ROTATION-INVARIANT OBJECT RECOGNITION USING EDGE PROFILE CLUSTERS

Ryan Anderson, Nick Kingsbury, and Julien Fauqueur

University of Cambridge
Signal Processing Group, Dept. of Engineering
University of Cambridge, UK
{raa37,ngk,jf330}@cam.ac.uk
web: http://www-sigproc.eng.cam.ac.uk

ABSTRACT

This paper introduces a new method to recognize objects at any rotation using clusters that represent edge profiles. These clusters are calculated from the Interlevel Product (ILP) of complex wavelets whose phases represent the level of "edginess" vs "ridginess" of a feature, a quantity that is invariant to basic affine transformations. These clusters represent areas where ILP coefficients are large and of similar phase; these are two properties which indicate that a stable, coarse-level feature with a consistent edge profile exists at the indicated locations. We calculate these clusters for a small target image, and then seek these clusters within a larger search image, regardless of their rotation angle. We compare our method against SIFT for the task of rotation-invariant matching in the presence of heavy Gaussian noise, where our method is shown to be more noise-robust. This improvement is a direct result of our new edge-profile clusters' broad spatial support and stable relationship to coarse-level image content.

1. INTRODUCTION

This paper describes a novel method of detecting and searching for specific edge structures in images, regardless of their orientation. Our "edge-profile clusters" allow us to detect and represent edges and ridges by their spatial properties as well as their profile. The profile of a feature indicates whether it is ridge or an edge, positive or negative. To our knowledge, no such attribute has been exploited in the literature for object recognition.

In general, most recent successful object recognition algorithms involve a) identifying features of an object that are invariant to transformation, and b) seeking near-matches of these features in potential candidate images. Such searches may be performed by reducing object images to a set of *interest points*, using Lowe's Difference of Gaussian (DoG) detector [6] or the Harris corner detector [3]. Local features are then calculated at these points with a variety of methods (several of which are compared in [8]), and correspondences between these feature sets are sought between all points calculated in the target image and a candidate search image. Finally, methods such as the generalized Hough Transform or RANSAC are used to calculate the affine transformation between the target and a candidate.

These techniques are appropriate and efficient for the corners and blobs detected by Harris and DoG methods, and have been applied to edge features as well, for detecting "wiry" objects [7]. We wish to adopt an approach that acknowledges that edge features do not possess a clearly defined "interest point" representation; they are entities that are distributed widely throughout space as well as scale. Therefore, we will represent edges with 2-dimensional entities in this paper. By doing so, we also distinguish our method from classic shape metrics such as the Hausdorff distance which, even when orientation information is included ([9]), describes points that may not be individually robust.

Our new method is motivated by observations in the co-efficients of the ILP (InterLevel Product), introduced in [1], a measure based upon the Dual-Tree Complex Wavelet [4]. We summarize the properties of the ILP in further detail in section 2, along with the ICP (InterCoefficient Product), introduced in [2], which is also used to identify the specific orientations of these features.

Once the abilities of the ILP and ICP functions are explained, we proceed in section 3 to cluster the ILP and ICP information into sets of entities that sparsely represents the major edge components of a target image. In section 4, we then outline the search algorithm to find these target entities in the ILP domain of the search data. We briefly compare our method against SIFT for a test target in section 5, and conclude in section 6 with a discussion of the results and the next steps for our research.

2. THE DT CWT TRANSFORM, AND ILP/ICP FUNCTIONS

In this section, we summarize the ILP and ICP functions, which transform both target and search images into the domain in which we will perform matching. We start with an overview of the the DT CWT upon which the ILP and ICP functions are based.

