Int. Conf. on Intelligent Robots and Systems, IROS 2011
September 25-30, 2011, San Francisco, USA

An Expected Perception Architecture using Visual 3D Reconstruction for a Humanoid Robot

N. Moutinho[†], N. Cauli [‡], E. Falotico [‡], R. Ferreira [†], J. Gaspar[†], A. Bernardino[†],
J. Santos-Victor[†], P. Dario[‡], C. Laschi[‡]

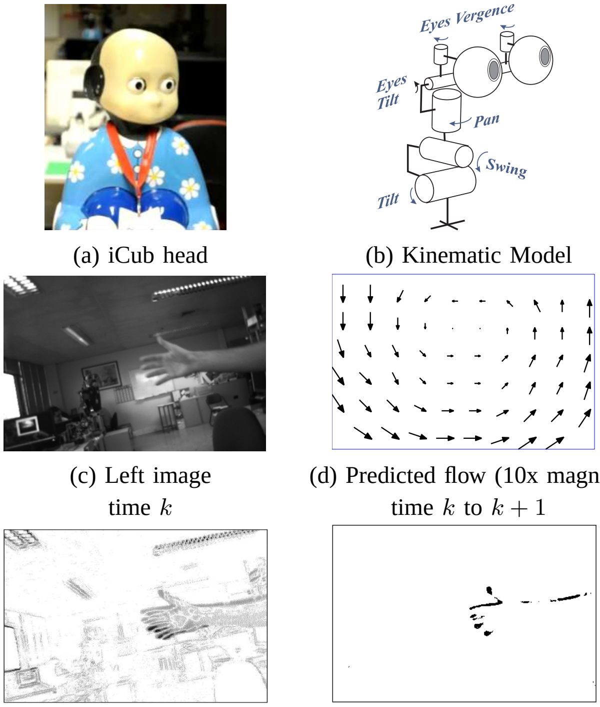
[†] Institute for Systems and Robotics / IST
Technical University of Lisbon
Lisbon, Portugal
{nmoutinho, ricardo, jag, alex}@isr.ist.utl.pt

[‡] The BioRobotics Institute
Scuola Superiore Sant'Anna
Pisa, Italy
{n.cauli, e.falotico, paolo.dario, cecilia.laschi}@sssup.it

Abstract— **The maintenance of a stable and coherent representation of the surrounding environment is an essential capability in cognitive robotic systems. Most systems employ some form of 3D perception to create internal representations of space (maps) to support tasks such as navigation, manipulation and interaction. The creation and update of such representations may represent a significant effort in the overall computation performed by the robot. In this paper we propose an architecture based on the concept of *Expected Perception* that allows lightweight map updates whenever the course of action happens according to the robot's expectations. It is only when the robot's predictions and the real world outcomes differ, that corrections must be done at its full extent. We performed experiments and show results in a real robotic platform with stereo (3D) perception where map corrections are proposed by simple image level (2D) comparisons.**

I. Introduction

In humans, perception is not just the interpretation of sensory signals, but a prediction of consequences of actions. Perception can be defined as a simulated action [1]: perceptual activity is not confined to the interpretation of sensory information but it anticipates the consequences of action, so it is an internal simulation of action. Each time it is engaged in an action, the brain constructs hypotheses about the state of a variegated group of sensory parameters throughout the movement. There are some experimental neuroscientific evidences supporting the presence of sensory anticipations in humans [1][2]. Such sensory anticipations are framed into more general schemes for perceptions and, ultimately, for sensorimotor coordination [2][3] and learning. Anticipation capabilities in humans are probably located in the cerebellum [4]. The predictions of consequences of action in the sensory space are very important for the control of movement, or more specifically for the so-called predictive control, which explain most of our sensory-motor behaviours: our brain does not base motor control on the sensory feedback, which is too slow for most everyday sensory-motor tasks, but on the predicted sensory input, i.e. an Expected Perception (EP), generated thanks to internal models built by experience [5]. Sensorimotor coordination schemes based on internal models offer a possibility for overcoming a significant difficulty related to feedback-based models. Various possibilities exist to combine information provided by the internal model and actual feedback. One of the first robotic implementation of sensory motor coordination model based on internal model is called Expected-Perception (EP) scheme discussed in [6][7]. The papers propose an anticipation mechanism to improve the perception-action loop of robots interacting with real-world environments. In this model the perception crucially involves comparison processes between incoming stimuli



(a) iCub head

(b) Kinematic Model

(c) Left image time k

(d) Predicted flow (10x magnif.) time k to $k+1$

(e) Image difference time k to $k+1$

(f) Unexpected perception time $k+1$

Fig. 1. Overview of the proposed methodology for unexpected perception detection implemented on a real iCub robot head.

