frontiers in Robotics and AI

TECHNOLOGY REPORT
published: 29 June 2016
doi: 10.3389/frobt.2016.00035

# Enabling Depth-Driven Visual Attention on the iCub Humanoid Robot: Instructions for Use and New Perspectives

*Giulia Pasquale[1,2,3]*, Tanis Mar[1,3], Carlo Ciliberto[2,4], Lorenzo Rosasco[2,3,4] and Lorenzo Natale[1]*

*[1] iCub Facility, Istituto Italiano di Tecnologia (IIT), Genova, Italy, [2] Laboratory for Computational and Statistical Learning, Istituto Italiano di Tecnologia, (IIT), Genova, Italy, [3] Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi, Universitá degli Studi di Genova, Genova, Italy, [4] Poggio Lab, Massachusetts Institute of Technology, Cambridge, MA, USA*

Reliable depth perception eases and enables a large variety of attentional and interactive behaviors on humanoid robots. However, the use of depth in real-world scenarios is hindered by the difficulty of computing real-time and robust binocular disparity maps from moving stereo cameras. On the iCub humanoid robot, we recently adopted the Efficient Large-scale Stereo (ELAS) Matching algorithm (Geiger et al., 2010) for computation of the disparity map. In this technical report, we show that this algorithm allows reliable depth perception and experimental evidence that demonstrates that it can be used to solve challenging visual tasks in real-world indoor settings. As a case study, we consider the common situation where the robot is asked to focus the attention on one object close in the scene, showing how a simple but effective disparity-based segmentation solves the problem in this case. This example paves the way to a variety of other similar applications.

**Keywords: disparity-based segmentation, visual tracking, disparity map, humanoid robotics, iCub**

OPEN ACCESS

***Edited by:***
*Alexandre Bernardino, Univ. Lisboa, Portugal*

***Reviewed by:***
*Juxi Leitner, Queensland University of Technology (QUT), Australia*
*Aamir Ahmad, Institute for Systems and Robotics, Portugal; Instituto Superior Técnico, Portugal*

****Correspondence:***
*Giulia Pasquale*
*giulia.pasquale@iit.it*

***Specialty section:***
*This article was submitted to Humanoid Robotics, a section of the journal Frontiers in Robotics and AI*

***Received:*** *30 December 2015*
***Accepted:*** *09 June 2016*
***Published:*** *29 June 2016*

***Citation:***
*Pasquale G, Mar T, Ciliberto C, Rosasco L and Natale L (2016) Enabling Depth-Driven Visual Attention on the iCub Humanoid Robot: Instructions for Use and New Perspectives. Front. Robot. AI 3:35. doi: 10.3389/frobt.2016.00035*

## 1. INTRODUCTION

The main obstacle to stereo vision lies in the process of matching 2D points in the images coming from the cameras on both eyes in order to compute the amount of displacement, or *disparity*. In this work, we consider the Efficient Large-scale Stereo (ELAS) Matching algorithm (Geiger et al., 2010) and incorporate it in the visual perceptual system of the iCub robot (Metta et al., 2008).

According to standard KITTI Stereo-Vision Benchmark (Geiger et al., 2012; Menze and Geiger, 2015), ELAS offers a reasonable trade-off between quality of the disparity estimation and computational time, which makes it particularly suited for applications that require real-time performance. Indeed, in the KITTI 2015 and 2012 benchmarks,[1,2] ELAS is the first method, among the ones that require less than 1s per frame processing time, which comes with an open-source implementation. This threshold is purely indicative but – given the relatively high resolution of the images constituting the KITTI benchmark – is aimed at excluding those methods that do not provide real-time performance. Those algorithms in the KITTI benchmarks performing better *and* faster than ELAS

[1] http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo
[2] http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=stereo

are either proprietary (Einecke and Eggert, 2010, 2013, 2015) or rely on Convolutional Neural Network architectures and require dedicated GPUs (Mayer et al., 2015).

Therefore, we have integrated the ELAS algorithm, which is available with an open-source license as a self-contained, optimized C++ library, LIBELAS,[3] into the iCub software system and tested it in different applications, making them readily available for the benefit of the community. In this technical report, we describe the software we have implemented and present a set of quantitative and qualitative experiments to assess the efficacy of the ELAS algorithm in a realistic robotic setting.

## 2. RELATED WORK

Depth is a natural cue for object segmentation. For example, consider this common situation for a humanoid robotic platform: a human stands in front of the robot showing to it an object to be recognized or grasped. Both motion- and appearance-based approaches to focus the robot's attention on the object of interest would impose many constraints even on this simple scenario. Color-based methods work under strict assumptions on the light conditions, kind of background (preferably a table or a wall) and generally fail in cluttered setting. Model-based methods can overcome many of the above limitations, but, more importantly, require the shape type of the object to be known *a priori* (Greggio et al., 2011). Motion cues are an alternative [see, e.g., Ciliberto et al. (2011) and Kumar et al. (2015)] but clearly require objects to be moving and ego-motion compensation. Perhaps the most distinguishing feature is simply the fact that the object of interest is closer to the robot than the background. Indeed, depth information has been exploited in similar robotics settings in the past (Goerick et al., 2005, 2006; Wersing et al., 2006, 2007, 2008; Leitner et al., 2012a; Rudinac et al., 2012).

The issues involved by estimation of the disparity map on a humanoid robotic head are related to calibration, speed, and robustness. Indeed, since performing 2-dimensional searches for matching points in the two cameras is computationally expensive, a commonly adopted approach for vergent stereo systems (Hartley and Zisserman, 2003) is to first *rectify* the left and right images, in order to bring corresponding points to lie on the same scanline. Then, disparity can be computed by performing only horizontal searches. However, accurately estimating the reciprocal position of the stereo pair, needed by the rectification step, can be difficult, particularly on a humanoid robotic head, where the pan and vergence of the robot eyes change continuously and the kinematic information is affected by uncertainty. The following horizontal disparity computation step then remains the main computational bottleneck. At present, solutions providing the disparity map with the right trade-off between speed and accuracy are active subject of research. Finally, robustness to lighting conditions, poorly textured regions and inaccuracies in the rectification is another key requirement for an algorithm to be usable on a robotic platform.

A different approach proposed the use of machine learning methods as Neural Networks to overcome the calibration problem. In Leitner et al. (2012b), the authors are able to learn a network that maps the 2D projections of a point onto the left and right cameras to its 3D position in the world, being the cameras orientation approximately provided by the robot kinematics. Combining this with an object detection technique that segments an object in both cameras, they are able to localize the object without the need for any calibration. However, by considering only a single point as the object's centroid, they do not solve the problem of 2-dimensional matching and disparity computation. Moreover, they rely on an appearance-based segmentation algorithm that suffers from the limitations of color-based methods mentioned above.

