From Pixels to Objects: Enabling a spatial model for humanoid social robots

Dario Figueira Manuel Lopes Rodrigo Ventura Jonas Ruesch

***Abstract*— This work adds the concept of object to an existent low-level attention system of the humanoid robot iCub. The objects are defined as clusters of SIFT visual features. When the robot first encounters an unknown object, found to be within a certain (small) distance from its eyes, it stores a cluster of the features present within an interval about that distance, using depth perception. Whenever a previously stored object crosses the robot's field of view again, it is recognized, mapped into an egocentrical frame of reference, and gazed at. This mapping is persistent, in the sense that its identification and position are kept even if not visible by the robot. Features are stored and recognized in a bottom-up way. Experimental results on the humanoid robot iCub validate this approach. This work creates the foundation for a way of linking the bottom-up attention system with top-down, object-oriented information provided by humans.**

Keywords:Object Recognition, Depth Perception; Stereo Computer Vision; Salience Map; Spatial Model

I. INTRODUCTION

For humanoid robots to autonomously act in our daily environment, they must be endowed with the capability of perceiving objects, e.g. required to handle. Thus, an appropriate representation in order to memorize and recognize these objects is mandatory. However, robot sensory apparatuses only provide raw sensory data. Taking vision as a sensor, how can it bridge the gap between raw pixels and the concept of object? And how can it realize their relative positions within the surrounding environment, even when they are temporarily out of the cameras field of view?

The system presented here addresses these problems, by taking a bottom-up, developmental approach. The developed module builds upon an existing low-level attention system. The previous work provides a salience map with respect to a robot-centric coordinate system (ego-sphere) [1]. This saliency map, together with a inhibition of return mechanism (IOR), allows the robot to saccade from salient point to salient point. However, these salience points correspond to preattentive features, e.g., movement, color, and shape, that do not incorporate the concept of object.

The goal of the work presented here is to endow the robot with the capability of learning and recognizing objects. By integrating this capability into the existing architecture, the attention module will be able to acknowledge the salience of known objects, because they are recognized as such. Moreover, the capability of recognizing known objects by visual features paves the way for higher level modules, such as language, to implement complex cognitive functions.

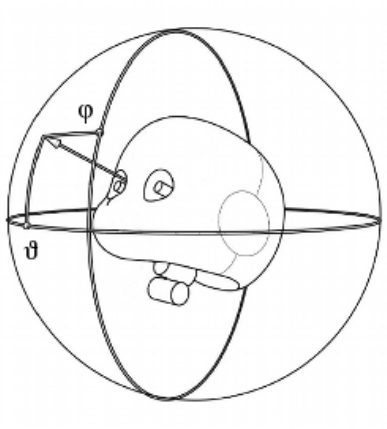


Fig. 1. Ego-sphere: a spherical map of the surroundings with a spherical coordinate system (azimuth ϑ and elevation φ.)

The robot considered here is the iCub humanoid robot[1]. However, just the head and torso modules were employed. While the head has the 6 degrees of freedom, the torso is fixed. We consider a robot centered coordinate system, specifically, a torso anchored coordinate system.

To fulfill the goal of enabling an agent to commute automatically its attention focus from recognized object to recognized object, we are modeling the environment with a salience map [2]. We project the surrounding space and objects into a spherical coordinate system centered in the neck of the robot, an egocentric sphere or ego-sphere, as defined in [1] (Figure 1).

A spatial model for the robot is here understood as a model representing the environment surrounding it, namely the known objects, together with their relative positions to the robot.

In this work we add to the spatial saliency map implemented in [1] and endow the system with an interpretation of its surroundings. The system now maps known visual objects, so it can know where they are after looking away. Instead of the short-time memory of the previous works [1], [2], the system remembers where important objects are at longer time scales.

This work was supported by the European Commission, Project IST-004370 RobotCub, and by the Portuguese Government - FCT (ISR/IST plurianual funding) through the POS_Conhecimento Program that includes FEDER funds, and through the project BIO-LOOK, PTDC / EEA-ACR / 71032 / 2006

D. Figueira, M. Lopes and R. Ventura are with Institute for System and Robotics (ISR), Instituto Superior Técnico, TULisbon, Portugal {dfigueira,macl,yoda}@isr.ist.utl.pt

J. Ruesch is with the Artificial Intelligence Laboratory, Department of Informatics, University of Zurich, Switzerland ruesch@ifi.uzh.ch

[1]http://www.robotcub.org/

We implemented an algorithm to automatically store to a database new objects as they get close to the eyes of the robot (the cameras). To do so, we compute a depth perception map [3], [4] of the image to determine if something is in close proximity and under the robot's scrutiny. If so, its representation is stored to a database to be later remembered and recognized.

