Published in IET Image Processing
Received on 7th March 2012
Revised on 4th October 2012
Accepted on 21st October 2012
doi: 10.1049/iet-ipr.2012.0104

ISSN 1751-9659

Framework for image retrieval using machine learning and statistical similarity matching techniques

Majid Fakheri[1], Tohid Sedghi[1], Mahrokh G. Shayesteh[1,2], Mehdi Chehel Amirani[1]

[1]*Department of Electrical Engineering, Urmia University, Urmia, Iran*
[2]*Wireless Research Laboratory, ACRI, Electrical Engineering Department, Sharif University, Tehran, Iran*
E-mail: m.shayesteh@urmia.ac.ir

Abstract: The aim of this study is to take advantage of both shape and texture properties of image to improve the performance of image indexing and retrieval algorithm. Further, a framework for partitioning image into non-overlapping tiles of different sizes, which results in higher retrieval efficiency, is presented. In the new approach, the image is divided into different regions (tiles). Then, the energy and standard deviation of Hartley transform coefficients of each tile, which serve as the local descriptors of texture, are extracted as sub-features. Next, invariant moments of edge image are used to record the shape features. The shape features and combination of sub-features of texture provide a robust feature set for image retrieval. The most similar highest priority (MSHP) principle is used for matching of textural features and Canberra distance is utilised for shape features matching. The retrieved image is the image which has less MSHP and Canberra distance from the query image. The proposed method is evaluated on three different image sets, which contain about 17 000 images. The experimental results indicate that the proposed method achieves higher retrieval accuracy than several previously presented schemes, whereas the computational complexity and processing time of the new method are less than those of other approaches.

1 Introduction

Content based image retrieval (CBIR) is a technique used for retrieving similar images from an image database. The most challenging aspect of CBIR is to bridge the gap between the low-level feature layout and high-level semantic concepts. CBIR systems have used local colour and texture features [1–3] for increasing the retrieval accuracy, but nevertheless they have reached to the desirable goal.

In the region-based image retrieval (RBIR) systems [4, 5], the image is segmented into regions based on the colour and texture features. The regions are closer to the human perception and are used as the basic building blocks for feature computation and similarity measurement. RBIR systems have been proven to be more efficient than the CBIR systems in terms of retrieval performance. In [4], images are compared based on the individual region to region similarity. Precise image segmentation has still been an open area of research. For example, the integrated region matching (IRM) algorithm introduced in [4], proposes an image-to-image similarity based on combining all corresponding regions of two images. In [4], each region is assigned significance worth based on its size in the image. In [6], fuzzy features are used to capture the shape information. Shape signatures are computed from the blurred images and then the global invariant moments are used as the shape features. The retrieval performance of Hirremath and Pujari [6] has been shown to be better than several systems such as [4, 7]. The studies mentioned in the above clearly indicate that, in CBIR, local features along with the shape information of objects play a significant role in determining the similarity of images. Precise segmentation is not only difficult to achieve but also is so critical in object shape determination. In [8], a windowed search over location and scale is shown to be more effective in object-based image retrieval than the methods based on inaccurate segmentation.

The objective of this paper is to develop a machine-learning technique which captures the local texture descriptors in a coarse segmentation framework of grids. The machine-learning technique used in this study is categorised in machine-learning applications, that is, search engines in image retrieval and pattern recognition. The new method also uses a shape descriptor in terms of invariant moments computed on the edge image. In our method, each image is partitioned into different sizes of non-overlapping tiles. A new framework is then used for texture analysis. The features computed on these tiles serve as the local descriptors of texture. Further, invariant moments calculated on the edge image are used to serve as the shape features. The combination of texture and shape features forms a robust feature set in retrieving application. Then, an integrated matching procedure based on the adjacency matrix of a bipartite graph among the image tiles, is provided, similar to the one discussed in [4], which yields the image similarity. Our method is rather similar to IRM, but so simple and less time

consuming. The reason is that all RBIR systems are complicated because of using different kinds of complicated algorithms that makes them time-consuming. Note that in RBIR systems, fast retrieval is the main objective. We evaluate the new method on three image sets that consist of about 17 000 images. The experimental results are compared with the methods of [3, 4, 6, 7, 9–12]. We also compare different methods from the view points of computational cost, feature length and processing time. The results indicate that the new method performs better than the mentioned methods.

The rest of this paper is organised as follows. In Section 2, we explain the proposed system that consists of image enhancement, regioning, feature extraction, integrated image matching and retrieval algorithm stages. In Section 3, we present the experiments, discuss about the results and compare them with the previously presented schemes. Finally, Section 4 concludes the paper.

2 Proposed method

Fig. 1 shows the comprehensive block diagram of the proposed CBIR system. In the following, we describe different parts of the new method in detail.

2.1 Image enhancement

In order to remove noise, each image is passed through a low-pass Gaussian filter. This stage is commonly used in most retrieval systems as the pre-processing step.

2.2 Regioning and blocking

Each image is partitioned into nine non-overlapping tiles. These tiles will serve as the local texture descriptors for image. The features extracted from the transformed tile images are used for texture similarity measurement. Feature extraction method will be completely discussed in Section 2.3. Three different image sets are used for experiments which comprise of images of sizes either 256×384 or 384×256 pixels. After image partitioning, the sizes of tiles will be 192×64, 96×128 or 192×128 pixels. Fig. 2*a* shows the partitioning of two different size images and their corresponding tiles. In this study, we have supposed that the most important part of image is its centre, that is, the centre contains the most valuable information compared with the side tiles. Hence, we apply region processing in such a way that the image centre is appeared in a large tile and the sides of image in small tiles. We then assign weights to each tile in order to demonstrate the significance of tiles. The assigned weights are shown in Fig. 2*a*. As expected, larger tiles are assigned larger weights. A typical image and its regioning for extracting the textural features are demonstrated in Fig. 2*b*. The logic for splitting into nine tiles can be justified as follows. Splinting into six and eight blocks causes the loose of centre information. Splitting into seven tiles makes the side tiles sizes larger. Further, more than nine tiles make the main information included in several tiles and also the method not computationally efficient. Therefore splitting image into nine blocks is a reasonable choice.

2.3 Feature set generation

The feature set consists of texture and shape descriptors, which are explained in the following subsections.