This work has been partially supported by the European Commission with the RoboSoM Project, FP7-ICT-248366, and by the Italian Ministry of Education and Research, in the PRIN programme. The authors would like to thank Italian Ministry of Foreign Affairs, General Directorate for Cultural Promotion and Cooperation, for its support to the establishment of the RoboCasa joint laboratory.

and EPs, built from previous perceptions, current motor commands, and internal models of the robot and the environment. Background knowledge plays here a helpful role, as it reduces the computational burden of perception and motor coordination tasks in partially structured environments. An application of internal model to the prediction of tactile feedback in grasping is presented in [8]. In [8], the sensory prediction is part of a grasping action, controlled by a scheme based on Expected Perception (EP) [6]. Internal models in an EP scheme can be built through experience of the real world by means of neural-network-based learning mechanisms. Gross et al. [9] provided a neural control architecture implemented on a mobile miniature robot performing a local navigation task, where the robot anticipates the sensory-motor consequences of all possible motor actions in order to navigate successfully in critical environmental regions such as in front of obstacles or intersections. Barrera and Laschi [10] provide a high-level architecture of sensory-motor coordination based on anticipatory visual perception and internal models built using a fuzzy neural network. This architecture for the Anticipatory Visual Perception (AVP) is depicted by figure 2 in terms of functional modules and flow of information between them. According to the AVP architecture, the locomotion of the robot relies on the combination of two perception-action cycles: the traditional and the AVP-based one.

In this paper we exploit the concept of Expected Perception to propose an architecture to address the problem of how to update a robot's internal representation of the surrounding environment. To be able to operate and plan in a dynamic environment, an autonomous robot must update, at each time step, the relative position of the items in the world with respect to itself, hence requiring efficient perception strategies. We consider the case of humanoid robots with stereo vision having to navigate and avoid obstacles in dynamic environments and discuss the role of the robot's motor information in the construction of a coherent and stable 3D representation of the world. In particular we are interested in using this knowledge for increasing the efficiency of robot perception with respect to the utilization of visual information alone.

Robot navigation, localization and 3D scene reconstruction are among the oldest and most researched areas in robotic science. However, the exploitation of Expected Perception ideas in these areas are not common. We argue that in complex systems, such as humanoid robots, Expected Perception based techniques will play a key role in the efficiency of the algorithms. The most common approaches to robot navigation rely solely on (visual) perception. Focusing on application to humanoid robots, [11] uses plane fitting methods in the depth maps acquired by the stereo setup at each time, to support the detection of the ground floor and obstacles required for navigation. In [12] visual odometry and 3D reconstruction are used to plan the footsteps of a humanoid robot. Another class of methods use motor information, to improve the quality of the of the navigation and reconstruction systems. Often, motor odometry is used together with visual information in a time filtering scheme to provide lower variance estimates of robot and landmark locations. Following the Simultaneous Localization and Map Building approach (SLAM) [13], the state of the robot (localization) and of 3D points in the environment (map) are updated through a combination of prior knowledge (the previous state) and the current sensory readings. In [14] monocular visual SLAM and the information of the robot planned movements are used to maintain a sparse map of the environment and localize the robot. More recently [15] used stereo vision and a grid based representation of the environment. In this paper we propose a step further in the utilization of motor information for 3D reconstruction systems. Beyond filtering and data association, motor information can be used to create predictions in the sensor space that can be very efficiently checked in run-time. If expectations are confirmed, no updates to the predictions are needed, thus allowing significant computational savings in average.