2.1 The DT CWT Transform

The Dual-Tree Complex Wavelet Transform (DT CWT) transforms an $N \times M$ image into a pyramid of L levels, where each level $l = 1 \ldots L$ contains $\frac{N \times M}{4^l} \times 6$ complex interlevel coefficients. The magnitude of a coefficient represents the strength of activity in the vicinity of its spatial location (x, y), scale l, and orientation d, where $d = 1 \ldots 6$ represents directional subbands approximately equally spaced between $15°$ and $165°$. The phase of DT CWT coefficients change linearly with the offset of a feature from the coefficient location. Note that the behaviour of DT CWT coefficients are similar to steerable pyramid coefficients [10]; however, the DT CWT can be implemented with linearly separable wavelet filter banks, providing improved computation speeds. However, this acceleration comes at the expense of losing "steerability"; the number and directions of the subbands are fixed.
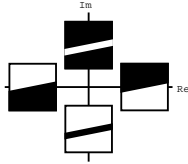
Figure 1: Relationship between the complex phase of an ILP coefficient in the 15° subband and the nature of a ∼ 15° feature in the vicinity.

Compared to Discrete Wavelet Transform, the DT CWT has two desirable properties suitable for object recognition: moderate shift invariance and better directional selectivity. However, while the magnitudes of complex wavelet coefficients provide valuable information for object recognition, the phases in their raw state are less helpful. If the image is shifted slightly, relative to the decimation reference, phase changes will be introduced that make matching difficult. It would be helpful, instead, if the phases of the coefficients were more directly dependent upon image content only. In the next section, we will see how the ICP and ILP functions create these dependencies.

2.2 The InterLevel Product: Feature Types

By looking at the difference in phase between a DT CWT coefficient $W(x,y,l,d)$ and a phase-doubled version of its coarser-scaled parent $W(x,y,l+1,d)$, one can see that the linear phase-offset relationships cancel to produce a phase difference that is relatively constant, regardless of spatial feature offset. As a result, this phase difference is related only to the nature of the multiscale feature present at the given location; this relationship is shown in Figure 1. An ILP coefficient, $\chi(x,y,l,d)$, creates this phase difference by multiplying the child coefficient with the conjugate of the parent; more details of this process can be found in [1]. Specifically, the ILP phase represents the type of *phase congruence* between even and odd Fourier components an octave apart. As an example, a positive real 2-D ILP ($\angle\chi = 0°$) [1] corresponds to the congruence of the positive sine (odd) Fourier coefficients, which form a positive step edge at this scale pair; similarly, a negative imaginary ILP ($\angle\chi = 270°$) indicates congruence of negative cosine (even) Fourier coefficients. In [5], Kovesi describes the relationship between Fourier components and complex wavelet coefficients in further detail (using complex log-Gabor wavelets).

Figure 2a shows an example of 15° ILP coefficients highlighting the unique edge profiles of the near-horizontal edges of an aerial building picture at level 2. Note that, as well as being shift-invariant, the ILP phase is moderately rotation invariant; features oriented within 15° of the central orientation of a given subband produce similar phase results. We demonstrate this in Figure 2b by rotating Figure 2a 30 degrees and observing that the phases of the 15° ILP coefficients remain relatively unaffected in the vicinity of the main edge features.

2.3 The InterCoefficient Product: Feature Angles

To determine the orientation of a feature (and, thus, the subband to find it in), we use a different phase-based function,

---

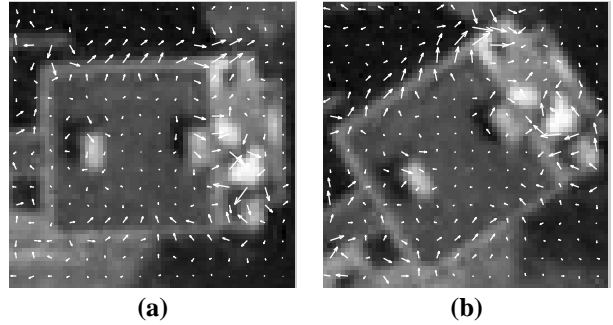[1] In this paper, we use $\angle x$ to denote $\arg(x)$.