2.3.1 Texture features: In this work, we use the coefficients of Hartley transform (HT) for generation of texture features. A brief description about the HT is given

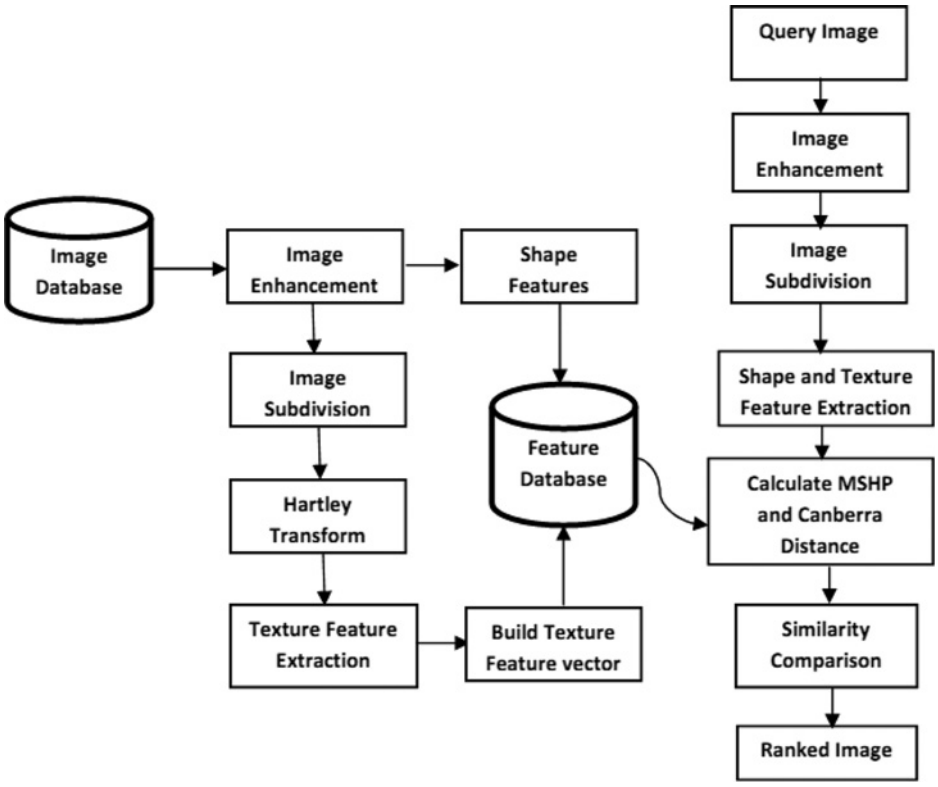

Fig. 1 Block diagram of the proposed retrieval system

doi: 10.1049/iet-ipr.2012.0104

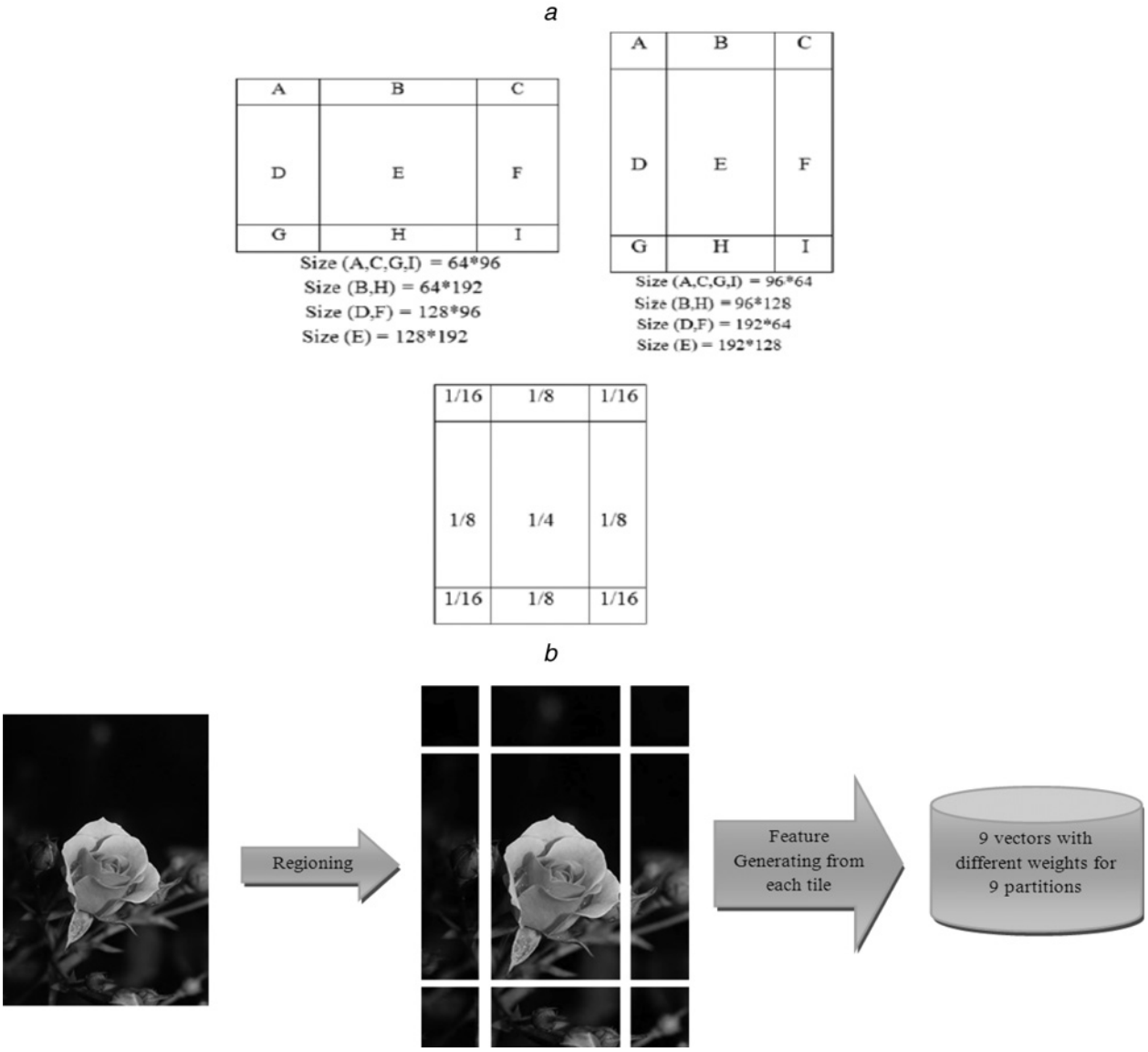

Fig. 2 *Assigned weights and image regioning for extracting the textural features*

