**Conference or Workshop Item:**

Hodge, Victoria Jane orcid.org/0000-0002-2469-0224, Hollier, Garry Phillip, Austin, Jim orcid.org/0000-0001-5762-8614 et al. (1 more author) (2008) Identifying Perceptual Structures In Trademark Images. In: Proceedings Fifth IASTED International Conference on Signal Processing, Pattern Recognition and Applications, 13-15 Feb 2008.

# IDENTIFYING PERCEPTUAL STRUCTURES IN TRADEMARK IMAGES

Victoria J. Hodge, Garry Hollier, Jim Austin & John Eakins
Advanced Computer Architectures Group
Dept of Computer Science, University of York
Heslington, York, YO10 5DD
UK
{vicky, hollier, eakins, austin}@cs.york.ac.uk

**ABSTRACT**
In this paper we focus on identifying image structures at different levels in figurative (trademark) images to allow higher level similarity between images to be inferred. To identify image structures at different levels, it is desirable to be able to achieve multiple views of an image at different scales and then extract perceptually-relevant shapes from the different views. The three aims of this work are: to generate multiple views of each image in a principled manner, to identify structures and shapes at different levels within images and to emulate the Gestalt principles to guide shape finding. The proposed integrated approach is able to meet all three aims.

**KEY WORDS**
Image segmentation and representation, perceptual shape finding.

## 1 Introduction

Computerised image retrieval takes a query image and attempts to find all matching images: images which might be deemed similar to the query image by a human analyst. Most experts agree that shape similarity is the most important determining factor for figurative (trademark) image similarity in humans [1]. In this paper, we focus on the task of using computerised methods to find shapes in trademark images to allow image similarity matching and retrieval that emulates human matching. However, human image similarity is not just determined by the similarity of simple image shapes but also encompasses higher-level patterns (structures) made by the individual shapes following the Gestalt principles such as similarity, proximity or continuity [2]. Thus, we introduce an approach for finding patterns (structures and shapes) in trademark images, at different perceptual levels emulating the Gestalt principles. The Gestalt principles refer to the shape-forming capability of human vision. In particular, they refer to the visual recognition of structures and whole shapes rather than just 'seeing' a simple collection of lines and curves. Hence a computerised image retrieval system must be able to identify and match the most salient aspects of an image's appearance including: the image's overall shape, the shapes of important image components or shapes defined by perceptually significant groupings of components.

Finding perceptual structures and shapes requires generating image representations (views) at different levels. This is a difficult task that requires a "semantic" level of understanding and a number of different processing methods as no one technique is ubiquitous. By integrating a series of techniques, we aim to overcome the limitations of each individual technique while exploiting their strengths. In IBM's QBIC system [3] each image in the database has multiple representations achieved through the use of different feature spaces of an image rather than by generating new views at different scales. French et al. [4] introduce an image retrieval system that employs multiple image representations and then consolidates the results of matching the different representations to produce a ranked list of results. We take our cue from French et al. [4] and generate multiple views of the image. We use scale space selection [5] and Gaussian pyramids [6] to blur the image followed by pixel clustering to extract the image structures at different levels. After clustering, we identify the shapes and structures within the image views using edge segmentation and linking that obeys the Gestalt principles of continuity and proximity. We thus have a set of image views for each image and each view has a set of shapes. These sets represent the shapes present in the image at different perceptual levels.

## 2 View generation and shape identification

Sections 2.1-2.4 describe how we merge lower level shapes and texture within the image to extract structures and produce perceptual views of the image. Section 2.5 describes a shape identification algorithm to determine the shapes present in these views and to identify other perceptual structures missed by the view generation step.

### 2.1 Scale Space representation

The first step for generating multiple perceptual views is image scaling. Scaling an image by different amounts allows us to identify different levels of structure within the image by blurring (merging) lower level structures and thus revealing the higher level structures, for example removing texture and grouping shapes. Here we develop the scale-space method of Lindeberg [5] which automatically selects the optimum scaling factor.

