Contour Localization based on Matching Dense HexHoG Descriptors

Yuan Liu and J. Paul Siebert
School of Computing Science, University of Glasgow, Glasgow, U.K.

Keywords: Feature Extraction, Local Matching, Object Detection, Edge Detection, Edge Contour Labelling, Segmentation Features, HexHoG Descriptors.

Abstract: The ability to detect and localize an object of interest from a captured image containing a cluttered background is an essential function for an autonomous robot operating in an unconstrained environment. In this paper, we present a novel approach to refining the pose estimate of an object and directly labelling its contours by dense local feature matching. We perform this task using a new image descriptor we have developed called the HexHoG. Our key novel contribution is the formulation of HexHoG descriptors comprising hierarchical groupings of rotationally invariant (S)HoG fields, sampled on a hexagonal grid. These HexHoG groups are centred on detected edges and therefore sample the image relatively densely. This formulation allows arbitrary levels of rotation-invariant HexHoG grouped descriptors to be implemented efficiently by recursion. We present the results of an evaluation based on the ALOI image dataset which demonstrates that our proposed approach can significantly improve an initial pose estimation based on image matching using standard SIFT descriptors. In addition, this investigation presents promising contour labelling results based on processing 2892 images derived from the 1000 image ALOI dataset.

1 INTRODUCTION

This paper addresses the issue of accurate object edge contour localisation given an initial estimate of an object's pose with respect to its pose captured within a reference image. Appearance-based methods (Dalal and Triggs, 2005; Lazebnik et al., 2006; Murphy et al., 2006; Felzenszwalb et al., 2010; Borji and Itti, 2012) and contour-based methods (Kontschieder et al., 2011; Schlecht and Ommer, 2011; Shotton et al., 2005; Xu et al., 2012) for object detection have been extensively studied in recent years. Appearance-based methods represent the dominant approach to object detection, and typically are based on a pipeline that first extracts local patch features, and then employs a sliding window to scan across the whole image to detect a target. Alternatively, the pipeline can be structured to employ local features in order to detect object parts, which can then be associated together to detect the whole target. Since an object's edge contours afford crucial information for visual perception, edge contour-based approaches have also been extensively developed. The edge contour representation could be represented by local curvature information, or by the spatial structural relationship between edge fragments. Such edge contour representations can be employed individually for part matching, or combined together to generate a shape model suitable for whole object detection. It is inherently difficult to extract the edge contours of an object directly, particularly when the object appears within a cluttered background, since background structures that intersect an object's boundary tend to corrupt, or distort, the extracted bounding edge contour. Therefore, appearance-based methods are predominantly used for object detection. However, the ability to localise an object's boundaries would allow the pixels representing the object to be specified, as opposed to merely knowing the approximate position of a bounding box containing the object, as currently afforded by sparse local feature-based methods. Therefore, accurately extracted edge contours could serve both to segment an object from the scene and also to provide a shape-based representation of the segmented object. Accordingly, the combination of appearance-based and edge contour-based methods (Schlecht and Ommer, 2011) has the potential to provide accurate object localisation and additional information describing an object's semantics.

The principal contribution of this paper is a new method for combining appearance and edge information to detect and localise an object's edge contours

Liu Y. and Siebert P..
Contour Localization based on Matching Dense HexHoG Descriptors.
DOI: 10.5220/0004744006560666
In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications* (VISAPP-2014), pages 656-666
ISBN: 978-989-758-003-1

within a cluttered background. A new feature descriptor, the HexHoG, based on a hexagonal, hierarchical grouping mechanism that confers it with sufficient reliability and distinctiveness to enable it to be used to sample the image at all detected edgel positions (as opposed to only corner locations). An initial pose estimation is first obtained by means of sparse local feature matching using a standard SIFT implementation. Based on this estimation result, a dense local edge matching process is then applied using our new HexHoG feature to refine the initial pose estimation, and this refined pose estimation is then used to constrain local dense edge matching to obtain object edge contour labelling (and *correspondences* between the contour edgels detected in the test and reference images). Therefore, in this work we are *not* employing HexHoG descriptors for object detection, but instead utilising HexHoG descriptors for edge contour matching, edge labelling and pose estimation refinement as a *post detection & classification process*.

The proposed method is validated using the dataset ALOI (Geusebroek et al., 2005). Our results show that our proposed method significantly improves pose estimation refinement and exhibits promising results for edge contour labelling. The remainder of this paper is organized as follows: Section 2 presents a brief review of related work. Section 3 introduces our complete system for object pose estimation and edge contour labelling. Our experimental results are presented in Section 4, followed by the paper's conclusions.