a Image regioning for extracting textural features for two different size images and weight assignment of partitions
b Example

below. More explanations are found in [13, 14]. The basic functions of HT are orthogonal and can be computed faster than the Gabor and wavelet filters. HT generates separate sub-bands for each of the positive and negative orientations. On the other hand, the conventional separable real wavelet transform suffers from the lack of shift invariance, provides just three orientations, has a poor directional selectivity and also cannot distinguish between 45 and −45° directions. In addition, the extra redundancy of HT allows a significant reduction of aliasing terms. Translation results in large changes in the phases of coefficients, but the magnitudes (and hence energies) are much more stable. By using the even and odd filters alternately in the HT, it is possible to achieve the overall complex impulse responses with symmetric real parts and anti symmetric imaginary parts.

HT is proposed for a more symmetrical Fourier analysis, which is expressed in a symmetric form between a function of a real variable and its transform. The HT expands a function in terms of real sine and cosine terms, whereas the Fourier transform expands into complex exponentials.

The two-dimensional (2D) discrete HT pair is given by the following equations

$$H(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} g(x, y)\text{cas}\left(\frac{2\pi xu}{M} + \frac{2\pi yv}{N}\right) \quad (1)$$
$$0 \le u \le M-1, \quad 0 \le v \le N-1$$

$$g(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} H(u, v)\text{cas} \quad \left(\frac{2\pi xu}{M} + \frac{2\pi yv}{N}\right) \quad (2)$$
$$0 \le x \le M-1, \quad 0 \le y \le N-1$$

where $g(x, y)$ is a function of two integer variables x and y, $H(u, v)$ is its HT and

$$\text{cas}(\alpha + \beta) = \cos(\alpha + \beta) + \sin(\alpha + \beta) \quad (3)$$

The above equations show that the HT exhibits a self-inverse property, that is, direct and inverse transformations use the same formulation. Consequently, both the HT and inverse HT can be computed using the same algorithm. Let $G(u, v)$ be the 2D Fourier transform of the function $g(x, y)$, then the relation between $G(u, v)$ and $H(u, v)$ is given by

$$H(u, v) = \frac{1+j}{2} G(u, v) + \frac{1-j}{2} G(-u, -v) \quad (4)$$

The Fourier transform of a real function is Hermitian, that is, $G(-u, v) = G^*(u, v)$ and $H(u, v) = \text{real}[G(u, v)] - \text{image}[G(u, v)]$. If the function g is centrosymmetric, that is, $g(x, y) = g(-x, y)$, then the HT will be equivalent to the Fourier transform [14].

In the following, we explain the texture feature extraction algorithm used in this study:

1. Decompose all tiles of image into several directional sub-bands using HT.
2. Do for $k = 1: K$ (K is the number of tiles, nine tiles in this study)
3. Do for $l = 1: L$ (L is the number of sub-bands, 20 sub-bands in this study)
4. Collect directional sub-band coefficients, that is, $\boldsymbol{H}_l^k$ ($\boldsymbol{H}_l^k$ is of size 20×20).
5. Compute the energy and variance of $\boldsymbol{H}_l^k$ as the sub-band features.
6. Collect the tile feature $\bar{\boldsymbol{f}}_{\sigma E}^k$ from the sub-bands features $\bar{\boldsymbol{f}}_{\sigma E}^k$ is of size 1×40).
7. Collect the feature vector $\boldsymbol{F}_{\text{texture}}$ from the tile features $\bar{\boldsymbol{f}}_{\sigma E}^k$ (size of $\boldsymbol{F}_{\text{texture}}$ is 1×40).

As mentioned above, the energy and variance of each sub-band of HT of each tile are computed separately. The energy (E_l^k) and variance (σ_l^k) of the lth sub-band of the tile k of image are calculated as follows

$$E_l^k = \frac{1}{MN} \sum_{u=1}^{M} \sum_{v=1}^{N} \left| H_l^k(u, v) \right|^2, \quad l = 1{:}L, \quad k = 1{:}K \quad (5)$$

$$\sigma_l^k = \sqrt{\frac{1}{MN} \sum_{u=1}^{M} \sum_{v=1}^{N} \left| H_l^k(u, v) - \mu_l \right|^2} \quad l = 1{:}L, \quad k = 1{:}K \quad (6)$$

where $H_l^k(u, v)$ are the coefficient values of the lth sub-band of HT for the tile k, $M \times N$ is the size of each sub-band, which is 20×20 in our experiments and μ_l is the mean value of the lth sub-band. Our proposed main feature vector is then constructed using the sub-features of different tiles of image. A sub-feature vector for the kth tile of the regioned image is constructed using the energies ($E_l^k s$) and variances ($\sigma_l^k s$) of its L sub-bands as follows

$$\bar{\boldsymbol{f}}_{\sigma E}^k = \left[\sigma_1^k, \sigma_2^k, \ldots, \sigma_L^k, \quad E_1^k, E_2^k, \ldots, E_L^k \right] \quad (7)$$

Noting (7), the dimension of sub-feature vector will be $2L$, so for $\boldsymbol{L} = 20$ used in this work, each tile of image has a sub-feature vector of size 40. Next, we assign different weights for each of sub-feature vectors (i.e. each tile), which is depicted in Fig. 2*a* It is observed that the larger tiles are assigned larger weights. Finally, we combine the nine weighted sub-feature vectors of nine tiles to form the main feature vector as shown below

$$\begin{aligned} \boldsymbol{F}_{\text{texture}} = {} & (1/16)\, \bar{\boldsymbol{f}}_{\sigma E}^1 + (1/8)\, \bar{\boldsymbol{f}}_{\sigma E}^2 + (1/16)\, \bar{\boldsymbol{f}}_{\sigma E}^3 + (1/8) \\ & + \bar{\boldsymbol{f}}_{\sigma E}^4 (1/4)\, \bar{\boldsymbol{f}}_{\sigma E}^5 + (1/8) \bar{\boldsymbol{f}}_{\sigma E}^6 + (1/16) \; + \bar{\boldsymbol{f}}_{\sigma E}^7 (1/8) \\ & + \bar{\boldsymbol{f}}_{\sigma E}^8 (1/16)\, \bar{\boldsymbol{f}}_{\sigma \text{E}}^9 \end{aligned} \quad (8)$$

The above proposed combination keeps the dimension of the main feature vector the same as the sub-feature vector (i.e. 40), which results in less memory. Note that if we appended the sub-feature vectors of nine tiles, the dimension would be 360. This would cause too significant increase in the usage memory, which is a great challenge in large databases.