For all these difficulties, active depth sensors have been often preferred when building complex behaviors as, e.g., interactive object learning (Lyubova et al., 2015). The goal of this work is to improve the stereo perception of the iCub robot in order to make dense 3D information usable in action–perception loops. In particular, we show an example object tracking application. Since the main problems affecting the stereo perceptual system of the robot were slowness and robustness, in this work we aim at improving the disparity computation step with respect to these aspects. For camera calibration and image rectification, we adopt the currently implemented technique on the iCub robot, described in Fanello et al. (2014) and briefly resumed in Section 3.1.

We decided to rely on the LIBELAS library because, comparing to other local dense stereo matching methods, which can be faster [see, e.g., OpenCV's Block Matching algorithm implementation (Bradski and Kaehler, 2013), for which a GPU accelerated version is also available], LIBELAS provides better matching results in texture-less regions. Indeed, the matching performance of algorithms based on local correspondences is affected by the window size, which is particularly critical in real-world application scenarios, where scenes are characterized by large texture-less elements. In this setting, small windows can be uninformative but too large windows cause border bleeding artifacts and heavier computational time (Geiger et al., 2010). Instead, the ELAS algorithm overcome this problem by propagating to non-textured regions disparity information derived on a set of robust correspondences. Moreover, when compared to semi-global methods, including OpenCV's implementation of Hirschmuller's Semi-Global Block Matching (SGMB) algorithm (Hirschmuller, 2008) currently in use on the iCub platform, ELAS scales better with respect to the disparity search space, which is generally large in robotics application where the robots must perceive both close and far objects. Indeed, in contrast with SGBM, which computes matching costs at each pixel for the full disparity space image, ELAS reduces the search only to plausible values between neighboring support points' disparities. Finally, according to recent experimental evidence (Sinha et al., 2014) (and to the KITTI benchmark), ELAS is remarkably fast on large images.

These are also the main reasons why LIBELAS has been the library of choice for many previous robotic applications. LIBELAS is adopted in Tombari et al. (2011) for an object recognition task, and in Van den Bergh and Van Gool (2012) it is used in conjunction with color and optical flow to provide a real-time

[3] http://www.cvlibs.net/software/libelas/

super-pixel segmentation of the scene. In Mitzel and Leibe (2012) and Baumgartner et al. (2013) LIBELAS is at the basis of an algorithm for people detection and inference engine that learns interactions between people and objects. Moreover, (Lin et al., 2012) show that LIBELAS can be implemented on an embedded ARM-based processor with real-time on mid-resolution images.

Based on these motivations, we decided to integrate LIBELAS into the iCub's depth perception system. The spirit of our work is close to Leitner et al. (2008), where the authors adapt the algorithm for disparity computation presented in Bernardino and Santos-Victor (2002) to work on a humanoid robotic head and benchmark its performance and computational efficiency. In this work, we validate the integration of LIBELAS in the iCub's stereo perceptual system. We focus in particular to situations where 3D information is used in visuo-motor tasks, specifically object detection and tracking in indoor settings. The contribution of the paper is twofold: we first improve the iCub stereo perceptual system, by providing the possibility of employing LIBELAS, beyond OpenCV's SGBM, for disparity computation; by building on faster and more robust 3D information, we are then able to implement a disparity-based tracker that allows for detection of never-seen objects on cluttered and dynamic background; this, finally, allows us to realize an interactive object learning application that we use also for semi-automatic ground-truth collection. All the code of the applications that we show in this work is made publicly available to the iCub community.

In Section 3 we briefly review the processing steps currently adopted on the iCub for depth estimation; in Section 4 we describe the application that we devised to focus the robot's attention on the closest object in its workspace. Finally, in Section 5 we experimentally demonstrate the effectiveness of this approach.

# 3. DEPTH ESTIMATION

In this section we briefly describe the depth estimation pipeline adopted in this work. Following the standard approach from multi-view geometry (Hartley and Zisserman, 2003), the process is organized into two main phases: image rectification and disparity computation.

The rectification step estimates the geometrical transformation matrix relating left and right image planes in order to align the epipolar lines with the image scanlines. After this operation, horizontal disparity is computed for each pixel in the left (right) rectified image, by searching its correspondent point in the right (left) rectified image along its scanline. The resulting disparity map provides an estimation of the 3D structure of the scene as a cloud of points (whose projections end up on the image pixels) with respect to the observer. To recover the 3D position of the point corresponding to a specific pixel, the camera's extrinsic parameters can be used, in combination with its disparity, to re-project it.

Regarding the estimation of the camera parameters, we follow the procedure described in Fanello et al. (2014), which involves online calibration starting from the initial off-line calibration (online re-calibration is required when the reciprocal position of the eyes change). As mentioned already, for disparity estimation, we adopt the Efficient Large-scale Stereo (ELAS) Matching algorithm proposed in Geiger et al. (2010).

## 3.1. Rectification

Image rectification consists in the process of transforming a set of multiple images onto the same plane and is a fundamental step to most depth estimation algorithms. Rectification requires knowledge of both the intrinsic (camera specific) parameters of the two (or more) cameras and extrinsic parameters, i.e., the position and orientation of the cameras with respect to the world reference frame. More formally, any 3D point with coordinates $\mathbf{X} = (x, y, z, 1)^\top$ with respect to the world reference frame is mapped on the camera image plane $\mathbf{x} = (u, v, 1)^\top$ via the transformation

$$s\mathbf{x} = P\mathbf{X} \tag{1}$$

where $s \in \mathbf{R}$ is a scaling factor, $P \in \mathbf{R}^{3\times4}$ is the *Projection Matrix* that can be factorized as $P = K[R|t]$, with $K \in \mathbf{R}^{3\times3}$ and $[R|t] \in \mathbf{R}^{3\times4}$ the matrices of intrinsic and extrinsic parameters, respectively.

Intrinsic and extrinsic parameters can be estimated off-line during a calibration phase. However, while intrinsic parameters are camera specific and do not change over time, on the iCub the relative pose of the cameras changes when the robot fixates objects at different distance. To circumvent this issue, Fanello et al. (2014) pre-compute the intrinsic parameters matrices $K_L$ and $K_R$ using standard calibration procedure (Hartley and Zisserman, 2003). Extrinsic parameters are then re-estimated at runtime. This calibration employs SIFT matching to estimate the *Fundamental Matrix* between the two camera planes [details are in Fanello et al. (2014)]. To achieve real-time performance, Fanello et al. (2014) exploit the robot's kinematics to approximate the camera transformation between subsequent frames and perform re-calibration using SIFT matching at a lower frame rate. This procedure is implemented in the `SfM (Structure from Motion)` module included in the iCub `stereo-vision` repository.[4] Once the projection matrices $P_L$ and $P_R$ associated with the left and right cameras are known, the corresponding images can be mapped onto the same plane, i.e., they are *rectified*. They are, therefore, ready for the subsequent stage: disparity estimation.

