

Reexamining Lucas-Kanade Method for Real-Time Independent Motion Detection: Application to the iCub Humanoid Robot

Carlo Ciliberto, Ugo Pattacini, Lorenzo Natale, Francesco Nori and Giorgio Metta

Abstract—Visual motion is a simple yet powerful cue widely used by biological systems to improve their perception and adaptation to the environment. Examples of tasks that greatly benefit from the ability to detect movement are object segmentation, 3D scene reconstruction and control of attention. In computer vision several algorithms for computing visual motion and optic flow exists. However their application in robotics is not straightforward as in these platforms visual motion is often dominated by (self) motion produced by the movement of the robot (egomotion) making it difficult to disambiguate between motion induced by the scene dynamics or by the own actions of the robot. Independent motion detection is an active field in computer vision and robotics, however approaches in this area typically require that some models of both the environment and the robot visual system are available and are hardly suitable for real-time control. In this paper we describe the *motionCUT*, a derivation of the Lucas-Kanade optical flow algorithm that allows detecting moving objects, irrespectively of the egomotion produced by the robot. Our method is purely visual and does not require information other than the images coming from the cameras. As such it can be easily adapted to any robotic platform. The system was tested on a stereo tracking task on the iCub humanoid robot, demonstrating that the algorithm performs well and can easily execute in real-time.

I. INTRODUCTION

Motion cues represent a dominant class of stimuli for autonomous agents operating in dynamic environments. Tasks such as tracking of moving targets or avoidance of active obstacles are just examples of situations in which it is fundamental to be able to quickly detect and react to changes in the surroundings. From psychophysics and in particular from studies on development ([1],[2]) it has further emerged that motion is employed by infants as a strong drive to understand the existing relationships between elements in the world.

Inspired by these observations, in robotics motion has been exploited for tasks ranging from control of attention ([3]) to autonomous object detection and recognition ([4]). From a more traditional perspective, computer vision shows several examples that demonstrate how motion cues can successfully be employed in a large variety of contexts, from active segmentation ([5]) to 3D scene reconstruction ([6]).

An interesting application comes from the field of developmental robotics in which motion information is exploited

This work is funded by the European Commission as part of the project ICT-FP7-215843 Poeticon.

C. Ciliberto, U. Pattacini, L. Natale, F. Nori and G. Metta are with the Robotics Brain and Cognitive Sciences Department, Italian Institute of Technology, Genova, Italy. carlo.ciliberto, ugo.pattacini, lorenzo.natale, francesco.nori, giorgio.metta@iit.it

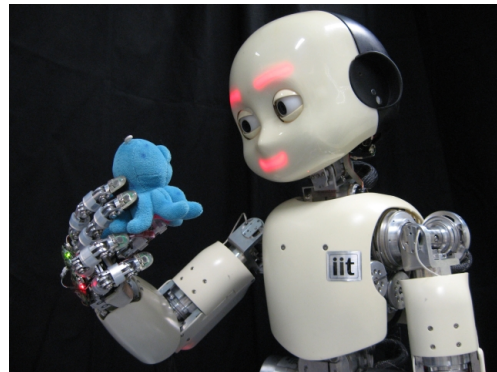


Fig. 1. The hardware platform: the iCub robot.

to discriminate the robot's own body from the environment (the problem of self discovery) ([7], [8], [9]).

Unfortunately it is difficult to use visual motion in robotics. In fact, motion induced by the robot own actions frequently introduces substantial disturbances in the motion signal produced by the other elements of the scene, causing the failure of typical algorithms that assume a static background.

In this work we present a novel approach to independent motion detection called *motionCUT* and demonstrate it on the humanoid robot iCub ([10]). The iCub is a robot shaped as a human child (Figure 1) and equipped with 53 degrees of freedom as well as force and tactile sensors, gyroscopes and stereo cameras mounted in the eyes.

We perform an extensive empirical evaluation of the *motionCUT* in particular to demonstrate that the algorithm can easily run in real-time. Furthermore we show that system performs independent motion detection coherently with respect to both time and viewpoint variability. Overall, the method allows accurate identification of novel objects moving in the scene and can be effectively used for real-time 3D tracking, even in presence of large disturbances induced by the own movement of the robot.

Notably we made the *motionCUT* software publicly available¹. This allows the whole research community to benefit from the work proposed in this paper, not only for performing research, but also for testing and evaluating the algorithm directly on their platforms.

¹The code is released with a GPL license and it can be downloaded from the iCub repository at the URL <https://robotcub.svn.sourceforge.net/svnroot/robotcub/trunk/iCub>.

II. PREVIOUS WORK

The field of independent motion can be roughly separated in two main streams: 2D and 3D techniques.