The proposed procedure is repeated for all images of database and the computed feature vectors are stored. We use the most similar highest priority (MSHP) rule for comparing the texture features of the query image and database images, which will be explained in Section 2.4.

We note that by utilising the textural features, only the textural information of image is captured and the other significant information is missed. Consequently, using the other features such as shape features will result in higher retrieval accuracy. Therefore in order to improve the accuracy rate, we add another set of features. In the following, we describe shape features.

2.3.2 Shape features: The gradient vector flow (GVF) is a static external force used in active contour method. GVF is computed as a diffusion of gradient vectors of a grey level or binary edge map derived from the image. GVF field [15] gives excellent results on concavities supporting the edge pixels with the opposite pair of forces, obeying force balance condition in one of the four directions (horizontal, vertical and diagonals) unlike the traditional external forces, which support either in the horizontal or vertical directions only. In addition, the authors in [6] used GVF for shape feature extraction. Simulation and analytical results showed the better performance of GVF than the other approaches such as hue and Zernike moments [15].

The algorithm for computation of edge image is given below [15]:

1. Convert the input colour image into the grey scale one.
2. Blur the image using a Gaussian filter.
3. Compute the GVF of the blurred image.
4. Filter out only strong edge responses using the threshold value th, where th is the variance of the GVF.
5. Repeat step d until achieving the main edge of image.

In this study, we first apply GVF and then extract the invariant moments. The translation, rotation and scale invariant 1D normalised contour sequence moments are computed on the edge image [15]. First, we consider the following moments

$$m_r = \frac{1}{B} \sum_{i=1}^{B} [z(i)]^r, \quad \mu_r = \frac{1}{B} \sum_{i=1}^{B} [z(i) - m_1]^r, \; \overline{\mu_r} = \frac{\mu_r}{(\mu_2)^{r/2}} \quad (9)$$

where $z(i)$ is the set of Euclidian distances between the centre of shape and all B boundaries of the shape, and m_r, μ_r and $\overline{\mu_r}$ are the mean, rth moment and normalised moment of $z(i)$,

doi: 10.1049/iet-ipr.2012.0104

respectively. The shape descriptors are then defined as by Xu and Prince [15]

$$F_{\text{shape\_1}} = \frac{(\mu_2)^2}{m_1}, \quad F_{\text{shape\_}}2 = \frac{\mu_3}{(\mu_2)^{3/2}}$$
$$F_{\text{shape\_}}3 = \frac{\mu_4}{(\mu_2)^2}, \quad F_{\text{shap\_4}} = \overline{\mu_5} \tag{10}$$

Therefore a total of four features ($F_{\text{shape\_1}}$, $F_{\text{shape\_2}}$, $F_{\text{shape\_3}}$ and $F_{\text{shape\_4}}$) are obtained from the above computations. The Canberra distance measure [9] is used for similarity comparison of shape features, which is given by

$$\text{canbdist} = \sum_{i=1}^{4} \frac{|V_i - W_i|}{|V_i| + |W_i|} \tag{11}$$

where $\boldsymbol{V}$ and $\boldsymbol{W}$ are the shape feature vectors of image set and query image, respectively, of size 4.

2.4 Integrated image matching

In the new method, a tile from the query image is allowed to be matched to any tile in the target image, but any tile of the query image will participate in the matching process only once. Although in the conventional methods, each tile of the query image is compared with all tiles of the target image. For clarification, consider the conventional bipartite graph [6] of tiles for the query and target images, which is demonstrated in Fig. 3. In this graph, the labelled edges of the bipartite graph indicate the distances between the tiles. Since this process involves many distance computations and comparisons, which are equal to K^2 (K is the number of tiles); hence, we propose an efficient algorithm, in which each tile participates in the matching just once. In our scheme, at first, the distances among a tile of the query image and all tiles of the target image are computed and the minimum one, (e.g. the distance between the tile i of query image and the tile j of the target), is recorded as d_{ij} and then the corresponding tiles of the query and target images are blocked. This blocking will prevent the blocked tiles of the query and target images from further participating in the matching process. In this way, each tile in the query and target images is allowed to take part in the matching only once. This process is repeated until each tile finds a good matching. Therefore the number of tiles participating in the matching in our scheme will be $K + (K-1) + \cdots + 3 + 2 + 1 = K(K+1)/2$. Comparing with the conventional matching method (K^2), the computational complexity of the new matching procedure and consequently the processing time are reduced about 1/2 which is a great advantage.

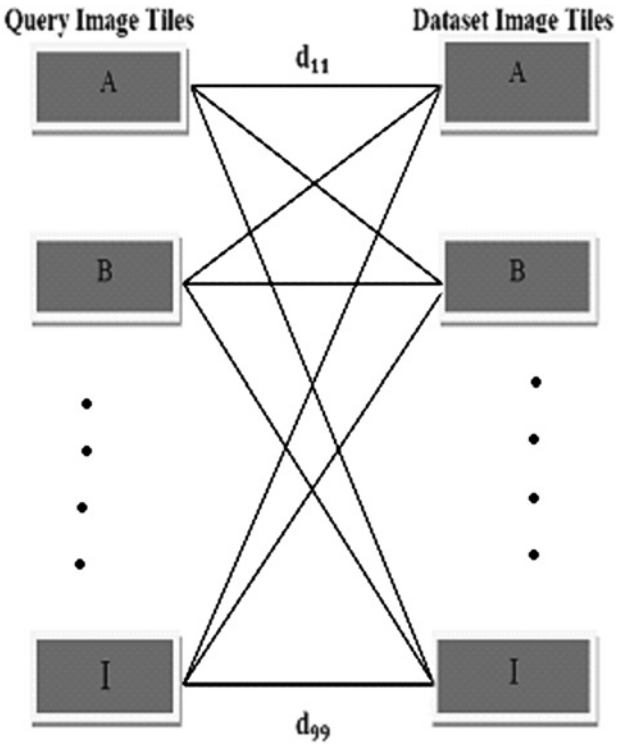

Fig. 3 *Typical bipartite graph used in the conventional methods*

We then form the adjacency matrix (D). The elements of this matrix are the computed minimum distances (d_{ij}). Next, we define the integrated minimum cost-matching distance (D_{qt}) between the query (q) and target (t) images as