## 3.2. Disparity Computation with ELAS

Disparity estimation consists in the process of evaluating the displacement of pixels from one (rectified) image to the other. Disparity is usually computed after rectification since at this stage the corresponding image points from the left and right cameras lie on the same scanline and, therefore, matching can be restricted to horizontal lines. A variety of disparity estimation methods have been proposed in the literature. The Efficient Large-Scale Stereo (ELAS) Matching algorithm proposed in Geiger et al. (2010) consists in the following two phases:

1. A set of robust support points is detected and matched across the two images.
2. Dense disparity on a uniform group of points is obtained from these support points in a Bayesian framework.

[4] https://github.com/robotology/stereo-vision

In this section we offer a very brief overview of the ideas underlying ELAS, while referring the reader to the original paper for a more detailed description of the algorithm (Geiger et al., 2010).

#### 3.2.1. Support Points

The first phase of ELAS is performed on a predetermined grid on the image plane, where candidate points are selected depending on their local appearance. To do so, the authors used a vector of local orientations (response to oriented Sobel filters) and performed robust matching between such feature vectors to eliminate unstable pairs of points. The outcome of this stage is a set $\mathbf{S}$ of points $s = (u, v, d)^\top$ that encode the position $(u,v)$ of a support point on the left (rectified) image and the disparity $d$ with respect to the matched point on the right (rectified) image.

#### 3.2.2. Bayesian Inference

The second phase relies on the two-view geometry parameters estimated in Section 1 and the support points $\mathbf{S}$ to predict the most likely disparity values for the remaining image pixels. In particular, the authors adopt a Bayesian framework to model the likelihood:

$$p(d \mid x^{(L)}, x_1^{(R)}, \ldots, x_n^{(R)}, \mathbf{S})$$

of observing a disparity $d$ for a given point $x^{(L)}$ on the left image and a set of candidate corresponding points $x_1^{(R)}, \ldots, x_n^{(R)}$ on the right image. The most likely disparity value is, therefore, estimated by factorizing such likelihood and performing a Maximum *A Posteriori* (MAP) procedure.

As pointed out in Geiger et al. (2010), this procedure can be carried out independently for each image point and it is fast and parallelizable. Clearly, this is a critical feature for the robotic setting, where the disparity estimation process must be computed at frame rate. LIBELAS offers two presets of parameters, `MIDDLEBURY` and `ROBOTICS`, the latter being specifically tuned for higher robustness to dynamic lighting conditions and real-world scenes. We integrated the `OpenMP` parallelization of LIBELAS, available at the same website of the library, in the `stereo-vision` repository.

## 4. OUR BENCHMARK: DISPARITY-DRIVEN ATTENTION

For validation, we consider the following benchmark. We designed a segmentation procedure based on the disparity map. This procedure identifies distinct 3-dimensional entities in the scene and it focuses the robot's gaze toward the object that is closest to the cameras. By following this strategy, we were able to implement a simple but effective tracking algorithm that continuously focuses the robot's attention and gaze toward the closest object in the scene, while at the same time providing also an approximate visual segmentation.

We first employed this basic tracking system to perform a qualitative and quantitative analysis of the disparity map produced by ELAS in a real-world indoor robotic setting (Sections 5.1 and 5.2). Then, within this general scenario we defined a reliable protocol to acquire ground-truth for visual object recognition (Section 6.1). We are exploiting this application to acquire a dataset of images depicting multiple objects held in the hand of a human teacher who is showing them to the iCub, which is going to be released soon. In fact, a similar strategy, based however on independent motion detection, was previously explored on the iCub robot (Fanello et al., 2013a,b) for previous releases of the *iCubWorld* dataset. We show that in such an application disparity information results in a more reliable and stable cue.

In the following, we describe the algorithm we devised to segment the object closest to the iCub cameras and focus the robot's gaze on it.

### 4.1. Foremost Object Segmentation

To achieve real-time performance, we reduced the post-processing operations on the disparity map to the minimum. Therefore, we implemented a proof-of-concept segmentation algorithm that can provide a reasonably stable and accurate blob around the closest proto-object in the scene. We are aware of the existence of more sophisticated algorithms, which may provide more precise segmentations [e.g., Li et al. (2013)]. These algorithms could be easily plugged in the present pipeline to realize other behaviors that require more accurate segmentation. Below we report the algorithm together with the parameters currently used on the robot. The code, implemented by using standard OpenCV functions, has been made available in the `dispBlobber` module, part of the iCub `segmentation` repository.[5]

1. *Filtering:* a $5 \times 5$ Gaussian filter ($\sigma_x = \sigma_y = 1.5$) is applied to the disparity map in order to smooth the surfaces. Then, as we are interested in the foremost object, background is suppressed by putting to black/zero all pixels whose grayscale value is under a threshold, set to $30 \div 50$, so that farther (darker) pixels are removed. A sequence of 4 dilation and 2 erosion operations with a $3 \times 3$ kernel matrix, interleaved by another $5 \times 5$ Gaussian filtering ($\sigma_x = \sigma_y = 2$) follows, in order to suppress small blobs and fill holes in large blobs.
2. *Blob selection:* a simple routine to localize the closest blob of "reasonable" size (i.e., larger than a predefined threshold to avoid spurious detections) is devised. Iteratively:
   (a) find the 2D image location of the brightest (closest) pixel as a vector $(u,v)$,
   (b) generate a candidate blob from the seed pixel by aggregating all neighboring pixels with disparity value between the brightest one ($d$) and a lower threshold defined by $0.9*d$,
   (c) suppress (putting to black/zero) the aggregated region if its size is lower than a threshold (in our experiment set to 300 pixels for $320 \times 240$ images and to 1400 pixels for $640 \times 480$ images),
   (d) start again from $a$ until one region satisfying the size requirement is found, and
   (e) check whether the returned region is composed by a single blob or by multiple connected blobs, by selecting only the largest one satisfying the size threshold fixed

[5] https://github.com/robotology/segmentation/tree/master/dispBlobber

above, in the latter case. This is achieved by first computing the contours of the region and then iterating on the connected components, discarding those whose area is under the threshold.

3. *Computing the blob's centroid and ROI:* if any blob is found, its center of mass and its smallest enclosing rectangular bounding box (ROI), with an arbitrary margin, are computed.
4. *Averaging over a temporal buffer:* finally, the centroid and the ROI are averaged over a buffer of $n$ frames (with, e.g., $n = 3$) in order to mitigate isolated mis-detections.

## 4.2. Foremost Object Fixation

At this stage, the foremost object's centroid on the reference (left) image is available for further usage. In this application, the `dispBlobber` module asks the `SfM` module to re-project the computed centroid to its corresponding 3D position in the Cartesian space. This 3D point is finally fed to the module in charge of controlling the robot's gaze [`iKinGazeCtrl` (Pattacini, 2011)], which moves the iCub's eyes accordingly. This solution is more accurate than performing the triangulation using the robot kinematics, because the `SfM` module performs a visual estimation of the relative orientation of the cameras [using the algorithm in Fanello et al. (2014)].

