xi}\cos\eta, \mathrm{e}^{\xi}\sin\eta\right]^t \tag{13}$$

In this work we are mainly concerned in tracking moving objects. Apart from other beneficial algorithmic properties, the main advantage of the log-polar geometry for tracking applications is the higher informative content of objects occupying the fovea with respect to coarsely sampled background elements in the

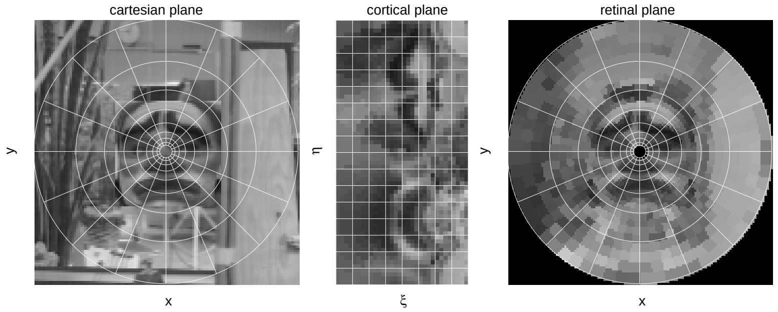

Figure 9: The log–polar transformation maps points from the cartesian plane (left) to the cortical plane (middle). The effective image resolution becomes coarser in the periphery, as can be observed in the retinal plane (right).

image periphery [8]. This embeds an implicit focus of attention in the center of the visual field where the target is expected to be most of the time.

The log-polar mapping preserves the shape of objects that undergo centered scale changes and rotations (it is scale and rotation invariant). This fact has been used to easily compute time–to–colision[12] and control vergence [7]. However, it is not "shift invariant" i.e. does not preserve the shape of objects under translation. This fact is usually referred a less desirable property because translations are very common transformations in tracking applications and there in no "easy" way to emulate them in the log-polar plane. While in cartesian coordinates a translation can be emulated by shifting the image coordinates, in log-polar it corresponds to a complicated and coupled transformation of coordinates. Notwithstanding, it can be computed and stored in a generic look–up–table.

Although cartesian coordinates were considered in the previous sections, the extension to log-polar coordinates is straightforward. Notice that 3D planar motion produces 2D transformations composed by intuitive deformations in the cartesian image plane (translation, rotation, etc.). In log-polar coordinates these deformations are not so intuitive and thus we prefer to define 2D transformations in cartesian coordinates. Then we express the corresponding log-polar deformations in terms of a map between the cartesian and log-polar warping fields. By using eqs. (12), (13) and (3), we obtain:

$$\mathbf{z}' = l\left(m\left(l^{-1}\left(\mathbf{z}\right)\right)\right) \Rightarrow m^{log} = l \circ m \circ l^{-1} \tag{14}$$

Both m and m^{log} are nonlinear functions of the motion parameters μ, and have closed-form expressions. Thus, the transformation of an image according to a correspondence map is basically the same in cartesian and log-polar domains. However, since log-polar images have less pixels, transformations are faster to compute, which is important to achieve high sampling rates and better tracking performance. To implement these transformations (both in cartesian and log-polar images) we use the *Intel Image Processing Library* [13]. This library contains optimized code with MMX instructions and performs linear or bicubic interpolation to avoid aliasing in the generated images, being capable of computing one log-polar warp in less than 10 msec in a PII 400Mhz machine. This way we can efficiently emulate not only translations but also any other types of transformations in log-polar images.

RESULTS

Several experiments are shown to evaluate the performance of the proposed methodologies. In particular we are interested in testing the motion estimation algorithm, evaluating the benefits of using foveated images and evaluating the planar model ability to approximate the geometry of other objects. For these purposes, we presents two simulated situations (the **open–loop** and **closed–loop** test) and one experiment with real images (the **robustness to non-planarity** test).