$$D_{qt} = \sum_{i=1}^{K} \sum_{j=1}^{K} d_{ij} \tag{12}$$

where D_{qt} is the summation of the elements of adjacency (distance) matrix. Therefore we obtain the minimum cost matching based on the MSHP rule.

Since MSHP method is more time consuming than the Canberra distance if we use it for shape features, the retrieval time will be increased (especially in huge database such as Internet). Hence, we utilise MSHP for texture features that has more valuable information.

2.5 Retrieval scheme

After extraction of texture and shape features, we compute the MSHP and Canberra distance of the query image and all images of used database, where the MSHP rule is used for texture features and Canberra distance is utilised for shape features. Then, we select the image from the database as the target (retrieved) image that has less MSHP and Canberra distance values from the query image.

3 Experiments and results

In this section, we evaluate the performance of the proposed method. For this purpose, we use three image sets and apply our method. We also compare the performance of the new method with the previously presented methods from different points of view. Further, we examine the properties of texture features extracted from the HT.

3.1 Image sets

Image set 1: The first used database was downloaded from the website http://www.wang.ist.psu.edu/docs/related, which contains 1000 Corel images. This image set consists of ten categories where each one includes 100 different images. The images are of sizes either 256×384 or 384×256 pixels. Fig. 4 depicts some images selected randomly from the ten categories of this database and Table 1 presents the semantic names of the ten categories.

Image set 2: We have used Lin *et al.* [9] complete image set as the second database, where there are

Fig. 4 *Sample images randomly selected from ten categories of image set 1*

Table 1 Ten categories of image set 1

Category	Semantic name
1	African people
2	beach
3	building
4	buses
5	dinosaurs
6	elephants
7	flowers
8	horses
9	mountains
10	food

two sets of images; set $D = \{I_1^d, I_2^d, I_3^d, \ldots, I_{1051}^d\}$ and set $Q = \{I_1^q, I_2^q, I_3^q, \ldots, I_{1051}^q\}$. Each set contains 1051 full colour images with sizes 256×384 or 384×256 pixels. The images in the set D are utilised as the target images and the images in the set Q are used as the query images. Fig. 5 shows some of the query and target images from the image set 2.

Image set 3: This image set is the Corel database, which consists of 15 000 images with 150 categories, where each category contains 100 images. Each image is of size 256×384 or 384×256 pixels. Corel image set has been widely used by CBIR research communities. It covers a variety of topics such as 'flowers', 'buses', 'beach', 'elephants', 'sunset', 'buildings', 'horses' etc. Fig. 6 illustrates some examples of image set 3.

3.1.1 Retrieval accuracy: The criterion for computing the retrieval accuracy is the average precision [3], which is defined as

$$p(i) = \frac{1}{100} \sum_{r(i,j)\le 100,\ 1\le j\le 100,\ ID(j)=ID(i)} 1 \qquad (13)$$

where $p(i)$ is the average precision of the query image i, and $ID(i)$ and $ID(j)$ are the category numbers of images i and j, respectively. $r(i, j)$ is the rank of the image j, that is, the position of the image j in the retrieved images for the query image i, which is an integer value between 1 and 100. This value, $r(i, j)$, depicts the percentage of the images belonging to the category of the image i in the first 100 retrieved images.

The average precision for the category t is given by

$$p_t(i) = \frac{1}{100} \sum_{1\le j\le 100,\ ID(i)=t} 1 \qquad (14)$$

Table 2 demonstrates the average retrieval accuracy for the three mentioned image sets when only the textural features have been used. In this table, we have also compared the performance of the new method with the other methods such as Gabor wavelet [4], standard wavelet [6] and complex wavelet [12] transforms. As Table 2 shows, the retrieval accuracy of the proposed method is higher than those of the mentioned methods. In the following, we assess the performance of the proposed method, in which both texture and shape features are utilised.

3.2 Performance evaluation on image set 1

As mentioned, we select the image from database as the target image that has the minimum values of MSHP and Canberra distance from the query image.

Fig. 5 *Examples of image set 2*

a Some query images
b Database images corresponding to the query images in Fig. 5*a*

doi: 10.1049/iet-ipr.2012.0104

Fig. 6 *Sample images randomly selected from 150 categories of image set 3*

Table 2 Average retrieval accuracy of image sets 1, 2 and 3 when only the textural features are used (100 returned images)

Image set	Gabor wavelets [4], %	Standard wavelets [6], %	Complex wavelet [12], %	Proposed method, %
1	42.2	40.1	52.3	55.78
2	75.7	70.3	80.1	83.57
3	39	36.5	49.65	53.8

The performance of the proposed method is shown in Table 3. For comparison, we have presented the performance of other retrieval systems introduced in [3, 4, 6, 7, 9]. It is obvious that the proposed method achieves more accuracy than the methods of [3, 4, 6, 7] in all categories. The fuzzy integrated region matching (FIRM) [3] and semantics-sensitive integrated matching for picture libraries (SIMPLICITY) [4] algorithms are both segmentation based methods. Since in these methods, the textured and non-textured regions are treated differently with different feature sets, their results are claimed to be better than the histogram-based method [7]. Also, the performance of edge-based system in [6] is better than the SIMPLICITY [4] and histogram based [7] methods and is a little less than the FIRM [3] method. Moreover, the method in [9] outperforms the methods of [3, 4, 6, 7]. Further, in comparison between the proposed method and the recently presented method of Lin *et al.* [9], we observe that in all categories except the category seven (flower category), our method has better performance. As an example of performance comparison, consider the African category which is a more complex category and many efforts have been made in the literature to improve its accuracy. We