2 RELATED WORK

Many object detection methods are able to achieve approximate localization of an object within a cluttered background. Borenstein & Ullman (Borenstein and Ullman, 2002) propose a Top-Down class-specific segmentation protocol to identify the structure of an object by means of high-level information, instead of using the traditional image-based criteria. Their method can detect an object which is labelled by means of previously learned 'building blocks', which do not precisely delineate the pixels comprising the detected object. Yu &Shi (Yu and Shi, 2003) present an integration model incorporating low-level edge detection and high-level patch detection to label an object of interest segregated from the background. However, no statistical evaluation of this method is presented in (Yu and Shi, 2003). Leibe et al (Leibe et al., 2008) contribute an Implicit Shape Model which affords their system a greater degree of flexibility by enabling it to learn different object shapes and use these shape models to categorize objects in novel images whilst inferring a probabilistic segmentation, which then in turn improves the robustness of the categorization and detection processes. Schlecht & Ommer (Schlecht and Ommer, 2011) propose a method for complementing appearance information with contour information in order to detect an object within a bounding box. Neither of these above two methods provide precise object boundaries which would allow the shape of segmented objects to be represented and recovered. Ferrari et al (Ferrari et al., 2010) provide a detection method by learning an object shape model represented using local contour features. Novel object instances could be localized in new images and the object boundaries were labelled rather than just being contained within a bounding box. A significant limitation of this system, however, is the computational cost of its learning process.

Feature extraction has been explored extensively in the context of object detection and localization. Gradient histogram-based descriptors have been researched intensively and applied widely for this purpose. Local densely sampled descriptors have been reported to give promising results in human detection (Dalal and Triggs, 2005) and wide-baseline matching (Tola et al., 2010), although such descriptors do not usually posses the property of rotation invariance. Sparse, distinctive features (Lowe, 2004; Mikolajczyk and Schmid, 2005; Alahi et al., 2012) achieve rotation invariance by rotating the local sampling coordinate frame according to the local dominant gradient orientation direction prior to compute an orientated gradient histogram distribution. Accordingly, this rotation normalization process is expensive to compute and is therefore inherently unsuitable when dense feature extraction is required. Furthermore, such features do not extract object edge information, which affords a crucial cue for visual perception.

3 APPROACH

In this section, we give the details of our proposed methods based on HexHoG feature extraction and dense local edge matching. The overview of our system is summarized in Fig.1.

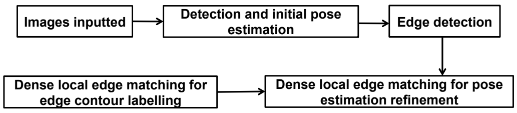

Images inputted
Detection and initial pose estimation
Edge detection
Dense local edge matching for pose estimation refinement
Dense local edge matching for edge contour labelling

Figure 1: The overview of our system.

3.1 Feature Extraction

3.1.1 SHoG Feature Extraction

Local image features based on the histogram of oriented gradients (HoG) representation have been widely adopted (Mikolajczyk and Schmid, 2005; Dalal and Triggs, 2005; Brown et al., 2011). Rotating the sampling coordinate frame according to the dominant local image gradient orientation provides a general way to achieve rotation invariance for local image features. In this work we adopt an alternative well established, but simpler, method to afford a substantial degree of rotation invariance within standard HoG. A single patch is first weighted by a Gaussian function and represented by a gradient orientation distribution histogram. In the histogram, the location of the highest bin, i.e. exhibiting the dominant gradient orientation, is barrel-shifted to the head of the histogram, which means the histogram starts with the frequency value of the dominant orientation, Fig.2. Therefore, we achieve rotation invariance by simply shifting the histogram rather than rotating and resampling the image coordinate frame as shown in Fig.6. We term this orientation normalised HoG as SHoG and the pseudocode for its construction is given in Algorithm 1.

Algorithm 1: SHoG Construction.

```
HoG: Histogram of Oriented Gradient
Num_Bin: Number of Bins in HoG
Max: Max HoG Bin value
Index: Index to the Max HoG Bin
$e \leftarrow 0$
for $i \leftarrow Index : Num\_Bin$ do
    $e \leftarrow e+1$
    $SHoG(e) \leftarrow HoG(i)$
end for
$r \leftarrow Num\_Bin - Index + 1$
for $i \leftarrow 1 : (Index-1)$ do
    $r \leftarrow r+1$
    $SHoG(r) \leftarrow H(i)$
end for
```

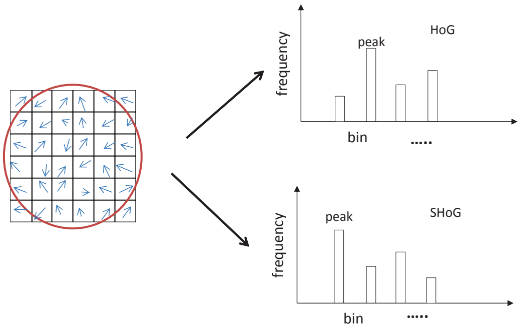


Figure 2: Local patch represented by HoG and SHoG.

3.1.2 HexHoG Feature Extraction

Based on SHoG, we investigate a hexagon grouping mechanism which is similar to DAISY (Tola et al., 2010) with the difference that this hexagon grouped local descriptor HexHoG can be recursively constructed to generate hierarchical descriptors. Moreover, unlike DAISY, HexHoG is substantially rotationally invariant.

A hexagon has its inherent rotational symmetry in geometry, which contributes to rotation invariance over a certain angular range. The hexagonally grouped local regions comprising HexHoG are constructed as shown in Fig.3. Each black circle represents a locally sampled region represented by an SHoG descriptor. Each black circle centre is a sampling point located on a hexagon vertex, and the centre point marks the sampling point at the centre of the hexagon on which each HexHoG group is constructed. Since we sample SHoG fields at not only the hexagon vertices but also the centre of the hexagon group, 7 rather than 6 SHoG fields are grouped together. Therefore, strictly we are computing a *septimal*, i.e. 7 element, grouping based on hexagonal geometry.