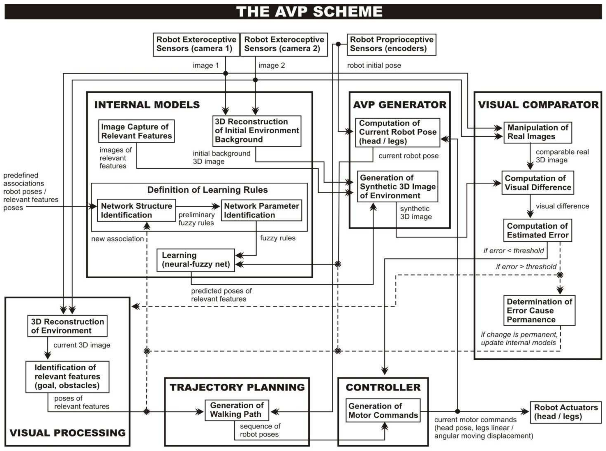

THE AVP SCHEME
Robot Exteroceptive Sensors (camera 1)
Robot Exteroceptive Sensors (camera 2)
Robot Proprioceptive Sensors (encoders)
image 1
image 2
robot initial pose
INTERNAL MODELS
Image Capture of Relevant Features
3D Reconstruction of Initial Environment Background
images of relevant features
initial background 3D image
predefined associations robot poses / relevant features poses
Definition of Learning Rules
Network Structure Identification
preliminary fuzzy rules
Network Parameter Identification
new association
fuzzy rules
Learning (neural-fuzzy net)
predicted poses of relevant features
AVP GENERATOR
Computation of Current Robot Pose (head / legs)
current robot pose
Generation of Synthetic 3D Image of Environment
synthetic 3D image
VISUAL COMPARATOR
Manipulation of Real Images
comparable real 3D image
Computation of Visual Difference
visual difference
Computation of Estimated Error
if error < threshold
if error > threshold
Determination of Error Cause Permanence
if change is permanent, update internal models
3D Reconstruction of Environment
current 3D image
Identification of relevant features (goal, obstacles)
poses of relevant features
VISUAL PROCESSING
TRAJECTORY PLANNING
Generation of Walking Path
sequence of robot poses
CONTROLLER
Generation of Motor Commands
current motor commands (head pose, legs linear / angular moving displacement)
Robot Actuators (head / legs)

Fig. 2. The AVP scheme[10].

The structure of the paper is as follows. Section II describes the visual expected perception scheme providing a detailed overview of the implemented architecture. Section III introduces the internal models for mapping the sensory odometry information to head pose and changes in the visual field. Section IV describes the Expected Perception Generator and Visual Comparator modules as well as how this information can be used to update a 3D scene representation in a computationally efficient manner. Finally, results obtained with a real iCub robot head (see figure 1) are shown in section V and conclusions are drawn in section VI.

II. Visual EP Scheme

The EP scheme architecture modelled and implemented in this work is shown in Figure 3. The scheme was thought to face a recent task for the EP applications. Differently from the most of previous works (like [6][7] for grasping), the Expected Perception is used to support a 3D vison system of a humanoid robot. More precisely, the prediction of camera images allows to simply update the 3D map of

