Object segmentation using Independent Motion Detection

Sriram Kumar[1,2], Francesca Odone[2], Nicoletta Noceti[2], Lorenzo Natale[1]

Abstract— **Independent motion detection aims at identifying elements in the scene whose apparent motion is not due to the robot egomotion. In this work, we propose a method that learns the input-output relationship between the robot motion – described by the position and orientation sensors embedded on the robot – and the sparse visual motion detected by the cameras. We detect independent motion by observing discrepancies (anomalies) between the perceived motion and the motion that is expected given the position and orientation sensors on the robot. We then perform a higher level analysis based on the available disparity map, where we obtain dense profile of the objects moving independently from the robot. We implemented the proposed pipeline on the iCub humanoid robot. In this work, we report a thorough experimental analysis that covers typical laboratory settings, where the effectiveness of the method is demonstrated. The analysis shows in particular the robustness of the method to scene and object variations and to different kinds of robot's movements.**

I. Introduction

Humanoid robots use vision as the primary source of information to interact with the environment. Motion cues are important for a variety of tasks, including *(i)* navigation, to avoid collision or self-localizing and *(ii)* foreground/background segmentation to detect objects and humans. Unfortunately, in humanoid robots, motion detection is made difficult by the fact that cameras are not stationary and the perceived visual motion is heavily affected by the robots' own movement during gazing or locomotion. Correct interpretation of the scene requires, therefore, to estimate the perceived scene motion and to separate the elements which are only apparently moving due to motion of the cameras (egomotion) from the ones that are moving independently (independent motion). This problem is usually referred to as *independent motion detection* (IMD [1], [2], [3], [4]).

Using the kinematics it is possible to estimate the visual motion of the scene based on the information provided by the motor encoders and the output of the inertial measurement units, when available. However a good kinematic model of the robot (including the placement of the inertial sensors) may be unavailable or difficult to derive, especially in presence of elastic elements at the joints (a situation that is quite common in walking humanoid robots). In this case it may be more practical to learn a model that predicts the visual motion that is induced by the robot's own movement.

[1]S. Kumar, L. Natale are with iCub Facility, Istituto Italiano di Tecnologia, Genova, Italy {sriram.kishore, lorenzo.natale} at iit.it

[2]S. Kumar, F. Odone, N. Noceti is with the Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi, Università degli Studi di Genova, Genova, Italy {francesca.odone, nicoletta.noceti} at unige.it

Our model learns the relationship between the actual robot velocity and the apparent motion estimated on the image frames. We assume that a global model is sufficient to represent the observed motion independently of the scene structure and that variability due to parallax can be captured by the flow statistics.

In this work we build on an early implementation of the method [5] and extend it to improve the quality of the segmentation and the robustness in various scenarios.

In the learning phase, we gather representative data by letting the iCub move randomly while it is observing a static environment. At run-time, we detect anomalies in the estimated optical flow, i.e., points which do not seem to move coherently with the robot's velocity. Such anomalies form a sparse description of the independent motion. Finally, we obtain a dense segmentation by combining anomalous points with an appropriately quantized disparity map.

Our algorithm has been implemented and validated on the iCub humanoid platform [6]. We provide a thorough experimental analysis of the method performances for different types of robot's movements, independent motions and scenes.

A. iCub Vision System

The iCub vision system consists of a pair of Dragonfly cameras mounted over an eye mechanism with 3 DoF. The cameras have a resolution of 320x240px and can stream at 33 Hz frame rate. The eye mechanism can pan, tilt and version (both eye pan independently based on depth of the fixation point). Eyes are mounted on a neck mechanism with 3 DoF (roll, pitch and yaw) which is further actuated by a 3DoF torso [7], [6].

In this work we use the the GazeControl software which coordinates the robot's eyes, neck and torso as explained in [8]. The observed egomotion is an outcome of pure rotation of the motors of the eyes and/or in combination with the rotation of those of the neck (in the experiments in this paper we do not actuate the torso). iCub is equipped with motor joint encoders and inertial sensors (accelerometer, gyroscope, magnetometer). In absence of external perturbations acting on the robot, egomotion can be estimated using the inertial measurements and/or by differentiating the encoder feedback.

The GazeControl coordinates the DoF of the robot to obtain a human-like coordinated movement. It allows moving the robot head by asking the it to gaze at random fixation points in the 3D world and obtain various velocity profiles. The following scenarios can be generated through GazeControl: (1) stationary neck with moving eyes, (2) coordinated

movement between the neck and eyes to achieve fixation in same direction (small velocity), (3) moving neck and moving eye in opposite direction (for stabilization), (4) moving neck and moving eye in random direction. Other combinations (like random neck and eye movements) are possible but unnatural and have not been considered in our analysis.