2D techniques generally assume that a global parametric transformation exists which can align the static background over multiple frames. This is generally true when the background is approximately planar, the observer is purely rotating or it is translating relatively slow with respect to the distance between the camera and the background.

3D techniques on the other hand, exploit the tridimensional information extracted from the image stream to have more stable results especially when visual data is characterized by strong parallax. Irani and Anandan [11] unified 2D and 3D techniques in a hierarchical fashion in order to face situations in which both described conditions arise.

In the work of Nelson [12], two 3D approaches are described. The first one is the classical use of epipolar geometry to detect velocities in the optical flow which do not behave coherently with respect to the evaluated epipole. The other one finds outliers in the optical flow under the assumption that observer egomotion induces relatively slow motion with respect to independent motion.

A more statistical way of tackling the independent motion problem was proposed by Jung and Sukhatme [13] which combined a classical approach to outlier detection in the optical flow computation together with particle filtering techniques [14] and an EM algorithm [15]. The results of this study appear to perform in real time on the three different robotic platforms over which tests were conducted.

Other techniques make use of the stereo information to improve detection performances. By combining the disparity map and optical flow information Talukder and Matthies [16] were able to obtain a better estimation of the observer egomotion and thus to isolate incoherencies due to independent motion. Argyros and Orphanoudakis [17] formulated the independent motion detection as a robust parameter estimation problem and used a linear model to approximate egomotion.

Differently from the approaches mentioned above, Rougeaux and Kuniyoshi [18] proposed a method which uses the additional information from the known joint velocities of a humanoid robotic head. By combining encoder information with the disparity map in a probabilistic framework they were able to segment an independent moving object which translated in front of the cameras. However this approach lacked the generality to consider scenarios in which motion was not occurring inside the camera's fovea and furthermore assumed that the independent motion projected only as a 2D translation on the image plane.

It can be noticed that all the described methods share the same underlying idea: first the flow induced by the observer on the image stream is evaluated, then independent motion is detected whenever elements on the image do not behave coherently with the estimated egomotion. Following this line of thought we developed a novel framework which exploits a classical optical flow computation technique in order to find

elements which do not behave correctly with respect to the actual observed egomotion.

III. THE *motionCUT* METHOD

Our method originates from an analysis of the optical flow problem, and in particular from the approach explored by Lucas and Kanade [19]. In the following we present a brief summary of this classic technique followed by the idea which led to the definition of our algorithm.

A. Lucas-Kanade Optical Flow

The study of optical flow deals with the problem of evaluating motion across streams of images. Such motion is usually induced on the image plane by the actual dynamics interesting the elements in the observed scene. A typical example illustrating this situation is a robot exploring its surroundings and consequently generating apparent motion in the image stream acquired from the embarked camera(s). In these contexts it can result necessary to understand exactly how the observed motion takes place. This can be partially inferred from the analysis of the optical flow, which is defined as the field of instantaneous velocities on the image plane reference frame.

Under the reasonable assumption that image sampling is performed at sufficiently high frequency, it can be accepted that the *appearance continuity* property holds. In other words, it is legitimate to expect that within small time intervals, the appearance of elements in the scene does not vary dramatically and it is thus possible to track points across subsequent frames based exclusively on their visual appearance. This property can be expressed more formally through the equation

$$I(p(t), t) = I(p(t + \delta t), t + \delta t) \quad \text{with } \delta t \ll 1 \quad (1)$$

where $I(\cdot, t)$ is the image function at time t (usually representing pixel brightness) and $p(t) = (x(t), y(t))$ is the projection on the image plane at time t of a point in the scene.

Deriving the image function $I(p(t), t)$ with respect to time, we obtain

$$\nabla_p I \cdot \dot{p} + \frac{\partial I}{\partial t} = 0 \quad (2)$$

where ∇_p is the gradient operator with respect to the x and y image plane directions and $\dot{p} = \partial p / \partial t$ represents the planar velocity of point p . As can be noticed, when solving with respect to \dot{p} , equation (2) has an infinite number of solutions, implying that the sole punctual information is insufficient to determine the exact velocity of any point on the image plane (resulting in the so-called *aperture problem*). However, it can be assumed that locally points behave similarly and thus that in sufficiently small neighborhoods, instant velocities are almost identical. Velocity \dot{p} of a point p can hence be approximated by setting a window W of size $w \times h$ around it and then solving the least-squares minimization problem

$$\hat{\dot{p}} = \arg \min_{v \in \mathbb{R}^2} \|\mathcal{I}_p \cdot v - \mathcal{I}_t\|_W^2 \quad (3)$$

where $\mathcal{I}_p = \mathcal{I}_p^W$ and $\mathcal{I}_t = \mathcal{I}_t^W$ are matrices whose rows are respectively the gradient $\nabla_p I$ and the derivative $\partial I / \partial t$ computed on each point q in W while $\|\cdot\|_W$ denotes the Euclidean norm in $\mathbb{R}^{w \times h}$.