Open-loop test – log-polar vs cartesian images In this simulated experiment we test implicit–tracking alone and compare the use of log-polar and cartesian images with objects of different sizes. The actual dimension of the object **is not known** *a priori*, therefore the system selects an initial template that occupies the full image. The target translates linearly in 3D space. At each time instance the algorithm estimates target position, which is used as initial guess to the next time step. In Fig. 10 we present plots of the estimated template position for the log-polar and cartesian versions of the algorithm. From these plots we can observe that the performance of both versions is good for large objects but degrades when

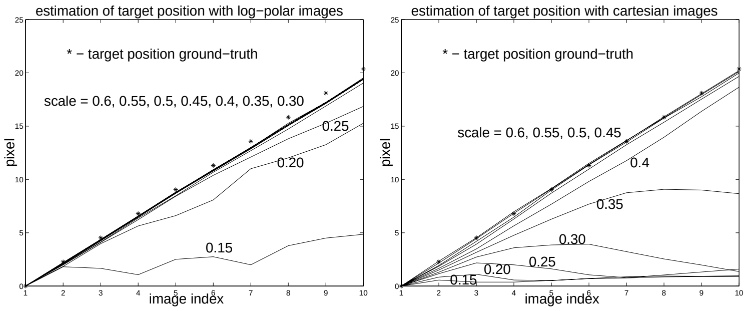

Figure 10: Comparison between log-polar **(left)** and cartesian **(right)** versions of the open-loop experiment. The true and estimated target position are represented for targets of several dimensions.

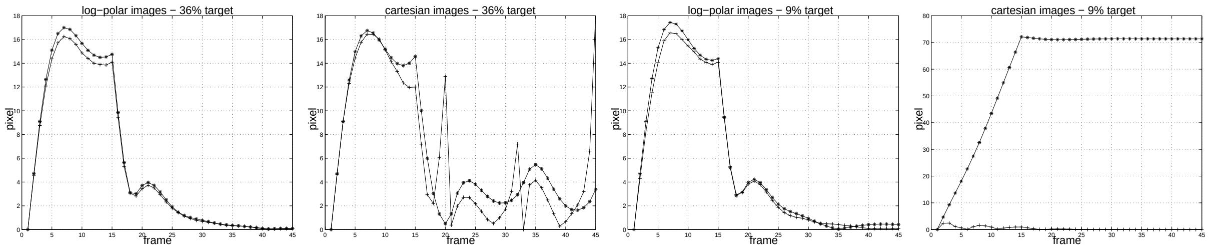

Figure 11: Estimated (+) *versus* true (∗) position of target the target. Comparison between log-polar and cartesian versions of the algorithm with 36% and 9% size objects

target size diminishes. Notwithstanding, the log-polar version copes smaller objects than the cartesian version.

Closed-loop test – cooperating behaviours This is also a simulated experiment and illustrate the integration of implicit and explicit–tracking behaviours. Simulated camera pan and tilt angles are controlled to keep the observation direction on the centre of the target. The target moves with constant velocity during the first 15 time steps and then stops. In this case the displacements can be larger than in the implicit–tracking experiments because the target is actively kept inside the field of view. Results are shown in Fig. 11 Notice in the plots that a 9% size object is not tracked by the cartesian algorithm. Even for 36% size cartesian tracking not very stable and sometimes looses track of target motion. The log-polar algorithm performs very well in both cases, presenting a tracking error less than two pixels in the image plane.

Real images – robustness to non-planarity This experiment is performed with a real non-planar target placed in front of the camera. The target was rotated along its vertical axis which corresponds to image transformations not following the assumed planar model. Results of the implicit–tracking module are presented in Fig. 12 in a qualitative way, since no ground truth data is available in this case. A quadrangular line surrounding the target illustrates the computed transformation. With log-polar images, the algorithm computes a coherent planar approximation for this transformation. Results for the cartesian case are also presented and show that it fails to reliably approximate the target deformation.