As a consequence, the head and eyes positions of the robot are continuously updated to keep the focus of attention fixed on the required 3D target, i.e., the closest object in the visual field, while the human moves it in front of the cameras. This pipeline is looped in real time so that the robot is able to follow the closest object with the gaze. It is then clear that relying on a fast and robust disparity map (eventually at the expenses of some subpixel precision) in this kind of application is critical, and the reported results confirm that LIBELAS is suited to this task.

We note also that this is a very basic (yet effective) implementation for a disparity-driven attention system and that further improvements as, e.g., applying a Kalman filter to the trajectory of the 3D centroid, could be introduced to smooth and stabilize the resulting tracking system.

# 5. EXPERIMENTAL EVALUATION

In this section, we present a qualitative as well as quantitative analysis of the depth estimation process described in Section 3, with particular focus on the improvements provided by the ELAS algorithm, which represents the novel element of the pipeline for disparity computation. As we are mainly concerned in assessing the possibility to employ this algorithm in practical, real-time, robotics applications, we first evaluate the disparity-based segmentation protocol introduced in Section 1 and then we evaluate this approach for disparity-driven visual attention behavior.

For our experiments, we employed the OpenCV (Bradski and Kaehler, 2013) implementation of the Semi-Global Block Matching algorithm (SGBM) (Hirschmuller, 2008) as a baseline to compare the performance of ELAS. Indeed, SGBM was considered the "off-the-shelf" disparity estimation algorithm for the iCub robot used in the `SfM` module in the iCub `stereo-vision` repository [see Fanello et al. (2014)].

## 5.1. Real-Time Disparity-Driven Segmentation

We collected a sequence of 200 frames at the resolution of $640 \times 480$, acquired across 2 minutes and recorded from the iCub cameras while a human subject was moving his hand in front of the robot. We computed the performance of the disparity-based segmentation protocol described in Section 1 when using ELAS or SGBM. In **Table 1,** we report the computational time required on our platform [Intel(R) Core(TM) i7 3770QM CPU at 3.40 GHz with 16-GB RAM] to perform the disparity estimation and segmentation, averaged over the whole acquisition sequence. We also report the ratio of "missed" blobs: the ratio of frames for which the segmentation algorithm failed to detect any blob. Our experiments show that ELAS significantly outperforms the baseline and moreover leads to better segmentation. In **Figure 1**, we report three exemplar images from this sequence: the first row depicts the rectified images acquired from the left camera (those from the right camera are not reported); the second and third rows report, respectively, the disparity maps and the resulting segmentation, obtained with ELAS (odd columns) or SGBM (even columns).

It can be noticed that the LIBELAS implementation is fast (achieving a 15 fps rate with respect to the 5 fps provided by SGBM) and robust enough to allow for further applications of the computed disparity, such as the disparity-driven attention behavior described in the following.

For these experiments, the disparity range was set to [0,127] for both ELAS and SGBM. In **Table 2**, we report the parameters of the SGBM algorithm, which have been tuned to the specific iCub's indoor setting. Those not reported were left to their default value [see OpenCV's documentation (Bradski and Kaehler, 2013)].

In the case of LIBELAS implementation, we chose the `ROBOTICS` preset of parameters offered by the library. In order to speed up computations we set the `post_process_only_left` and the `subsampling` parameters to `true` and employed the `OpenMP` accelerated version of the library. All other parameters were left to their default value.

## 5.2. Disparity-Driven Visual Attention

In this section we test the presented simple attention system driven by disparity information, whose underlying principle is to keep the robots' gaze focused on the closest object in the scene. In particular, we consider the following setting: a human moves an object in front of the robot camera, and we evaluate the stability of the resulting "tracking" application.

We employed the pipeline described in Section 4. In the current experiment, we used low-resolution images ($320 \times 240$) in

**TABLE 1 | Average computational time (splitted for disparity computation and segmentation) and percentage of blobs missed by LIBELAS and SGBM over the sequence represented in Figure 1.**

| | SGBM | ELAS |
|---|---|---|
| Time disparity [ms] | 190 | 60 |
| Time segmentation [ms] | 20 | 5 |
| Time total [ms] | 210 | 65 |
| Missed blobs [%] | 11.2 | 2 |

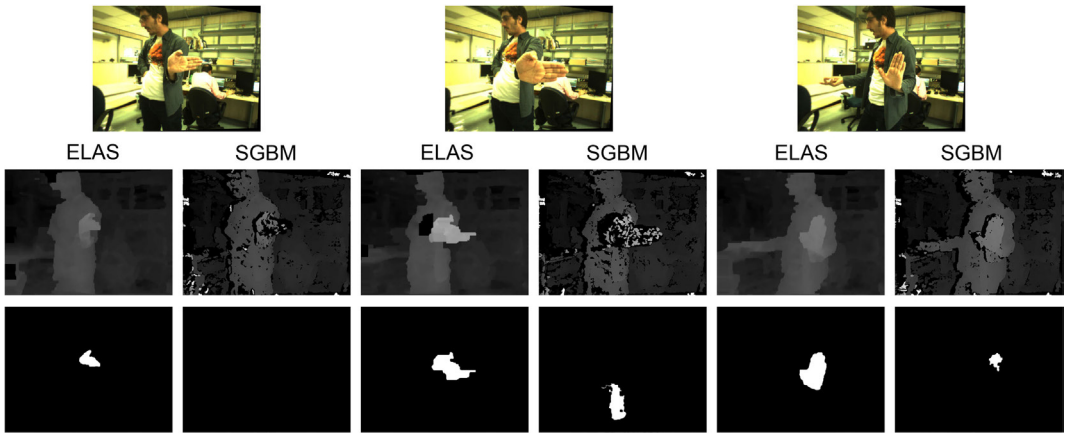


**FIGURE 1 | Examples from a sequence recorded on the iCub's cameras, with fixed eyes and head**. Top: left rectified images. Middle: disparity maps computed by LIBELAS and OpenCV's SGBM methods. Bottom: segmentation of the closest blob in the scene.

**TABLE 2 | SGBM parameter setting.**

| OpenCV's parameter name | Value |
|---|---|
| `preFilterCap` | 63 |
| `SADWindowSize` | 7 |
| `P1` | 8 · 7 · 7 |
| `P2` | 32 · 7 · 7 |
| `uniquenessRatio` | 15 |
| `speckleWindowSize` | 50 |
| `speckleRange` | 16 |
| `disp12MaxDiff` | 0 |

*The left column reports the parameter name in the OpenCV implementation and the right column its value in our application.*

order to provide the gaze controller with a more frequent feedback and the disparity range was reduced to [0, 95]. For SGBM, we used the same parameter set of the previous experiment. For LIBELAS, there was no need to enable the `subsampling` since the lower resolution already allowed to achieve frame-rate performance (30 fps) on our platform. Experiments with SGBM were performed off-line because the lower efficiency (~10 fps) did not allow for a smooth tracking.

