Object Edge Contour Localisation based on HexBinary Feature Matching

Yuan Liu, Gerardo Aragon-Camarasa and J. Paul Siebert

Abstract— **This paper addresses the issue of localising object edge contours in cluttered backgrounds to support robotics tasks such as grasping and manipulation and also to improve the potential perceptual capabilities of robot vision systems. Our approach is based on coarse-to-fine matching of a new recursively constructed hierarchical, dense, edge-localised descriptor, the HexBinary, based on the HexHog descriptor structure first proposed in [1]. Since Binary String image descriptors [2]–[5] require much lower computational resources, but provide similar or even better matching performance than Histogram of Orientated Gradient (HoG) descriptors, we have replaced the HoG base descriptor fields used in HexHog with Binary Strings generated from first and second order polar derivative approximations. The ALOI [6] dataset is used to evaluate the HexBinary descriptors which we demonstrate to achieve a superior performance to that of HexHoG [1] for pose refinement. The validation of our object contour localisation system shows promising results with correctly labelling $\sim 86\%$ of edgel positions and mis-labelling $\sim 3\%$.**

I. INTRODUCTION

Simultaneous detection and localisation of object boundaries in images is a fundamental requirement to support robotics tasks such as grasping and manipulation. Currently, the standard approach is to localise the centroid, or some reference point, of object of interest within a bounding box [7]–[10], which specifies an approximate object position, but does not afford any explicit edge contour information. A number of researchers [11]–[13] have reported investigations into object localisation that affords edge contour labelling based on shape, or contour, model learning. Not only is edge contour information important in direct robotics interation tasks, numerous reports [8], [14]–[16] also indicate that edge contour information can capture crucial shape information that plays an essential role in visual advanced perception. Accordingly, in this paper we also employ object edge contour information, from which we construct a new hierarchical hexagon-based binary descriptor. Based on this descriptor we present a complete framework for pixel-level localisation of the objects edge contours.

Local feature extraction has been explored extensively in the field of computer vision and can be generally divided into two dominant methods: Classical approaches to local feature extraction derive from the orientated gradient histogram [17]–[20], generated from a local patch represented by a histogram of quantised gradient orientations weighted by their corresponding gradient magnitude values. SIFT [17] has served as a standard benchmark for evaluating local feature performance because of its good performance in many computer vision applications and widespread availabilty. While SURF [21] was developed to improve the computational efficiency of the feature extraction process by employing Haar-wavelet filters, efficiently implemented by means of *integral images*. PCA-SIFT [22] was then proposed to achieve faster matching by reducing the descriptor dimensions from 128 to 36 elements via Principal Components Analysis. Computational cost has been an issue with these descriptors for real-time/on-line robotics applications. Initially to address computation efficiency, Binary String descriptor (BS) have recently been devised and intensively explored since their first inception, BRIEF [2], appeared. BS descriptors are generated by computing pairwise intensity comparisons within a local sampling pattern and have been demonstrated to exhibit lower computation and storage requirements and to improve feature matching properties. ORB [3] extends BRIEF by coupling with the orientated FAST keypoint detector [23]. This descriptor exploits the intensity centroid to measure the keypoint orientation, according to which, a steered and hence in-plane rotation invariant BRIEF descriptor is generated accordingly. In BRISK [4], the set of compared point-pairs sampled within each local descriptor window are arranged within certain sampling configurations, typically exhibiting polar geometry (i.e. rotational symmetry) to facilitate rotational invariance, as adopted by DAISY [24]. The FREAK [5] descriptor utilises a retina-like sampling pattern for selecting compared pairs of image intensities. In this scheme image data is sampled using a Gaussian window that increases exponentially in size with the radial distance from the descriptor centre to avoid aliasing as the spatial distance between sampling points increases. The similarity between these BS descriptors can be computed efficiently by means of their Hamming distance. While these binary-based descriptors have made a considerable contribution to improving local feature extraction execution rates and matching performance rates, they have not yet been utilised to address the task of directly encoding object edge contours.

There has been growing interest in hierarchically grouped descriptors within the computer vision community [25]–[28] where local descriptors, used to express the local parts of an object, are successively combined to form new, combined, descriptors. These hierarchically grouped descriptors are potentially not only more distinctive, but also provide a mechanism for capturing the topological relationships between object fragments, which in turn opens the possibility

The authors acknowledge financial support from the Chinese Scholarship Council, China, and the European Union within the Strategic Research Project Clopema, Project No. FP7-288553.