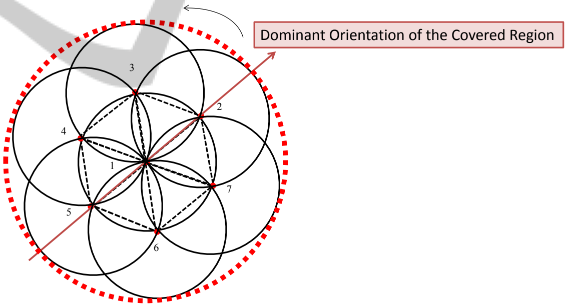


Figure 3: The first level HexHoG structure.

We can freely set both the radius of the circular regions denoting each SHoG field and the distance between neighbouring sampling points. These parameters control the overlap between the SHoG fields of each grouping, which influences the degree of rotation invariance of the final HexHoG descriptor and also the distinctiveness of this representation. We compute the dominant orientation of the region covered by red dashed circle by computing a HoG field spatially weighted by a Gaussian envelope, and thereafter selecting the peak HoG orientation bin, as performed in SIFT.

The above protocol determines where to sample the 6 vertexes of the hexagon once the hexagon center has been fixed. Three sampling points, including the center point, are co-aligned in the direction of the dominant orientation. Then we can generate this hexagonally grouped feature by concatenating $SHoG_i$(*i=1,2,...7*) by first assigning the central SHoG descriptor to the head of the grouped descriptor, followed by the SHoG descriptor which is aligned to the dominant orientation. All of the remaining $SHoG_i$ descriptors will subsequently be concatenated in anti-clockwise order. The complete Level 1 HexHoG descriptor is constructed about its centre point as follows:

$$L1\_HexHoG = SHoG_1, SHoG_2, SHoG_3, \ldots, SHoG_7 \quad (1)$$

The feature is then normalized by its magnitude to achieve robustness to illumination variations. This process can be applied recursively to generate higher level hexagonal descriptors using the same concatenating mechanism. Accordingly, *L2_HexHoG* is generated based on the seven *L1_HexHoGs* centred on the red points in Fig.4. For clarity, we have enlarged the first level hexagon edge length to make it easier to illustrate. The ordering mechanism used to concatenate the *SHoG* for *L1_HexHoG* is consistent with the above description. However, the dominant orientation for each region covered by a *L1_HexHoG* group is defined differently here, except for the central *L1_HexHoG* group which retains its original dominant orientation, computed when it was originally extracted, as described above. The pseudocode to generate *L1_HexHoG* and *L2_HexHoG* is given in Algorithm 2 and 3 respectively.

The blue arrow in Fig.4 shows the dominant orientation of the central region covered by a *L1_HexHoG*. The dominant orientations of all the other 6 $L1\_HexHoG_i$ are defined by the red arrows, respectively, each of which illustrates the direction from the whole group centre to the vertex of each corresponding hexagon. The right figure in Fig.4 illustrates how we generate a *L1_HexHoG* feature for the red dashed region. Finally, the second level hexagonal feature is constructed by:

$$L2\_HexHoG = L1\_HexHoG_1, L1\_HexHoG_2, \\ L1\_HexHoG_3, \ldots, L1\_HexHoG_7 \quad (2)$$

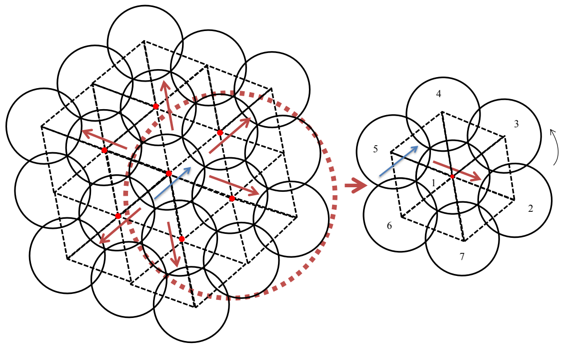


Figure 4: The second level HexHoG structure.

Algorithm 2: L1_HexHoG Construction.

```
$< Px_1, Py_1 >$: HexHoG centre, i.e. Sample Point location
$r$: Hexagon Side Length
$\theta$ :Dominant Orientation of the Sampled Point $< Px_1, Py_1 >$
$< Px_i, Py_i >$ $(i \leftarrow 2...7)$: Six Vertex Positions of the hexagon centred on $< Px_1, Py_1 >$
$ts \leftarrow 2pi/6$
for $i \leftarrow 1:6$ do
    $tv \leftarrow (i-1)ts + \theta$
    $Py_{i+1} \leftarrow Py_1 + r\sin(tv)$
    $Px_{i+1} \leftarrow Px_1 + r\cos(tv)$
end for
for $i \leftarrow 1:7$ do
    Construct $SHoG_i$ at Point $< Px_i, Py_i >$
end for
$L1\_HexHoG \leftarrow Normalize(SHoG_1, SHoG_2, ..., SHoG_7)$
```