(a)   **(b)**

Figure 2: Complex ILP coefficients χ_2 at Level 2, subband 15°, representing an aerial image of a building at two different angles. Note the distinctive, coherent phase profiles associated with the top and bottom edges of the building, and that these phase profiles are relatively invariant to rotations within the subband.

named the ICP (InterCoefficient Product). While the ILP calculates conjugate products (and hence phase differences) across scales in the same location, the ICP calculates conjugate products across space; specifically, between two *adjacent* coefficients at the same scale and orientation. Any dominant feature that spans the support regions of both DT CWT coefficients will cause these coefficients to have phases whose difference is proportional to the orientation of the feature by a fixed constant. Thus, by dividing by this constant, one can cause the complex argument of the ICP coefficients ψ to equal the angle of the underlying feature. This relationship is a direct trigonometric result of the phase/offset relationship between a feature and a coefficient, and is demonstrated explicitly in [2].

We now have two multiscale shift-invariant sources of phase information that we can use to characterize edge features of an object.

3. BUILDING A ROTATION INVARIANT TARGET MODEL

We start by transforming the target image T with the ILP and ICP functions to produce the pyramid of $\chi^{(T)}$ and $\psi^{(T)}$ coefficients respectively, and isolating the regions where we believe the coefficient phases will be stable.

3.1 ILP Phase Coherence and Stability

Empirical observations of ILP coefficients indicate that edge and ridge objects of interest occur where ILP coefficients, within the same subband, possess the following qualities:

1. Large magnitude, indicating that activity is present; and
2. Spatial adjacency of a number of coefficients with approximately the same complex phase ("coherent" ILP coefficients), implying that the same, dominant feature is influencing all coefficients.

Under these circumstances the relationship between ILP phase and image content is *stable*; that is, it is invariant to relatively small affine transformations of the content, such as may occur with rotation or translation. To enforce the latter criteria, and hence effectively separate edges from textures, we first create a new set of coefficients $R(x,y,l,d)$ at

each subband and level that demands phase similarity between neighbouring coefficients, as dictated by requirement 2 above:

$$R(x,y,l,d) = \begin{cases} \frac{R_{\text{sum}}(x,y,l,d)}{4}, & \text{if} \frac{|R_{\text{sum}}(x,y,l,d)|}{\Sigma_{a=0}^{1}\Sigma_{b=0}^{1}|\chi^{(T)}(x+a,y+b,l,d)|} > \beta \\ 0, & \text{otherwise.} \end{cases}$$

(1)

where $R_{\text{sum}}(x,y,l,d) = \Sigma_{a,b=0}^{1}\chi^{(T)}(x+a,y+b,l,d)$ and β is a threshold that controls the strictness with which one can enforce phase coherence; we use a value of $\beta = 0.8$. The resulting R coefficients are either an average of four neighbouring ILP coefficients, if they possess similar phase. Regions where R is zero (i.e. with inconsistent ILP coefficients) correspond to smooth or textured image regions. Figure 3 shows the new coefficients.

3.2 Clustering Coherent ILP Coefficients

After thresholding out coefficients of inconsistent phase, we look to a clustering algorithm to sparsely represent the largest of the remaining non-zero coefficients in R, which we expect to be stable features. In this paper, we use a region growing algorithm to seed and grow clusters until no neighbouring ILP coefficients can be found that are non-zero and within a phase threshold (say, $\pm 30°$) of the seeded coefficient. The weighted locations of the resultant labelled coefficients are then used to calculate the cluster parameters; if R_c represents the coefficients of R that are in cluster c, then the mean μ_c and covariance Σ_c are calculated from the locations of these coefficients, with cluster weight $\alpha_c = |\sum R_c|$ and overall cluster ILP phase profile $\theta_c = \angle(\sum R_c)$.
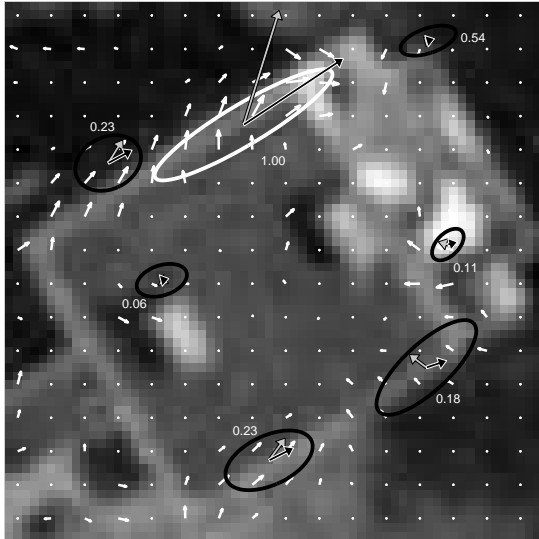