The authors are with the School of Computing Science, University of Glasgow, United Kingdom, G12 8QQ. E-mail: y.liu.3@research.gla.ac.uk, gerardo.aragoncamarasa@glasgow.ac.uk, paul.siebert@glasgow.ac.uk.

of recognising an object by analysing the (grouped) parts of the object. HexHoG [1] is a recursively constructed hierarchical descriptor employing seven HoG descriptors spatially grouped in a hexagonal configuration shown in Fig. 1, which samples image data only at edge contour locations. In this paper, we also employ the hierarchical hexagonal grouping framework of HexHoG but investigate substituting the computationally expensive HoG descriptors with less expensive and potentially more robust BS descriptors to construct a new feature we term HexBinary. Hierarchical descriptor grouping serves two underlying objectives: firstly, to avoid fixed hand-configured descriptors groupings and instead generate groupings driven by the underlying data being represented and matched. Secondly, by constructing the descriptor by recursively grouping vectors generated by relatively small "base" descriptors, a hierarchy of descriptors can be constructed to allow coarse-to-fine descriptor matching strategies, and also different levels of visual "concept" to be clustered, e.g. by means of VQ.

Accordingly, the principal contributions of this paper comprise a new binary sampling scheme explored based on sign encoding first and second order derivatives in polar coordinates (applied to both Gaussian and Laplacian filtered image data) to form the HexBinary descriptor, and a framework for object edge contour localisation by means of coarse-to-fine edge feature matching. We present both the qualitative and quantitative results for pixel-level edge contour localisation using the ALOI [6] dataset.

II. APPROACHES

In this section, a complete framework for object contour localisation is introduced in detail, based on dense local edge matching by our new hierarchical descriptor HexBinary, introduced in Subsection A. In Subsection B we then describe our pose refinement process using the HexBinary descriptor. Finally based on the refined pose estimation, we describe in Subsection C the object contour localisation process required to localise object edge contours both directly and accurately.

A. The HexBinary Descriptor

The hierarchical hexagonal grouping configuration combined with HoG fields [1] has been demonstrated to give a viable degree rotation invariance and sufficient matching reliability to achieve pose estimation refinement. Several different binary string descriptor sampling configurations [2]–[5] have been reported, ranging from regular symmetric to randomly sampled. The binary bit computed by comparing the sign of difference in the intensities pairs of sampling points in effect encodes the sign of the first order derivative. Intuitively, this encoding mechanism captures the relative local spatial configuration of light and dark in the local image region sampled by the descriptor. Furthermore, if we compute the sign of the second order differences, this would correspond to the local spatial configuration of the intensity gradients. In order for the descriptor to sample efficiently, we proposed to utilise point-wise differences that correspond to approximations of the orthogonal first and second polar

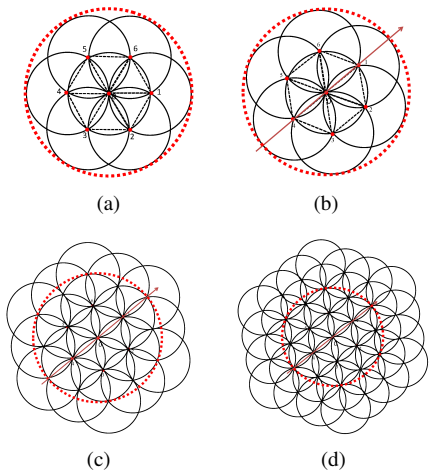


Fig. 1. The recursive hierarchical hexagon structure:(a) the original first level sampling pattern of the hexagon structure;(b) the first level rotated sampling pattern of the hexagon structure (the red arrow shows the dominant orientation of the red dotted region); (c) the second level sampling pattern of the hexagon structure; (d) the third level sampling pattern of the hexagon structure. Each black circle represents the Gaussian kernel size which could be freely parameterised.

derivatives. Therefore we proposed two different comparison schemes to construct HexBinary, one of which is first order and the other is second order. Furthermore, we investigate two image pre-filtering methods: simple Gaussian low-pass filtering to suppress image noise and aliasing and Laplacian of Gaussians isotropic filtering that captures second order gradient information, potentially useful for encoding boundary information.