Algorithm 3: L2_HexHoG Construction.

```
$< Px_1, Py_1 >$: HexHoG centre, i.e. Sample Point location
$r$: Hexagon Side Length
$\theta_1$ : Defined Dominant Orientation for the Sampled Point $< Px_1, Py_1 >$
$< Px_i, Py_i >$ $(i \leftarrow 2...7)$: the Six Vertex Positions of the hexagon centred on $< Px_1, Py_1 >$
$\theta_i$ : Defined Dominant Orientation for the Sampled Point $< Px_i, Py_i >$
$ts \leftarrow 2pi/6$
for $i \leftarrow 1:6$ do
    $tv \leftarrow (i-1)ts + \theta$
    $Py_{i+1} \leftarrow Py_1 + r\sin(tv)$
    $Px_{i+1} \leftarrow Px_1 + r\cos(tv)$
    $\theta_{i+1} \leftarrow tv$
end for
for $i \leftarrow 1:7$ do
    Construct $L1\_HexHoG$ at Point $< Px_i, Py_i >$
end for
$L2\_HexHoG \leftarrow Normalize(L1\_HexHoG_1, L1\_HexHoG_2, ..., L1\_HexHoG_7)$
```

3.2 Detection and Edge Contour Labelling

3.2.1 Detection with Pose Estimation

The objective of this paper is to localize the edge contours of an object within a cluttered background based on a dense local edge matching process, where this object has already been detected by conventional sparse feature matching. Accordingly, this edge segmentation process relies on a correct prior object detection and classification result and the quality of the pose estimation obtained during the prior detection and classification process. Since SIFT (Lowe, 2004) is an established benchmark for state-of-the-art performance in object detection, in this paper, we adopt SIFT in our experiments for object detection and initial pose estimation purposes. We directly match sparse SIFT descriptors extracted from a test image to the corresponding SIFT descriptors extracted from a reference image, grouped and filtered using the GHT and RANSAC respectively. In order to obtain a more accurate pose estimation, we perform a further refinement step by means of dense local HexHoG matching, described as follows:

- 1. Compute edge label (edgel) maps for both the test image and the corresponding reference image using the Canny Edge Detector;
- 2. Project the edgels of the reference image into the test image edgel map according to the initial pose estimation;
- 3. Find the set of the test image edgels that neighbour each projected edgel from the reference image edgel map, within a constrained search area for each projected edgel;
- 4. From the set of neighbouring test-image edgels, find the best matching test image edgel for each projected edge point by comparing their HexHoG features, computed from the input images;
- 5. Re-estimate the pose transformation from the reference image to the test image, based on all the matched edgel-pair correspondences obtained above.

The constrained search area reduces false-positive matches between background clutter edgels and the reference object's edgels, while the use of edgel-located feature matching provides many more feature correspondences than corner-based features alone, especially when the reference object inherently lacks corners, i.e. contains mainly smooth edge contours.

Validation. A validation method is required to evaluate how well the proposed pose estimation refinement method performs. For each test image, we record ground-truth information specifying the rotation and translation used to embed the reference object pixels into a background image. Therefore, we know the precise location of edge contours of the reference object in the test image. According to the pose estimation provided by the image matching process (either SIFT or dense HexHoG), the estimated object edgel positions are obtained by projecting the reference edgels into the test image. For each reference edgel, the distance between its estimated position and its ground-truth position is then computed to give its pose estimation error. The mean and standard deviation of matched point displacement error for the test set is used to evaluate pose estimation performance.

3.2.2 Edge Contour Labelling

Object edge contour labelling is implemented following pose estimation refinement. Estimated edgel positions in the test image are found by projecting the reference edgels using the refined pose estimation transformation. The search process, constrained to a limited range in X and Y, is then repeated to match between the edgels positions estimated using sparse matching and the edgels in the test edgel map. The edgels within the test image which match to the projected reference image edgels are then labelled in the test image as being *contour edgels*. An edge connectivity post-process is then executed as follows: If an edgel in the test image is labelled as contour edgel, all connected edgels (comprising its 8 nearest neighbours) will be likewise labelled. This process is then repeated for each newly labeled contour edgel. We perform 6 iterations in our experiment in order to label those edgels which comprise the object's edge contours and thereby potentially capture the shape of the detected object in terms of observed edgels.

4 EXPERIMENTAL RESULTS

The data employed in our validation experiments has been obtained from the Amsterdam Library of Object Images (ALOI) (Geusebroek et al., 2005). A selection of test object images is shown in Fig.5. The top row comprises objects randomly selected from ALOI; the middle row shows in-plane rotated versions; the bottom row shows rotated objects embedded into a background. We fix the Gaussian weighted patch size to be 7 pixels wide for SHoG, and the sampling hexagon edge length to 3 pixels, which results in the HexHoG grouping structure shown in Fig.3.

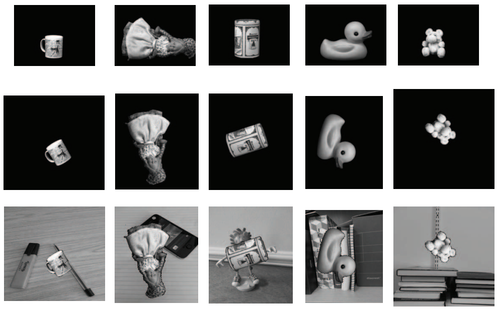


Figure 5: Examples of the data used in our experiments.

