2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Real Time Vision for Robotics Using a Moving Fovea Approach with Multi Resolution

Rafael Beserra Gomes and Luiz Marcos Garcia Gonalves and Bruno Motta de Carvalho

Abstract— **We propose a new approach to reduce and abstract visual data useful for robotics applications. Basically, a moving fovea in combination with a multi-resolution representation is created from a pair of input images given by a stereo head, that reduces hundreds of times the amount of information from the original images. With this new theoretical approach we are able to compute several feature maps, including several filters, stereo matching, and motion, in real time, that is at more than 30 frames per second. As the main contribution, the moving fovea allows, most of the time, a robot to avoid performing physical motion with the cameras in order to get a desirable region in the images center. We present mathematical formalization of the moving fovea approach, the algorithms, and details of the implementation of such schema. We validate it with experimental results. This approach has demonstrated to be very useful to robotics vision.**

I. INTRODUCTION

We propose a moving fovea approach for low-level vision that works in combination with a multi-resolution, multi-feature representation applied to robotics stereo vision. While using a conventional attention system, a robot would have to move its resources (cameras) in order to get a desired point at the images center (the fovea). So the basic idea of our method is to change the region of interest in both images without performing unnecessary physical motions for that. Basically, once a region other than the current focus of attention is chosen inside both images, image processing and computer vision techniques are applied to provide data reduction and feature abstraction around the chosen points building the multi-resolution, multi-feature representation.

The proposed algorithm for data reduction allows our system to achieve real-time processing (more than a 30 fps frame rate) running in a conventional 2.0 GHz Intel processor. This processing rate allows a robotics platform to perform tasks involving attention control (tracking) and recognition behaviors. So the basic contribution of the proposed approach is to allow the robot to select regions of interest in its environment, that is, to foveate (verge) its robotics eyes on the selected regions, without the need of moving resources. Only by software calculations, by moving the fovea inside a current view of a scene, the system is able to keep its attention on the selected region as necessary, for example, to recognize or manipulate objects, and to eventually shift its focus of attention to another region, once a task has been finished.

R. Gomes rafaelbg@dca.ufrn.br, L. Gonalves lmarcos@dca.ufrn.br and B. de Carvalho bruno_m_carvalho@yahoo.com are with Universidade Federal do Rio Grande do Norte, Natal, Brazil.

On the top of this model, high-level vision tasks, such as attention strategies, for tracking a ball are developed. Recognition tasks could also be successfully performed based on feature extraction from the resulting representation. Both tasks validate the proposed model and its use in robotics applications. We remark that we do not want to design a system to perform specific tasks. We want a behaviorally active system that may be able to perform different tasks in different environments or situations, automatically responding in real-time, to environment changes. In this way, we believe that data reduction and abstraction are the main key of the system. The model for reducing data and abstracted features, plus the moving fovea proposed in this article, that allowed us to develop a system with these requirements, are the main issues that will be treated in this paper.

II. RELATED WORKS

Vision is so far the most powerful biological sensory system. Since computers appeared, several vision systems have been proposed trying to provide vision sense to machines. However, the heterogeneity of techniques for modeling a complete vision algorithm makes the implementation of a real-time vision system a complex task. The necessary quantity of visual features grows very fast depending on the task and consequently the amount of processing to recover them. For example, if stereo vision is used, the main goal is to recover the disparity of object projections, given two different images of the same scene [1], [2]. Disparity calculation is the main issue here, making stereo a complex problem. Several algorithms have been implemented in order to reduce its complexity or to enhance its precision [3], [4], [5], [6], [7].

Of course, recovering disparity feature (or depth) [8], [9], [10], [11] is not the only purpose of using vision in robots. Several tasks can rely on vision, based on features such as intensity, texture, edges, motion, wavelets, Gaussians, stereo, and motion between other several ones that can be extracted from visual data. For example, simple tasks involving attention and recognition behaviors can use Gaussian derivatives [12], or a combination of them with disparity and motion [13].

The use of full resolution images complicates the feature extraction processing, if real time is a requirement. Several models have been proposed in the literature for image data reduction and feature abstraction. Most of them treat visual data as a classical multi-resolution image, a pyramidal structure, or as a scale space. The credit for the idea of using the classical multi-resolution model in visual search

can be given to Leonard Uhr [14]. The scale space theory is formalized by Witkin [15] and further by Lindeberg [16]. The Laplacian pyramid was introduced by Burt and Adelson [17]. Multi-resolution was integrated into an argument for visual attention by Tsotsos [18], [19]. A problem when calculating the classical pyramid is the processing time. It does not allow real-time execution, mainly for robotics purposes, unless dedicated architectures for vision processing are used. In fact, most works do not explicitly deal on-line with the real-time constraints experimented in robotics problems.