First order HexBinary: As shown in Fig. 1 (a), the initial first level sampling configuration is located at a hexagon centre p_0 with the hexagon vertices p_1 and p_4 aligning with the x axis. In order to make the descriptor invariant to rotation, we first determine the local dominant orientation. We have adopted essentially the same mechanism as utilised in BRISK [4] to compute the orientation based on the sample pairs defined between the hexagon centre point and the vertexes. The image is first filtered by a Gaussian kernel of standard deviation σ, and the smoothed intensity values with respect to the hexagon centre and vertexes are I_i(*i=0,1,...6*). We then compute the local gradient of the red dotted circle covered region in Fig. 1 (a) using:

$$g(\mathbf{p}_i, \mathbf{p}_j) = \frac{1}{N} \sum_{p \in \mathcal{S}} (\mathbf{p}_j - \mathbf{p}_i) \cdot \frac{I_j - I_i}{\|\mathbf{p}_j - \mathbf{p}_i\|^2}. \qquad (1)$$

where $\mathbf{p}$ is the position vector of the hexagon centre and vertexes, and N =12 which is the number of pairs in set $\mathcal{S}$ composed by two subsets of point pairs:the subset approximating the polar tangential derivatives comprises adjacent pairs of samples taken at the hexagon vertexes and subtracted in a clockwise direction $\mathcal{S}_1 = \{(p_6, p_1), (p_i, p_{i+1})$(*i=1,...5*)$\}$;the subset approximating the radial polar derivatives comprises the group of sample subtractions taken from the hexagon centre to each single vertex $\mathcal{S}_2 = \{(p_0, p_i)$(*i=1,...6*)$\}$.

According to the computed local dominant orientation, we

resample the hexagon vertex points as Fig. 1 (b) shows, and extract a new set $\mathcal{Q}$ which has the same sampling scheme and binary encoding method as in $\mathcal{S}$. The 12 point pairs in $\mathcal{Q}$ are used to generate a 12 bit binary string, where each bit τ corresponds to:

$$\tau\left(I; i, j\right) = \begin{cases} 1 & \text{if } I_i < I_j \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

A 12 bit first level descriptor *HexBinary1* sampled at point p_0 is generated as above, and the second level descriptor *HexBinary2* is generated by concatenating the *HexBinary1*$_i$ descriptors sampled at positions p_i(*i=0,1,...6*). The second level and third level hexagon structures are shown in Fig. 1 (c) and (d). And the steps for computing the hierarchical descriptor recursively are described in Algorithm 1.

Second order HexBinary: In order to differentiate the first order HexBinary descriptor and the second order HexBinary descriptor, we denote them as *FHexBinary* and *SHexBinary*, respectively. The process of constructing the *SHexBinary* descriptors follows that of constructing the *FHexBinary* descriptors, except that we now compare pairs of first order intensity difference values. In the rotated first level sampling hexagon structure, a set of first order intensity differences is computed using pairs from set:

$\mathcal{S}' = \{(p_0, p_1), (p_4, p_0), (p_0, p_2), (p_5, p_0), (p_0, p_3),$
$(p_6, p_0), (p_6, p_1), (p_i, p_{i+1})$(*i=1,...5*)$\}$.

Accordingly, we get a corresponding first order intensity difference value set $\mathcal{D}$, from which we select 9 pairs to generate *SHexBinary* with each bit τ corresponding to :

$$\tau\left(D; i, j\right) = \begin{cases} 1 & \text{if } D_i < D_j \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

where$(\mathbf{D}_i, \mathbf{D}_j)$ is a spatially adjacent pair, e.g., $(\mathbf{D}_i = I_1 - I_0, \mathbf{D}_j = I_0 - I_4; \mathbf{D}_i = I_1 - I_6, \mathbf{D}_j = I_2 - I_1)$. The *SHexBinary* descriptor is recursively generated using the same construction scheme used to compute *FHexBinary*.

We have also concatenated the *FHexBinary* and *SHexBinary* descriptors together, based on first order and second order intensity derivative information respectively, to construct the *CHexBinary* descriptor that encapsulates these information sources into a potentially more powerful descriptor.