II. Related works

In the earlier methods, the egomotion was modeled from the scene only. Irani et al. [1] combined 2D affine transformation for 2D scenes with parallax analysis for 3D scenes in a unified approach for intermediate scenes. Nelson et al. [2] proposed two complementary approaches: constraint ray filtering which constraints the velocity of the scene due to moving camera and animate motion detection which constraints the rate of change of the projected velocity to detect rapid movements. Argyros et al. [3] used LMedS to find the model parameters of stereo and optical flow. Jun et al. [4] computed the egomotion by reducing the error of bilinear transformation on optical flow positions on a subset of feature points. MotionCUT [9] inspects the locations where optical flow is disrupted by background occlusions and disocclusions generated by moving object(s). These methods estimate egomotion from the assumption that moving objects occupy a larger portion of the scene and may fail otherwise.

One way to resolve the dominant motion problem is to use the robotic motor encoder values and the scene together. Currently, two such methods have been proposed for the iCub robotic platform and we outlined them here:

1) Heteroscedastic Approach to Independent Motion Detection (HIMD) [10] relies on a 3D calibrated stereo system and assumes knowledge of the robot's kinematics. The method compares the tracking of sparse points with the projected movement of corresponding 3D points transformed using the robot kinematic model. This method performs well in the presence of many outliers, occlusions and cluttered background.

2) Learning egomotion with sensor data [5] method is a simplified version of HIMD, where the robot kinematics is not available and, instead of a full 3D vision system, a 2D estimate of apparent motion (optical flow) is exploited. The method models egomotion by learning the relationship of the observed optical flow statistics and the robot's motor velocities; then it detects anomalies by thresholding the Mahalanobis distance of the optical flow estimated at runtime and the learned model. The assumption in this case is that egomotion is dominated by camera rotation and that objects in the scene are located far away from the robot. With respect to HIMD, this simplified version is faster and, as demonstrated empirically, able to deal with most practical circumstances. In this paper we build on the latter approach where we enrich the learned motion model with additional information. We extend the input vector to the motion predictor to include inertial data. We also use depth information in the processing to improve the result of the segmentation. Finally, we perform an extensive experimental validation to validate empirically to what extent the learned model manages to capture the variability of the observed motion.

III. The method

The framework for independent motion detection we propose and evaluate is illustrated in Fig. 1. It is composed of 5 steps: *(A)* data generation, *(B)* training (egomotion learning), *(C)* testing (sparse anomaly detection), *(D)* post-processing, *(E)* dense segmentation. Training and testing follow [5].

A. Data generation

We now discuss how we obtain data from the robot sensor, encoder values and flow statistics from image pairs.

1) Motion generator: We use the iCub GazeControl to generate different head configurations and various velocity profiles of the head joints; to do this we randomly sample a predefined rectangular 3D area in the workspace to select targets for the controller. This allows us to sample effectively the input space and thus to produce a representative set of data to be fed to the machine learning algorithm.

As mentioned in Sec. III-A.2 there are certain cases in which the head is controlled to produce opposite motion of the eyes and the neck joints. This happens because the eyes are faster than the neck to achieve target fixation; therefore after the eyes have achieved fixation the remaining part of the motion is dedicated to move the neck to orient the head to face the target while the eyes compensate this movement to maintain stable fixation. This is visible, for example, in Fig. 2: $V1$ represents the horizontal velocity of the head and $V4$ represents the horizontal velocity of one of the eyes (both along the X-axis, in reference to Cartesian coordinate system of the robot). To stabilize the eyes while the neck moves, the controller maintains the relation $V4 = -V1$. A similar relation is enforced between neck and eye along the vertical direction (the Y-axis), i.e. $V5$ and $V2$. These are special cases for our motion model, since the motor velocities are not null, but the observed apparent motion is zero because the acquired images are identical. Examples of these relationships need to be included in the training set, but would be very unlikely to be explored if training was purely random in the space of the velocity of the individual joints.