Table 3 Comparison of average precision (%) of the proposed method and other retrieval systems presented in [3, 4, 6, 7, 9] when 100 images of set 1 are retrievable

Semantic name of class	FIRM [3], %	SIMPLICITY [4], %	Edge based [6], %	Histogram based [7], %	Lin method [9], %	Proposed method, %
Africa	47	48	45	30	48	54
beaches	35	32	35	30	44	46
building	35	35	35	25	36	39
bus	60	36	60	26	69	70
dinosaur	95	95	95	90	96	99
elephant	25	38	25	36	55	60
flower	65	42	65	40	89	83
horses	65	72	65	38	70	75
mountain	30	35	30	25	42	48
food	48	38	48	20	53	58
average	50.5	47.1	50.3	36	60.2	63.0

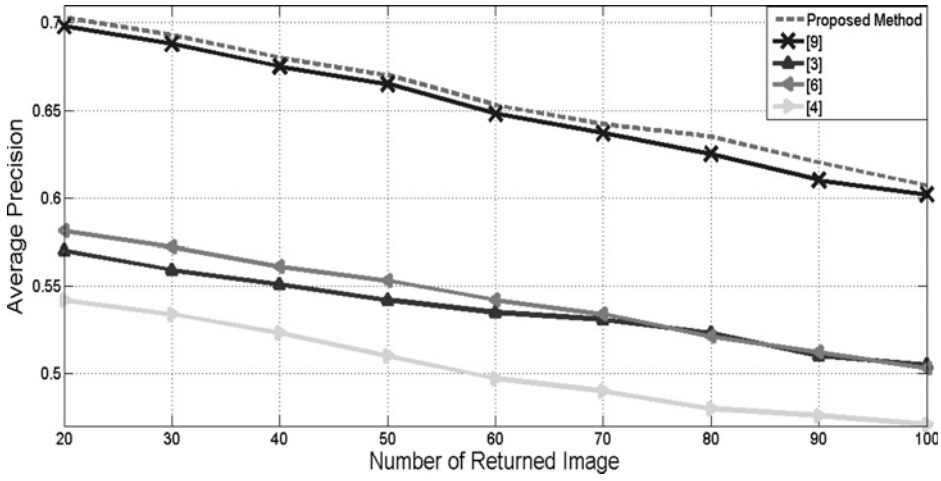

Fig. 7 *Average precision of different methods for image dataset 1*

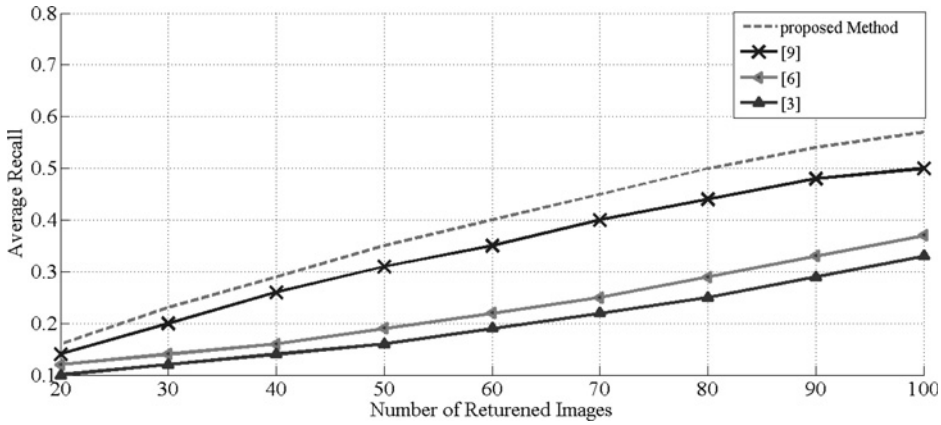

Fig. 8 *Average recall of different methods for image dataset 1*

observe that for this category, our method achieves 54% accuracy, while the other methods including [9] yield less accuracy.

The average accuracy of retrieval and the average recall of different methods against the number of returned images are demonstrated in Figs. 7 and 8, respectively, for image sets 1. The experimental results clearly reveal that for the first 20–100 returned images of the 1000 images of the ten categories, the new method outperforms the methods of [3, 4, 6, 9].

3.3 Performance assessment on image set 2

As mentioned, in image set 2 there are two sets of images, the images in set D are employed as the target images and those in set Q are used as the query images. Each image pair (I_i^d, I_i^q) is randomly chosen from the set D and set Q, respectively. Each I_i^q is used as the query image and for each query image, the system finds a predefined number of images from the set D, which have less Canberra distance and MSHP from the query image. If I_i^d exists among the found images, we say that the system has correctly recognised the expected image, otherwise, the system has failed.

Table 4 demonstrates the retrieval accuracy of different methods. We observe that the proposed method has much better performance than the methods of [9, 10, 11]. It is also obvious that the retrieval precision increases as the number of returned images increases.

3.4 Performance evaluation on image set 3

Figs. 9 and 10 demonstrate the average precision and the average recall of different methods for image set 3,

Table 4 Comparison of the average recall (%) of the proposed method and the other commonly used retrieval systems when different numbers of images are retrieved (image sets 2 and 3)

Number of recalls	1	2	3	4	5	10	20	30	50	100
Huang and Dais [10]	65.3	72.2	74.7	77.0	78.1	83.5	86.2	88.4	92.0	94.7
Jhanwar *et al.* [11]	62.4	70.7	74.8	76.6	79.0	84.0	87.7	90.2	92.3	94.6
Lin *et al.* [9]	85.5	90.5	92.3	93.2	93.6	95.1	96.7	97.7	98.9	99.2
proposed method (image set 2)	88.6	93	94.9	95.4	95.6	96.7	97	98.1	99	99.7
proposed method (image set 3)	86.1	90.9	91	93.75	94.2	94	97.10	96.8	97	99.5