4.1 Rotation Invariance Performance

The performance of local feature matching in terms of rotation invariance is evaluated for both HoG and our proposed features. We randomly select 20 different images from ALOI as a reference set, and rotate each image by 1° per step in range $[0,90]^\circ$to generate a set of test images, respectively. For each rotation, a set of keypoints is detected using the Fast Corner Detector (Rosten and Drummond, 2006). The descriptor for each keypoint in each reference image is computed and compared to the descriptor of the corresponding point in each test image. We record the dot product of the corresponding descriptors and compute the average dot product over 20 different test images as a function of degree of in-plane rotation. The performance obtained using HoG and our proposed features to match local features is illustrated in Fig.6. In our system, 8 histogram bins are used to record the relative frequency of 8 local gradient orientation directions. This explains the periodic performance observed every 45° for all our proposed features in Fig. 6, Although the rotation invariance of the feature is getting weaker with the grouping level increased, *L3_HexHoG* can still give the matching dot product greater than 0.8, which is the matching threshold we applied through our system. On the other hand, the performance of HoG declines monotonically with object rotation, falling below an average dot product of 0.8 at around 25° of in-plane rotation.

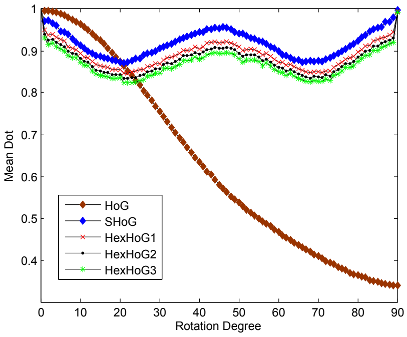


Figure 6: Local feature matching performance.

4.2 Pose Refinement Performance

Before the pose estimation refinement process can be implemented, we must first decide which level HexHoG feature to adopt for local edgel matching. We devised the following experiment to determine the displacement error resulting from local edge matching: 20 different images from ALOI are randomly selected as a reference set and then rotated incrementally to form a test set. Therefore, for each edgel in each test image we generate, we know the corresponding edgel in the reference image original. The HexHoG feature for each reference image edgel is then computed and compared to the features computed within a local neighbourhood of 2 pixels in radius, centered on the corresponding test image edgel. We find the best dot product match and record its position. The spatial distance between the matched position and the corresponding true feature position is computed for each reference edgel as the displacement error for local matching. Thereafter the average error is computed over 20 reference images and we obtain the displacement error distribution as a function of rotation for 3 levels of feature grouping, as shown in Fig.7. The level3 HexHoG feature gives the smallest displacement error for all applied rotations, which suggests that *L3_HexHoG* will give better localisation performance compared to our other, less grouped, features for the purpose of pose estimation refinement.

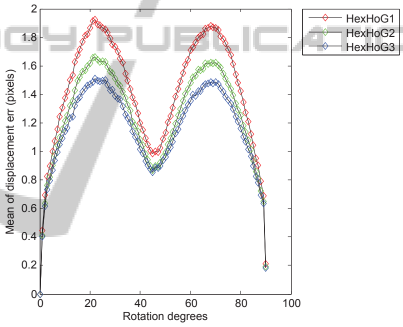


Figure 7: Displacement error for local HexHoG matching.

We investigated pose refinement performance with respect to the constrained search bounds, by varying the X,Y search range from ± 1 to ± 10 pixels, and computing the refined pose estimation error accordingly. By comparing the refined pose estimation error to the initial pose estimation error for each test image, we can determine the number of test images which exhibit an improvement in pose estimation due to the refinement process. Both the average pixel error and the standard deviation for the entire test set of initial estimations, and refined estimations, are also computed. We employed all 1000 different objects from the ALOI database as reference images to validate our local matching approach to contour edgel labelling. Each of these reference images is randomly rotated in-plane and embedded into 5 different backgrounds respectively to generate a test dataset comprising 5000 images. Fig.5 illustrates examples of the image sets described above.

In Table.1, we present (in pixel units) the mean error and standard deviation of the refined pose estima-

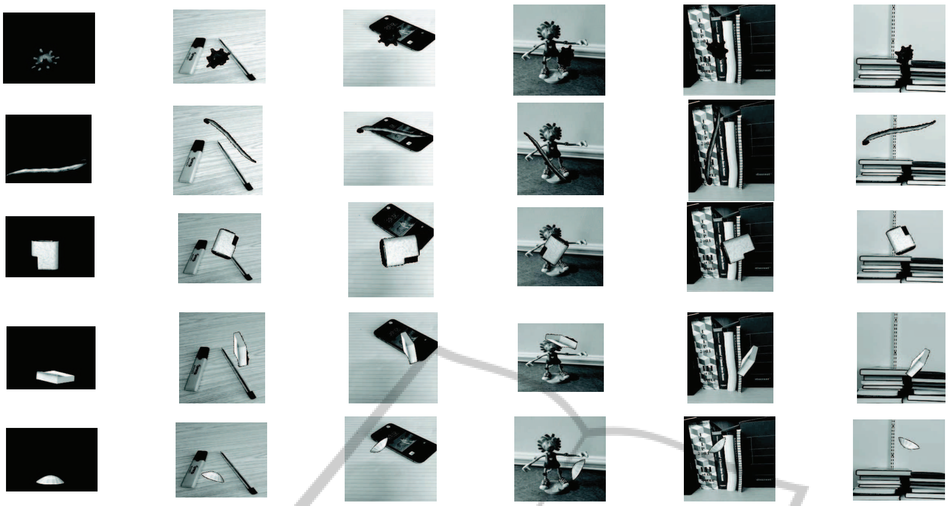


Figure 8: Failed examples of detection by SIFT: the first column shows the reference objects; the remaining columns show the test objects with backgrounds.