2) Velocity vector: Velocity vector $\dot{\mathbf{q}}_t$ is a vector representation of velocity motor encoders and sensors values that represent the egomotion of the robot at time t. Method [5] uses the iCub motor encoders as the input velocity vector. Apart from the motor encoders, iCub is equipped with inertial sensors. Our velocity vector comprises of *(i)* velocity of 3 joint encoders of the eyes, *(ii)* velocity of 3 joint encoders of the neck, *(iii)* linear acceleration and difference of Euler angle values of inertial sensor places on the head. Different combinations of these quantities are used to test their effectiveness: 6D (speed of motor encoders, i.e. $d = 6$), 9D (speed of motor encoders and linear acceleration, i.e. $d = 9$), 12D (speed of motor encoders, linear acceleration and difference of Euler angle values, $d = 12$) and 9Di (eye motor encoders, linear acceleration and difference of Euler

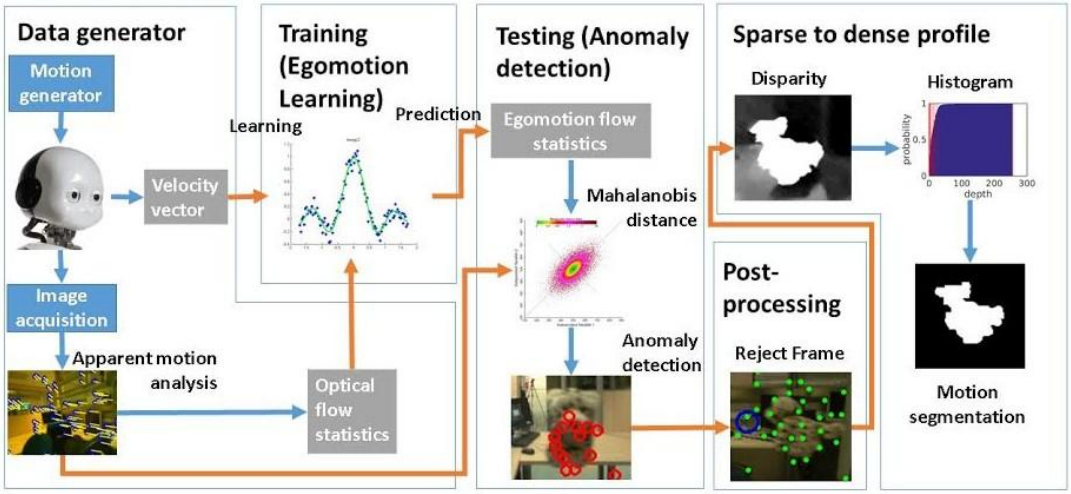


Fig. 1: Pipeline of independent motion detection algorithm.

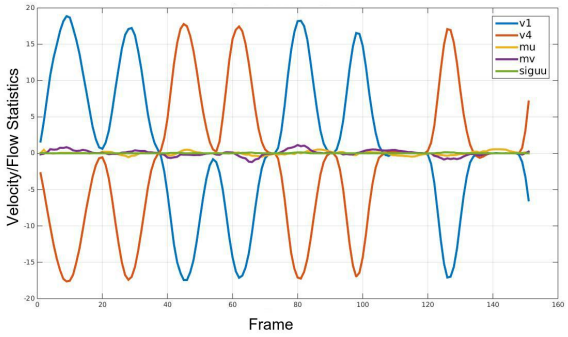


Fig. 2: Correlation between neck and eyes motion. This figure shows thats the GazeControl joint movements are not independent, in particular here we show the the stabilization effect when the neck and the eyes are moved in opposite direction (notice that the mean and standard deviation of the optical flow components m_u, m_v, sig_{uu} are close to zero).

angle values, $d = 9$). These results are discussed in Sec. IV-A

3) Image acquisition: RGB images of the stereo camera pair are captured by the iCub vision system. We use a single (left) camera to learn the egomotion and the stereo pair for computing the disparity map which is later used for dense independent motion segmentation.

4) Apparent motion analysis: We consider two consecutive images I_{t-1} and I_t. We compute a sparse optical flow following a very standard procedure: we extract Harris corner points from each image [11] $z_i(t) = (x_i(t), y_i(t))$, with $i = 1$ to N.

Each corner $z_i(t+1)$ detected in image I_t is mapped to its corresponding point $z_i(t)$ using KLT optical flow technique [12] i.e. $v_i(t) = z_i(t) - z_i(t+1)$. The quality of the obtained results depend on a number of factors, as we will discuss in the experimental analysis. If the robot moves at a high speed the effects of motion blur and inadequate frame rate could influence negatively on the estimated motion.

From the set of optical flow vectors for a frame I_t we obtain a global optical flow statistics vector μ_t, Σ_t, describing optical flow mean and variance.