We chose the Scale Invariant Feature Transform (SIFT) [5] algorithm to enact our recognition. We also used this algorithm to match corresponding points in pairs of stereo images, thus computing the disparity or depth of these points. By using this algorithm for detecting distance, we define a new object as a cluster of SIFT features in close proximity to the cameras, while already saved objects are detected continuously in the input images by the SIFT algorithm. Finally, the recognized objects in the robots field of vision are inserted into the egocentric map.

In the next section we shall describe our spatial model in a general sense, leaving out the implementation details to be discussed in the third section. Then, in the forth section, we describe an experiment in which the functioning of our work is illustrated, and present experimental results to validate the approach. Finally, we finish with some conclusions and future work.

II. ARCHITECTURE

The architecture, displayed in Figure 2, has several interconnected modules to form a sensing-deliberation-actuation chain. It is motivated on the Itti and Koch model [6] where stimuli from various sources are represented and combined into a single salience map. Then, the point that maximizes this map is selected, as the one winning the robot attention. Finally the robot gazes towards the new selected attention point.

The ego-sphere keeps a short-memory of the previously looked upon positions, in the form of an inhibition-of-return mechanism (IOR). The IOR information reduces the salience levels of the already observed locations. The resulting behavior is the capability of the robot to fully explore its environment without being stuck on the absolute saliency maxima of the salience maps.

Our work adds a level of abstraction, the concept of objects, to the previous architecture. Before, the world consisted in basic salient signals, now there are already distinct object. To acquire this knowledge the robot has to solve two problems. *What* and *when* to learn a new object. A new object is learned when it is detected in close proximity of the robot. The object is assumed to consist in the image patch that is close to the robot. The proximity is defined has a arms length distance, *i.e.,* the reachable objects.

saved, the objects are stored together with a "label" that is simply a number (the order of appearance). When the robot has the possibility to ask humans around him for the names of the objects it is discovering and storing, new possibilities concerning associating object representations to names arise.

III. IMPLEMENTATION

In this section we provide the details to perform object segmentation and recognition.

Many different approaches have been used in computer vision to enable recognition, for instance, eigenspace matching has been used successfully by Schiele [7], others have used Speeded Up Robust Features (SURF) [8], and many have benefited from David Lowe's Scale Invariant Feature Transform (SIFT) [5]. Our approach, the latter approach, solves both problems of object segmentation and recognition. We chose SIFT over eigenspace matching for reasons such as invariance to scale and excelling in cluttered or occluded environments (as long as three SIFT features are detected, the object is recognized). And while the SURF algorithm is faster and performs generally well, SIFT's recognition results are still superior [9]. The setback about using this algorithm is that it takes a lot of processing time, the most efficient implementations are not able to run it in real time (24 FPS) [10].

SIFT [5] is an algorithm that extracts, features from an image. These features are computed from histograms of the gradients around the key-points, and are not only scale invariant features, but also invariant to affine transformations (*e.g.,* rotations invariant). Furthermore, they are robust to changes in lighting, robust to non-extreme projective transformations, robust up to 90% occlusion, and are minimally affected by noise. We use the SIFT algorithm to enable the recognition in our system because of all these powerful characteristics. Due to the nature of the SIFT features, its second drawback is the inability to extract features from a texture-less object, as shown in Figure 3: few or no features, in yellow dots, are found in areas with homogeneous color, such as on the