The classical multi-resolution approach has been adapted using its positive aspects, as the nice property of multi-scale processing for feature enhancement, but fitting all data into a much more compact structure [20]. Only small images are used, in the above approach, in the pre-processing phase. Then, features can be calculated from these images using any desirable filtering over the small images. This approach has proven to be fast enough to allow real-time processing, as shown in previous work [20]. A problem with the above approach is that the fovea (the highest resolution image) is always centered in the stereo images, so a physical motion is necessary if another region (or scene point) has to be put on it, which takes time. We note that this approach some how imitates what happens in biological system, where saccadic movements are constantly realized.

We propose the use of a moving fovea as an enhancement the above model. Instead of forcing the robot to perform a physical motion, in order to get a new interest point in the center of the images given by the attention process, we perform a new calculation of the above structure changing the original position of each small image in the acquired images. When the fovea gets close to the borders of the current view, a physical motion has to be suggested to the robot anyway. However, this simple idea has allowed a much better performance of the system as will be shown, mainly for tracking and recognition.

III. The Multi-resolution Multi-feature Using Moving Fovea Structure

The multi-resolution model allows visualization of the whole scene with different resolution images with the same size. Images with higher resolution include more detail of the scene objects however covering a smaller region of the scene. This allows real-time implementation of attention and recognition behaviors. The complex problem solved in this work is how to move the fovea in this model.

A. Data Reduction and Abstraction in Robots

To reduce visual data, we use a light structure, regarding data reduction and abstraction, made of multi-features (MF) extracted from a multi-resolution (MR) representation of the scene. This technique has been used by a robot equipped with a stereo head [21] shown in Figure 1. In this version of the MRMF approach [21], we use an embedded PC in the robot with the two cameras connected to it. This PC has two frame grabbers that get as input the two video streams from the cameras on the stereo head shown in Figure 1. Both structures (MR and MF) represent the mapping of topological/spatial indexes from the sensors to multiple attention or recognition features.

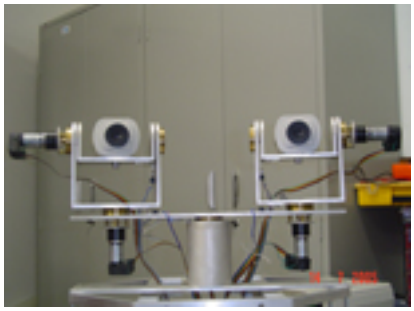

Fig. 1. Stereo Head platform with 5 mechanical degrees of freedom

As the MRMF with moving fovea is an improvement over the classic multi-resolution pyramid, we describe it using the notation that supports the description of the schema described next. Each frame captured by the cameras is re-sampled in levels numbered from 0 to m ($m+1$ levels of resolutions), each level centered at the original image I of size $U = (U_x, U_y)$. Each k_{th} level of resolution R_k is the mapping of an area of size $S_k = (S_{k,x}, S_{k,y})$ in the original image to an area of size $W = (W_x, W_y)$ constant for all resolutions. The size of the first level should have a resolution equals to the original image I. To this effect, $S_0 = U$ and $S_m = W$. The intermediate levels are interpolated between these two final ones. Once all levels are centered maps in I, each mapped area has a shift $\Delta = (\Delta X, \Delta Y)$ in relation to the origin $(0,0)$ of I. As R_0 maps the whole original image, $\Delta R_0 = (0,0)$. In I, R_m maps W pixels. Then, $\Delta R_m = (U-W)/2$. With the interpolation of Δ we obtain:

$$\Delta R_k = \frac{k(U-W)}{2m} \tag{1}$$

The Figure 2 shows those concepts using 4 levels.