B. Training phase or Egomotion Learning

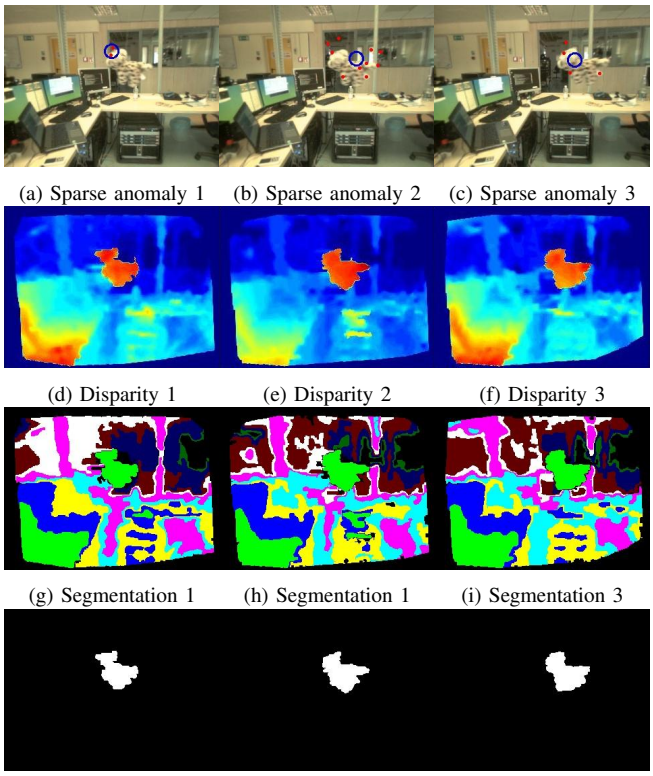
We learn a vector-valued input-output relationship $f : X \rightarrow Y$ with $X \subseteq \mathbb{R}^d$ and $Y \subseteq \mathbb{R}^p$. A velocity vector $\dot{\mathbf{q}}_t \in X$ describing the robot movement represents the input of the learning algorithm. In this paper, we investigate the cases in which $d = 6, 9, 12$. The computed global optical flow statistics $\mathbf{p}_t \in Y$ is the output of the learning algorithm, in this work $\mathbf{p}_t = (\mu_u, \mu_v, \sigma^2_{uu}, \sigma^2_{uv}, \sigma^2_{vv})$, thus $Y \subseteq \mathbb{R}^5$. For the learning algorithm, we adopt vector-valued RLS [13] with an RBF kernel — implementation available in the GURLS library [14].

C. Anomaly detection

In testing phase, multiple objects could be moving independently. The velocity vector of the robot is given as input to the learning model and the flow statistics are predicted for the egomotion. The independent moving objects can be distinguished from the egomotion by thresholding the Mahalanobis distance between the computed optical flow and the predicted egomotion flow statistics. More details can be found in [5].

D. Post processing

In this step, we discard isolated frames where more than 80% of their key-points are detected as anomalies. There are at least two possible causes for this to happen. First, the learned model may not be able to generalize to the observed scene because the latter is not well represened in the training set. Second, when the iCub is starting to accelerate its head, the abrupt change in the motion model is in contrast with the smoothing effect of the regularizer in the regression model. On an average 8 frames are rejected in 100 frames when the robot is moved with maximum speed that is 30% of the limit; but this can increase if the egomotion is larger. In future works, we will address these problems by means of



(a) Sparse anomaly 1 (b) Sparse anomaly 2 (c) Sparse anomaly 3

(d) Disparity 1 (e) Disparity 2 (f) Disparity 3

(g) Segmentation 1 (h) Segmentation 1 (i) Segmentation 3

(j) Dense Profile 1 (k) Dense Profile 2 (l) Dense Profile 3

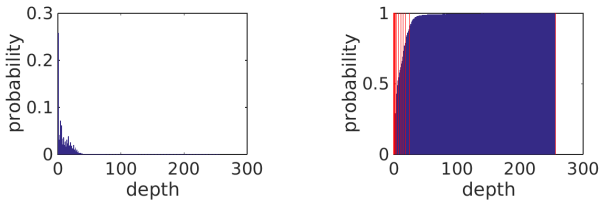
Fig. 3: Dense Anomaly segmentation steps (Blue circles represent the center of mass).

a more complex scene modeling and by exploring a larger part of the input space with online learning.

E. Sparse to dense segmentation

So far apparent motion is estimated only at the corners and therefore the detected independent moving objects are sparse. We now observe that we can exploit complementary visual information, i.e. the disparity map (we use the software described in [15] which computes a robust dense disparity map using the approach proposed in [16]).