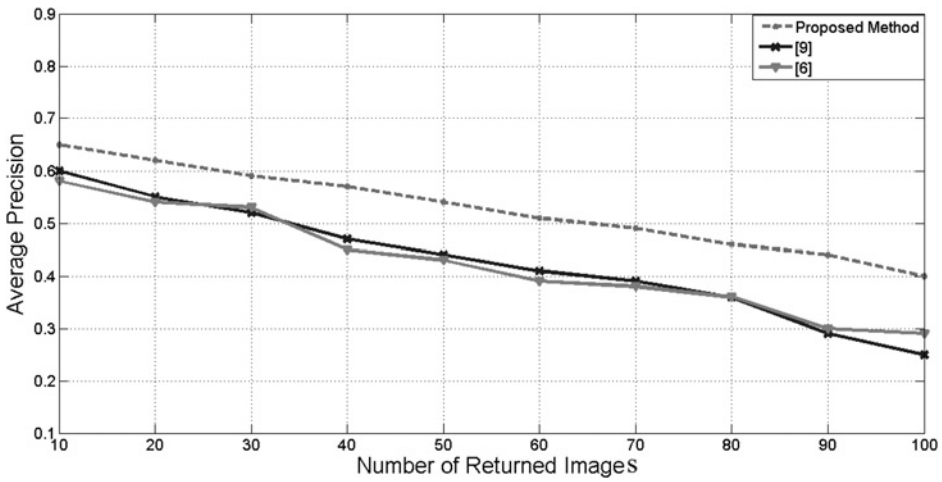

Fig. 9 *Average precision of different methods for image set 3*

doi: 10.1049/iet-ipr.2012.0104

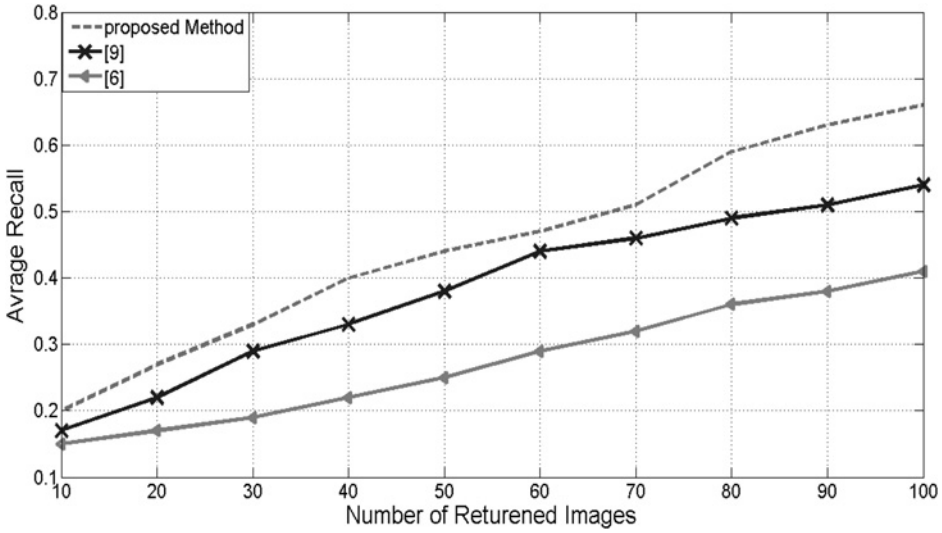

Fig. 10 *Average recall of different methods for image set 3*

respectively. It is obvious that the proposed method outperforms the methods of [6, 9]. Also, from Table 4, we observe that the average recall accuracy of the proposed method is much better than the methods of [9, 10, 11].

3.5 Computational cost

Table 5 depicts the computational complexity of our method and the method of Lin *et al.* [9], which has better performance than the methods of [3, 4, 6, 7] for image set 1. It is observed that the proposed method has the same number of multiplications and comparisons as those of Lin *et al.* [9]. However, the number of additions in our method is reduced by the factor of 1.4.

We have also compared the processing times of our method and Lin *et al.* [9]. It is worth mentioning that we have simulated the methods of [6, 9] on the same computer that we evaluated our work. Therefore the comparison is fair. The methods were executed on a 1.7 GHz Corei7 CPU processor with 4 Gigabyte RAM. The experimental results show that the proposed retrieval method takes 38 ms and the method of Lin *et al.* [9] needs 44 ms for each image in the image set 1. Thus, our approach is 1.2 times faster than the method of Lin *et al.* [9]. Consequently, the processing time of the new retrieval system will be much less than that of Lin *et al.* [9] for huge image sets. Since the methods of [6, 9] reported that their run times are less than [3, 4], hence we can conclude that the proposed method needs less run time compared to the methods of [3, 4].

Table 6 provides the feature vector length and processing time for feature vector computation. It is also worth mentioning that, in our approach, the dimension of the main feature vector is less than the Gabor wavelet [4] and method of Lin *et al.* [9] and equals to that of the standard

Table 5 Computational complexity per pixel for image set 1

	Number of additions	Number of multiplications	Number of comparisons
Lin *et al.* [9]	47.9	18.6	1.5
proposed method	38.2	18.6	1.5

Table 6 Feature vector length and feature extraction time of different methods for a query image in image set 1

	Gabor wavelets [4]	Standard wavelet [6]	Lin method [9]	Proposed method
feature vector length	42			
(6 orientations, 7 scales)	40	variable from 50 to larger lengths	40	
feature generation time, s	3.54	0.61	1–3	0.7

Table 7 Rotation invariant property of HT

Class no.	Rotation in degree	Average value of σ	Average value of E
1	0 (no rotation)	1.125	0.764
	20	1.130	0.654
	45	1.121	0.681
	55	1.127	0.673
	90	1.124	0.762
2	0 (no rotation)	3.298	0.948
	20	3.254	1.004
	45	3.292	0.975
	55	3.227	0.899
	90	3.200	0.999
3	0 (no rotation)	2.443	1.693
	20	2.554	1.558
	45	2.834	1.449
	55	2.367	1.561
	90	2.659	1.487
4	0 (no rotation)	1.547	0.359
	20	1.582	0.398
	45	1.345	0.437
	55	1.576	0.429
	90	1.465	0.354
5	0 (no rotation)	4.212	1.012
	20	4.324	0.924
	45	4.491	0.896
	55	4.285	0.999
	90	4.349	1.101

Table 8 Effect of shift of features extracted from HT

Shift values	$MSPE_1$		$MSPE_2$		$MSPE_3$	
	For σ features	For E features	For σ features	For E features	For σ features	For E features
5	0.190	0.209	0.192	0.207	0.183	0.192
10	0.210	0.213	0.182	0.197	0.191	0.184
15	0.219	0.208	0.194	0.213	0.186	0.189

wavelet [6]. We note that increasing the dimension of feature vector will produce higher retrieval results but at the cost of increase in memory usage, that is, there is a trade-off between the accuracy and complexity. It is also observed that our method needs less feature generation time than the Gabor wavelets [4] and Lin method [9] while it has better performance.