CONCLUSIONS

We have presented a framework for attention fixation on moving targets composed by the integration of two behaviours: implicit–tracking and explicit–tracking. These behaviours are analogous to the covert and overt visual attention mechanisms in what is related to eye movements. The main conclusions to retain from this work are the following:

- Target location is based on motion continuity and geometric models (planar), reducing the search range to a vicinity of the initial solution, and the dimension of the search space to the number of model parameters.
- Targets are defined and detected on log-polar images, increasing computational efficiency and robustness to target scaling and non-planarity.

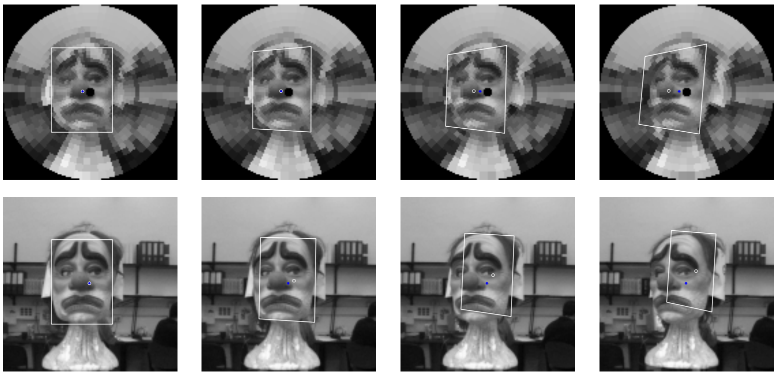

Figure 12: Computing an approximation to a non-planar transformation with real images – the overlaid window represents the computed transformation. **(top row)** With log-polar images. **(bottom row)** With cartesian images.

- An optimization strategy (the appearance prototypes) is introduced, representing a compromise between local/global optimization methods and evidencing good convergence/range properties with controlled computational cost.
- The explicit–tracking module moves the observation direction towards the target, increasing the amount of target related information available and augmenting the amplitude of possible target motions.

Experimental results, obtained in simulated and real setups, support these conclusions. Further work will focus on the integration of other visual cues for target detection and validation (color, shape, motion, etc.).

References

[1] E. Schwartz. Spatial mapping in the primate sensory projection : analytic structure and relevance to perception. *Biological Cybernetics*, 25:181–194, 1977.

[2] G. Sandini and V. Tagliasco. An Antropomorphic Retina-like Structure for Scene Analysis. *Computer Vision, Graphics and Image Processing*, 14(3):365–372, 1980.

[3] G. Sandini, C. Braccini, G. Gambardella and V. Tagliasco. A Model of the Early Stages of the Human Visual System: Functional and Topological Transformation Performed in the Peripheral Visual Field. *Biological Cybernetics*, 44:47–58, 1982.

[4] C. Weiman. Log-polar vision for mobile robot navigation, In *Proc. of Electronic Imaging Conference*, pages 382–385, Boston, USA, November 1990.

[5] C. Weiman and G. Chaikin. Logarithmic Spiral Grids for Image Processing and Display. *Comp Graphics and Image Proc*, 11:197–226, 1979.

[6] J. Santos-Victor and G. Sandini. Visual behaviors for docking, *CVIU*, 67(3), September 1997.

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

[8] A. Bernardino and J. Santos-Victor. Binocular visual tracking: integration of perception and control. *IEEE TRA*, 15(6), December 1999.

[9] B. Horn. *Robot Vision*, MIT Press, McGraw Hill, 1986.

[10] A. Gelb. *Applied Optimal Estimation*, The M.I.T Press, 1974.

[11] A. Bernardino, J. Santos-Victor and Giulio Sandini. Tracking planar structures with log-polar images In *Proc. SIRS2000*, Reading, UK, July 2000.

[12] M. Tistarelli and G. Sandini. On the advantages of polar and log-polar mapping for direct estimation of the time-to-impact from optical flow. *IEEE Trans. on PAMI*, 15(8):401–411, April 1993.

[13] Intel Corporation. Intel image processing library. http://developer.intel.com/vtune/perflibst/IPL.