B. Pose Refinement

We employ the HexBinary descriptor to perform pose refinement using the same process proposed in [1]: SIFT is applied to give an initial pose estimation of a detected object located in a cluttered background. The object's edge contour points are employed to sample the HexBinary features used to refine this pose information, since they anchor a large set of samples which define the structure of the object. These

Algorithm 1 Hierarchical HexBinary Descriptor Generation

$p_i(i \leftarrow 0, 1...6)$: First Level Hexagon Centre and Vertex Positions
θ_0 : Computed Local Dominant Orientation for the Region Centred at p_0
$\theta_i(i \leftarrow 1...6)$: Defined Local Dominant Orientation for the Region Centred at p_i
$ts \leftarrow pi/3$
for $i \leftarrow 1 : 6$ **do**
  $tv \leftarrow (i-1)ts + \theta_0$
  $\theta_i \leftarrow tv$
end for
for $i \leftarrow 0 : 6$ **do**
  Generate First Level Descriptor *HexBinary1*$_i$ Centred at p_i with Local Dominant Orientation θ_i
end for
Generate Second Level Descriptor *HexBinary2*$_0$ Centred at p_0 by Concatenating *HexBinary1*$_i(i \leftarrow 0, 1...6)$
Generate *L*th Level Descriptor *HexBinaryL*$_0$ Centred at p_0 by Concatenating *HexBinary(L-1)*$_i(i \leftarrow 0, 1...6)$

descriptors are used to specify the relative pose of the object as follows:

1) The edgels of the reference image (black background) and the test image (cluttered background) are first be detected by the Canny edge detector, and a morphological operation is applied to the reference and test edge maps to remove isolated edgels. This process also eliminates some noise points and renders the detected reference object edgels more consistent with the detected test object edgels.
2) The reference edgels are projected into the test image based on the initial pose estimation. A small local area of the test image surrounding each reference edgel is then searched for the test edgel which has the best match to a (corresponding) reference edgel and which also exceeds a certain matching threshold.

A set of corresponding edgel pairs is generated based on the above steps and used to re-estimate the pose of the object by means of RANSAC.

C. Edge Contour Localisation:

Our primary objective is to label directly the contour edgels detected in the test image by finding matching correspondences with the edgels in the reference image (rather than projecting edgels from the reference image into the test image based on the refined pose estimation), as follows:

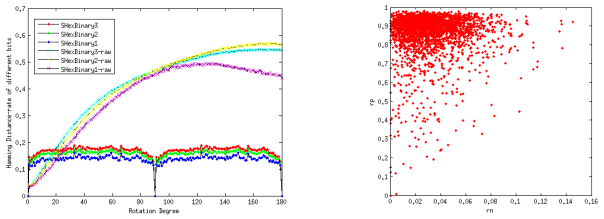
1) For each black-background reference image *R1*, generate a second reference image *R2* with a white background (to allow the boundary descriptors to match over positive or negative background contrast phases). Detect edgels for both *R1* and *R2*, and transform the edgel positions into the test image according to the refined pose estimation.
2) Classify each reference object edgel as being an interior edgel or a boundary edgel.
3) For each reference object interior edgel from *R1*, the best matching test edgel is searched for in a local area in the test image, as in the pose refinement process. If the match score exceeds a detection threshold, this matched test edgel and its edgel neighbours within 1 pixel distance will be all labeled as test object edgels.
4) For each reference object boundary edgel from *R1*, we perform the same search as in step 3, but use a different strategy to undertake feature extraction, as described below. This process is repeated again for the corresponding reference object boundary edgel from *R2*.
5) We use a coarse-to-fine approach to match the HexBinary features sampling on edgel locations. Contour edgel matching commences by first matching the highest level of grouped HexBinary descriptor, and then proceeding to attempt to match using the next lower level of descriptor grouping. If no match is detected at the current grouping level, and if the descriptors at all (lower grouping) levels have been used without finding a successful edgel match, then no test object edgel will be labeled to a corresponding reference edgel.

The idea behind step 2 above is inspired by [29]. A strong gradient magnitude is more likely to occur at boundary positions. Therefore, we employ χ^2 distance to classify boundary edgels and their corresponding tangential directions. This process is applied to the reference edgels. In the reference image *R1*, we extract a local patch which is centred on an edge point, and divide it into two halves in the horizontal direction. The intensity histogram of each half patch is computed and compared by χ^2 distance to measure the gradient magnitude between the two parts. Each such patch is rotated for every 10 degrees per step in order to find the position where the biggest χ^2 for this patch occurs. When a maximum χ^2 is found, and if one of the half patches A_1 includes a sufficient number of background valued pixels, exceeding the number of those found in the other half patch A_2, A_1 is deemed to cover pixels from the background, and this edge point will be defined as a reference boundary point on the object. Otherwise it is defined to be a reference edgel inside the object. HexBinary descriptors will be centred on those interior reference edgels to serve edgel matching. A different approach is taken for reference boundary edgels: In order to avoid background clutter disrupting descriptors located on the object bounding contour, we displace the centre of these HexBinary descriptors in a direction normal to the edge boundary contour towards the reference object interior by r pixels (the first level hexagon side length). We are thus able to substantially eliminate background clutter pixels from object boundary descriptor samples. Descriptors located on bounding contour edgels within the corresponding search area in the test image will be shifted in the same manner to generate a matching descriptor.