As a reference ground-truth to compare with the result of the disparity-based segmentation, we used the output of a model-based object tracker (Taiana et al., 2010). In particular, we used a red ball for which a well-established particle filter tracker is implemented in the `pf3dTracker` module, part of the `icub-basic-demos` repository.[6] As the operator moved the red ball in front of the robot, the gaze was focused toward it (since it was the closest object in the scene). The information about its estimated position was acquired independently using the disparity-based segmentation procedure described above and the color/shape-based particle filter tracker (see **Figure 2**). We recorded the coordinates ($u_{disp}$, $v_{disp}$) of the closest blob's centroid on the left image plane as provided at each frame by the segmentation module on top of ELAS disparity map. At the same time, we recorded the coordinates ($u_{model}$, $v_{model}$) of the center of the red ball in the same image plane, provided by the red-ball detector.

**Figure 3** reports the image plane coordinates (top rows) with red and blue colors, respectively, for disparity and model-based tracker, and their difference (bottom rows). Notice that, in some frames, ELAS estimates the wrong position for the blob (sudden jumps in the red curves); instead, the red ball tracker fails to detect its target within a 2 s interval around $t = 18$ s in the plot. In **Figure 2**, we provide a short sequence showing the ELAS failure around $t = 2.6$ s. Indeed, it does happen that, since the robot is moving and we are in an uncontrolled setting, the disparity map is affected by noise that cause false blob detections. Notice, however, that these errors occur on isolated frames and can be removed by filtering the 2D image position detected by raw segmentation. In **Figure 4**, we report instead a short sequence extracted from the interval in which the red ball detector fails: in this case, the error is due to a constant mis-detection caused by the slightly changed lighting condition. This unexpected behavior by the way offers the occasion to highlight that the proposed approach based on the disparity cue for tracking and segmentation is not only a viable solution but can also be even more robust than appearance-based information when the 3D position of the target is a more stable signal than its color.

**Figure 5** shows the same quantities of **Figure 3** but computed on the disparity map provided by SGBM. We computed the disparity map off-line on the same set of rectified images acquired when tracking with the ELAS algorithm. As can be clearly noticed, the unstable behavior of the disparity produced by SGBM is not sufficient to provide a fast and reliable signal to track the ball.

## 6. APPLICATIONS OF DISPARITY ON HUMANOID ROBOTS

In this section we show how the disparity-driven attention system described in Section 5 can be employed to improve the robot perception of the surrounding environment. In particular, we consider a basic interaction between the robot

[6]https://www.github.com/robotology/icub-basic-demos

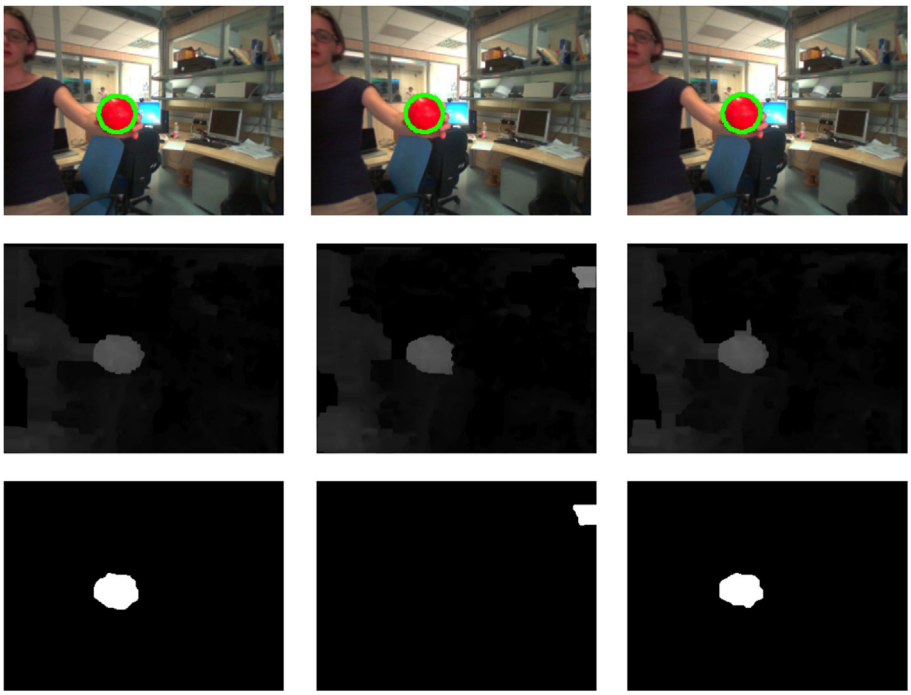

**FIGURE 2 | Frames extracted from the sequence represented in Figure 3 around $t = 2.6$ s, when LIBELAS fails to detect the closest object**. Top: output of the red ball tracker. Middle: disparity map. Bottom: disparity segmentation.

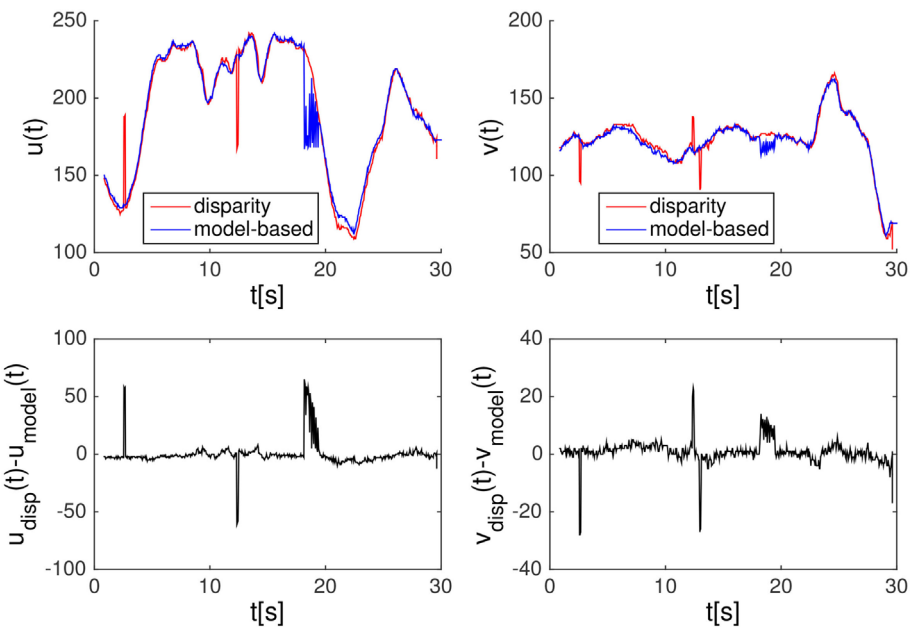


**FIGURE 3 | Coordinates of the red ball target, recorded while the human operator was moving it in front of the robot**. Top: $u$ and $v$ coordinates of the closest blob's centroid on the left image plane (red trace), provided at each frame by the disparity segmentation module, and of the center of the red ball in the same image plane (blue trace), provided by the red ball detector. Bottom: difference between the two. LIBELAS is used to provide the disparity map.

and a human teacher or the situation in which the robot needs to visually select objects on a table. In this section, our observations are mainly qualitative.