When the left pseudoinverse \mathcal{I}_p^\dagger of \mathcal{I}_p exists, equation (3) is solved by taking $\hat{p} = \mathcal{I}_p^\dagger \mathcal{I}_t$. Existence of \mathcal{I}_p^\dagger depends on the spatial appearance of the neighbors of p . In particular it can be easily shown that in order for \mathcal{I}_p^\dagger to exist, both partial derivatives of the image along the x and y axes need to be different from zero in some points of the neighborhood of p . By analyzing this condition it has been observed that the so-called *corners* [20] – that is points whose neighborhood has both strong partial derivatives – are the most robust elements over which apply the Lucas Kanade algorithm.

B. Failure Analysis

From the discussion above, corners appear the most suitable points over which compute optical flow with the Lucas-Kanade algorithm. However in practice, even corner tracking is not always successful. As a general rule, in order to verify that the instant velocity v of a point p has been correctly estimated, the patch W around that point in the image I_t is compared to the patch of the same size at $p + v$ in the new image I_{t+1} (where the original point is supposed to have moved). Given a suitable threshold Θ_M , the discrepancy measure

$$M(p) = \sum_{q \in W} (I_t(p+q) - I_{t+1}(p+v+q))^2 \quad (4)$$

is then used to evaluate whether tracking was correctly performed ($M(p) < \Theta_M$) or not ($M(p) \geq \Theta_M$).

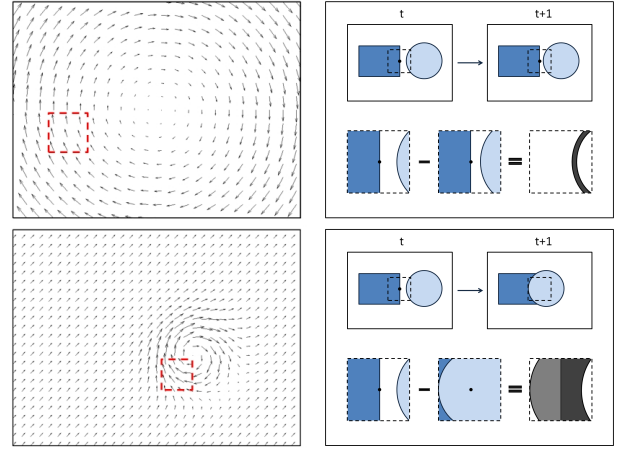
It is thus interesting to analyze experimentally when the Lucas-Kanade algorithm tends to fail and why. Conclusions from this investigation will lead directly to the method we are proposing to perform independent motion detection.

The main empirical circumstances in which errors in the evaluation process of the optical flow arise are three:

- **Speed.** The instantaneous velocity of the point is too large with respect to the window where motion is being considered. Hence, \mathcal{I}_t loses its meaning of temporal derivative.
- **Rotations.** The motion around the point has a strong rotational component and thus, even locally, the assumption regarding the similarity of velocities falls.
- **Occlusions.** The point is occluded by another entity and obviously it is impossible to track it in the subsequent frame.

Tracking failures caused by high punctual speed depend exclusively on the scale of the neighborhood where optical flow is computed. This issue is usually solved by the so-called *pyramidal approach* [21] which applies the Lucas-Kanade method at multiple image scales. This allows to evaluate iteratively first larger velocities and then smaller ones.

Empirically we determined that when rotations cause failures in the tracking process, this is usually a consequence of the motion independent from the observer. Figure 2(a)



(a) Rotations.

(b) Occlusions.

Fig. 2. Illustration of typical failure circumstances for the optical flow computation: (a) Egomotion rotations (top) usually induce homogeneous and uniform flows on the image stream while independent rotations (bottom) frequently cause the Lucas-Kanade assumption of local constant velocity to not usually hold (the red square windows encompass a detail of such behavior). (b) Failures due to occlusion depend on the relationship between image sampling frequency and the speed at which such event happens: if the occluding object is moving slowly (top), differences in the neighborhood of the occlusion are smaller than those exhibited in the case of higher velocities (bottom).

depicts a comparison between the typical rotational effects on the image generated respectively by egomotion (upper image) and independent motion (lower image). As it can be noticed, egomotion has a global effect on the image stream, producing optical flows that result smooth and locally almost constant. Circular flow due to independent motion, on the other hand, affects only specific areas of the image plane and introduces local distortions.