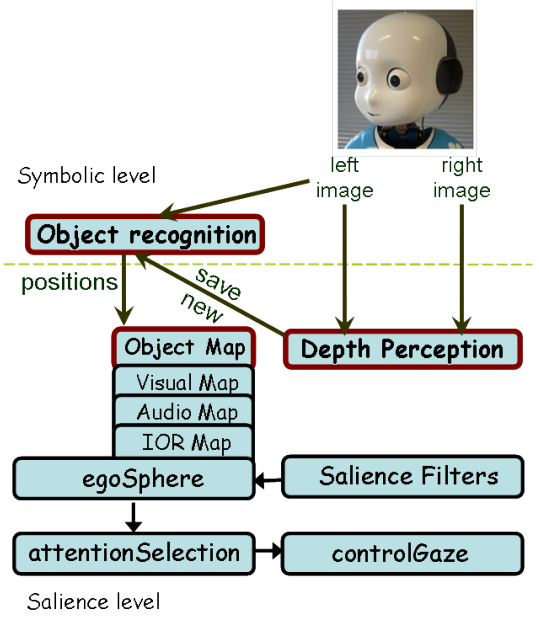


Fig. 2. System architecture, after introducing the modules presented here (dark red border): object recognition, depth perception, and a new map, the objects map.

Fig. 3. Example of SIFT feature extraction; the yellow dots correspond to the extracted features positions.

table, on the ground, or on the wall.

A. Depth Perception

The common way to determine depth, with two stereo cameras, is by calculating disparity. Disparity is defined as the subtraction, from the left image to the right image, of the 2D coordinates of corresponding points in image space. To calculate depth we require the knowledge of the following camera parameters:

- focal length f;
- camera baseline β;
- pixel dimension γ.

Also, we need to correctly match a point of the environment, seen in both stereo images, with pixel coordinates (x_1, y_1) in the first image and (x_2, y_2) in the second. The point's coordinates in the camera references are (X_1, Y_1, Z) for the first camera and (X_2, Y_2, Z) for the second. Armed with all this information we can calculate how far away the matched point is (depth Z) by derivation (1), and illustrated in Figure 4.

$$\begin{cases} \gamma\, x_1 = f\, \frac{X_1}{Z} \\ \gamma\, x_2 = f\, \frac{X_2}{Z} \end{cases} \Leftrightarrow \gamma\, Z\, (x_1 - x_2) = f\, (X_1 - X_2) \Leftrightarrow Z = \frac{f\, \beta}{\gamma\, (x_1 - x_2)} \tag{1}$$

where $\beta = X_1 - X_2$.

Some ways to match corresponding points can be: pixel by pixel probabilistic matching with a Bayesian formulation [4]; or histogram matching of the neighborhood of the pixel [11].

The SIFT features, with their invariance and robustness, enact a way to solve the problem of matching corresponding points in stereo images. We generate a sparse disparity map by extracting the SIFT features from stereo images, and look for matches between both sets. Assuming that the robot's eyes are roughly aligned in the horizontal (*i.e.,* mis-alignment of under 30 pixels) we compute the disparity between matching features from the pair of stereo images. Matches that have a high horizontal disparity are assumed to be part of an object in close proximity to the robot's face and matches with low horizontal disparity belong to the "background." Matches with high vertical disparity or negative horizontal disparity are outliers.

Using a batch of real images we get the following results summarized in Table I. In the first column we have the number of features detected in the left image, in the second column we display the number of matches found between the left image features and the right image features, while on the third column we show how many of those matches were outliers.

TABLE I
SUMMARY OF DISPARITY MATCHING RESULTS.

images	features	matches	outliers
total	27564	9545	165
percentage	100	34.6	0.5

Comparing the extracted features of different images in different resolutions, a threshold for the horizontal disparity T_h was found empirically to be the width of the image divided by 6.4. Moreover, the vertical threshold T_v to determine outliers was also empirically found to be the height of the image divided by 16.

If the matches between detected features are close enough (each match having its horizontal disparity greater than the threshold), the group is stored to the database as a new object. Only the features that are correctly matched between the two stereo images with high horizontal disparities are stored, because only these features are believed to belong to the