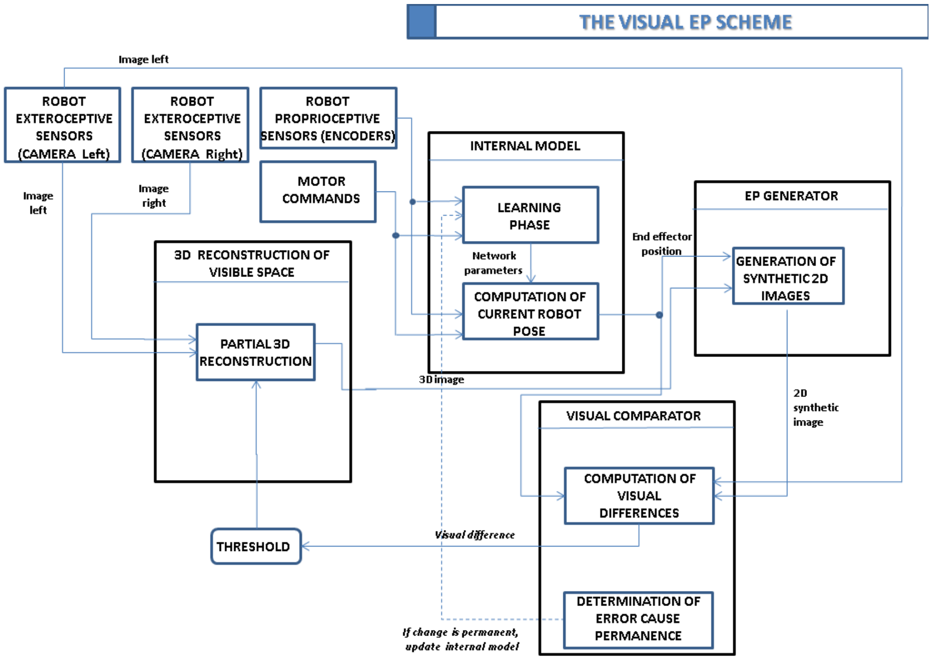

THE VISUAL EP SCHEME
Image left
ROBOT EXTEROCEPTIVE SENSORS (CAMERA Left)
ROBOT EXTEROCEPTIVE SENSORS (CAMERA Right)
ROBOT PROPRIOCEPTIVE SENSORS (ENCODERS)
INTERNAL MODEL
MOTOR COMMANDS
LEARNING PHASE
EP GENERATOR
Image left
Image right
End effector position
3D RECONSTRUCTION OF VISIBLE SPACE
Network parameters
GENERATION OF SYNTHETIC 2D IMAGES
COMPUTATION OF CURRENT ROBOT POSE
PARTIAL 3D RECONSTRUCTION
3D image
2D synthetic image
VISUAL COMPARATOR
COMPUTATION OF VISUAL DIFFERENCES
Visual difference
THRESHOLD
DETERMINATION OF ERROR CAUSE PERMANENCE
If change is permanent, update internal model

Fig. 3. Visual EP Scheme.

the environment, without calculate it again in every vision step.

Unlike the classical control loops, that base their response only on the past/current sensory data, the EP architectures use the sensory anticipation to control the actuators. In [10] the authors start to investigate this task for the control of a humanoid robot locomotion. This scheme is divided in two main loops:

- classical perception-action loop
- prediction loop

The AVP architecture represents the starting point for the design of the scheme implemented in this paper. Although the two schemes apparently solve the same problem, they are actually different. First of all, the scheme proposed by A. Barrera and C. Laschi describes an entire control loop based on visual anticipation, whereas the architecture of this paper focus mainly on the prediction loop. Both of the schemes generate the Expected Prediction from a visual 3D reconstruction of the environment. The AVP architecture produces a tridimensional Expected Perception, whereas the proposed scheme generates a 2D prediction, easier and faster to process and analyze.

In the scheme depicted in Figure 3, a loop step consists in the 3D reconstruction of the environment captured by the stereo cameras. During each step the cameras response will be anticipated, generating a predicted left camera image. An internal model of the robot kinematics permits to obtain the prediction. The 3D reconstruction is, then, simply updated using the comparison between the predicted image and the actual one. The zone of the 3D reconstruction to be updated for the next step are calculated applying a threshold on the difference image.

The EP scheme operation is modelled by a series of blocks:

- The *Internal Model* block calculates the new end effector position (rotation matrix and translation vector of the left camera) from the encoder angles and motor commands. The block is realized using a multilayer feedforward neural network, trained with the backpropagation algorithm. Additionally, internal model is updated only if the change registered within the environment is permanent. In this case, the structure and learning parameters of the neural network are adjusted to consider a new association between robot body movements and changes in the visual field.
- The *EP Generator* block calculates the predicted left camera image from the 3D reconstruction (generated/updated by the *3D Reconstruction of Visible Space*) and the left camera position (given by the *Internal Model*).
- The *Visual Comparator* block calculates the differences between the predicted and actual images. Furthermore this block evaluate if the errors are permanent and a new training phase of the internal model network is necessary.
- The *3D Reconstruction of Visible Space* block updates the most unpredictable zone of the 3D reconstruction. Only at the beginning of the control loop the architecture has to build the entire 3D reconstruction of the visible environment.

.