III. PERFORMANCE EVALUATION

Although our primary application domain is in robot vision systems, we have evaluated our proposed methods using the Amsterdam Library of Object Images (ALOI) dataset [6]. By basing our validation on the ALOI database, we are able to evaluate the HexBinary descriptors over a far wider range of objects and shapes than we have available and can capture within our own laboratory-based robot workcell. In order to make comparisons with the HexHoG descriptor, we employ the same set of test images and backgrounds provided by the author of [1]. For each reference image, five test images are composited with different backgrounds and randomly assigned object poses. Therefore, our validation test set comprises 1000 reference and 5000 test images. We set the HexBinary descriptor parameters empirically: The hexagon edge length is 3 pixels and the Gaussian kernel size for smoothing the image is 9 pixels with standard deviation 2, to suppress sampling aliasing. The matching threshold is 0.2, in this case a *dissimilarity* threshold in the range [0 1], as described below.

Rotation invariance evaluation: In order to evaluate the rotation invariance property of our HexBinary descriptor, we select 20 different reference images at random from the ALOI set and generate rotated versions in $1\,^\circ$steps over $180\,^\circ$. We then extract HexBinary descriptors at keypoints detected by the Fast Corner Detector [23] and compare the descriptor matches from the reference image to those of each of its rotated images. The total number of bit differences between compared binary strings normalised by the bit string length is used as the matching *dissimilarity* score which is averaged over the 20 different reference images, and plotted as a function of rotation. For all the descriptors we propose, we observe similar performance results to those in Fig. 2 (a), illustrating the degree of rotation invariance of three different grouping levels of second order HexBinary descriptor from Gaussian filtered images. The results demonstrate



(a) (b)

Fig. 2. (a) HexBinary rotation invariant matching performance; (b) Distribution of pixel-level localisation: Each red dot represents an edge labelling ROC point, *rp* vs *rn*.

that our proposed descriptors provide rotation invariance, returning matching (dissimilarity) scores smaller than 0.2 for all compared rotations, while the performance of the raw (rotation variant) descriptors decreases gradually with increasing compared rotation.

Pose refinement evaluation: The pose refinement process is based on an initial pose estimation by standard SIFT. 5000 test images were tested and only 2892 of these images were successfully detected by SIFT to provide an initial pose estimation. Initial pose estimation failures tended to be due to lack of detected keypoints. We applied the pose refinement process to those images which provided an initial pose estimation based on each of the proposed hierarchical HexBinary descriptors in isolation. Following the validation protocol in [1], a search range of ± 5 pixels gives superior performance for HexHoG. Therefore, we use the same search range for each HexBinary descriptor to make comparisons with HexHoG. We also test the HexBinary descriptors generated by sampling LoG filtered images, rather than Gaussian filtered images. The edgel displacement error is computed by the distance between the estimated edgel position and the ground truth edgel position, from which we provide the mean and the standard deviation of the local edgel displacement error after pose refinement and corresponding to the initial pose estimate value, for all images whose pose estimation improved after refinement. The performance results for different hexagon-based descriptors are shown in Table I and Table II. The number of improved images after pose refinement for different descriptors is also provided in Table III. From these tables we can observe that all the proposed descriptors based on hierarchical hexagon configurations give improved pose refinement, and that the pose refinement results improve with increasing levels of hierarchical descriptor grouping. Although the first level HexHoG out-performs all the first level HexBinary descriptors, as the level of HexBinary grouping increases, HexBinary descriptors give better performance than HexHoG.