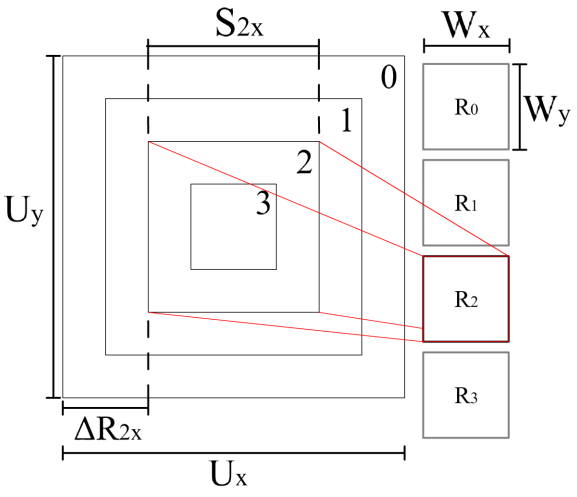

Fig. 2. MRMF using 4 levels

The mapping of I to R_k associates S_k pixels from the original images with W pixels of R_k. For this, a rate $P_k = (P_{k,x}, P_{k,y})$ is fixed, such that $P_k W = S_k$, that can be rewritten as:

$$P_k = \frac{mU + Wk - kU}{mW} \tag{2}$$

A block $B_k(x,y) : \{0...W_{k,x}-1\}x\{0...W_{k,y}-1\} \rightarrow \mathbb{PN}^2$ is defined by:

$$\begin{aligned} B_k(x,y) \quad & = \{(a,b) \in I| \\ & \Delta R_{k,x} + P_{k,x}x \leq a < \Delta R_{k,x} + P_{k,x}(x+1), \\ & \Delta R_{k,y} + P_{k,y}y \leq b < \Delta R_{k,y} + P_{k,y}(y+1)\} \end{aligned} \tag{3}$$

This way, a block B_k of P_k pixels is mapped into a single pixel in R_k. This mapping is a function $\psi : \mathbb{PN}^2 \rightarrow \mathbb{N}$ so that:

$$R_k(x,y) = \psi(B_k(x,y)) \tag{4}$$

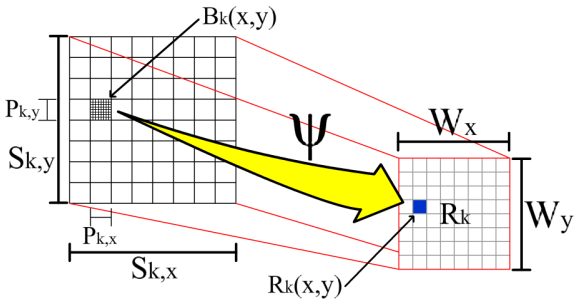

Fig. 3. ψ function maps each block B_k to a R_k pixel

For this ψ function, we have chosen the 4 pixels distant $1/3P_k$ of the central pixel of each block. This results in a good representative value for the pixel group at the same time that is computationally fast.

B. Introducing the moving fovea concept

In the moving fovea model, the center of the region that maps R_k may vary its position, in such a way that it can be any region of the original image. The center of the better resolution R_m is indicated by a fovea vector F. As R_0 already maps the whole image, this level keeps the same pixel values, no matter where the fovea is placed. To the level R_m, the fovea center may move from its origin $(0,0)$ (the center of I) up to a limit on which the mapping is close to the border on the acquired image. Hence, the fovea center should be at a distance $W/2$ from any border. So the center of the fovea F should be between $(W-U)/2$ and $(U-W)/2$. Note that whenever $F = (0,0)$, we have the same sampling schema of the MR model without moving fovea. The positions for the intermediate levels are interpolated from the first and last resolutions, summed to the shifting when $F = (0,0)$:

$$\delta R_k = \Delta R_k + \frac{kF}{m} \tag{5}$$

which can be rewritten as:

$$\delta R_k = \frac{k(U - W + 2F)}{2m} \tag{6}$$

The Figure 4 shows those concepts using 4 levels. With the moving fovea concept, a block is now defined by:

$$\begin{aligned} B_k(x,y) \quad & = \{(a,b) \in I| \\ & \delta R_{k,x} + P_{k,x}x \leq a < \delta R_{k,x} + P_{k,x}(x+1), \\ & \delta R_{k,y} + P_{k,y}y \leq b < \delta R_{k,y} + P_{k,y}(y+1)\} \end{aligned} \tag{7}$$