III. Internal Model

The role of the internal model, inside the Visual EP Scheme, is to learn the correlation between robot body movements and changes in the visual field. We tested the

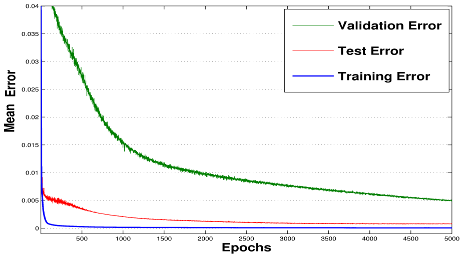


Fig. 4. The graph shows the learning phase of the neural network. The Mean Error of 5000 epochs is plotted for training set, validation set and test set of data.

internal model moving the iCub's head (neck pan, neck swing and neck tilt). To compute visual shifts due to the head movements the internal model learns the position and orientation of the end effector (the left camera) related to the head position. So, being the camera position known, it is possible to generate a predicted synthetic image for the comparison with the real one. The internal model has been implemented using a feedforward artificial neural network, a Multilayer Perceptron. This network has been developed with 3 neurons in the input layer (related to neck swing, tilt and pan angles), 10 neurons in the hidden layer and 12 neurons in output layer that correspond with the values of the transformation matrix $\mathbf{T}$ calculated using odometry to estimate the position and the orientation of the left camera, given a certain rotation R and translation t.

$$\mathbf{T} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

The network uses the non linear activation sigmoid function with backpropagation learning rule. It has been taken into account 300 simulation results, the training set for the neural network is the 70% of these values, the 15% is used for the validation set and another 15% is used for the test set. These values have been shuffled to increase the variability of the training set. The results of the learning are shown in figure 4. The training error is almost nullified after few epochs. This is due to the simple kinematics chain estimated for this task. We stopped the training when the error for the validation set is less then $6 \cdot 10^{-6}$.

IV. Estimating 3D

The general setup assumed in this paper consists of a stereo camera mounted at the end of a kinematic chain, described as a set of I joint angles $\theta = (\theta_1, \theta_2, \cdots, \theta_I) \in \mathbb{R}^I$. Assume that the odometry is calibrated in the sense that a map $T : \mathbb{R}^I \to \mathsf{SE}(3)$ is available, where $\mathsf{SE}(3)$ denotes the set of rigid transformations, which converts the odometry values into camera position and orientation as described in the previous section. At each time instance k a pair of images L^k and R^k, henceforth refered to as left and right images respectively, are obtained from cameras mounted on the last segment of the kinematic chain together with a set θ^k of odometer readings which provide an estimate of the camera position as $\mathbf{T}^k = T(\theta^k)$.

The objective of this section is to incrementally obtain a disparity map D^k at each time instant, using the expected perception concepts to avoid recomputing the whole map every time. This can be broken into two sub-tasks: 1) discovering which areas require an update; 2) recomputing the disparity map on the invalidated regions and fusing this information with the previously available information. Each of these will be described in greater detail next.

A. Detecting Invalid Regions

In the Expected Perception framework, intensive sensor processing should be delayed until a simpler detection algorithm flags this data as no longer valid. Since obtaining 3D measurements from stereo image data is considered a computational burden, it is a prime candidate for this approach. Here the question of how to detect when the previous data is no longer valid is addressed.

Since the detection process should be as computationally inexpensive as possible this detection is done by comparing brightness information at consecutive time instances L^k and L^{k+1}. Note that since the cameras are moving, a direct brightness comparison is impossible (see figure 1.(e)) and a prediction step is needed to propagate the previous image to time instant $k+1$ before the comparison. This task is simplified by the availability of the camera position estimates $\mathbf{T}^k$ and an estimate of the 3D world represented as a disparity image D_-^{k+1}, but since these readings are not perfect (sensor discretization, kinematic errors, etc.), a means of attenuating predictable false positives is needed. Refer to figure 5 for an overview.

When new odometry information is available at time $k+1$ the system is able to compute a prediction for the left image

$$L_-^{k+1} = p(L^k; \mathbf{T}^{k+1}, D_-^{k+1})$$

by mapping each pixel on the image at time k to a certain position on the image space at time instant $k+1$ by reprojecting the previous brightness and 3D information on the new camera pose. In perfect conditions, where the real transformation is well known and there is no noise, the predicted image matches perfectly with the acquired image (L^{k+1}), except for non-predictable image points either not visible in the previous image or belonging to a moving object. In these conditions a brightness difference

$$E^{k+1} = \left| L_-^{k+1} - L^{k+1} \right|$$

between these two images should result in a well defined segmentation of the areas in need of being updated.