TABLE I

MEAN ERROR OF SINGLE EDGEL POSITION FOR EACH LEVEL OF DESCRIPTOR AND THE CORRESPONDING REDUCED VALUE Δ FROM THE INITIAL MEAN ERROR (G_HexB MEANS HEXBINARY DESCRIPTOR GENERATED FROM GAUSSIAN FILTERED IMAGES; L_HexB MEANS HEXBINARY DESCRIPTOR GENERATED FROM LOG FILTERED IMAGES).

Descriptors	Level1	Δ_1	Level2	Δ_2	Level3	Δ_3
G_FHexB	10.7297	0.5345	0.9023	1.3188	0.6043	1.5178
G_SHexB	2.3705	0.4036	0.6399	1.5096	0.4687	1.7259
G_CHexB	2.0208	0.4860	0.6373	1.5099	0.4898	1.6978
L_FHexB	2.0475	0.6178	0.7049	1.5707	0.4200	1.7901
L_SHexB	2.2066	0.5557	0.6809	1.5634	0.4070	1.8031
L_CHexB	1.5340	0.9767	**0.5822**	**1.6370**	**0.3743**	**1.8327**
$HexHoG$	**0.9008**	**1.3570**	0.7508	1.4612	0.6853	1.5068

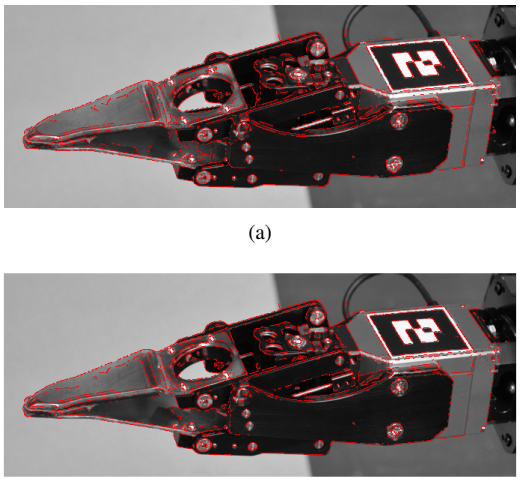
TABLE II

STANDARD DEVIATION OF SINGLE EDGEL POSITION ERROR FOR EACH LEVEL DESCRIPTOR AND THE CORRESPONDING REDUCED VALUE Δ FROM THE INITIAL STANDARD DEVIATION .

Descriptors	Level1	Δ_1	Level2	Δ_2	Level3	Δ_3
G_FHexB	12.9189	0.0912	2.2734	0.0846	1.3909	0.3558
G_SHexB	3.6426	0.0177	1.9085	**0.2653**	1.1681	**1.1439**
G_CHexB	2.4594	0.0149	**1.7439**	0.2542	1.2305	1.0427
L_FHexB	2.7983	-0.0099	2.4379	0.1373	1.4875	0.6502
L_SHexB	3.0726	0.0159	2.0399	0.1334	1.2455	0.8879
L_CHexB	2.6506	-0.0048	1.8427	0.2386	**0.9965**	1.0803
$HexHoG$	**1.9002**	**0.1166**	1.8327	0.2145	1.7235	0.3265

Edge contour localisation evaluation: Because all the HexBinary descriptor variants perform almost equally well in pose refinement, we employ $G\_SHexBinary$ due to its smaller vector length, for edge contour localisation using a search range= ± 3 pixels. In order to evaluate the results, we rotate the reference mask according to the ground truth pose information and project it into the test image with 1 pixel dilation. The number of test edgels inside the mask and the number of the labeled test edgels inside the mask are computed as NTP and NPL, respectively. We also compute the number of the reference edgels inside the mask as NR, and the number of the labeled test edgels outside the mask as NNL. The rate *rp=NPL/NTP* and *rn=NNL/NR* are used to evaluate the pixel-level localisation performance. rp and rn of each pose refined image are computed and the results are shown in Fig. 2 (b). Through our proposed method, the edge contour labelling process achieves viable results with a mean rp=0.8654, and mean rn=0.0314.

We present representative examples of the object edge



(a)

(b)

Fig. 3. (a) Edge contour localisation for a robot gripper under 3°of out-of-plane rotation ; (b) Localisation result for gripper with 5°of out-of-plane rotation.

contour localisation process applied to images in the ALOI data set in Fig. 4. A number of edgels have been miss-labelled because the background is similar in appearance to the object, while a number of object edgels detected in the test image might not have beeen detected in the reference image due to the edge detector not producing consistent edge lables between these views, and therefore inconsistent edge lables will be missed. If corresponding edgels are not found within the adopted search range, this also results in missing labels in the test image.