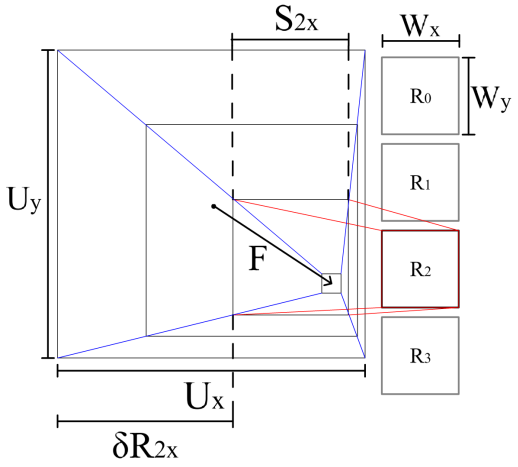

Fig. 4. The MRMF with moving fovea using 4 levels. F is the fovea vector.

IV. Mapping

Due to the possibility of a mapping between positions referring to the original image (including the fovea position), positions referring to each level, and positions from level to level, it is important to use the model successfully, as will be described by an algorithm in a later section.

A. between level and the original image

Let be υ a function that, given $p = (a,b)$ in I and a level k, results in a position q in R_k so that the block $B_k(q)$ contains p. By the definition of B_k 7 we have:

$$\begin{aligned} x' \leq \frac{a - \delta R_{k,x}}{P_{k,x}} < (x'+1), \\ y' \leq \frac{b - \delta R_{k,y}}{P_{k,y}} < (y'+1) \end{aligned} \tag{8}$$

As we want integer positions, $(x', y') \in \mathbb{N}^2$, we can write (8) so that:

$$\upsilon_k(a,b) = \left(\lfloor \frac{a - \delta R_{k,x}}{P_{k,x}} \rfloor, \lfloor \frac{b - \delta R_{k,y}}{P_{k,y}} \rfloor \right) \tag{9}$$

Let now be ω a function that, given a pixel $p = (a,b)$ in R_k, results in the possible positions q in I so that the block $B_k(p)$ contains all q. By the definition of B_k 7 we have:

$$\omega_k(p) \subseteq \bigcup_t \delta R_k + P_k p + t \tag{10}$$

, where $t \in \{0...W_{k,x} - 1\} \times \{0...W_{k,y} - 1\}$.

B. between levels

Suppose that we want to map a pixel at level k to the level j. A pixel (x, y) at level k is a result of $\psi(B_k(x, y))$. Each pixel in $B_k(x, y)$ is also in a block $B_j(x', y')$ if $j < k$. However, this is not necessarily true for each pixel if $k < j$ (see Figure 5). Let be Φ the function that results in the set of those index (x', y'):

$$\Phi_{k,j}(x, y) = \{(x', y') | \exists (a, b).(a, b) \in B_k(x, y), \\ (a, b) \in B_j(x', y')\} \tag{11}$$

Note that Φ can be $\emptyset$ if $k < j$.

We can evaluate the Φ function, evaluating each pixel in B_k. Given a pixel $p = (c, d)$ that is in R_k, using 10 we have a set $A = \omega(p)$, and we can convert each pixel in A referring to I to level j using 9:

$$\Phi_{k,j}(p) \subseteq \bigcup_t \upsilon_j(t) \tag{12}$$

, where $t \in \omega_k(p)$.

Note that if $\upsilon_j(t) \notin \{0...W_{k,x} - 1\} \times \{0...W_{k,y} - 1\}$, there's no correspondent block at level j, since these values are not defined by B_j.

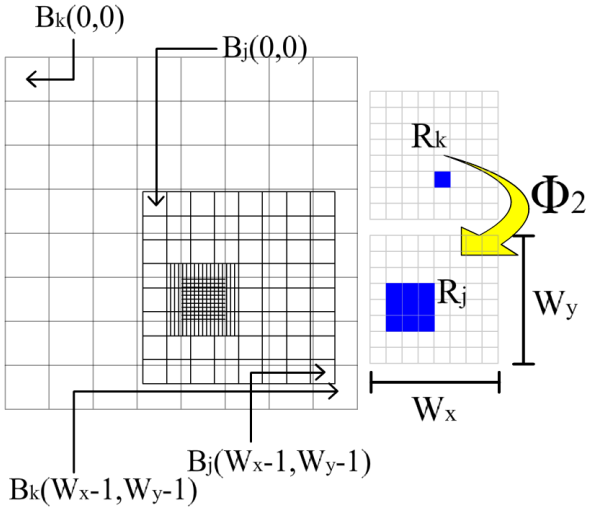

Fig. 5. Matching pixels in level k to a next level j. The vertical hatch indicates blocks in level j that contains one or more pixels that are also in the block hatched by horizontal lines.