## 6.1. On the Fly Object Recognition

We focus on the setting used in Fanello et al. (2013), where a human teacher shows new objects to the iCub in order for the

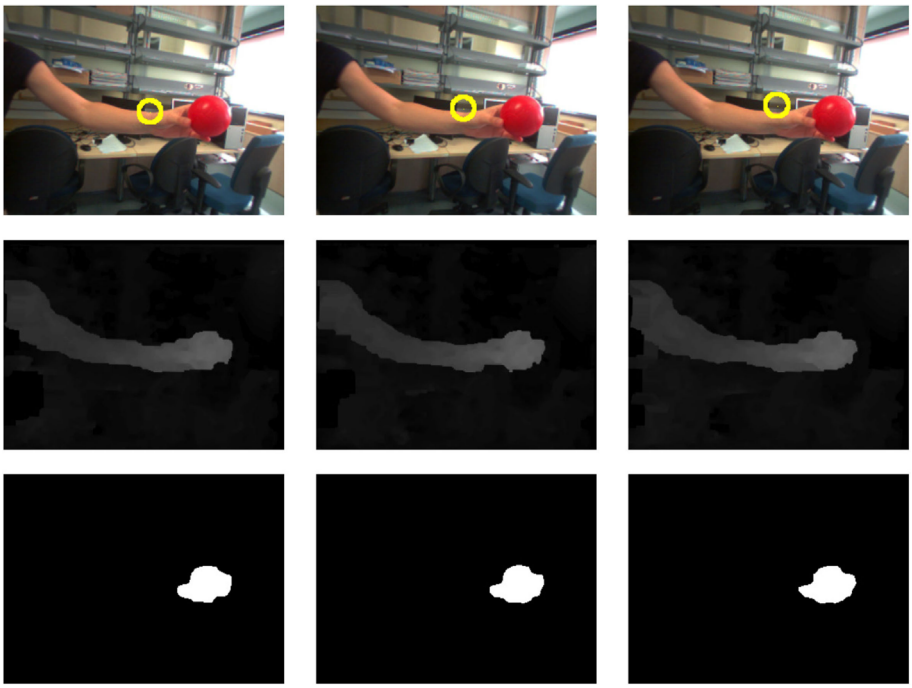

**FIGURE 4 | Frames extracted from the sequence represented in Figure 3 in the period from $t = 18$ s to $t = 20$ s, when the red ball tracker fails to detect its target**. Top: output of the tracker. Middle: disparity map. Bottom: disparity segmentation.

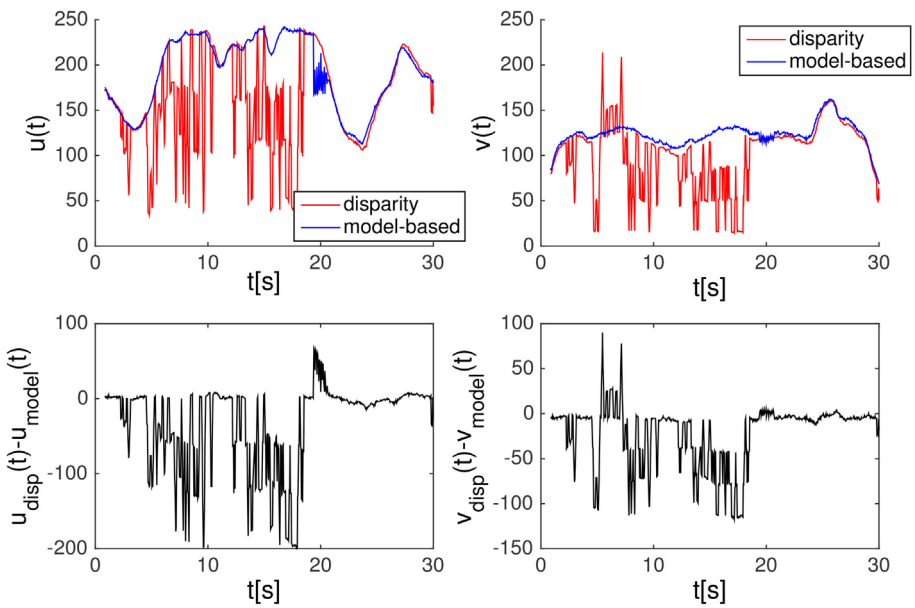


**FIGURE 5 | Similar to Figure 3, but in this case SGBM provides the disparity map**. The blue trace is the $(u,v)$ coordinate compute from the color tracker, while the red trace is obtained using the blob detector on the disparity map. Because of the low efficiency, the blob detector is computed off-line. For comparison in this experiment, we use the same sequence of **Figure 3**.

robot to focus its attention toward them and learn their visual appearance. Communication between the human and the robot occurs through speech, i.e., commands and object labels are verbally provided by the human teacher [see Fanello et al. (2013b) for more details and a throughout overview of the system]. We show how, by replacing the motion-based segmentation and tracking, used in Fanello et al. (2013b), with the disparity-based approach described in Section 4, we are able to remarkably improve the


**FIGURE 6 | Three frames (extracted from the attached video) showing the effectiveness of the proposed segmentation system**. Top: resulting crop in the left rectified frame, labeled by the operator using speech. Middle: disparity map. Bottom: segmented disparity blob, its centroid, and the enclosing ROI.

usability of the resulting application and the naturalness of the human-robot interaction.

In **Figure 6**, we report three frames extracted from three corresponding sequences, recorded while tracking three different objects following the disparity-based strategy previously described. The top row shows the output of the pipeline: the object (in this case a cup, a toy octopus, and a lemon squeezer) is localized in the scene using the ROI provided by the disparity segmentation module, the label being provided verbally by the human teacher. The middle row reports the associated disparity map by ELAS and the bottom row reports its segmentation, together with the centroid, that is used for the tracking (red dot, average over three frames; green dot, current frame) and the ROI, used for the segmentation (averaged over three frames to account for spurious mis-segmentations). The ROI is computed as the smallest rectangular region enclosing the segmented blob, with a margin of 20 pixels.

The full video, recorded from the iCub cameras while the robot was focusing on the objects to be learned and showing the different stages of the tracking pipeline, is available as Supplementary Material to this paper. Notice that the disparity-driven control of attention results in stable object tracking.

The uses of this tracking application can be multiple: first of all, we plugged it into our object recognition pipeline to teach new objects to the robot (code available in the iCub `onthefly-recognition` repository[7]). Then, we are currently employing it as a fast and natural method to collect large-scale annotated datasets of images containing objects "as seen by the robot," to be used to train/benchmark off-line visual recognition systems on the robot's visual experience. This is particularly useful because it allows to collect object recognition ground-truth by consistently reducing the effort of the manual annotation phase: indeed not only the label is provided verbally by the teacher but also the ROI around the object is automatically provided by the disparity segmentation. The acquisition application through which we are currently building a large-scale visual recognition dataset of objects part of the iCub's world is available online at the `iCubWorld`[8] repository.