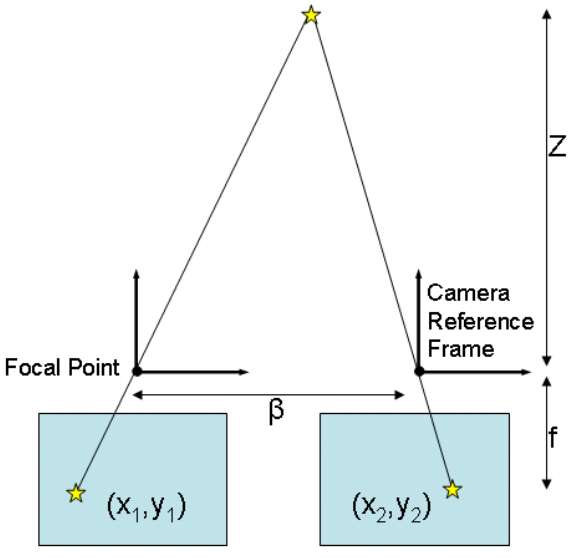


Fig. 4. Simple camera model to calculate depth, f: focal distance, β: distance separating the parallel cameras, γ: pixel-to-meter ratio in the camera sensors, (x_1, y_1): pixel coordinates of point we wish to calculate depth, Z: depth.

(a) Left image

(b) Right image

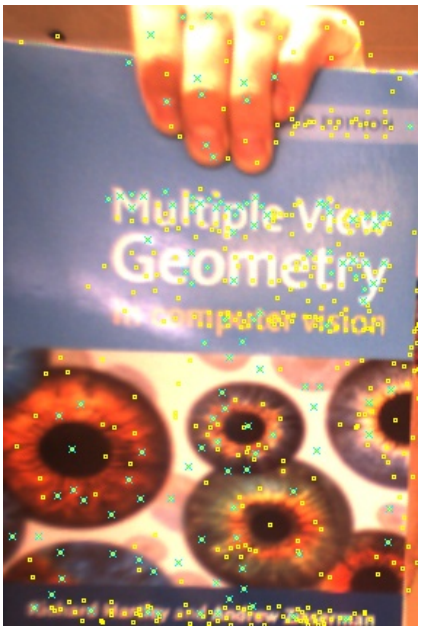
Fig. 5. Matching SIFT features in a stereo images: features in yellow; matched features in blue.

close object. For instance, the features from the background being seen by a hole in the object are discarded.

Figure 5(a) and Figure 5(b) exemplify in blue crosses the features that are correctly matched between the two stereo images as being the same, and therefore stored to the database as a new object (if not recognized as part of an already known object).

B. Recognition

To decide upon the presence of an object in the image, SIFT relies on a voting mechanism that is implemented by a Hough transform. Defining pose as the position, rotation and scale of an object, each match votes on an object-pose pair in the image. The Hough transform is computed to identify clusters of matches belonging to the same object. Finally, a verification through least-mean-squares is conducted for



(a) Object saved to database; Yellow: SIFT features, blue: SIFT features saved to the database

(b) Red: recognized object, yellow: SIFT features, purple: SIFT features matched with the database

Fig. 6. Recognition of saved object in the environment

consistent pose parameters along all matches (verifying if the matches found have correct relative positions).

After experimenting with several objects, having the robot store them to the database and then holding them farther and farther away, the algorithm was able to recognize them until roughly two meters away, when the number of extracted features declines significantly. Of the many features stored in the database and shown in blue crosses in Figure 6(a), only the few extracted ones, depicted in purple filled squares, are needed to recognize the object (encased in a red frame) in Figure 6(b).

C. Database and Mapping

New objects are stored into a database, which links object identifiers (labels) to sets of SIFT features. Each set contains a label, if the labels are the same then the different sets are considered to be of the same object. When known objects are encountered in the environment, their positions are mapped into the ego-sphere [1]. Thus, an object representation is stored in the database, while their positions, whenever recognized by the robot, are represented solely in the ego-sphere.

The egocentric saliency map used for attention selection is obtained from the composition of several specialized maps: a visual map (M_{vis}), containing saliency information extracted from visual features (e.g., motion, color), and an auditory map (M_{aud}), obtained from sound stimuli captured by the robot's microphones [1]. These maps cover the entire space surrounding the robot with a spherical coordinate system (azimuth $\vartheta \in [-180\,^\circ; 180\,^\circ]$ and elevation $\varphi \in [-90\,^\circ; 90\,^\circ]$). The saliency information stored in these maps is continuously decayed ($M_{vis}(k+1) = d_{vis}\ M_{vis}(k)$, $M_{aud}(k+1) = d_{aud}\ M_{aud}(k)$), according to a forgetting factor ($d_{vis} = d_{aud} = 0.95$). This factor coupled with a maximum frame-rate of 20 FPS, yields a half-life of less than a second, 14 frames.