V. Feature Extraction

In order to test the proposed methodology, we further extract some desired features to perform tracking and recognition. Basically, we convolve each resolution level of the above fovea representation with several filters and further stereo and motion features are also calculated from it.

A. Calculation of stereo disparity

Given the position of the fovea in one of the images (say I_1), the position of the fovea in the other image, I_2, has to be determined, and also the opposite, fovea of I_2 in I_1. We decided to choose the fovea in both images to be at the same point in the scene, besides referring to the original image when stereo disparity is calculated. This search is done by using cross-correlation, considering epipolar restrictions, that is, $F_{2,y} = F1, y$.

To find the position of the fovea at the other image, the score that maximizes correlation on the first level is used.

Once the two images are calculated, disparity is computed using correlation scores, between pixels in both images, that is in practice implemented by several convolution operations [1]. Performance is yet enhanced by using one level to predict disparity for the next one.

As the position of the center of the fovea may be in different positioning at the two images, the displacement of level k that was δR_k will be denoted by $\delta R_{1,k}$ and $\delta R_{2,k}$ for the resolution k on the left and right images respectively.

For calculating stereo correspondence using multi-resolution with moving fovea, we first compute it at level zero, of lower resolution. Disparity for the other levels can be calculated using an estimation from the previous level through a simple refining schema, what is given by the following algorithm:

Algorithm 1 Stereo correspondence for the level k ($k > 0$) of I_1, where D is the disparity map and $corr(y, x_1, x_2)$ means the correlation between a window centered at (x_1, y) in I_1 and a window centered at (x_2, y) in I_2

```
t ⇐ ⌈P_{k-1} / P_k⌉
for i = 0 to W_y do
  for j = 0 to W_x do
    p ⇐ Φ_{k,k-1}(i, j)
    q ⇐ p + D_{1,k-1}(p)
    r ⇐ Φ_{k-1,k}(q)
    if r is inside R_{2,k} then
      maxscore = −∞
      for k = −t to t do
        if (r_x + k, r_y) is inside R_{2,k} then
          if corr(i, j, r_x + k) > maxscore then
            maxscore = corr(i, j, r_x + k)
            D_{1,k}(i, j) = (r_x + k) − j
          end if
        end if
      end for
    end if
  end for
end for
```

Figure 6 shows disparity maps for one of the images in the Tsukuba image database. Ground truth is shown in Figure 7.

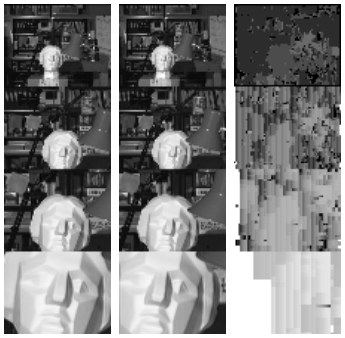

Fig. 6. Equalized disparity calculated only for the first level, a refining from this one is done for the following levels.

Fig. 7. Ground truth

VI. Experiments and results: testing attention and recognition behaviors

Basically, in order to validate the proposed methodology, several experiments involving performance, attention, and recognition behaviors were performed.

Remember that what is been testing is the time performance if our multi-resolution with moving fovea approach is applied, not the algorithms themselves.

If an algorithm can be applied to a classic image representation, then it's possible to apply it to each resolution in a straight forward manner, since each resolution can be stored in the same conventional way. In most case, someone would have to adapt her/his algorithm if it's location dependent, as the stereo algorithm. In these cases, he/she has to convert positions between levels to evaluate indexes to their real values.

A. Implementation issues

It's worth to remark that most expression developed can be expressed only in function of a few parameters, assuming that W, U, m, k (a different block of processing for each level) are all constants and can be pre-processed using meta-programming. All experiments that are described on next use this approach. For example: $\Phi_{k,k-1}$ can be rewritten to:

$$f_k(p) = \frac{WU - W^2 + 2WF + P(2mU + 2kW - 2kU) + t}{2(W(k-1) + U(m-k+1))}$$

B. Performance experiments

We applied the algorithm on a real time acquired video stream, whose images have the following parameters:

$U = (640, 480)$ (images size)
$W = (32, 24)$ e $(64, 48)$ (resolutions for two MRMF)
$m = 3$ (both MRMF with 4 levels of resolution)

Time results for each phase are shown in Table I. A 2.0 GHz processor was used in this experiment. Overall, a gain of about 1800% in processing time was observed from the original images to the reduced ones. As said, used filters were: gradient (x, y and threshold of their magnitude), Gaussian, Gaussian gradient (x, y and threshold of their magnitude), and Laplacian. In the Table I, *Stereo simple* is without estimating disparity from one level of less resolution to the next, that is, it starts from zero at all levels. *Stereo predict* is with estimation of disparity from one level to the next.