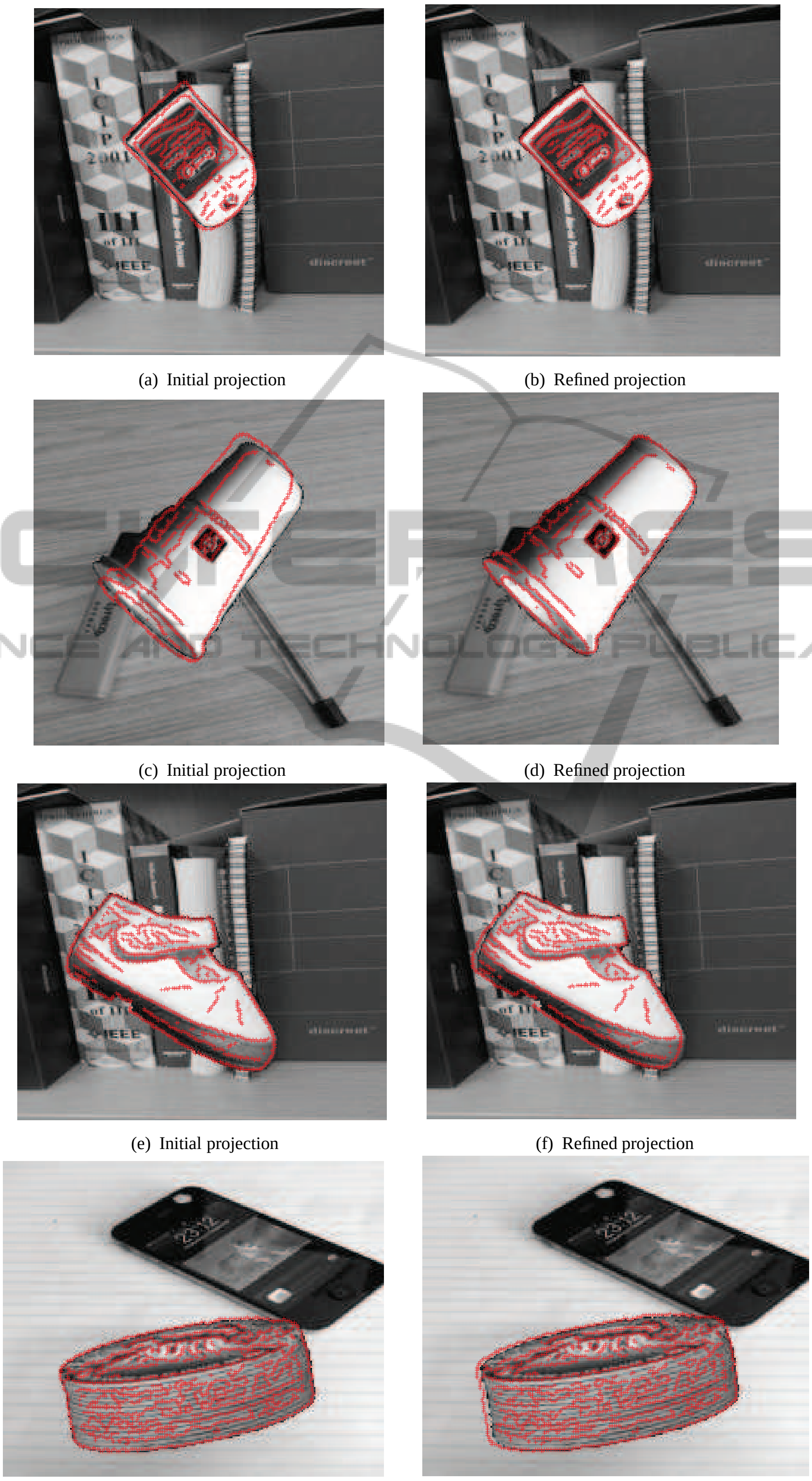
Table 1: Pose estimation refinement performance.

Search range ±	Mean	StdDev	No. Improved Pose Est.	Improv. Ratio
1	1.35	2.68	2807	97.06
2	0.94	2.93	2771	95.82
3	0.84	2.89	2757	95.33
4	0.91	6.42	2738	94.67
5	0.83	2.84	2720	94.05
6	0.84	2.59	2696	93.22
7	0.86	2.56	2669	92.29
8	0.89	2.57	2637	91.18
9	0.92	2.58	2607	90.15
10	0.99	2.73	2560	88.52
Initial Pose Estimate	2.20	2.69	0	0