We first quantize the disparity map into set of bins computed from equally split cumulative histogram. Then we find the segment(s) containing anomalies. To contrast the effect of anomalies lying at object boundaries, we consider only 50% of those which are close to the center of mass. Fig. 3 shows three consecutive RGB frames with sparse



(a) Average (b) Cumulative

Fig. 4: Depth probability distribution.

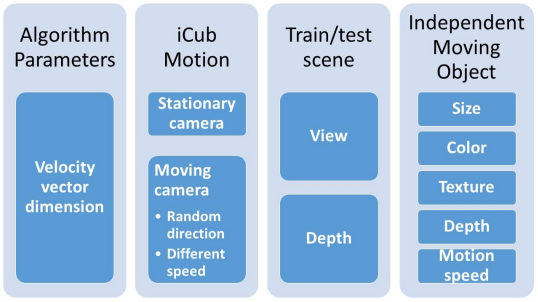


Fig. 5: Various train/test scenarios

anomalies and their disparity maps, disparity quantization and dense motion segmentation.

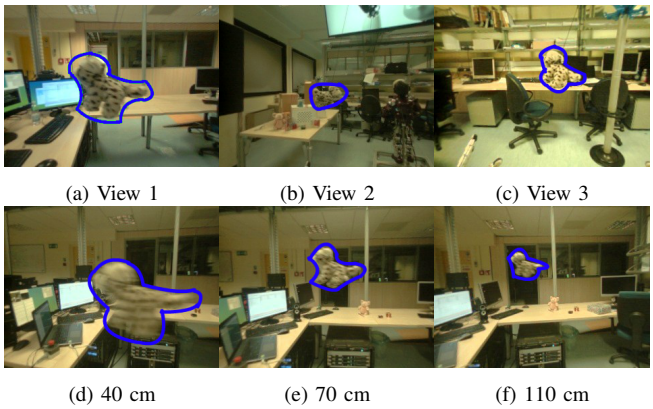
An analysis of the average depth distribution (Fig. 4a) reveals that most of the pixels are located far away from the camera; which means that the probability of detecting the points far away is higher. If equal sized bins are used to quantize the disparity map, then the moving object might be split between two or more depths. To get a balanced quantization of the disparity map, we compute the histogram bins size by equally splitting the cumulative histogram in 0.1 size portions (Fig. 4b).

IV. Experimental analysis

To our knowledge, there is no available benchmark which provides appropriate scenarios for our analysis. We acquired sets of data in-house – Fig. 5 shows different configurations in the training and test sequences used to explore various parameters configuration and how the algorithm performs in different situations. Fig. 6 shows the different test environments, which depict a typical laboratory setting with various depths and degrees of complexity/clutter. The object boundaries of moving objects in the testing sequence (200 images) are annotated using LabelMe annotation [17]. The training and test sequences are captured at 11 fps. The drop of frame rates is due to the computation time dedicated to calculating the disparity map.

In the learning stage parameters, we set the regularization parameters with 10 fold cross validation (provided by GURLS libraries). To explore the input space completely, 3000 data of velocity vector (motor encoders and inertial sensors) as input and flow statistics as output are collected.

Performance of the method is reported in terms of ROC curve. In a ROC Curve, the True Positive (TP) detection is plotted against the False Positive (FP) detections. In our case, these values are computed at point and frame level. To calculate the ROC Curve at frame level, a frame is considered as TP if more than 80% of the points are TP, the same for FP, False Negative (FN), True Negative (TN). The frame level ROC Curve shows how considering the points of a frame collectively helps in making a decision. The two measures have shown to be very coherent (since one is derived from the other). Thus, for space reasons, most experiments in the section are described in terms of frame based ROC.



(a) View 1 (b) View 2 (c) View 3

(d) 40 cm (e) 70 cm (f) 110 cm

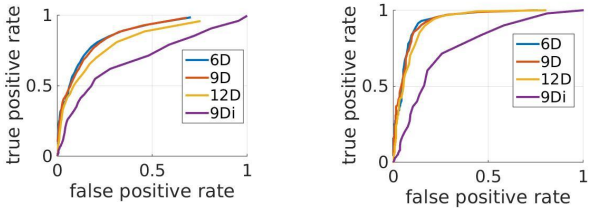
Fig. 6: Test scenarios. Objects with different views: (a),(b),(c); Objects with different depths: (d), (e), (f)

A. *Different input dimensions*

In reference with Sec. III-A.2, the comparative study of using different combinations of input space for modeling egomotion is performed and Fig. 7 shows point-based and frame-based ROC curves. From these plots, we can derive the following observations. First: performance in the case 6D, 9D and 12D are similar. On the other hand, detection rate remains acceptable when inertial input is used instead of the neck motor velocity. This suggests that the inertial input does carry useful information, however, in our experimental scenario it was redundant with respect to the one carried by the motor encoders. Inertial input may turn out to be critical in cases in which motor encoders are not informative, i.e. during locomotion or in presence of external disturbances (these situations will be explored in the future).