The *scale-space representation* for a 512x512 pixel 2-D image ( $I \in \Re^2$ ) of continuous $f : \Re^2 \rightarrow \Re$ where *f (x, y)* is the pixel intensity at *(x, y)* is $L : \Re^2 \times \Re_+ \rightarrow \Re$ which is given by the solution of the diffusion eqs 1 and 2.

$$\partial_t L = \frac{1}{2}\nabla^2 L = \frac{1}{2}\sum_{i=1}^{D} \partial_{x_i x_i} L \quad \textbf{(1)}$$

with $L(\mathbf{x},0) = f(\mathbf{x})$ where $\mathrm{x} = (x, y) \in \Re^2$

$$L(\mathbf{x},t) = (g(\cdot,t) * f(\cdot))(\mathbf{x}) \quad \textbf{(2)}$$

$g(\mathbf{x},t) = \frac{1}{(2\pi t)^{D/2}} \mathrm{e}^{-\mathbf{x}^{\mathrm{T}}\mathbf{x}/(2t)}$ and $*$ is the convolution operation. The scale parameter $t \in \Re_+$ corresponds to the square of the standard deviation of the kernel $t=\sigma^2$. We are interested in the significant structures' edges in the image so we choose the normalised Laplacian which is a "general purpose" edge-detector. We look for maxima (with respect to *t*) of $t\nabla^2_{\mathbf{x}} L(\mathbf{x},t)$ , where *L* is the scale-space representation of *f*, and *f* is the pixel intensity pattern of our image. In terms of the more usual spread of a Gaussian, we look for maxima (with respect to $\sigma^2$) of $\sigma^2 \nabla^2_x L(x,\sigma^2)$ .

To look for these maxima, Lindeberg either: selects a fixed point (e.g., the image centre), or follows the spatial maxima through the image as they move with increasing *t*. To avoid the heavy processing required by the second approach while also reducing the possibility of missing scales by using the first approach, we choose several fixed points in the image. Therefore, the values $\sigma^2_{ij}, j=1,\ldots,J_i$ are our candidate scales taken from 25 equally spread sample points $\mathbf{x_i}$. We also limit the permissible scales in $\{\sigma\}$ to between 2 and 24. Allowing higher values causes the image to be too blurred to be useful for image structure segmentation purposes.

We now have a set of candidate scales $\{\sigma\}$ for the 25 sample points. We take the histogram of $\{\sigma\}$ to identify the optimal scale to use to process the image and smooth this histogram with a 3-value kernel *{1, 2, 1}* to remove perturbations. The *{1, 2, 1}* kernel assigns a higher weighting to the central (chosen) value and a lower weighting to its two direct neighbours thus allowing us to select our optimum scale. The $\sigma$ corresponding to the first highest peak in the histogram is taken as our final scale.

## 2.2 Gaussian Pyramids

In this stage the aim is to determine informative image scales to identify structures in images. Scale-space selection identifies informative scales but can be inconsistent due to the chance placement of the 25 sample points leading to under or over generalisation of the regions surrounding each sample point. Conversely, the Gaussian Pyramid is consistent across images but uses fixed scale values meaning it cannot adapt to different scales and may miss structures. Therefore, we introduce the pyramid as a pre-processor to provide consistency by pre-smoothing images to increase their similarity prior to scale selection.

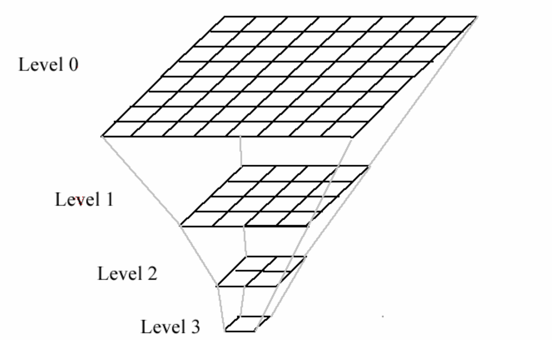



**Fig. 1. The multiple levels of the Gaussian pyramid where the filtered image levels effectively form an inverted pyramid structure.**

The pyramid takes an image $G_0(x, y)$ and convolves the image with a Gaussian kernel (low-pass filter) to produce image $G_1(x, y)$. The derived image $G_1(x, y)$ is then convolved with the kernel to produce $G_2(x, y)$ which is then processed to produce $G_3(x, y)$. For our pyramid implementation, we use 4 levels $G_0$, $G_1$, $G_2$, $G_3$ with dimensions 512x512, 256x256, 128x128, 64x64 pixels respectively as shown in Fig. 1.