tion for the test dataset matched using different search bounds and also the initial error in pose estimation obtained using SIFT. The results in Table.1 confirm that the pose estimation refinement process improves the mean pose estimation error for the whole test dataset by approximately a factor of 2. All test images were first classified by means of SIFT matching, employing the GHT and RANSAC for pose estimation. When the object of interest has less distinctive corners and is not sufficiently distinguishable from the background, SIFT will fail to detect such an object. In this experiment, 2892 images were successfully detected out of 5000 images in total. A selection of failed examples is shown in Fig.8. Consequently, we only apply our pose estimation refinement and edge labelling process to test image examples containing successfully detected object instances. The number of improved object pose estimations and their corresponding fraction of the test set is also presented in Table.1. When the search range for edgel matching was constrained to less than 10 pixels, the HexHoG based pose estimator achieved an improvement in over 90% of the initially successful object detections. We can observe in Table.1 that the mean pose estimation error is least for a search range in the region of ±5 or ±6 pixels (as a reference point for comparison, the *L3_HexHoG* used for matching is 28 pixels in diameter). However, the number of pose estimations that exhibit an improvement declines monotonically with search range. Therefore, there is a tradeoff between the degree of pose refinement and the number of object detections that are improved. For subsequent edge contour labelling experiments, reported below, we choose a search range of ±6 pixels. A selection of examples of post pose estimation refinement is illustrated in Fig.9.

4.3 Edge Labelling Performance

Finally, we re-applied dense local edge matching in order to label directly the edgels detected within the test image that comprise the contour edgels of the object of interest, rather than project edgels from the reference image into the test image, according to the recovered pose estimation (using a ± 6 pixel search range). Fig .10 shows examples of the labelling results we obtained by matching three different grouping levels of HexHoG descriptor. When the image background is very cluttered, or the object outer boundary is not easily distinguished from the background, missed object boundary detections can result and background edgels close to the object can be mis-labelled as belonging to the object. We can observe in Fig .10 that each level of HexHoG descriptor produces slightly different labellings, making it difficult to conclude which level HexHoG feature grouping gives best results. It would appear that the distraction from the background is greater for larger higher level descriptors (which straddle both the object boundary and the background to a greater de-



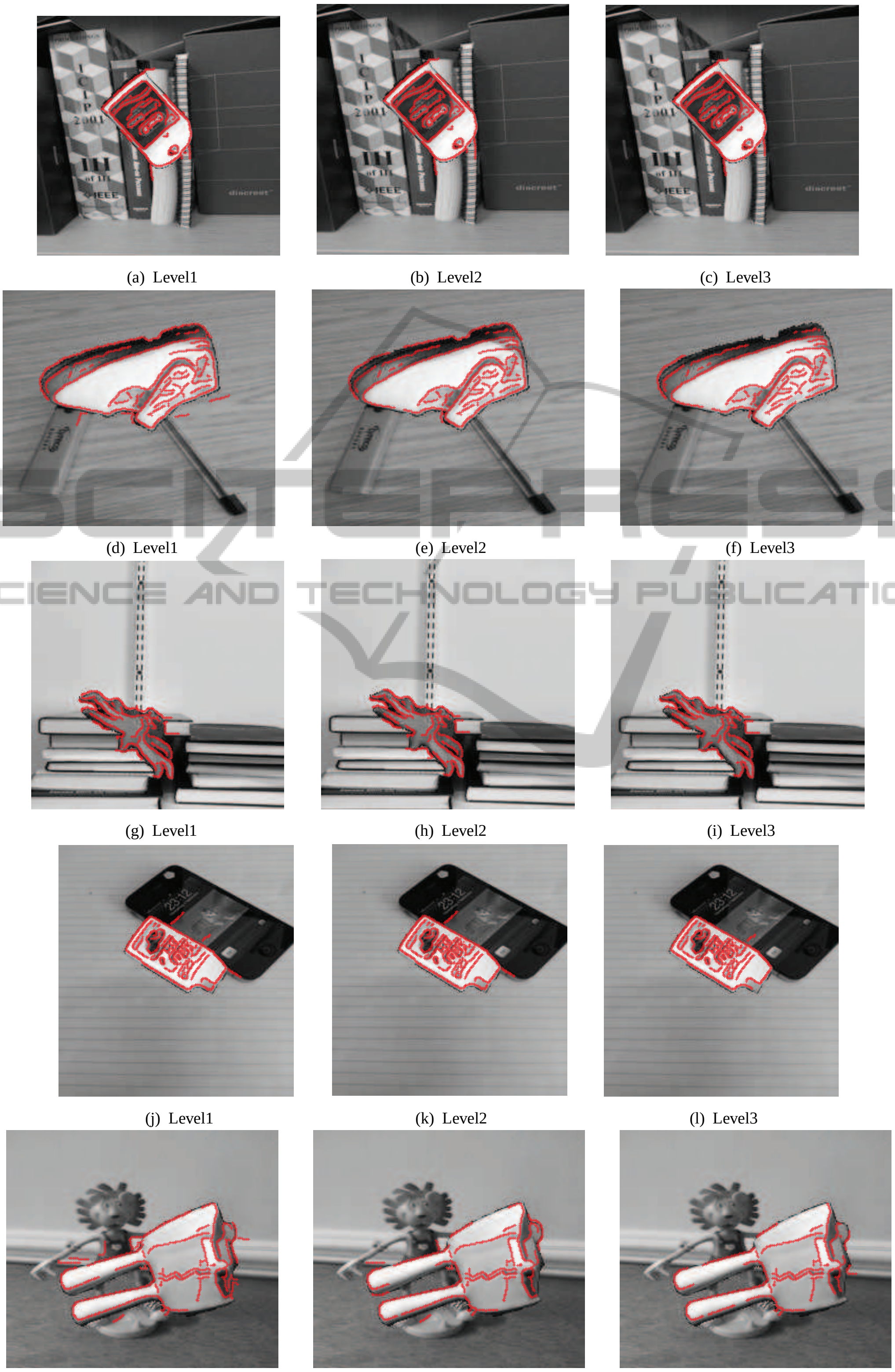
(a) Initial projection (b) Refined projection