Figure 3: An example of the R coefficients corresponding to the level 2 15° subband ILP coefficients of Figure 2a. These coefficients are clustered according to section 3.2, and ICP orientations are assigned. Grey arrows indicate the ILP phase of each cluster; black arrows indicate their ICP orientations, and the number indicates the normalized α_c weight of the cluster. The cluster with the highest α_c across all subbands (the "primary cluster") is present in this subband and is indicated with a white cluster boundary.

We also require an orientation for each cluster to a) identify the correct subband in which to search for transformed instances of the cluster, and b) calculate the oriented location of subsequent clusters appropriately. Thus, we add ICP orientations to each cluster. For cluster c, we calculate $\overline{\psi_c} = \arg(\sum \psi_c)$, where ψ_c are all of the ICP coefficients co-located with R_c members of cluster c.

3.3 Summary of Edge-Profile Clusters

We now have clusters corresponding to the visually salient and consistent edge/ridge features in a target image. More precisely, we define an "Edge-Profile Cluster" c to be a cluster of coherent ILP coefficients which efficiently represents an edge or a ridge with five parameters: its center (μ_c), size/shape (Σ_c), orientation ($\overline{\psi_c}$), weight (α_c), and edge profile (θ_c).

An example of edge-profile clusters for a target object is shown in Figure 3 for the 15° ILP coefficients of Figure 2b; the other five subbands of Level 2 will possess similar clusters around the detected features at different orientations.

We now introduce a method to detect rotated instances of this constellation of clusters.

4. THE ILP CLUSTER MATCHING ALGORITHM

We first search for possible instances of a dominant edge/ridge feature of an object, regardless of its orientation, and then attempt to "build" the rest of the object around it.

4.1 Primary Cluster Selection

In this paper, we will simply assume that the dominant feature of a target image is represented by the cluster with the highest α_c across all subbands, whose value reflects both the magnitude and spatial extent of its ILP coefficients. We name this cluster the *primary cluster*, c_p, with associated parameters μ_p, Σ_p, α_p, θ_p, and $\overline{\psi_p}$. In Figure 3, this cluster is indicated by the white cluster boundary. The remaining clusters we name *secondary clusters*, whose presences we detect in section 4.3 after finding candidates for each primary cluster.

In the next section we search the ILP coefficients of the candidate image, $\chi^{(S)}$, for rotated instances of the primary cluster.

4.2 Building a List of Primary Cluster Candidates

First, we transform the search image S into the ILP and ICP pyramids, $\chi^{(S)}$ and $\psi^{(S)}$ respectively.

In a search image, potential candidates for our primary cluster will have the same ILP phase; we ignore the ICP orientations and search across all subbands, as we are looking for instances that occur at any angle. For each subband, we construct an ellipse of ILP coefficients in the shape of the primary cluster, oriented at an appropriately rotated angle. If we define this new rotated cluster ellipse by $\Sigma_{p,d}$, we then template match it against the decimated ILP coefficients in each subband. The result of this match, $r(x,y,l,d)$, is a value between -1 and 1 that represents the similarity between the primary cluster and the indicated location (x,y), scale l, and general orientation d of the search image.