Finally, with the same edge contour localisation framework as described above, we present the results of an initial investigation into localising the edge contours of our robot's gripper as it appears within its workcell. In this case examples of directly labelling the gripper under 3°and 5°out-of-plane rotations (with respect to a reference image) are shown in Fig. 3. Although our proposed approach has not been specifically designed to be invariant to out-of-plane rotations, it is still able to make a reasonable attempt at matching and localising the grippers edge contours when the appearance of the gripper has been deformed within a small range of pixels.

IV. CONCLUSIONS AND FUTURE WORKS

In this paper, we present a complete framework for object edge contour localisation based on matching a dense set of novel hexagonally sampled and hierarchically composed HexBinary descriptors, generated from Gaussian filtered or LoG filtered images. Our pose refinement validation results indicate that the HexBinary descriptor outperforms HexHoG descriptor for in-plane transformed images while offering faster computation. Moreover, coarse-to-fine edge contour matching by HexBinary descriptors offers a promising level of edge contour localisation performance on which to base robotic grasping and manipulation behaviours. Our future work will incorporate a multi-scale image representation and new shape contour representations to investigate new descriptors which could extend our proposed framework into out-of-plane and non-rigidly transformed matching applications in robot vision.

TABLE III

NUMBER OF IMPROVED IMAGES AFTER POSE REFINEMENT

Descriptors	Level1	Level2	Level3
G_FHexB	31	2683	2789
G_SHexB	766	**2800**	**2826**
G_CHexB	1258	2795	2815
L_FHexB	1060	2088	2734
L_SHexB	896	2064	2707
L_CHexB	1549	2115	2728
$HexHoG$	**2598**	2685	2720

REFERENCES

[1] Y. Liu and J. P. Siebert, "Contour localization based on matching dense hexhog descriptors," in *International Conference on Computer Vision Theory and Applications (VISAPP 2014)*, 2014, pp. 656–666.

[2] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: binary robust independent elementary features," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 778–792.

[3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2564–2571.

[4] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2548–2555.

[5] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 510–517.

[6] J.-M. Geusebroek, G. J. Burghouts, and A. W. Smeulders, "The amsterdam library of object images," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, 2005.

[7] K. Murphy, A. Torralba, D. Eaton, and W. Freeman, "Object detection and localization using local and global features," in *Toward Category-Level Object Recognition*. Springer, 2006, pp. 382–400.

[8] J. Schlecht and B. Ommer, "Contour-based object detection," in *Proceedings of the British Machine Vision Conference. BVA Press*, 2011.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1627–1645, 2010.

[10] O. Barinova, V. Lempitsky, and P. Kholi, "On detection of multiple object instances using hough transforms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 9, pp. 1773–1784, 2012.