We conclude that the strategy proposed in this report is a viable alternative to the motion-based tracker employed in Ciliberto et al. (2013), Fanello et al. (2013a,b) and Pasquale et al. (2015). Another advantage is that this strategy is more effective because it does not require the human teacher to continuously shake the object in front of the robot (as it is the case when motion is used instead). This results in a more natural interaction, where the user is free to move the object slowly or keep it still at all. The segmentation is more accurate, especially because it allows to improve the detection using low-pass filtering, and the quality of the acquired images is higher.

## 6.2. Object Exploration and Manipulation

Finally, we consider a setting in which the robot is standing in front of a table and uses disparity to distinguish separate objects. In this setting, by relying on LIBELAS disparity map, we could reconstruct the scene in front of the robot and the system could determine the optimal hand pose for a reliable grasp (Gori et al., 2014).

[7]https://github.com/robotology/onthefly-recognition

[8]https://github.com/GiuliaP/icubworld

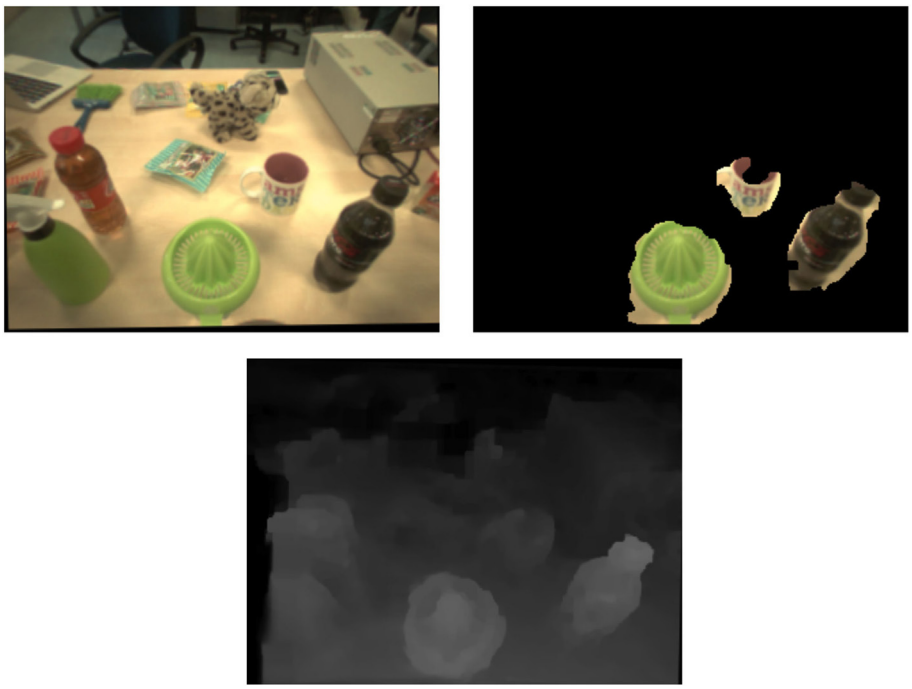

**FIGURE 7 | Top left: rectified frame recorded from the iCub's left camera while the robot was looking at a table in front of it**. Top right: segmentation of the three closest objects on the table obtained from the disparity map of the scene (bottom).

In **Figure 7**, we report the left rectified image (top left) and the corresponding segmentation (top right), obtained by putting a threshold on the disparity map (bottom). For the purpose of demonstration, such threshold was chosen manually; however, in a real application more sophisticated processing of the disparity map could applied to cluster 3D point clouds and better detect separate objects.

## 7. CONCLUSION

In this work, we have described the current system implemented on the iCub robot to perform depth estimation and how it benefits from the recent incorporation of the state-of-the-art disparity computation algorithm ELAS (Geiger et al., 2010). We have evaluated a few real applications of the information provided by the disparity map produced by ELAS to typical robotics settings. We experimentally demonstrated that this approach is computationally efficient and robust for the real-world scenario. The work presented in this report may be at the basis of more complex behaviors of the humanoid robotic system, such as interaction with the human or with the surrounding environment. The system described in this paper for depth estimation is made publicly available: it can be used as an off-the-shelf solution for the benefit of the whole iCub community and, with minor adaptations, on other robots.

## AUTHOR CONTRIBUTIONS

All authors have contributed to the conceptual design or development of the work described in this paper and participated in drafting and revising its content. They also approve publication of the paper and agree to be accountable for all aspects of the work described therein.

## SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at http://journal.frontiersin.org/article/10.3389/frobt.2016.00035

## REFERENCES


Baumgartner, T., Mitzel, D., and Leibe, B. (2013). "Tracking people and their objects," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Portland, OR: IEEE Computer Society), 3658–3665.

Bernardino, A., and Santos-Victor, J. (2002). "A binocular stereo algorithm for log-polar foveated systems," in *Biologically Motivated Computer Vision, Second International Workshop BMCV, Proceedings*, eds H. H. Bülthoff, C. Wallraven, S.-W. Lee, and T. A. Poggio (Berlin, Heidelberg: Springer), 127–136.

Bradski, G., and Kaehler, A. (2013). *Learning OpenCV: Computer Vision in C++ with the OpenCV Library*. 2nd Edn. O'Reilly Media, Inc.

Ciliberto, C., Fanello, S., Santoro, M., Natale, L., Metta, G., and Rosasco, L. (2013). "On the impact of learning hierarchical representations for visual recognition in robotics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Tokyo.

Ciliberto, C., Pattacini, U., Natale, L., Nori, F., and Metta, G. (2011). "Reexamining Lucas-Kanade method for real-time independent motion detection: application to the iCub humanoid robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA.

Einecke, N., and Eggert, J. (2010). "A two-stage correlation method for stereoscopic depth estimation," in *IEEE International Conference on Digital Image Computing: Techniques and Applications (DICTA)* (Sydney: IEEE Computer Society), 227–234.

Einecke, N., and Eggert, J. (2013). "Stereo image warping for improved depth estimation of road surfaces," in *IEEE Intelligent Vehicles Symposium (IV)* (Gold Coast, QLD: IEEE Computer Society), 189–194.

Einecke, N., and Eggert, J. (2015). "A multi-block-matching approach for stereo," in *IEEE Intelligent Vehicles Symposium (IV)* (Seoul: IEEE Computer Society), 585–592.

Fanello, S. R., Ciliberto, C., Natale, L., and Metta, G. (2013a). "Weakly supervised strategies for natural object recognition in robotics," in *IEEE International Conference on Robotics and Automation (ICRA)*. Karlsruhe.