In order to integrate the system described in this paper with the attention selection mechanism, the recognized objects are projected onto a third map (an object map M_{obj}). This map, combined with the other two, contributes for the egocentric saliency map: $M_{ego} = max(M_{vis}, M_{aud}, M_{obj})$. As the others, this map is also subject to a continuous decay of its information, albeit with a much longer forgetting factor ($M_{obj}(k+1) = d_{obj}\ M_{obj}(k)$, where $d_{obj} = 0.9995$). How long should the robot remember where objects of interest were? How long before such information is unreliable? Those are not trivial questions to answer. Therefore, to fulfill the practical goal of this work, of enabling the robot to switch its attention focus from recognized object to recognized object, even when such objects are not continuously in the robots field of view, this simple decaying memory with such a forgetting factor, that gives an half-life of little over one minute, is sufficient.

To verify the repeatability of the mapping coordinates (x, y) of an object in the image to coordinates in the ego-sphere (ϑ, φ) several experiments were conducted. An object was left on the table in front of the robot, while the robot's head slowly turned. From these experiments we conclude that when the object is away from the limits of the image, it is repeatedly mapped to the same location with an error under one degree elevation and two degrees azimuth. When on the verge of leaving the image, the error in mapping jumps up to two degrees elevation and four degrees azimuth.

The objects are mapped into the ego-sphere as gaussian peaks in salience. To cope with the mapping error, the gaussian parameters used were $\sigma_\vartheta = 30$ and $\sigma_\varphi = 15$.

IV. RESULTS

One of the experiments set up to show the correct recognition and mapping consisted of:

1) showing two objects, in turn, for it to learn and store to the database, a book and a magazine cover;
2) Then setting them up in front of him separated wide enough so that when the robot's attention would be on one of this objects his field of vision will not cover the second object as well;
3) Observing the resulting behavior.

The robot, upon recognizing the both previously known objects in Figure 7(a), adds interest peaks in the ego-sphere [1] (Figure 7(b)). The Attention Selection [1] module informs the robot where to look, thus fixing its attention in the first object (Figure 7(c)). After some time, a inhibition region is added to the Inhibition-of-Return map [1] (Figure 7(d)) which nudges the attention selector to continue exploring the environment of interesting points, the recognized objects. The Attention Selection module then indicates the robot to look at the now most salient region in the memory, the second blob in the ego-sphere, the second object in Figure 7(f). After some more time, the IOR-map decays and another IOR region is added, now on top of the second object location, effectively changing the robots attention back to the first object. The behavior is a continuing switching of the robot's attention through the objects in the observable environment, pausing to gaze at each object in turn.

V. CONCLUSIONS AND FUTURE WORK

This work aimed at the implementation of a spatial model of the space surrounding the iCub robot that included the salient objects which the robot encounters in its explorations. This model is used to commute the robot's attention focus automatically between objects, while not being dependent on the robot field of vision or on the objects visibility conditions.

To this end, we mapped recognized objects by introducing salience peaks on the ego-sphere [1]. The robot can now explore its environment based on low-level saliency but also on high-level information, i.e. objects.

With this long-term memory implemented, the goal of making this spatial model non-dependent on the robot's field of vision was achieved. As depicted in the results, the robot returns its focus to previously observed objects that were at the moment not in its field of view.

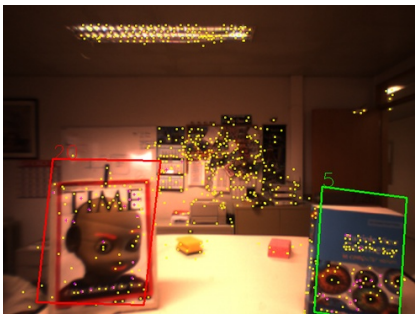
Currently, only the positions of the recognized objects are used in our work. One avenue is tracking specific objects in the environment. Another, is the search for specific objects in the surroundings to ascertain its existence or not. Additionally, the estimation of how far an object actually is, in absolute terms, is another avenue of possible work to be done.

VI. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of Alexandre Bernardino's reviews, comments and insights in the matter of depth perception.