If $I \in \Re^2$ is the original 512x512 pixel 2-D image then the pyramid is computed as eqs 3 and 4:

$$G_0(x, y) = I(x, y) \quad \textbf{(3)}$$
$$G_{i+1}(x, y) = FILTER(G_i(x, y)) + RESIZE(G_i(x, y)) \quad \textbf{(4)}$$

For the *FILTER* function, we use the standard Gaussian function in eq 5:

$$f_{\sigma,n}(x) = \frac{\partial^n}{\partial x^n}\frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{e^2}{2\sigma^2}} \quad \textbf{(5)}$$

where we set $\sigma^2 = 3$.

Filtering is followed by *RESIZE* which resizes $G_i$ by scale factor 0.5 to give $G_{i+1}$ using separable spline interpolation algorithm described in [7]. We found that resizing without interpolation over-emphasises jagged lines in images by increasing the aliasing.

The next processing step is to divide each blurred variant of the image into regions (structures). We use pixel intensity categorisation to identify the structures.

## 2.3 Categorisation

To categorise (cluster) the pixels, we take our cue from Lu and Chung [8] who proposed a hill-clustering method for determining the number of texture clusters. So, for each pyramid level $G_i(x, y)$, the scale ($\sigma$) is selected and the image is blurred with a Gaussian kernel of size $\sigma$

giving $B_i(x, y)$. From $B_i(x, y)$, we generate a histogram of pixel greyscale intensity values (divided into 255 bins). This raw histogram needs smoothing using a one-dimensional Gaussian with standard deviation 10 bins (1 pixel width) before it is usable. We then choose the $N$ highest peaks ($N$ categories) of the smoothed histogram and set thresholds midway between neighbouring peaks which should reflect the larger-scale structures in the image as shown in Fig. 2.

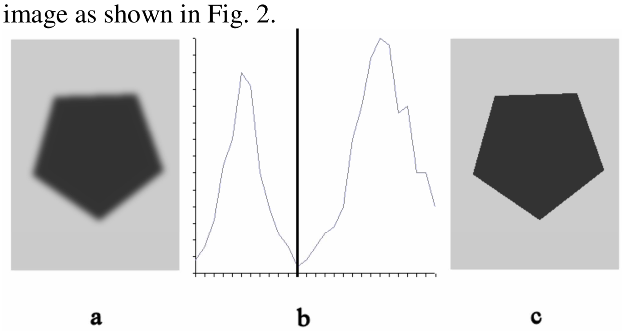



**Fig. 2. (a) is the source image. (b) is this image's pixel intensity histogram with the pixel intensity threshold drawn for k=2 categories - the trough in the histogram identifies the threshold (category boundary). (c) shows the result of categorisation.**

Previous pixel categorisation work [9] tends to rely upon a pre-specified maximum number of categories $M_{max}$. The optimum number of categories is then determined by segmenting the image into $k$ categories for $2 \leq k \leq M_{max}$ and using some suitable criterion to select the optimum [9] which is laborious. We employ a simple heuristic which we developed following detailed analysis of the pixel intensity peaks of 450 trademark images used in [10]: sort the peaks into peak intensity order and if the peak value is less than *100* then do not include the peak. This resets $M_{max}$ to the $k$ peaks with values greater than *100*. This value (*100*) was derived through a series of analyses. It is a trade-off: too high a value causes some images to have too few or even 0 categorisations. Too low a value causes too many categorisations for some images. We then identify the *2* highest peaks, *3* highest peaks up to $M_{max}$ highest peaks and divide the image into a series of views (image representations) with $2, 3 \ldots M_{max}$ categories per view. The result is a series of categorised views where pixels of similar intensity are grouped to reveal the structures within the image.