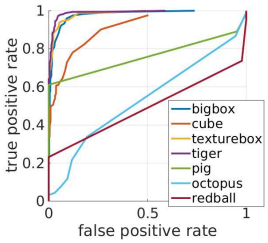
B. *Object variance*

The purpose of this test is to see whether the algorithm has any bias towards objects of different appearance. The appearance of objects affects the corner detection density, the accuracy of optical flow and disparity maps. For this test, seven different objects having different size, color and texture are used (see Tab. I). All the moving objects are placed at a constant distance 70 cm away from the robot. Fig. 8a shows the frame level analysis of these objects. Since texture is very important for corner detection, pig, red ball and octopus are the most challenging and the overall performances are rather

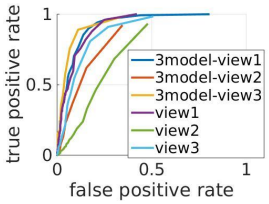


(a) Point based ROC Curve (b) Frame based ROC Curve

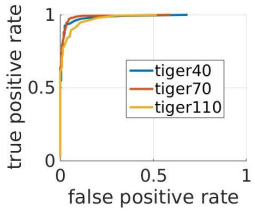
Fig. 7: Comparison of ROC curves when the learned model uses different input velocity vectors.



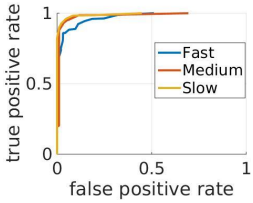
(a) Frame based ROC Curve when different objects are used.



(b) Frame based ROC Curve when an object move against a different scene structure.



(c) Frame based ROC Curve when an object move at different distance.



(d) Frame based ROC Curve when an object moves at different speed with small egomotion.

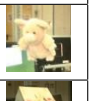
Fig. 8: ROC Curves

poor. Objects presenting a relatively rich texture, no matter the size, have instead been detected accurately.

C. *Different scene view*

In this case, the framework is trained and tested at different scenes (i.e. different environments) as shown in Fig. 6a, 6b, 6c. The three views which are selected, have different and complex depth scenarios in the scene. We learned 4 different models: one was trained including all views collectively, the other three were trained on each individual view. We then tested each view against its corresponding model, as well as the collective model. A collective trained model of all three views was trained with 5000 data and

TABLE I: Object Characteristics

	Size	Color	Texture	Image
Rubik cube	Small	Multi	Yes	
Octopus	Small	Blue	No	
Red ball	Small	Red	No	
Tiger	Medium	Grey	Yes	
Textured box	Medium	Black&White	Yes	
Pig	Medium	Pink	No	
Big box	Large	White	No	

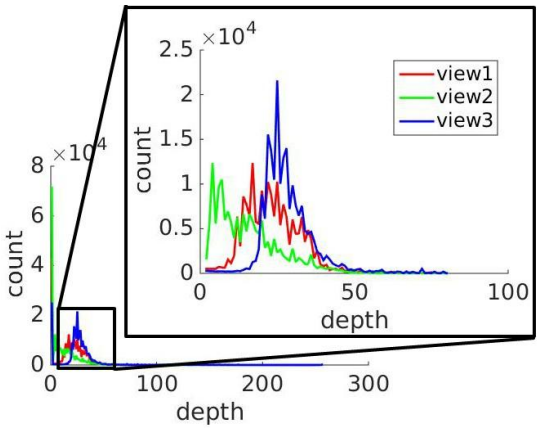
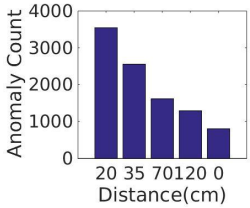
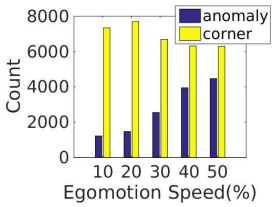


Fig. 9: Depth histogram of 100 frames from 3 different views



(a) Detected anomaly count. In this case the head moves horizontally and a stationary object is placed at increasing distance. The point at zero distance corresponds to the control situation in which the object is absent.