In real conditions however, the odometry and disparity information are not perfect and small misalignments are to be expected. Unfortunately these result in high intensity errors, particularly noticeable near the intensity edges and corners of the observed image, and require some mechanism to eliminate their influence. Fortunately these edges and corners occur at predictable positions allowing for a mechanism which attenuates their influence to be implementable. To this end, these error images are used to generate an accumulated error which is propagated and accumulated along time using a certain forgetting factor λ as

$$C^{k+1} = (E^{k+1} + \lambda C_{-}^{k+1})/(\lambda + 1)$$

where

$$C_{-}^{k+1} = p(C^k; \mathbf{T}^{k+1}, D_{-}^{k+1})$$

is the information propagated along time. In the described setup a forgetting factor of $\lambda = 10$ provides good results. The effect of this low pass filtering is that it detects and propagates image areas known to be noisily unpredictable and this information can be used to attenuate each pixel in the error image before deciding whether a region needs to be updated or not. Thus, an attenuated error is generated as

$$A^{k+1} = E^{k+1} \exp(-\alpha \left(C_{-}^{k+1}\right)^2),$$

where a value of $\alpha = 0.1$ provides good results. This attenuation results in a good compromise between detection of unexpected changes and ignoring repeatably unexpected image areas. The segmentation of the unexpected event is then done in the image space, detecting what is not predictable by the system, by applying a threshold (in this case the hand moving independently).

B. The Disparity Map

The previous description makes use of a precomputed disparity map (inverse depth information) to be able to predict both the image at the next time instant L_{-}^{k+1} and the cumulative error C_{-}^{k+1}. This map needs to be updated and is the proposed output of the algorithm here presented. Using a stereo pair of images at the initial time instant, L^0 and R^0, we obtain an initial disparity map D^0 that is used to bootstrap the algorithm. This information is used as long as it is valid and generates predicted images consistent with the ones acquired. At each time instant the disparity map is updated from the odometry information through a function p_D as

$$D_{-}^{k+1} = p_D(D^k; \mathbf{T}^{k+1})$$

which performs a rigid transformation of the 3D points represented as the disparity map D^k. This prediction is then used to apply the expected perception concept as described previously to detect which areas require full disparity re-computation. Only these areas are updated reducing the computation time.

For the disparity map information used in this paper we used the Semi-Global Block Matching algorithm available in OpenCV, based on [16].

V. Results

In order to test the proposed perception methodology in a real setup, an experiment was conducted with the iCub robotic head (figure 1(a)). The robotic head allows the synchronous acquisition of stereo images and odometry information, namely the angles associated to each joint (see figure 1(b)). The intrinsic and extrinsic parameters of the cameras have been previously calibrated using a Matlab's calibration toolbox [17]. During the experiment, the robot's head moves randomly and an independently moving object (hand) appears in the fields of view of the cameras (see figure 6(a), (b), (c) and (d)). The movement of the hand is also random and thus there is no information describing it.