### 2.4 View Generation

It is desirable to differentiate line/region images from noisy/textured images and treat the two types differently. Line and region images require merging of lower level image structures (shapes) to infer the higher level structures. Textured and noisy images require the texture or noise to be effectively blurred out to produce a homogeneous region to represent the structure (shapes and regions) in the image. We specify $M_{max}$ as 2 for line and region–based images that are bicolour (black and white) and $M_{max}$ as 4 for texture/noisy or grey-scale images. Note $M_{max}$ may be reset if there are fewer than 4 peaks over *100*. We have erred on the side of caution by allowing 4 categories to ensure all views are found while potentially some unwanted views may be generated.

For this operation we use the Laplacian pyramid $L_0$, operator, which represents the difference of Gaussians ($G_0$-$G_1$) [6]. This is essentially an edge detection of $G_0$ and is given in eq 6:

$$L_0(x, y) = G_0(x, y) - RESIZE(G_1(x, y)) \qquad \textbf{(6)}$$

We can exploit the energy of $L_0$ to differentiate the types as textured/noisy images will have a higher energy (more edges) compared to line/region images. Following visual analyses of the energy levels of: the decompositions seen by humans in 84 trademark images in a set of experiments [11], the decompositions seen by humans in 63 trademark images in a set of experiments [12] and a further set of 450 images comprising clean, noisy and textured images [10], we use the following processing steps for the two types of images:

First, calculate the energy of $L_0$ as in eq. 7.

$$Energy = \sqrt{\sum_{\forall x,y} p(x, y)^2} \qquad \textbf{(7)}$$

where $p(x, y)$ is the greyscale value of pixel ($x, y$) in $L_0$.

Then apply the following decision rules:
If *energy* < *9600* then process the image as a region-based/line-based.
If *energy* ≥ *9600* then process the image as a textured/noisy image.

We then process these selections as follows:

**For region/line-based images**

- $G_0$ – unprocessed.
- $G_2$ – straight categorisation of $G_2$ image – no scale selection.
- $G_3$ – select scale (kernel width), convolve Gaussian (σ) with $G_3$ image, categorise resulting convolved image.

**For texture/noisy images**

There is a tendency for $\sigma_0 == \sigma_2$ in textured/noisy images where $\sigma_0$ is the scale selected for $G_0$ and $\sigma_2$ is the scale selected for $G_2$. During our analyses, we found that $G_0$ and $G_2$ were the best levels of the Gaussian pyramid to process for textured images. However, if $\sigma_0 == \sigma_2$ this would produce virtually identical outputs when $G_0$ and $G_2$ were convolved with equivalent kernels and is not desirable. Accordingly, we test for equivalence and alter our processing strategy accordingly.

- If ($\sigma 0$ <> $\sigma 2$) then
    - $G_0$ – select scale *(kernel width)*, convolve Gaussian ($\sigma_0$) with $G_0$ image, categorise resulting convolved image.

- o $G_2$ – select scale *(kernel width)*, convolve Gaussian ($\sigma_2$) with $G_2$ image, categorise resulting convolved image.

- If ($\sigma_0 == \sigma_2$) then
  - o $G_0$ – select scale *(kernel width)*, convolve Gaussian ($\sigma_0$) with $G_0$ image, categorise resulting convolved image.
  - o $G_3$ – straight categorisation of $G_3$ – no scale selection.

## 2.5 Shape Identification

In sections 2.1-2.4, we have produced various views of an image with the aim of merging lower level shapes and texture to pinpoint perceptual structures. Next we identify shapes in this data. Our image structure-finding approach uses a closed shape identification algorithm. The method adapts and refines Saund's closed shape identification algorithm [13]. By doing this, the approach can find higher level (perceptual) shapes.

Initially, the closed shape algorithm requires an underlying technique to identify the edge segments within an image and to detect the relationships between those edge segments. We resize the multiple views generated to 2048x2048 pixels from 512x512 to ensure edge separation as all structures will be at least 4 pixels wide and the structure's edges will not be adjacent. If the edges are in adjacent pixels then tracing the shapes is difficult as it is not clear which edge a pixel belongs to. We resize with no interpolation to prevent blurring of the edges in the view as blurred edges will confuse the edge detector. We find the edges in the image using a simple Laplacian edge detector before subdividing these edges into constant curvature segments (CCSs) using the Wuescher & Boyer [14] curve segmentation algorithm. This aggregates edge primitives into more perceptually-oriented CCSs. We have refined and improved the technique by increasing the tidying of the edges prior to edge segmentation to ensure there are no gaps or errors in the edges and tailoring the parameter settings to trademark images to improve the quality of the CCSs produced.