(b) False positive and corner count for different head speed without independent motion. Head speed limit is expressed as % of maximum motor speed and the anomaly count in multiples of ten.

Fig. 10: False positives analysis

test results against the three conditions separately (3model-view1, 3model-view2 and 3model-view3, respectively). The frame based ROC curves are shown in Fig. 8b. Notice that all models produced lower performance when tested against the scenario view2 (including the model tested collectively on all three views). This is because view2 is a more challenging scenario in which depth has higher variability than view1 and view3 (see depth histograms in Fig. 9). From this figure it is evident that for view2 the depth is distributed from 0 to 50 with a large variance compared to view1 and view3. In future work, we will include representation of the scene structure in the learned model so to capture larger variability in the observed optical flow.

D. Scene structure variance

In support of Sec. IV-C, a further scene structure variance experiment is conducted. In this experiment the system is trained on view1 and tested on a stationary object placed at different depths while the robot moves horizontally. In this case, there is no independent motion and the system should detect no anomalies. In Fig. 10a we plot the false positive count as a function of the different depth of the object in the scene. As the object is introduced in the scene at a particular depth, the average depth of the test sequence differs from the training sequence. The only exception is when distance is zero, in which the object is absent (the testing and training sequence are the same). The difference between the testing and training sequences decreases as the distance increases and it is observed that the number of false positive decreases. This shows that the average depth in the scene of test data should be similar to that of the training data.

E. Depth variance of independent motion

In this experiment, the system is trained on view1 and tested on objects which are moved at different depths from the robot as shown in Fig. 6d ($40\ cm$), Fig. 6e ($70\ cm$) and Fig. 6f $110\ cm$. The performance measure in Fig. 8c shows that, irrespective of the depth, the algorithm performs well on a well trained data set. So, the algorithm is invariant to a considerable depth in the scene provided that the average variance of the test scene is approximately equal to that of the training set.

F. Motion variance in speed

Humans have a tolerance limit of the speed of independent moving object and the egomotion at which the independent motion is detected [18], [19]. A similar tolerance check is conduced on the robot.

1) Independent moving object: This experiment is performed to test how well the independent motion detection performs, by moving an object at different speeds before the camera with slow egomotion (10% of the maximum speed limit). The result observed in Fig. 8d shows that on increasing the speed of independent moving object it becomes harder to detect the anomalies.

2) Egomotion: In this experiment, the robot is moved horizontally at a constant speed with no observed independent motion in the scene. This experiment is conducted to explore the minimum and maximum egomotion that the algorithm can withstand by moving the robot at different speed. The result seen in Fig. 10b, shows that as the speed of egomotion increases the anomaly count increases. This is because the camera moves fast and the pixels get blurred. This in turn results in less textured scene and detection of fewer corners. In this case the optical flow does not approximate the motion field well which results in more anomalies being detected.

G. Frame level inspection

In this experiment, we investigate the relation between the anomaly detection, depth image and ground truth annotation to understand the reason behind some of the incorrect detections. Fig. 11 shows the detection for 20 frames image sequence and 2 situations in which motion detection failed. Further inspection shows that in frames 17, 18 the object is correctly segmented using depth but no anomalies are detected because the object is stationary (motion goes to zero because the object is about to invert the direction of movement). This situation is responsible for 6% of the false negatives and is indeed due to incorrect ground truth/labeling. In frame 6, a similar situation occurs. In this case, however, the object is incorrectly segmented using depth. This is due to

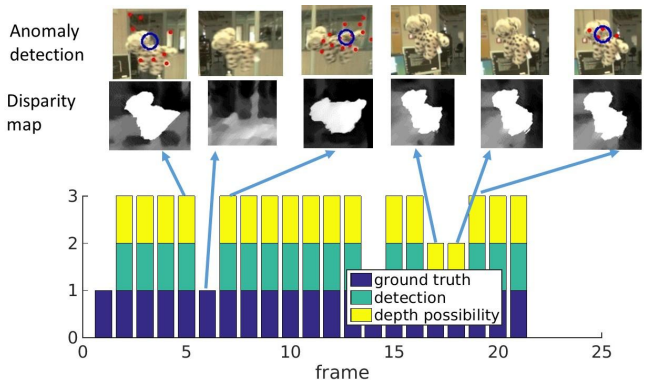


Fig. 11: Comparison of anomaly detection, annotation and disparity map to understand frame level detections

the image blur generated by the movement of the object. This is again supported by the results of the experiment reported in Sec. IV-F.1.