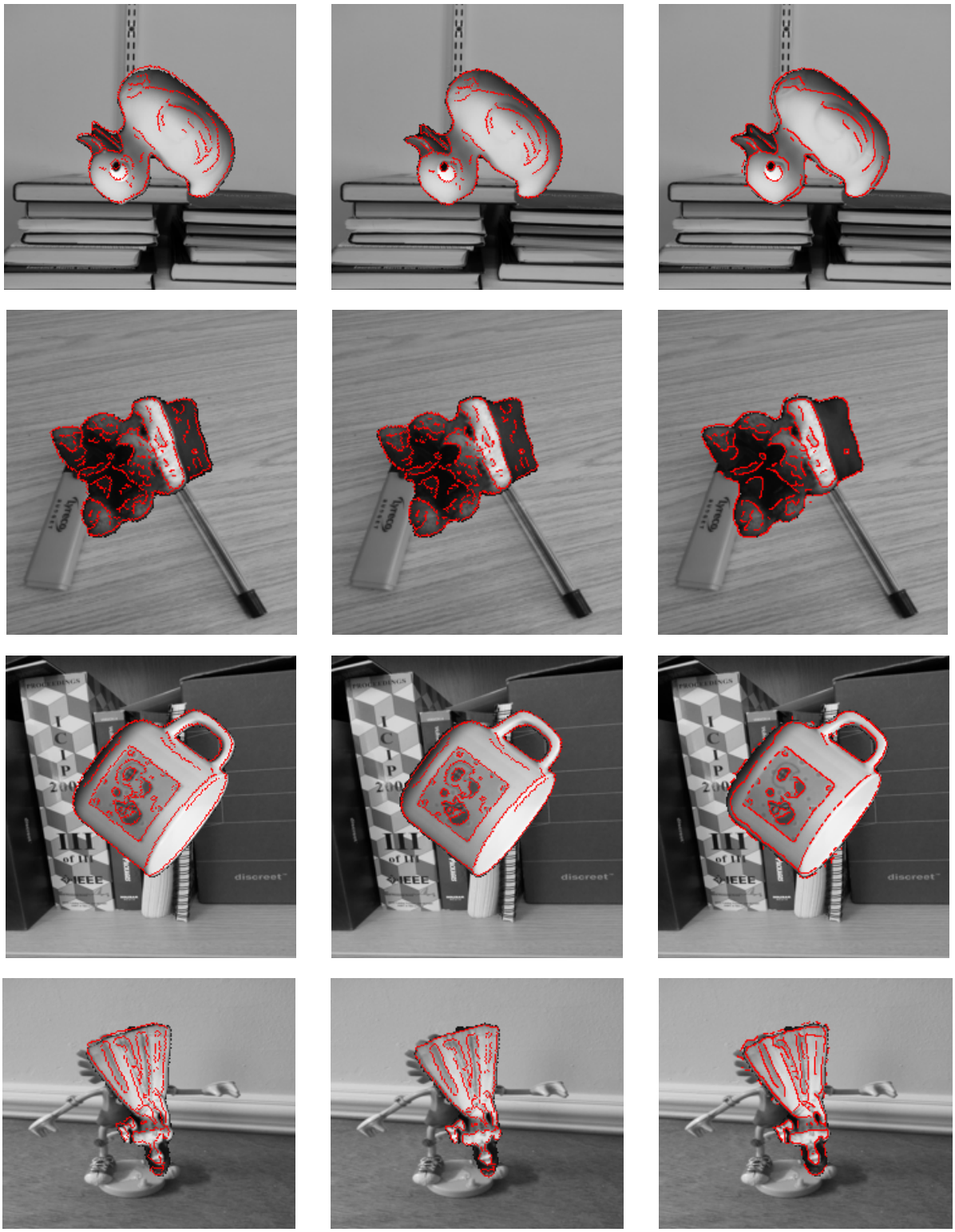


Fig. 4. Pose estimation and edge contour localisation examples: the first column shows reference contour projection based on initial pose estimation by SIFT; the second column shows reference contour projection based on refined pose estimation by $G\_SHexBinary3$; the third column shows directly labelled test object edge contour pixels.

[11] A. C. Berg, T. L. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondences," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 26–33.

[12] V. Ferrari, F. Jurie, and C. Schmid, "Accurate object detection with deformable shape models learnt from images," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[13] P. Kontschieder, H. Riemenschneider, M. Donoser, and H. Bischof, "Discriminative learning of contour fragments for object detection." in *BMVC*, 2011, pp. 1–12.

[14] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 1–15.

[15] J. Shotton, A. Blake, and R. Cipolla, "Contour-based learning for object detection," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1. IEEE, 2005, pp. 503–510.

[16] M. Maire, P. Arbeláez, C. Fowlkes, and J. Malik, "Using contours to detect and localize junctions in natural images," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[19] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1615–1630, 2005.

[20] M. Brown, G. Hua, and S. Winder, "Discriminative learning of local image descriptors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 1, pp. 43–57, 2011.

[21] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 404–417.

[22] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–506.

[23] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 430–443.

[24] E. Tola, V. Lepetit, and P. Fua, "Daisy: An efficient dense descriptor applied to wide-baseline stereo," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 5, pp. 815–830, 2010.

[25] G. Bouchard and B. Triggs, "Hierarchical part-based visual object categorization," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 710–715.

[26] A. Agarwal and B. Triggs, "Hyperfeatures–multilevel local coding for visual recognition," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 30–43.

[27] S. Fidler, M. Boben, and A. Leonardis, "Learning hierarchical compositional representations of object structure," *Object Categorization: Computer and Human Vision Perspectives, Cambridge University Press, Cambridge*, 2009.

[28] A. Leonardis and S. Fidler, *Learning hierarchical representations of object categories for robot vision*. Springer, 2011.

[29] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 5, pp. 898–916, 2011.