These CCSs thus provide the building blocks for our closed shape identifier as in fig 3. Our aim is to group these CCSs using Gestalt-like methods to produce a graph of CCS relations which will underpin the Saund closed shape identification algorithm. Each CCS becomes a node in the graph with two ends (first point - denoted as an x, y coordinate and last point - also denoted as an x, y coordinate). We find all segments that are end-point proximal. We extract endpoint proximity by comparing CCSs. We have evaluated various distances (in pixels) to use for end-point proximity calculations and found the following performed optimally with respect to finding perceptual shapes and structures.

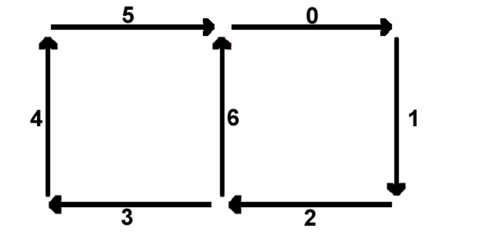



**Fig. 3. A set of CCSs (0-6). The arrow heads denote the first end of the line segment and the opposite end of the line segment is hence the last end.**

If dist($CCS_1$,$CCS_2$) < 32 pixels then $CCS_1$ and $CCS_2$ are end-point proximal. If dist($CCS_1$,$CCS_2$) < 256 and the difference between the gradients of the lines (or the terminal gradients of curves) is within ±5° then $CCS_1$ and $CCS_2$ are end-point proximal (and continuous). This effectively joins the graph by linking the proximal end-points and mimics human perception by allowing a wider gap between continuous pairs than non-continuous pairs of CCSs. Note that we differentiate CCS ends (first, last) and only allow one end-point proximity between $CCS_{1\_last}$ and $CCS_2$ to prevent cycles. We always use the closest so if dist($CCS_{1\_last}$, $CCS_{2\_first}$)=10 and dist($CCS_{1\_last}$, $CCS_{2\_last}$)=11 then the proximity is $CCS_{1\_last} \rightarrow CCS_{2\_first}$ even though dist ($CCS_{1\_last}$, $CCS_{2\_last}$) < 32.

Our closed shape algorithm overlays this graph. The search commences from each end (first and last) of each node (CCS). For each end (first then last) in turn, all possible paths are followed. This effectively forms a search tree with paths through the tree representing the possible shapes present in the image, see fig 4.

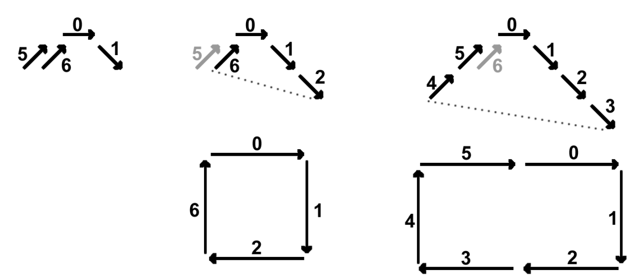


**Fig. 4. The search tree for the set of CCSs in Fig. 3. The left tree shows the tree after expanding each end of node 0 (root). The middle tree shows how, when the tree is expanded by node 2, a closed path is found - 0126. When 2 is expanded, although 6 is end-point proximal it is not added as it is already present on the opposite side of the tree. The right tree shows the tree expanded by node 4 and node 3. A second closed path is identified - 012345.**

The search is managed through the use of scores for ranking possible paths through junctions such as t-junction or crossroads, see table 1. We have revised the junction scores used by Saund to improve the quality of the results for figurative images and to make the algorithm more consistent. We used the results from our previous work involving human experiments [11] to derive our new junction scores. During path search and scoring, we separate straight paths from turning paths using the table of scores depending on whether the path is: turning clockwise (CW) or anticlockwise (ACW); OR straight clockwise or anticlockwise. Each path accumulates a score using the score from each junction it passes through. Our path scores are an average of the junction scores. Saund's uses a cumulative (product) calculation but this favours short paths whereas we allow longer paths to be explored. We have a minimum score threshold (0.6 for straight paths and 0.8 for turning paths), compared to 0.6 and 0.9 respectively for Saund. As soon as the average score for a path falls below the minimum score, we terminate the search on that path. These minimum scores were derived from a series of analyses using the images from [10].