3.6 Properties of features extracted from HT

We examine the properties of features obtained from HT in this study and show that they are rotation invariant and rather shift invariant. For this purpose, we have carried out the experiments for five classes of image set 1.

To demonstrate the rotation invariant characteristic, we perform the following steps:

(a) Rotation of image by the specified values of 0, 20, 45, 55 and 90°.
(b) Feature extraction of each rotation using HT.
(c) Computation of the values of variance σ and energy E in each rotation for different images of each class and then averaging.
(d) Comparison of the average values of σs and Es obtained in step (c).

Simulation results are listed in Table 7. It is observed that the average values of σ and E for different rotations in each class are almost the same which shows that the features extracted from HT in this study are invariant against rotation.

We have examined the effect of shift on the extracted features by the mean-squre percentage error (MSPE) criterion which is defined as the error between the extracted features and the features circularly shifted by a defined value, then averaged over database:

$$\text{MSPE} = \frac{1}{\boldsymbol{T}} \sum_{t=1}^{\boldsymbol{T}} \frac{||F_t - A_t||}{||F_t||} \tag{15}$$

where F_t and A_t are the features before and after shift, respectively, and $\boldsymbol{T}$ is the length of feature vector. Less values of MSPE indicate that the features are more invariant to shift. In this experiment, the extracted features [(7) and (8)] are divided into two sub-features.

1. Sub-feature vector for $\boldsymbol{\sigma}$ with size of 20.
2. Sub-feature vector for $\boldsymbol{E}$ with size of 20.

The reason for dividing the main feature vector into two sub-feature vectors is that the natures of $\boldsymbol{\sigma}$ and $\boldsymbol{E}$ features are different. Hence, using MSPE for the main feature vector will not be meaningful. Therefore we consider the MSPE for each sub-vector. We have used the shifts of 5, 10 and 15. In Table 8, we have shown the MSPE for the orientations of 30° (denoted by $MSPE_1$), 45° (shown as $MSPE_2$) and 65° (depicted as $MSPE_3$). It is observed from the table that, in this work, the obtained features from HT are rather invariant to shift.

4 Conclusion

We proposed a new method for image retrieval. The images were partitioned into nine non-overlapping tiles. An efficient integrated matching scheme among the tiles of the query and database image was implemented for image similarity measurement, which reduced the matching complexity by 1/2. The HT for each of the non-overlapping tiles was computed and the energy and variance of coefficients of tiles were used as the texture features. Then, the invariant moments of edge image were used to describe the shape features. A combination of texture and shape features provides a robust feature set for image retrieval. The experiments performed on three image sets demonstrate the efficiency of the proposed method in comparison with the existing methods, that is, the new method achieves more retrieval accuracy. The dimension of feature vector in our approach is the less than the other approaches. Further, the computational complexity and processing time of the proposed method are less than the previously presented methods, which is an advantage in CBIR systems.

5 References

1 Sedghi, T., Fakheri, M., Shayesteh, M.G.: 'Region and content based image retrieval using advanced image processing technique'. Proc. IEEE, Machine Vision and Image Processing Symp., 2010
2 Sedghi, T., Fakheri, M., Amirani, M.: 'Features composition for proficient and real time retrieval in content based image retrieval system'. Proc. IEEE, Machine Vision and Image Processing Symp., 2010
3 Chen, Y., Wang, J.Z.: 'A region-based fuzzy feature matching approach to content-based image retrieval', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2002, **24**, (9), pp. 1252–1267
4 Li, J., Wang, J.Z., Wiederhold, G.: 'IRM: integrated region matching for image retrieval'. Proc. Eighth ACM Int. Conf. on Multimedia, October 2000, pp. 147–156
5 Rui-zhe, Z., Jia-zheng, Y., Jing-hua, H., Yu-jian, W., Hong, B.: 'A novel generalized SVM algorithm with application to region-based image retrieval'. Proc. Int. Forum on Information Technology and Applications, 2009, pp. 280–283
6 Hirremath, P.S., Pujari, J.: 'Content based image retrieval based on texture and shape features using image and its complement'. Proc. IJCSS, 2008, vol. 2, pp. 25–35
7 Rubner, Y., Guibas, L.J., Tomasi, C.: 'The earth mover's distance, multi-dimensional scaling, and color-based image retrieval'. Proc. DARPA Image Understanding Workshop, 1997, pp. 661–668
8 Hoiem, D., Sukhtankar, R., Schneiderman, H., Huston, L.: 'Object-based image retrieval using statistical structure of images'. Proc. CVPR, 2004
9 Lin, C., Chen, R., Chan, Y.: 'A smart content-based image retrieval system based on color and texture feature', *Image Vis. Comput.*, 2009, **27**, pp. 658–665
10 Huang, P.W., Dai, S.K.: 'Image retrieval by texture similarity', *Pattern Recognit.*, 2003, **36**, pp. 665–679

doi: 10.1049/iet-ipr.2012.0104

11 Jhanwar, N., Chaudhurib, S., Seetharamanc, G., Zavidovique, B.: 'Content based image retrieval using motif co-occurrence matrix', *Image Vis. Comput.*, 2004, **22**, pp. 1211–1220
12 Kingsbury, N.: 'Complex wavelets for shift invariant analysis and filtering of signals', *J. Appl. Comput. Harmon. Anal.*, 2008, **10**, (3), pp. 234–253
13 Rajavel, P.: 'Directional Hartley transform and content based image retrieval', *Signal Process.*, 2010, **90**, pp. 1267–1278
14 Millane, R.P.: 'Analytic properties of the Hartley transform and their implications'. Proc. IEEE, 1994, pp. 413–428
15 Xu, C., Prince, J.L.: 'Snakes, shapes, and gradient vector flow', *IEEE Trans. Image Process.*, 2005, **7**, pp. 359–369