TABLE I
Time performance (MR-MF).

Resolution	32x24	64x48	96x72	128x96
MR-MF	0.2ms	1.2ms	1.9ms	2.6ms
Filtering	0.7ms	4.9ms	11.7ms	24.1ms
Stereo simple	0.7ms	9.0ms	50.2ms	110.0ms
Stereo predict	0.8ms	4.3ms	20.0ms	40.2ms

C. Testing features in tracking behavior

In tracking experiment, a hand holding a ball appears in front of the camera mount, an user signals the initial position of the fovea (at the ball) and the system should track it by only changing the position of the fovea in the image given by the current viewing position. This is done using the less resolution level, where the fovea can be tracked and the corresponding one in the other image can be easily calculated by using a correlation measure. When the ball is almost leaving the visual field during the tracking, the system suggests a movement. Figure 8 shows the tracking procedure. By using the moving fovea approach, it is possible to disengage attention from actual position and to engage it to another position from a frame to another, in real time. If moving robot resources is required every time that attention changes, this could take some 500 ms [20].

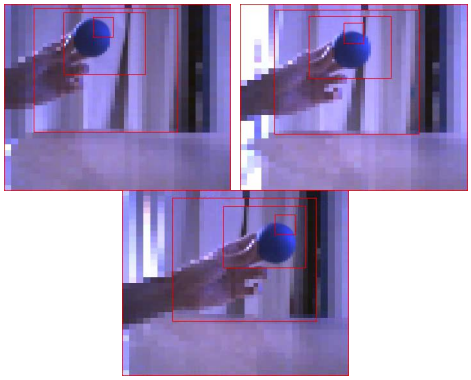

Fig. 8. Tracking a ball using a moving fovea.

D. Testing features in recognition behavior

In another experiment involving recognition, two objects, a tennis ball and a domino, were presented in several positions to the system. About 35 images were taken for each one, on-line. Then, the above model was applied to all of them and a neural network was trained with 1300 epochs, using a

threshold function of the gradient as the net input data and a bit vector as the net output data with each bit indicating a correspondent object. . The same objects were presented again to the cameras and the activation calculated in the net. The result was in about 70% of positive identification for the ball and about 85% for the domino, even though the object was present at the peripheral region of the camera.

VII. Conclusion and future work

We have built a useful mechanism involving data reduction and feature abstraction that could be integrated and tested in attention control and recognition behaviors by adding a moving fovea. To do that, the first step is the determination of the position in which the fovea must rely. Then, parameters are determined for data reduction and feature abstraction. By using an efficient down-sampling schema, a structure derived from the classical pyramid, however much more compact, is constructed in real-time (some 1.2ms in a PC 2.0 GHz, for 4 $\times$ 64 $\times$48 resolution images). Then, computer vision techniques, as shape from stereo, shape from motion, and other feature extraction processes are applied in order to obtain desired features (each single filter costs less than 500 μs).

By using the proposed model, we tested behaviors that have accomplished real-time performance mainly due to the data reduction and abstraction performed. Also, the moving fovea representation proposed has allowed to perform tasks as overt attention to be done in real-time, that can be applied to accelerate some tasks. So the main contributions are the enhancement done over a previous schema for data reduction and feature abstraction with the inclusion of a moving fovea. We remark that experiments in attention and recognition, using small images (some with low resolution) were done. Based on this fact (robustness), we believe that this approach can be used in other high-level processes, in order to accomplish other tasks, as navigation for example.

The ability of changing attention focus is the basis not only for the tasks described, but also for other more complex tasks involved in robot cognition [20]. This model changed a previous approach somewhat inspired by the biological model in the sense that the more precise resolution levels are located in the center of the image. In this way, the less resolution levels can be used for example to detect motion or features to be used in navigation tasks (mainly bottom-up stimuli) and the finer levels of resolution can be applied to tasks involving recognition as a text reading or object manipulation. A search task can use a combination of one or more levels. Of course, in this case, the moving fovea does play an important role, avoiding the head of performing unnecessary motions.