The third situation in which Lucas-Kanade fails, is caused when an object covers another one (or the background). In this context the main role in determining whether optical flow has been successfully computed is played by the speed at which such occlusion takes place. As shown in Figure 2(b), if the occluding object moves slowly with respect to the image sampling frequency, the window around the target point remains almost unaltered even when the point disappears behind the object. However, as the event happens faster, larger portions of the neighborhood get covered, eventually causing the discrepancy measure between original and tracked window to increase over the threshold Θ_M .

In principle, this situation can be the result of both an independent object movement or the observer egomotion, when the latter has a strong translation components. However, aside from pathological cases, elements in the scene lay far enough from the observer so that the translation components due to camera motion become negligible.

C. The Cover-Uncover Trick

We propose a method to detect independent motion which derives directly from our observations on optical flow. As a matter of fact, the fundamental idea underlying our approach

is to take into consideration the points in the image where the Lucas-Kanade algorithm fails, as they are likely to identify image patches where independent motion is actually occurring, rather than those over which it succeeds.

The main concern is the selection of those points over which to compute the optical flow. In our context we look for points where tracking is likely to fail as soon as one of the conditions discussed in Section III-B is met, i.e. flow inconsistencies due to rotations or occlusions. With these premises we named our method *motionCUT*, being *CUT* the acronym for *Cover-Uncover Trick*: independent moving agents in the scene can be recognized by inspecting the effect of their motion at the frontier with stationary elements of the environment (see Figure 2(b)) where occlusions of parts of the background (cover) and their corresponding re-emergence (uncover) mostly appear.

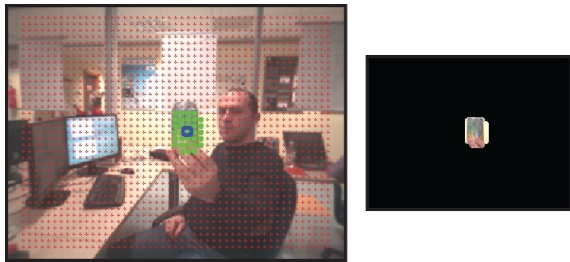


Fig. 3. *Left*. The uniform static grid of nodes depicted in red. The operator waves a can generating independent motion cue that in turn stimulates the superimposed nodes resulting in green; the centroids of the detected blob is also automatically extracted and highlighted in blue. *Right*. The output of *motionCUT* can drive a coarse segmentation of the moving object.

To this end, we consider a dense grid of points uniformly distributed over the image plane and compute the optical flow with these grid nodes as pivots (Figure 3). In this framework point selection is done regardless of their appearance; for this reason the majority of them will not be corners but rather weaker points in the sense of Lucas-Kanade tracking. This process exposes the system to the risk of having points where the pseudoinverse of \mathcal{L}_p does not exist and it is therefore impossible to compute the optical flow. However in practice, the aperture problem occurs when the window W around the point is not large enough. This problem is mitigated by the adoption of the pyramidal approach [21] since the most crucial components of motion are correctly captured by windows at higher scales.

Being less robust than typical corners, the points of the regular grid are quite susceptible to motion inconsistencies, responding positively – in the sense that the optical flow computation fails over them – when they lie near an image patch where independent motion occurs. This lack of robustness however makes these points also more likely to produce false positives. This issue can be avoided by filtering the result of motion and considering only areas where clusters of nodes respond at the same time.

Our independent motion detection procedure can be finally described by the following steps:

- 1) **Track.** A uniform grid $G \subseteq I$ is placed over the image

plane I and for every couple of consecutive frames the Lucas-Kanade optical flow is computed over each grid element solving Equation (3).

- 2) **Compare.** The discrepancy measure expressed by the index M in Equation (4) is determined for each grid node in G . Provided a suitable threshold Θ_M , nodes $(x, y) \in G$ for which $M(x, y) \geq \Theta_M$ identify the subset $G_M \subseteq G$ of potential independent motion locations.
- 3) **Skim.** False positive are eliminated by removing nodes in G_M which are not located nearby at least n other positive responding nodes of G_M (with n a parameter chosen *a priori*). The remaining points in G_M are considered locations over which independent motion is occurring.

The simplicity of the *motionCUT* makes it is well suited for one of the main goals of this work, namely to achieve independent motion detection in real-time. Computing optical flow according to the original Lucas-Kanade algorithm requires the identification of the corners and the successive least-squares solution of Equation (3); by contrast, our method discards the first search phase by adopting a static grid of interest points over which the least-squares computation is performed. As result, *motionCUT* runs in real-time² processing input images of 320x240 pixels at the same frame rate of the cameras acquisition, that is 30 Hz.

IV. PERFORMANCE ANALYSIS

In order to verify the assumptions underlying the proposed method and to obtain a concrete picture of what happens in a real scenario, we performed experiments in a controlled environment. We designed a specific setup where a small soft toy (a stuffed blue octopus) is placed on top of a cart that in turn can travel on a rail track at a given speed. At the same time, the robot iCub stands in front of the setup, steering its head irrespectively of the object along its yaw axis (i.e. from left to right and vice-versa) while acquiring images of the scene through its stereo cameras.