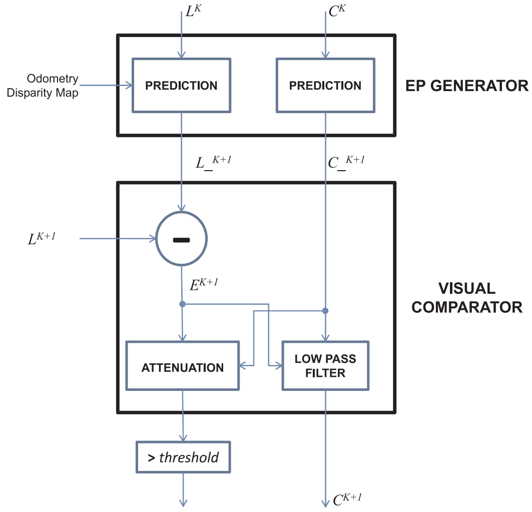
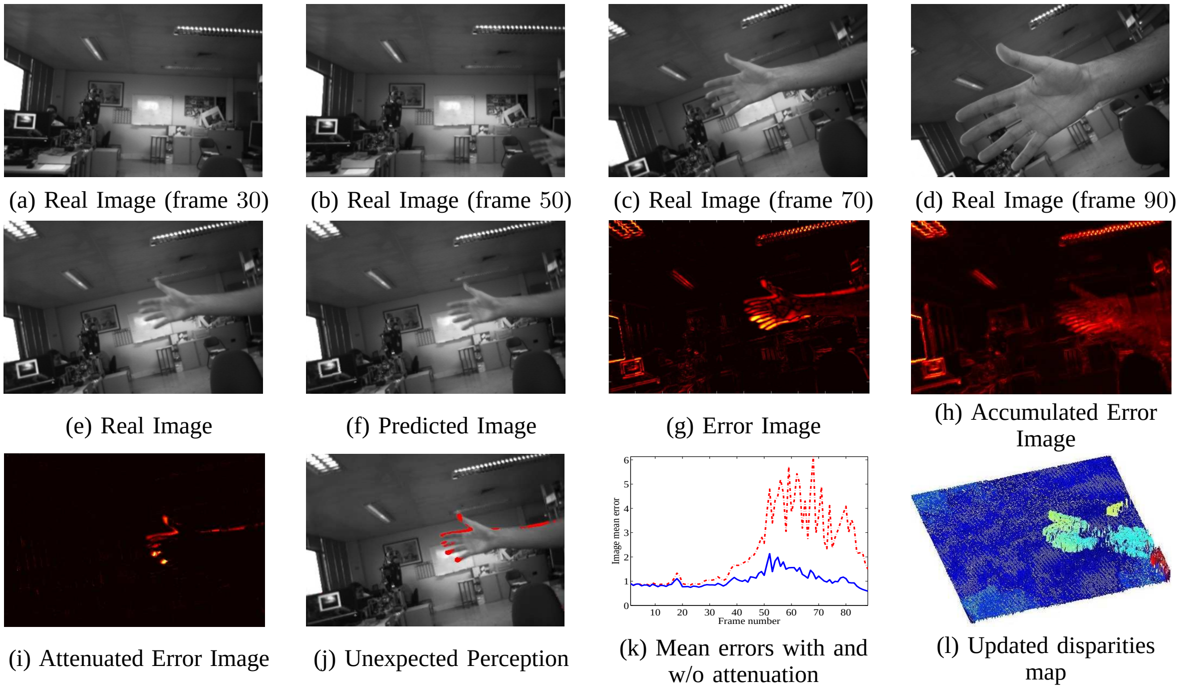


Fig. 5. Implementation of the EP Generator and Visual Comparator modules.

Given an input image acquired at the time k, the angles of the joints observed at time $k+1$ and a dense disparities map representing the imaged world at time k, one can predict the input image at $k+1$ (figure 6(f)). This image is expected to have compensated most of the self-motion, however one finds that due to some illumination variance and to some imprecision on the encoders, there are some differences between the predicted and the real images, mostly near the image edges (flickering lights are examples of image differences not motivated by motion).

Figure 6(g) shows the brightness difference between the real and predicted images. This error image shows not only the hand but all the small misalignments near the edges. However, the accumulated error image (figure 6(h)) takes these misalignments into account since they are predictable, constantly appearing in the same places, allowing their attenuation along time (figure 6(i)). Figure 6(k) compares the mean errors before and after the attenuation process (figure 6(g) and (i)), where it is noticeable the decrease of the mean error value after this process, meaning that mostly of the image error is now due to the presence of unpredictable events. Now the hand appears clearly salient. This clear saliency allows to find independent motion by simple thresholding, as shown in figure 6(j).

Figure 6(l) shows the updated disparity map. This map is constructed in an iterative manner, being guided by unexpected perception detected on the image (differences). The larger disparities correspond to the hand, which in fact is closer to the iCub's cameras than the rest of the background. The qualitatively correct information on the disparities map shows that it can be constructed iteratively using just some specific information at each time sample.



(a) Real Image (frame 30) (b) Real Image (frame 50) (c) Real Image (frame 70) (d) Real Image (frame 90)

(e) Real Image (f) Predicted Image (g) Error Image (h) Accumulated Error Image

(i) Attenuated Error Image (j) Unexpected Perception (k) Mean errors with and w/o attenuation (l) Updated disparities map

Fig. 6. Updating the expected disparities map given stereo images and the joints motion. Input stereo-sequence of images, frames $30, 50, 70, 90$ (a,b,c,d). Real Image at time $k+1$ (e). Predicted Image at time $k+1$ (f). Error image (brightness difference between (e) and (f)) at time $k+1$ (g). Accumulated error images up to time k (h). Attenuated error image at time $k+1$ (i). Unexpected perception at time $k+1$ (j). Means of (g) and (i) along time, resp. in dash-dotted-red and blue-continuous lines (k). Updated disparities map at time $k+1$ (l).