We retain locations at which the match-score r is above τ. The threshold τ controls the proportion of candidates retained for further processing; we use $\tau = 0.2$, a fairly liberal threshold.
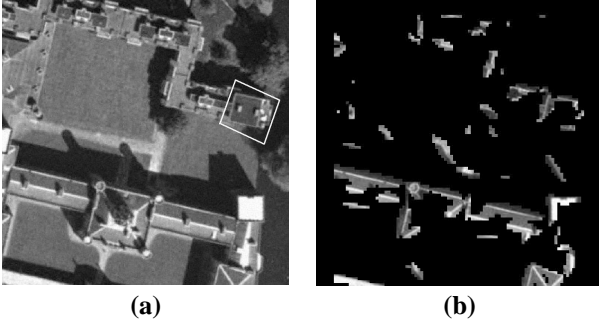
(a) **(b)**

Figure 4: In a), a search image is shown in which we will seek the target object of Figure 3. In b), we show the locations of the means for the set $\mathcal{K}$ of candidates (at any angle) for the primary cluster indicated in Figure 3. Note that the correct primary cluster match (and thus, the correct target match) is located at the right middle of the search image.

The results of our primary cluster candidate search is a set $\mathcal{K}$ of candidates, where each individual candidate $\kappa \in \mathcal{K}$ is a potential location around which we may find the target. For the primary cluster shown in Figure 3, we highlight all of its candidates in a search image in Figure 4. If the primary cluster is a positive ridge, we expect all the candidates to be positive ridges, and no step edges or negative ridges. In our illustrated example, our primary cluster possesses an ILP profile part way between an edge and a positive ridge; our candidate list will contain features with an equivalent profile.

Having selected and ranked our areas to search for the desired object, we use the secondary clusters to create and test hypotheses that the object exists at the location and orientation specified by each primary cluster candidate.

4.3 Ranking and Selecting the Best Candidate

Because of our liberal threshold for the primary cluster search, we are likely to have potentially several thousand primary cluster candidates. For each candidate κ, we wish to check if the ILP phases of the secondary clusters agree with the ILP phases in the corresponding locations in the search image.

Our method of searching all directional subbands for the desired edge profile is broadly accommodating of the feature angle within the subband. For example, any matching edge between $0°$ and $30°$ will be identified in the $15°$ subband. However, to fit the secondary clusters in the proper spatial orientation, we need a more specific orientation to assign to the primary cluster. Thus, for each primary cluster candidate (which we name κ_p) we calculate the cluster ICP by weighting the ICP coefficients co-located with the ILP cluster candidate (we name this set of ICP coefficients ψ_{κ_p}) from their distance to the center of the Gaussian cluster and taking the argument of the sum:

$$\overline{\psi_{\kappa_p}} = \angle \left(\sum_{x,y \in c_{\kappa_p}} \frac{e^{-[xy]\Sigma_c^{-1}[xy]^T}}{2\pi|\Sigma_c|^{\frac{1}{2}}} \cdot \psi_{\kappa_p} \right)$$

We now use this angle $\overline{\psi_{\kappa_p}}$ as a canonical orientation for the primary cluster that is precise enough to calculate the expected locations of the secondary clusters, relative to μ_{κ_p},

the mean of the candidate primary cluster. This calculation is a straight-forward rotation of the secondary target clusters' offsets, relative to the primary target cluster. If we define the rotation angle $\Delta\psi_\kappa$ and rotation matrix R_κ as follows:

$$\Delta\psi_\kappa = \overline{\psi_{\kappa_p}} - \overline{\psi_p}$$

$$R_\kappa = \begin{bmatrix} \cos\Delta\psi_\kappa & -\sin\Delta\psi_\kappa \\ \sin\Delta\psi_\kappa & \cos\Delta\psi_\kappa \end{bmatrix}$$