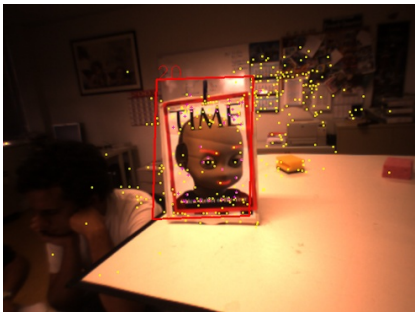
REFERENCES

[1] J. Ruesch, M. Lopes, A. Bernardino, J. Hornstein, J. Santos-Victor, and R. Pfeifer, "Multimodal saliency-based bottom-up attention, a framework for the humanoid robot icub," *IEEE International Conference on Robotics and Automation Pasadena, CA, USA, May, 2008*, May 2008.

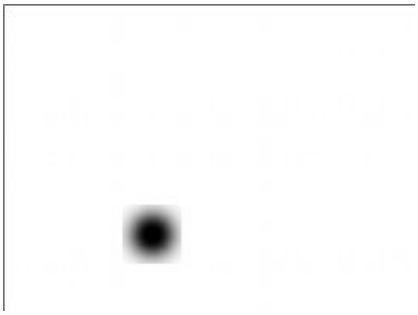


(a) Red: recognized first object, green: recognized second object

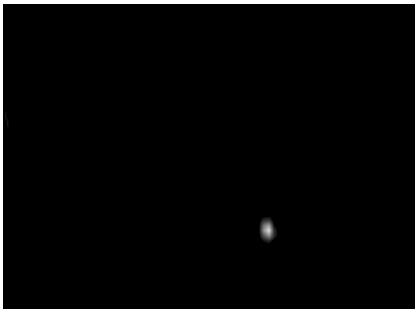
(b) Saliency map. White: recognized objects



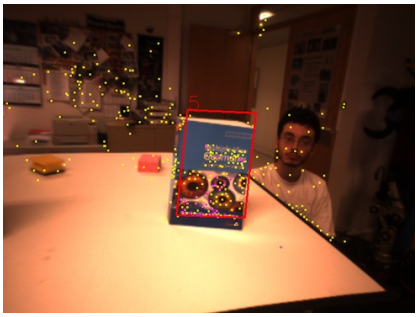
(c) Red: recognized object



(d) Inhibition-of-return map. Black: inhibition-of-return blob



(e) Updated saliency map. white: recognized objects



(f) Red: recognized object

Fig. 7. Recognizing and gazing at objects in the environment. The robot recognizes the two objects in fig.(a), adds the corresponding salience peaks to the saliency map (fig.(b) and gazes at one object (fig.(c)). After a little while he gets bored (adds an IOR region to the IOR map in fig.(d)), updates the salience map accordingly (fig.(e)) and gazes at the next object in fig.(f).

[2] A. Dankers, N. Barnes, and A. Zelinsky, "A reactive vision system: Active-dynamic saliency," *Proc. of the 5th International Conference on Computer Vision Systems, ICVS, Bielefeld, Germany*, March 2007.

[3] K. Muhlmann, D. Maier, J. Hesser, and R. Manner, "Calculating dense disparity maps from color stereo images, an efficient implementation," *IJCV*, vol. 47, no. 1-3, pp. 79–88, April 2002.

[4] A. Bernardino and J. Santos-Victor, "A binocular stereo algorithm for log-polar foveated systems," *Biological Motivated Computer Vision, Tuebingen, Germany*, November 2002.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, November 2004.

[6] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vision Research*, vol. 40, no. 10-12, pp. 1489–1506, May 2000.

[7] B. Schiele and J. L. Crowley, "Object recognition using multidimensional receptive field histograms," in *European Conference on Computer Vision*, 1996, pp. I:610–619.

[8] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Computer Vision (ECCV)*, vol. 3951, pp. 404–417, 2006.

[9] J. Bauer, N. Sunderhauf, and P. Protzel, "Comparing several implementations of two recently published feature detectors," in *International Conference on Intelligent and Autonomous Systems (IAV)*, Toulouse, France, 2007.

[10] C. Wu, "SiftGPU: A GPU implementation of david lowe's scale invariant feature transform (SIFT)."

[11] K. Prazdny, "Detection of binocular disparities," *Biological Cybernetics*, vol. 52, no. 2, pp. 93–99, June 1985.