Fanello, S., Ciliberto, C., Santoro, M., Natale, L., Metta, G., Rosasco, L., et al. (2013b). "iCub world: friendly robots help building good vision data-sets," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Portland, OR.

Fanello, S., Pattacini, U., Gori, I., Tikhanoff, V., Randazzo, M., Roncone, A., et al. (2014). "3D stereo estimation and fully automated learning of eye-hand coordination in humanoid robots," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)* (Madrid: IEEE Computer Society), 1028–1035.

Geiger, A., Lenz, P., and Urtasun, R. (2012). "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Providence, RI.

Geiger, A., Roser, M., and Urtasun, R. (2010). "Efficient large-scale stereo matching," in *Asian Conference on Computer Vision (ACCV)* (Queenstown: Springer), 25–38.

Goerick, C., Mikhailova, I., Wersing, H., and Kirstein, S. (2006). "Biologically motivated visual behaviors for humanoids: learning to interact and learning in interaction," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)* (Genova: IEEE Computer Society), 48–55.

Goerick, C., Wersing, H., Mikhailova, I., and Dunn, M. (2005). "Peripersonal space and object recognition for humanoids," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)* (Tsukuba: IEEE Computer Society), 387–392.

Gori, I., Pattacini, U., Tikhanoff, V., and Metta, G. (2014). "Three-finger precision grasp on incomplete 3D point clouds," in *IEEE International Conference on Robotics and Automation (ICRA)*, 5366–5373.

Greggio, N., Bernardino, A., Laschi, C., Santos-Victor, J., and Dario, P. (2011). Real-time 3D stereo tracking and localizing of spherical objects with the iCub robotic platform. *J. Intell. Robot. Syst.* 63, 417–446. doi:10.1007/s10846-010-9527-3

Hartley, R., and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge: Cambridge university press.

Hirschmuller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 328–341. doi:10.1109/TPAMI.2007.1166

Kumar, S., Odone, F., Noceti, N., and Natale, L. (2015). "Object segmentation using independent motion detection," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)* (Seoul: IEEE Computer Society), 94–100.

Leitner, J., Bernardino, A., and Santos-Victor, J. (2008). "A benchmark on stereo disparity estimation for humanoid robots," in *IEEE Conference on Autonomous Robot Systems and Competitions (ICARSC/Robotica)*. Aveiro.

Leitner, J., Chandrashekhariah, P., Harding, S., Frank, M., Spina, G., Forster, A., et al. (2012a). "Autonomous learning of robust visual object detection and identification on a humanoid," in *IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)* (San Diego, CA: IEEE Computer Society), 1–6.

Leitner, J., Harding, S., Frank, M., Forster, A., and Schmidhuber, J. (2012b). Learning spatial object localization from vision on a humanoid robot. *Int. J. Adv. Robot. Syst.* 9. doi:10.5772/54657

Li, C., Lu, L., Hager, G. D., Tang, J., and Wang, H. (2013). "Robust object tracking in crowd dynamic scenes using explicit stereo depth," in *Asian Conference on Computer Vision (ACCV)* (Daejeon: Springer), 71–85.

Lin, K. W., Lau, T. K., Cheuk, C. M., and Liu, Y. (2012). "A wearable stereo vision system for visually impaired," in *IEEE International Conference on Mechatronics and Automation (ICMA)*, 1423–1428.

Lyubova, N., Ivaldi, S., and Filliat, D. (2015). From passive to interactive object learning and recognition through self-identification on a humanoid robot. *Auton. Robots* 40, 33–57. doi:10.1007/s10514-015-9445-0

Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., et al. (2015). A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. *ArXiv e-prints*.

Menze, M., and Geiger, A. (2015). "Object scene flow for autonomous vehicles," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA.

Metta, G., Sandini, G., Vernon, D., Natale, L., and Nori, F. (2008). "The iCub humanoid robot: an open platform for research in embodied cognition," in *8th Work. on Performance Metrics for Intelligent Systems*. Available at: http://www.icub.org

Mitzel, D., and Leibe, B. (2012). "Close-range human detection for head-mounted cameras," in *British Machine Vision Conference (BMVC)*. Guildford.

Pasquale, G., Ciliberto, C., Odone, F., Rosasco, L., and Natale, L. (2015). "Teaching iCub to recognize objects using deep Convolutional Neural Networks," in *Proceedings of the 4th Workshop on Machine Learning for Interactive Systems, 32nd International Conference on Machine Learnin* (Lille), 21–25. Available at: http://www.jmlr.org/proceedings/papers/v43/pasquale15

Pattacini, U. (2011). *Modular Cartesian Controllers for Humanoid Robots: Design and Implementation on the iCub*. Ph.D. thesis, RBCS – Istituto Italiano di Tecnologia, Genova, IT.

Rudinac, M., Kootstra, G., Kragic, D., and Jonker, P. P. (2012). "Learning and recognition of objects inspired by early cognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vilamoura: IEEE Computer Society), 4177–4184.

Sinha, S. N., Scharstein, D., and Szeliski, R. (2014). "Efficient high-resolution stereo matching using local plane sweeps," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Columbus, OH: IEEE Computer Society), 1582–1589.

Taiana, M., Santos, J., Gaspar, J., Nascimento, J., Bernardino, A., and Lima, P. (2010). Tracking objects with generic calibrated sensors: an algorithm based on color and 3D shape features. *Rob. Auton. Syst.* 58, 784–795. doi:10.1016/j.robot.2010.02.010

Tombari, F., Gori, F., and Di Stefano, L. (2011). "Evaluation of stereo algorithms for 3d object recognition," in *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* (Barcelona: IEEE Computer Society), 990–997.

Van den Bergh, M., and Van Gool, L. (2012). "Real-time stereo and flow-based video segmentation with superpixels," in *IEEE Workshop on Applications of Computer Vision (WACV)* (Breckenridge, CO: IEEE Computer Society), 89–96.

Wersing, H., Kirstein, S., Götting, M., Brandl, H., Dunn, M., Mikhailova, I., et al. (2006). "A biologically motivated system for unconstrained online learning of visual objects," in *Artificial Neural Networks–ICANN 2006* (Athens: Springer), 508–517.

Wersing, H., Kirstein, S., Gotting, M., Brandl, H., Dunn, M., Mikhailova, I., et al. (2007). Online learning of objects in a biologically motivated visual architecture. *Int. J. Neural Syst.* 17, 219–230. doi:10.1142/S0129065707001081

Wersing, H., Kirstein, S., Schneiders, B., Bauer-Wersing, U., and Körner, E. (2008). "Online learning for bootstrapping of object recognition and localization in a biologically motivated architecture," in *Computer Vision Systems: 6th International Conference, ICVS, Proceedings*, eds A. Gasteratos, M. Vincze, and J. K. Tsotsos (Berlin, Heidelberg: Springer), 383–392.

**Conflict of Interest Statement:** The work described in this paper was conducted in the absence of any commercial or financial relationship that can be construed as a potential conflict of interest.