(c) Initial projection (d) Refined projection

(e) Initial projection (f) Refined projection

(g) Initial projection (h) Refined projection

Figure 9: Edge projection from the reference objects into the test images according to the initial and refined pose estimation: the first two rows show the examples with improvement from the refined projection; the last two rows show the examples which failed to achieve improvement from the refined projection



(a) Level1 (b) Level2 (c) Level3

(d) Level1 (e) Level2 (f) Level3

(g) Level1 (h) Level2 (i) Level3

(j) Level1 (k) Level2 (l) Level3

(m) Level1 (n) Level2 (o) Level3

Figure 10: Object edge contour labelling results: from the first column to the third column, edge labelling results by using *L1_HexHoG*, *L2_HexHoG*, *L2_HexHoG*, are shown respectively.

gree) while the lower level descriptors have less reliability. Therefore, in our future research we propose to combine the different level features, perhaps within a coarse-to-fine search framework, in order to optimize the labelling performance. In this case the largest grouping would be first matched and then the search process repeated using successively low-level groupings which are then matched using increasingly constrained search bounds.

5 CONCLUSIONS

In this paper, we present a new hexagonally grouped and rotationally invariant image descriptor, the HexHoG, that can be computed recursively to generate hierarchical features. Hierarchical grouping affords sufficient discriminability to allow HexHoG descriptors to be sampled at all detected edgel positions (as opposed to only corner locations) in order to match edge contours between a reference and test image. Given an initial class and pose for a detected object, we are then able to apply dense local HexHoG matching, to both improve the detected object's pose estimation and also directly label the edge contours of the object as they appear in a test image. Therefore our proposed methodology supports segmentation-through-matching.

Our validation experiments show that matching HexHoG features, which are based only on appearance information computed at edgel locations, has the potential to improve the performance of object pose estimation by approximately a factor of 2. By improving the accuracy of the pose estimation process, it is then possible to project contours from the reference image into the test image and annotate the location of a detected object with sufficient accuracy for many practical tasks such as grasping in robotics. However, improved pose estimation also improves the search constraints required to match test image edge contours directly, to allow HexHoG matching to offer the possibility of recovering the actual edgel labels detected in the test image that correspond to contour edgels in the reference image, as described above.

Our results indicate that for purely affine pose transformations, the proposed scheme can recover a significant fraction of edgel labellings in the test image. In many situations, where for example the pose relationship between the target object contained in the reference and test images is non-affine, e.g. for out-of-plane rotation or under projective distortion, dense HexHoG feature matching has the potential to maintain pixel-accurate correspondences between the edge contours detected within the test and reference object images.

Our future work will focus on incorporating an improved edge detector, hierarchical approaches to matching the HexHoG features and improved post-lablling processing for determining edgel connectivity and edgel contour shape representation.

ACKNOWLEDGEMENTS

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.

REFERENCES

Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517. IEEE.

Borenstein, E. and Ullman, S. (2002). Class-specific, top-down segmentation. In *Computer VisionECCV 2002*, pages 109–122. Springer.

Borji, A. and Itti, L. (2012). Exploiting local and global patch rarities for saliency detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 478–485. IEEE.

Brown, M., Hua, G., and Winder, S. (2011). Discriminative learning of local image descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(1):43–57.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645.

Ferrari, V., Jurie, F., and Schmid, C. (2010). From images to shape models for object detection. *International Journal of Computer Vision*, 87(3):284–303.

Geusebroek, J.-M., Burghouts, G. J., and Smeulders, A. W. (2005). The amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112.

Kontschieder, P., Riemenschneider, H., Donoser, M., and Bischof, H. (2011). Discriminative learning of contour fragments for object detection. In *BMVC*, pages 1–12.

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer*

Society Conference on, volume 2, pages 2169–2178. IEEE.

Leibe, B., Leonardis, A., and Schiele, B. (2008). Robust object detection with interleaved categorization and segmentation. *International journal of computer vision*, 77(1-3):259–289.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630.

Murphy, K., Torralba, A., Eaton, D., and Freeman, W. (2006). Object detection and localization using local and global features. In *Toward Category-Level Object Recognition*, pages 382–400. Springer.

Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer.

Schlecht, J. and Ommer, B. (2011). Contour-based object detection. In *Proceedings of the British Machine Vision Conference. BVA Press*.

Shotton, J., Blake, A., and Cipolla, R. (2005). Contour-based learning for object detection. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 503–510. IEEE.

Tola, E., Lepetit, V., and Fua, P. (2010). Daisy: An efficient dense descriptor applied to wide-baseline stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(5):815–830.

Xu, Y., Quan, Y., Zhang, Z., Ji, H., Fermuller, C., Nishigaki, M., and Dementhon, D. (2012). Contour-based recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3402–3409. IEEE.

Yu, S. and Shi, J. (2003). Object-specific figure-ground segregation. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–39. IEEE.