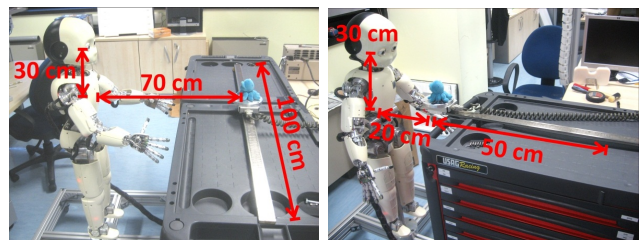


Fig. 4. Setup for the *motionCUT* performance analysis relative to the parallel (left) and orthogonal (right) motion of the cart.

The experimental apparatus is visible in Figure 4 along with the relative distances among components in two different configurations that have been employed to evaluate

²A multi-core Intel (R) Xeon machine with 2.27 GHz of clock frequency has been employed for the experiments. Remarkably, on this machine we measured an average processing time for a single 320x240 pixels image of 15 ms, meaning that *motionCUT* is capable of running up to the speed of 66 Hz.

the behavior of *motionCUT* with respect to the two most common types of motion: a target that moves parallel to the camera plane and a target that travels towards the robot in a direction approximately orthogonal to the camera plane.

With this parameters we observed that 40 deg/s is the maximum angular speed achievable by the neck joint before the *motionCUT* algorithm starts showing small false positives. Similarly, the minimum linear speeds at which the cart has to move before it is detected was identified to be 10 and 20 cm/s respectively for the parallel and orthogonal direction. These values support the intuitive idea that motion along the orthogonal direction is more difficult to detect as it induces smaller changes on the image stream given the same velocity of the cart. Remarkably such lower bounds remained constant for all neck angular velocities.

In order to have a better idea of how fast the projection of the cart on the image plane was actually moving, we analytically computed its pixel per frame velocity when the head was not controlled, the cart was moving parallel to the image plane and its projection laid approximately on the image center. It resulted that in our system – which samples video inputs at 30 fps – the cart linear speed of 10 cm/s corresponded to approximately one pixel per frame.

Experiments were conducted for neck angular velocities ranging from 0 to 40 deg/s and for cart linear speeds ranging from 20 to 100 cm/s. Under these experimental conditions, in the following sections we analyze two main aspects of the system performance: smoothness of the response along time and robustness to slight viewpoint changes (e.g. stereo vision).

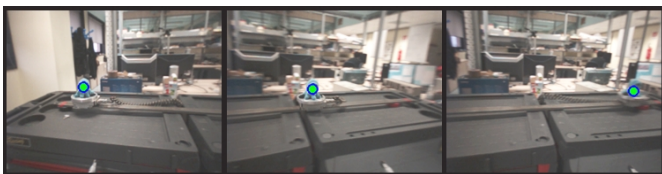


Fig. 5. Example of the *motionCUT* detection precision: the green-blue dot represents the center of mass of the area detected as moving independently by the algorithm while both the head and the cart are moving.

A. Temporal Smoothness

In most cases, motion is not an impulsive event but rather a smooth process. It follows that a desirable property for an independent motion detector would be to respond coherently to changes in time and that its output should not vary excessively from frame to frame.

In order to evaluate the *motionCUT* response along time, we considered the center of mass of the 2D independent moving area detected by the algorithm and then analyzed the trajectory traced by such point on the image plane while both the cart and the head were moving. Typical examples of the detected centroids are shown in Figure 5 where they are identified as the green-blue dots over the image plane.

Figure 6 reports the trajectories of both x and y coordinates of the detected centroids for different cart speeds (from left to right 20, 40 and 100 cm/s) while the head was not

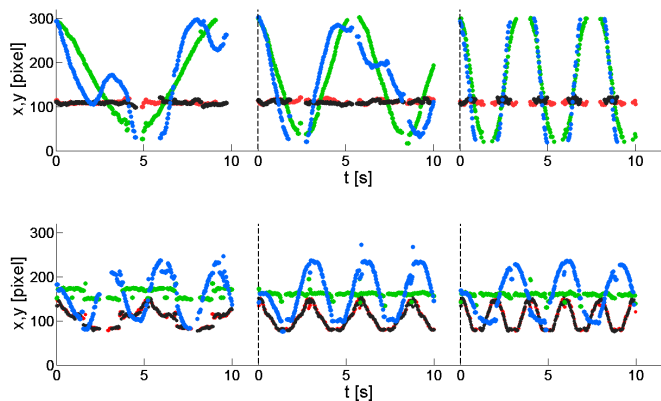


Fig. 6. Trajectories of the x , y coordinates of the center of mass of the areas detected as moving independently. The cart is moving parallel (up) or orthogonal (down) with respect to the image plane. The plots are reported for the following cart speeds: from left to right 20, 40, 100 cm/s. Colors legend: (1) green for x and red for y in the case of a static head; (2) blue for x and black for y in the case of a head rotating at 20 deg/s.

controlled (green for x and red for y) or rotating around the yaw axis at 20 deg/s (blue for x and black for y).