VI. Conclusion

In this paper we have applied the Expected Perception concept to the problem of maintaining a valid 3D reconstruction of visible space. The current 3D knowledge of the system is used to create predictions of the future perceptions at the sensor level. This allows a very quick assessment of the validity of the prediction through direct sensor based correlations. We have shown in a real robotic head, that predictions of image data can support direct comparison with freshly acquired images to detect regions where the world has changed. Even with a significant amount of ego-motion, camera noise and encoder uncertainty, Expected Perceptions mechanisms can be implemented at a low-level, very close to the direct camera readings. This is a key finding for the development of truly efficient and automated cognitive systems.

References

[1] A. Berthoz, in *The sense of movement*. Harvard University Press, 2002.

[2] R. Johansson, “Sensory input and control of grip,” in *Sensory Guidance of Movements*, 1998, pp. 45–59.

[3] D. Miall, D. Weir, D. Wolpert, and J. Stein, “Is the cerebellum a smith predictor?” in *Journal of Motor Behavior*, vol. 25(2), 1993, pp. 203–216.

[4] D. Wolpert, D. Miall, and M. Kawato, “Internal Models in the celebellum,” in *Trends in Cognitive Sciences*, vol. 2(9), 1998, pp. 338–347.

[5] D. Miall and D. Wolpert, “Forward models for physiological motor control,” in *Neural Networks*, vol. 9(8), 1996, pp. 1265–1279.

[6] E. Datteri, G. Teti, C. Laschi, G. Tamburrini, P. Dario, and E. Guglielmelli, “Expected Perception: an anticipation-based perception-action scheme in robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2003, pp. 934–939.

[7] ——, “Expected Perception in robots: a biologically driven perception-action scheme,” in *Proceedings of ICAR 2003, 11th International Conference on Advanced Robotics*, vol. 3, 2003, pp. 1405–1410.

[8] C. Laschi, G. Asuni, E. Guglielmelli, G. Teti, R. Johansson, M. Carrozza, and P. Dario, “A bio-inspired neural sensoty-motor coordination scheme for robot reaching and preshaping,” in *Autonomous Robot*, vol. 25, 2008, pp. 85–101.

[9] H. Gross, V. Stephan, and T. Seiler, “Neural architecture for sensorimotor anticipation,” in *Cybernetics and Systems Research*, vol. 2, 1998, pp. 593–598.

[10] A. Barrera and C. Laschi, “Anticipatory visual perception as a bio-inspired mechanism underlying robot locomotion,” in *EMBC*, 2010.

[11] K. Sabe, M. Fukuchi, J.-S. Gutmann, T. Ohashi, K. Kawamoto, and T. Yoshigahara, “Obstacle avoidance and path planning for humanoid robots using stereo vision,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004*, 2004, pp. 592–597 Vol.1.

[12] R. Ozawa, Y. Takaoka, Y. Kida, K. Nishiwaki, J. Chestnutt, J. Kuffner, S. Kagami, H. Mizoguch, and H. Inoue, “Using Visual Odometry to Create 3D Maps for Online Footstep Planning,” in *2005 IEEE International Conference on Systems, Man and Cybernetics*, 2005, pp. 2643–2648.

[13] J. Leonard and H. Durrant-Whyte., “Simultaneous map building and localisation for an autonomous mobile robot.” in *Proc. IEEE Int. Workshop on Intelligent Robots and Systems (IROS)*, Osaka, 1991, pp. 1442–1447.

[14] O. Stasse, A. Davison, R. Sellaouti, and K. Yokoi, “Real-time 3d slam for humanoid robot considering pattern generator information,” in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006, pp. 348–355.

[15] N. Kwak, O. Stasse, T. Foissotte, and K. Yokoi, “3D grid and particle based SLAM for a humanoid robot,” in *2009 9th IEEE-RAS International Conference on Humanoid Robots*, Dec. 2009, pp. 62–67.

[16] H. Hirschmller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 328–341, 2008.

[17] J.-Y. Bouguet, “Camera calibration toolbox for matlab,” http://www.vision.caltech.edu/bouguetj.