Then the parameters of each candidate secondary cluster are calculated as follows:

$$
\begin{aligned}
\mu_{\kappa_c} &= \mu_{\kappa_p} + R_\kappa(\mu_c - \mu_p) & \Sigma_{\kappa_c} &= R_\kappa \Sigma_c R_\kappa^T \\
\overline{\psi_{\kappa_c}} &= \overline{\psi_c} + \Delta\psi_\kappa & \alpha_{\kappa_c} &= \alpha_c \\
\theta_{\kappa_c} &= \begin{cases} \theta_c, & \psi_{\kappa_c} < \pi \\ -\theta_c^*, & \psi_{\kappa_c} > \pi \end{cases}
\end{aligned}
\tag{2}
$$

We also use $\overline{\psi_{\kappa_c}}$ to determine the subband $d_{\kappa_c} = 1\ldots6$ in which the target ILP is compared to θ_{κ_c}.

We now have the location, shape, and expected ILP phase of each secondary cluster c for candidate κ. To compare the expected ILP to the actual ILP for each secondary cluster, we once again perform a template correlation between the predicted cluster ellipse and the actual image content at the expected location, producing m_{κ_c}, a value between -1 and 1 that measures the correlation between the expected ILP phase of the secondary cluster and the observed ILP phase at the candidate image location.

For a given candidate κ, we now have $\mathcal{C}$ clusters that will "vote" for the likelihood that κ is the best candidate. However, the votes are not equal; some of our clusters are larger and more stable than others. Accordingly, we weight each candidate cluster (including the primary candidate cluster) by α_{κ_c} and sum:

$$M_\kappa = \sum_{c \in \mathcal{C}} \alpha_c m_{\kappa_c}$$

And, finally, we select the most appropriate match by taking the candidate with the maximum value of M:

$$\text{Best Match} = \arg\max_{\kappa \in \mathcal{K}} M_\kappa \tag{3}$$

5. TESTING AND RESULTS

We demonstrate our matching algorithm by matching the 64×64 object of Figure 3 in the 384×384 search image of Figure 4 under increasing additive Gaussian noise, at level 2 (a decimation of 4×4). For our tests of rotation invariance, we use quadratic interpolation to rotate the target in $5°$ increments from $0°$ to $180°$ before clustering; we then apply the Gaussian noise to the search image, prior to application of the ILP function. We use the same setup to test the SIFT method[2], and compare the two methods' abilities to successfully match the target at each noise level. We also demonstrate our matching method's invariance to uniform illumination changes, by performing our tests under a non-linear gamma distortion: if the pixel values s of the search image S are normalized to a range from 0 to 1, we apply the distortion $s_\gamma = r^\gamma$ before transform, for $\gamma = 0.5, 1$. A correct match

---

[2]We use the Matlab SIFT code available from D. Lowe at http://www.cs.ubc.ca/~lowe/keypoints.

for our method occurs when the best match of equation 3 is the κ candidate at the correct location and orientation, and a correct SIFT match occurs when at least three interest points have been correctly located in the search image. In Figure 5, one can see the superior ability of the ILP clustering method to cope with heavy noise. It also possesses a more gradual decrease in performance, when compared to the swift decline in performance of the SIFT features at 20% Gaussian noise. In Figure 6, we see an example correct match.
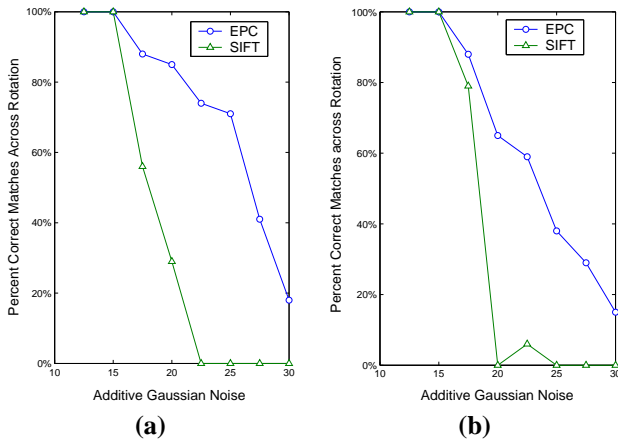