As desirable, for both parallel (top) and orthogonal (bottom) cases, independent motion appears as a remarkably smooth process and is detected without noise excluding time intervals during which the cart moved outside the field of view of the camera. The only exception to this observation results for the cart translating at 20 cm/s in the orthogonal direction. In this case cart motion is too slow and thus not always detected.

It can be further noticed that when the cart moves significantly faster than the neck (e.g. parallel case, cart moving at 100 cm/s), the trajectories traced by the x component is approximately identical to the one registered when the head is stationary.

Finally it is worth mentioning that for the orthogonal case, the y component of the centroid trajectory is not influenced by head rotations and thus it results identical to the y trajectory acquired when the head was not controlled. This observation is evidence of the fact that the *motionCUT* algorithm does not lose information in presence of egomotion.

B. Stereo Coherence

Computing the correspondence between areas in a pair of stereo images is fundamental for depth-perception. Concurrent detection of motion in both images can improve the 3D localization of the moving object.

However, in order to exploit independent motion detection in contexts where stereo vision is involved, an independent motion detection algorithm needs to exhibit coherent performances on the acquired stereo images.

In Figure 7 an indicative example of the *motionCUT* output is provided for synchronized left (top) and right (bottom) image samples acquired while the iCub neck was rotating at 20 deg/s and the cart was moving at 40 cm/s. The informal observation suggests that as desirable, even though not exactly identical in shape, the areas detected as moving

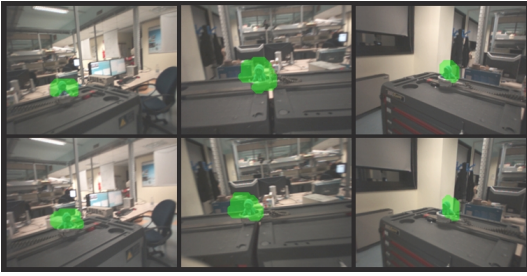


Fig. 7. Three examples of the *motionCUT* output (colored in green) for simultaneous left (top) and right (bottom) image samples. Even if not a proof of coherence to stereo inputs, these images suggest that the *motionCUT* framework is robust to slight viewpoint changes.

independently (colored in green) cover approximately similar parts of the 2D projection of the object.

To have a more formal evaluation of this property we analyzed the discrepancies between moving areas detected simultaneously in left and right images. Measure of such disparity is estimated by matching the 2D centers of mass of these blobs to the complementary image (e.g. left centroids to right images) using a trivial method such as cross-correlation (manually supervised to avoid potential mismatches) and then computing the Euclidean distance between such match and the original centroid of that image.

In Figure 8 are plotted the mean values and standard deviations of the described measure computed for different neck and cart velocities. As it can be noticed, the errors between stereo detected moving areas are relatively low with respect to the actual size of the object’s projection which was evaluated for the parallel case by a human observer to be a window of approximately 45×45 pixels on a total image size of 320×240 . This measure is reported as a red vertical dashed line as a reference to better interpret the numerical results of the stereo coherence measure.

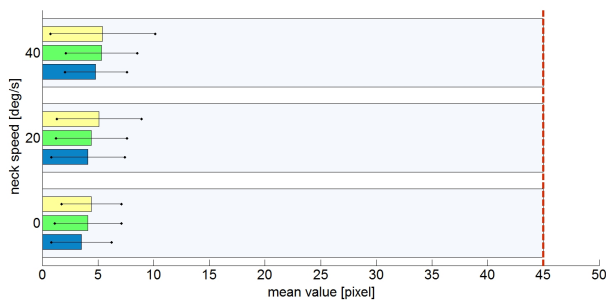


Fig. 8. Mean values of the *motionCUT* measure of stereo coherence for neck angular velocities equal to 0 (bottom), 20 (middle) and 40 (top) deg/s and cart speeds equal to 20 (blue), 40 (green) and 100 (yellow) cm/s. The standard deviations computed during the experiments are plotted as error bars (black lines). The red dashed line represents the approximate side length of the window containing the octopus toy.