V. Conclusion

Independent motion detection allows exploiting motion cues while the robot is moving. This is a fundamental capability that simplifies the visual interpretation of the scene to detect moving obstacles. In this work, we considered a sparse representation of the scene (a set of corner points) and the robot sensor values to learn the egomotion; we then complement the sparse model with a dense representation of the scene (disparity maps) to segment the object boundaries of independently moving objects. The contribution of this paper are: *(i)* the identification of a set of experimental scenarios that represent a typical indoor setting, *(ii)* the construction of a model able to exploit a complex set of sensor information available on the robot, both for learning the egomotion and for identifying anomalies, *(iii)* a simple and efficient computational process to extract the object contour from anomalous points and depth map. Our method was able to perform well on rich textured objects provided the speed of the robot and the structure of the environment do not differ between the training and testing conditions. The method described in this paper assumes that egomotion is dominated by camera rotation and that flow statistics can model variability due to parallax. This assumption works well in our experimental conditions but may be too restrictive in those cases in which egomotion is induced by external perturbations or is due to complex whole-body movements (i.e. walking or balancing). We will address these issues in future works by incorporating depth and inertial information into the learned model.

References

[1] M. Irani and P. Anandan, "A unified approach to moving object detection in 2d and 3d scenes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 6, pp. 577–589, 1998.

[2] R. C. Nelson, "Qualitative detection of motion by a moving observer," *International journal of computer vision*, vol. 7, no. 1, pp. 33–46, 1991.

[3] A. A. Argyros and S. C. Orphanoudakis, "Independent 3d motion detection based on depth elimination in normal flow fields," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE, 1997, pp. 672–677.

[4] B. Jung and G. S. Sukhatme, "Detecting moving objects using a single camera on a mobile robot in an outdoor environment," in *International Conference on Intelligent Autonomous Systems*, 2004, pp. 980–987.

[5] S. R. Fanello, C. Ciliberto, L. Natale, and G. Metta, "Weakly supervised strategies for natural object recognition in robotics," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4223–4229.

[6] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The icub humanoid robot: an open platform for research in embodied cognition," in *Proceedings of the 8th workshop on performance metrics for intelligent systems*. ACM, 2008, pp. 50–56.

[7] R. Beira, M. Lopes, M. Praga, J. Santos-Victor, A. Bernardino, G. Metta, F. Becchi, and R. Saltarén, "Design of the robot-cub (icub) head," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 94–100.

[8] U. Pattacini, "Modular cartesian controllers for humanoid robots: Design and implementation on the icub," Ph.D. dissertation, Ph. D. dissertation, RBCS, Italian Institute of Technology, Genova, 2011.

[9] C. Ciliberto, U. Pattacini, L. Natale, F. Nori, and G. Metta, "Reexamining lucas-kanade method for real-time independent motion detection: Application to the icub humanoid robot," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 4154–4160.

[10] C. Ciliberto, S. R. Fanello, L. Natale, and G. Metta, "A heteroscedastic approach to independent motion detection for actuated visual sensors," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 3907–3913.

[11] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.

[12] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. IEEE, 1994, pp. 593–600.

[13] L. Baldassarre, L. Rosasco, A. Barla, and A. Verri, "Multi-output learning via spectral filtering," *Machine Learning*, 2012.

[14] A. Tacchetti, P. K. Mallapragada, M. Santoro, and L. Rosasco, "Gurls: a least squares library for supervised learning," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 3201–3205, 2013.

[15] S. R. Fanello and G. Pasquale, "Stereo vision," https://github.com/robotology/stereo-vision, 2015.

[16] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Asian Conference on Computer Vision (ACCV)*, 2010.

[17] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *International journal of computer vision*, vol. 77, no. 1-3, pp. 157–173, 2008.

[18] O. Kawakami, Y. Kaneoke, K. Maruyama, R. Kakigi, T. Okada, N. Sadato, and Y. Yonekura, "Visual detection of motion speed in humans: spatiotemporal analysis by fmri and meg," *Human brain mapping*, vol. 16, no. 2, pp. 104–118, 2002.

[19] L. Ferman, H. Collewijn, T. Jansen, and A. Van den Berg, "Human gaze stability in the horizontal, vertical and torsional direction during voluntary head movements, evaluated with a three-dimensional scleral induction coil technique," *Vision research*, vol. 27, no. 5, pp. 811–828, 1987.