A problem found in our approach is that Φ is a set of potential pixels, and choosing anyone can introduce location errors that can be accumulated along a refining schema. A solution to this problem is to use some interpolation method that, on the other hand, could introduce more computational costs. Besides, if integer operations are always desired, each δR_k can also introduce error, but no more than $2m - 1$ (acceptable for real application) and each P_k no more than $mW - 1$. However, in most cases it's possible to choose suitable resolution dimensions and m value so that P_k is always an integer.

References

[1] D. Marr, *Vision – A Computational Investigation into the Human Representation and Processing of Visual Information*. Cambridge, MA: The MIT Press, 1982.

[2] B. K. P. Horn, *Robot Vision*. MIT Press, 1986.

[3] Y. Ohta and T. Kanade, "Stereo by intra and inter-scanline searching using dynamic programming," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 2, p. 139, 1985.

[4] D. Reimann and H. Haken, "Stereo vision by self-organization," *BioCyber*, vol. 71, no. 1, pp. 17–26, 1994.

[5] R. D. FREEMAN and I. OHZAWA, "On neurophysiological organization of binocular vision," *Vision Research*, vol. 30, pp. 1661–1676, 1990.

[6] D. J. FLEET, H. WAGNER, and D. J. HEEGER, "Neural encoding of binocular disparity: Energy models, position shifts and phase shifts," Personal Notes, Tech. Rep., 1997.

[7] C. L. Zitnick and T. Kanade, "A cooperative algorithm for stereo matching and occlusion detection," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 7, pp. 675–684, July 2000.

[8] J. Hespanha, Z. Dods, G. Hagger, and A. Morse, "Decidabilit of robot positioning tasks using stereo vision system," *37th Conference on Decision and Control*, 1998.

[9] Y. Matsumoto, T. Shibata, K. Sakai, M. Inaba, and H. Inoue, "Real-time color stereo vision system for a mobile robot based on field multiplexing," *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1997.

[10] E. Hubber and D. Kortenkamp, "Using stereo vision to pursue moving agents with a mobile robot," *proceedings on Robotics and Automation*, 1995.

[11] D. Murray and J. Little, "Using real-time stereo vision for mobile robot navigation," *Autonomous Robots*, 2000.

[12] R. P. N. Rao and D. H. Ballard, "Probabilistic models of attention based on iconic representations and predictive coding," *In Neurobiology of Attention, L. Itti, G. Rees, and J. Tsotsos (editors)*, 2004.

[13] L. M. G. Gonçalves and R. A. Grupen, "Integrating attention and categorization behaviors in robotics," in *Proc. of the 6th International Conference on Intelligent and Autonomous Systems*. IEEE Computer Society Press, July, 25-27th 2000.

[14] L. Uhr, "Layered 'recognition cone' networks that preprocess, classify and describe," in *IEEE Transactions on Computers*, 1972, pp. 758–768.

[15] A. P. Witkin, "Scale-space filtering," *Proc. 8th International Joint Conference on Artificial Intelligence*, vol. 1, no. 1, pp. 1019–1022, 1983.

[16] T. Lindeberg, "Scale-space theory in computer vision," *Kluwer Academic Publishers*.

[17] P. Burt and T. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 9, no. 4, pp. 532–540, 1983.

[18] J. K. Tsotos, "A complexity level analysis of vision," in *Proceedings of International Conference on Computer Vision: Human and Machine Vision Workshop*, I. Press, Ed., vol. 1, June 1987.

[19] J. K. Tsotsos, "Knowledge organization and its role in representation and interpretation for time-varying data: the alven system," pp. 498–514, 1987.

[20] L. M. G. Gonçalves, R. A. Grupen, A. A. Oliveira, D. Wheeler, and A. Fagg, "Tracing patterns and attention: Humanoid robot cognition," *The Intelligent Systems and their Applications*, vol. 15, no. 4, pp. 70–77, July/August 2000.

[21] S. Segundo and L. M. G. Gonçalves, "Redução e abstração de dados no projeto robosense," in *SIBGRAPI '97 - X Brazilian Symposium of Computer Graphic and Image Processing - WTDCGPI*, October 2004.