V. REAL TIME STEREO TRACKING

A natural application for independent motion detection is real-time control of gaze to track a moving target. With gaze tracking is intended the action of controlling the robot’s head

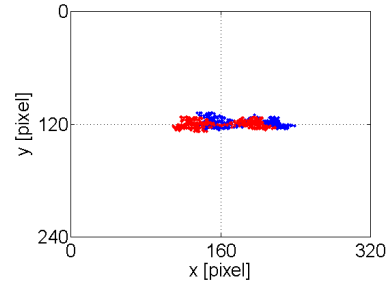


Fig. 9. Scatter plots for the left (blue) and right (red) camera, showing the (x, y) positions of independent moving centroids detected by the *motionCUT* while the robot performed stereo gaze tracking of the toy.

motors in order to keep the projection of the moving object within the camera planes as close as possible to the images center. In Section IV we showed that the center of mass of the detected area moves smoothly with respect to time, and as such it is suitable signal for tracking; furthermore, we also observed that *motionCUT* responds similarly when applied to left and right images thus implying that gaze tracking can be performed on stereo cameras.

The iCub gaze controller can control the robot’s neck and the eyes individually to follow a target detected on both cameras. This controller is purely feedback and does not perform anticipatory movements based on a prediction of the position of the target. Thus, as in our system the visual signal frequency is bounded to 30 fps we do not expect the controller to completely reduce the tracking error to zero (i.e. it will always lag slightly behind the moving target).

Figure 9 reports the scatter plots of the centroids collected throughout the experiment of the resulting regions identified as belonging to the toy moving at 40 cm/s parallel with respect to the image plane: the diagram of the superimposed left and right traces illustrates that the centroids lay in the neighborhood of the images center. This confirms that the robot achieves its goal successfully and manages to keep the object in the fovea even in the presence of a cluttered background.

Finally, Figure 10 provides an excerpt of a complete images sequence recorded during a more traditional tracking task where the robot is requested to gaze at a person walking through a typical laboratory environment. This experiment exposes the motion detector to challenging environment containing distractors such as occluding elements, rapid changes in light conditions, as well as unexpected emergence of moving objects. Despite the complexity of the background, it is evident from the strip images that our method produces robust detection of the moving target with a behavior that varies smoothly in time and is consistent with respect to the two different views acquired from the left and right cameras of the robot. In particular, the movement of the target is effectively tracked both when the person is far from the robot (frames 1 and 6) and when he gets closer to it (frames 2-5). Furthermore in the images sequence there exists a substantial modification in light conditions with a maximum of brightness reached approximately at frame 4.

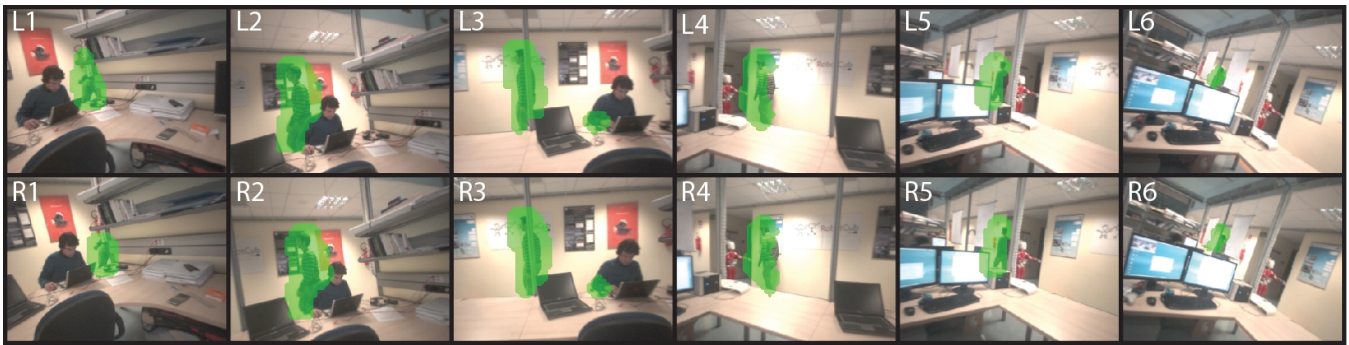


Fig. 10. A strip of images recorded during the real time stereo tracking of a walker: six images are shown in their temporal sequence from L1 to L6 as taken from the left camera, whereas images from R1 to R6 represent the corresponding acquisitions from the right camera. The walking person is highlighted with a green blob using the result of *motionCUT* detection.

The algorithm is robust to occlusions, this is visible at the frames in which pillars and posters cover the walker. Notably, at frame 3 the person sitting at the table produces a secondary blob of motion with his hand. This distractor is of limited size and it does not interfere with the task since the tracker is instructed to follow the largest blob in the stereo pictures.