(a)　　　　　　　**(b)**

Figure 5: A comparison of the proposed Edge-Profile Cluster (EPC) method and SIFT for rotation-invariant object matching, for the target and search image shown in Figure 4a. At each level of noise (x-axis), we attempt to determine object matches for 36 different rotations of the target, and record the proportion of correct matches. Results are shown for an undistorted search image in a) and with $\gamma = 0.5$ distortion in b).
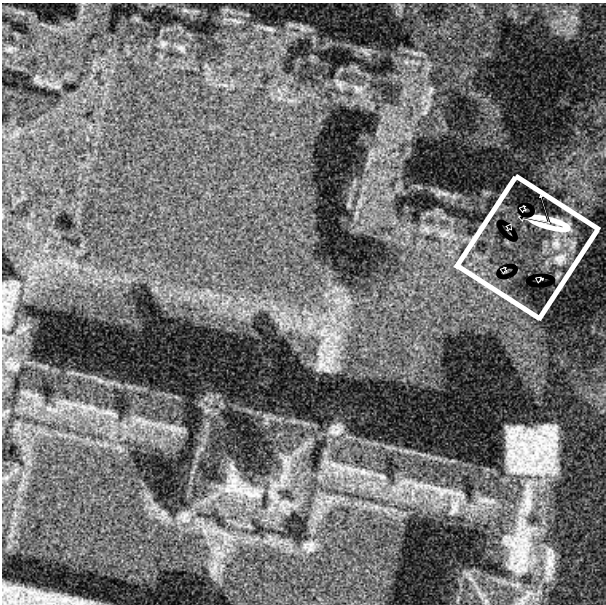

Figure 6: An example match of the target at 17.5% Gaussian noise, rotated $45°$. We display the correct match along with the associated candidates for the $15°$ clusters from Figure 3, including the primary cluster.

6. CONCLUSIONS

In this paper, we proposed a new method of object recognition based upon edge-profile clusters in the ILP transform domain whose ILP phases are invariant to rotation, and dependent only upon the edge profiles that they represent. We feel that edges have a natural advantage in robustness when compared to interest points and we illustrate this advantage by showing our matching algorithm's superiority in matching in heavy noise. In the future, we plan to show its ability to match in a scale- and affine-invariant manner as well. We will also investigate the extent to which interest points and our clustered edge profiles are complimentary.

We also feel that the proposed method has much in common with the human psychovisual system; our system relies heavily upon interaction between directional filters of adjacent scale and space, much like the V1 neurons of the visual cortex. Thus, we will investigate the use of our edge-profile clusters as a cortical processing model as well.

REFERENCES

[1] R. Anderson, N. Kingsbury, and J. Fauqueur. Coarse level object recognition using interlevel products of complex wavelets. In *International Conference on Image Processing (ICIP)*, September 2005.

[2] R. Anderson, N. Kingsbury, and J. Fauqueur. Determining multiscale image feature angles from complex wavelet phases. In *International Conference on Image Analysis and Recognition (ICIAR)*, September 2005.

[3] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

[4] N.G. Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Journal of Applied and Computational Harmonic Analysis*, (3):234–253, 2001.

[5] P. Kovesi. Image features from phase congruency. *Videre: Journal of Computer Vision Research*, 1(3):1–26, 1999.

[6] David G. Lowe. Object recognition from local scale-invariant features. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 1150, Washington, DC, USA, 1999. IEEE Computer Society.

[7] K. Mikolajczyk, A. Zisserman, and C. Schmid. Shape recognition with edge-based features. In *Proceedings of the British Machine Vision Conference*, 2003.

[8] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[9] C. F. Olson and D. P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, 6(1):103–113, January 1997.

[10] Eero P. Simoncelli and William T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *International Conference on Image Processing*, volume 3, pages 444–447, 23-26 Oct. 1995, Washington, DC, USA, 1995.