VI. CONCLUSIONS AND PERSPECTIVE WORK

In this work we have presented the *motionCUT* framework whose purpose is to detect motion occurring in visual scenes independently of the egomotion generated by moving cameras. We performed the experimental validation of this technique on the humanoid robot iCub. The results reported in the paper show that the output of the *motionCUT* is sufficiently informative to control the gaze of the robot to track moving targets with both eyes. Remarkably, we determined that tracking on stereo images can be performed with an average error of few pixels (less than 5) and, importantly, the algorithm is computationally simple to run in real-time on normal computers. Our tests were performed on a single platform; however, since the method relies only on vision and does not require any knowledge about the environment or the kinematics of the robot, it is easily portable on other platforms.

The *motionCUT* is a bottom-up method whose output consists of local observations regarding the visual stream behavior; it is thus clear that it could largely benefit from the integration with top-down techniques that exploit the image appearance ([22], [16]) and/or joint velocities ([18]) as complementary information, eventually enhancing the algorithm performances and, specifically, the rejection of false positives for high speed of the camera motion. Our future goal is therefore to blend all these different data in the probabilistic framework of a Markov Random Field ([23]) by combining the topological structure of the *motionCUT* nodes grid with the global observations on the images appearance and prior knowledge of cause-effect relations for head movements.

Additionally, we are currently researching on the possibility to employ *motionCUT* to learn eye-hand coordination for precise reaching tasks. In fact, due to the stability demonstrated when applied to stereo images, this algorithm

provides effective motion cues that can be exploited to detect and control the hand of the robot while autonomously exploring the workspace.

REFERENCES

- [1] E. S. Spelke, R. Kestenbaum, D. Simons, and S. D. Wein, "Spatiotemporal continuity, smoothness of motion and object identity in infancy," *The British Journal of Developmental Psychology*, vol. 12, pp. 113–142, 1995.
- [2] W. C. Simith, S. C. Johnson, and E. S. Spelke, "Motion and edge sensitivity in perception of object unity," *Cognitive Psychology*, vol. 46, pp. 31–64, 2003.
- [3] J. Ruesch, M. Lopes, A. Bernardino, J. Hoernstein, and R. P. J. Santos-Victor, "Multimodal saliency-based bottom-up attention a framework for the humanoid robot icub," 2008.
- [4] P. Fitzpatrick and G. Metta, "Grounding vision through experimental manipulation," *Philosophical Transactions of the Royal Society: Mathematical, Physical, and Engineering Sciences*, vol. 361, pp. 2165–2185, 2003.
- [5] M. G. Ross and L. P. Kaelbling, "A systematic approach to learning object segmentation from motion," September 2003.
- [6] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun, "Structure from motion without correspondence," 2000.
- [7] L. Natale, F. Orabona, G. Metta, and G. Sandini, "Exploring the world through grasping: A developmental approach," 2005.
- [8] C. Kemp and A. Edsinger, "What can I control?: The development of visual categories for a robot's body and the world that it influences," in *IEEE International Conference on Epigenetic Robotics*, Paris, France, September 2006.
- [9] K. Gold and B. Scassellati, "Using probabilistic reasoning over time to self-recognize," *Robotics and Autonomous Systems*, vol. 57, no. 4, pp. 384–392, 2009.
- [10] "The icub web site," <http://www.icub.org>.
- [11] M. Irani and P. Anandan, "A unified approach to moving object detection in 2d and 3d scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 6, pp. 577–589, 1998.
- [12] R. C. Nelson, "Qualitative detection of motion by a moving observer," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 33–46, 1991.
- [13] B. Jung and G. S. Sukhatme, "Detecting moving objects using a single camera on a mobile robot in an outdoor environment," *Intelligent Autonomous Systems*, pp. 980–987, 2009.
- [14] D. Fox, "Kld-sampling: Adaptive particle filter," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 33–46, 1991.
- [15] C. M. Bishop, "Pattern recognition and machine learning," *Springer*, 2006.
- [16] A. Talukder and L. Matthies, "Real-time detection of moving objects from moving vehicles using dense stereo and optical flow," *IEEE Conference on Intelligent Robots and Systems*, 2004.
- [17] A. A. Argyros and S. C. Orphanoudakis, "Independent 3d motion detection based on depth elimination in normal flow fields," 1997.
- [18] S. Rougeaux and Y. Kuniyoshi, "Velocity and disparity cues for robust real-time binocular tracking," 1997.

- [19] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," 1981.
- [20] J. Shi and C. Tomasi, "Good features to track," 1994.
- [21] J. Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker description of the algorithm," 2004. [Online]. Available: http://robots.stanford.edu/cs223b04/algo_tracking.pdf
- [22] H. Sawhney, Y. Guo, and R. Kumar, "Independent motion detection in 3d scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1191–1199, 2000.
- [23] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1984.