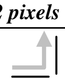
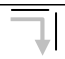
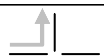
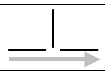
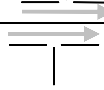
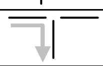
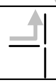
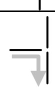
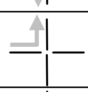
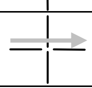
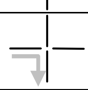
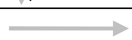
| Junction | Turning ACW | Turning CW | Straight ACW | Straight CW |
|---|---|---|---|---|
| ***dist($CCS_1$,$CCS_2$) < 2 pixels*** | ***1.0*** | ***1.0*** | ***1.0*** | ***1.0*** |
|  | 1.0 | 0.7 | 1.0 | 0.7 |
|  | 0.7 | 1.0 | 0.7 | 1.0 |
|  | 1.0 | 0.5 | 0.9 | 0.5 |
|  | ***1.0*** | 1.0 | 1.0 | 1.0 |
|  | 1.0 | ***1.0*** | 1.0 | 1.0 |
|  | 0.5 | 1.0 | 0.5 | 0.9 |
|  | 1.0 | 0.5 | 0.9 | 0.5 |
|  | 0.5 | 1.0 | 0.5 | 0.9 |
|  | 1.0 | 0.5 | 0.6 | 0.5 |
|  | ***1.0*** | ***1.0*** | 1.0 | 1.0 |
|  | 0.5 | 1.0 | 0.5 | 0.6 |
|  | 1.0 | 1.0 | 1.0 | 1.0 |

**Table 1. A table of the shape finding junction scores. Each row represents a junction configuration such as t-junction or crossroads. The arrow indicates the path direction through the junction. The bold scores differ from Saund's scores.**

As each leaf node in the tree is expanded, new child nodes are compared with child nodes in the opposite side of the tree. If they are end-point proximal then a closed path (a cycle) has been identified and its nodes and boundary pixels are added to the list of candidate paths. To produce the set of shapes for each image in this paper, we accept all candidate paths; only repetitions are removed. We have produced a perceptual relevance classifier that can rank or classify shapes as perceptually relevant or irrelevant [15] and discard perceptually irrelevant shapes.

## 3 Results

We present some results of our methods. Fig 5 shows that higher-level structure (a ring shape) is extracted using blurring and categorisation. In fig 6, we show the result of blurring and categorising a textured and noisy image to demonstrate that the texture is clustered and the higher-level structure of the image is revealed. Finally, in fig 7, we show that perceptual shapes are found using our methods. We thus prove that by using our processing pathway to blur, categorise, edge segment and identify the shapes, perceptually relevant shapes may be extracted.

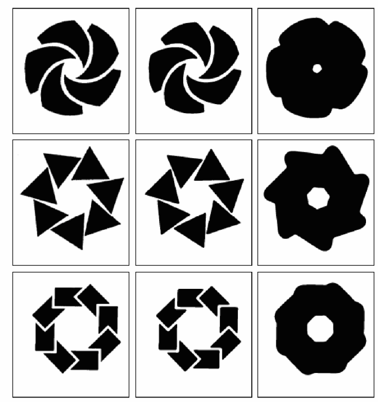



**Fig. 5. Three images (a, b and c) and their respective outputs. All images were classified as line/region by the energy-based classifier.**

In fig 5, the views produced from each image are similar when compared visually by a human observer on a column basis. The ring–structure has been found. If the three images in fig. 5, column 3 were matched the ring structures would be similar. If the three original images in column 1 were matched they would not be similar.

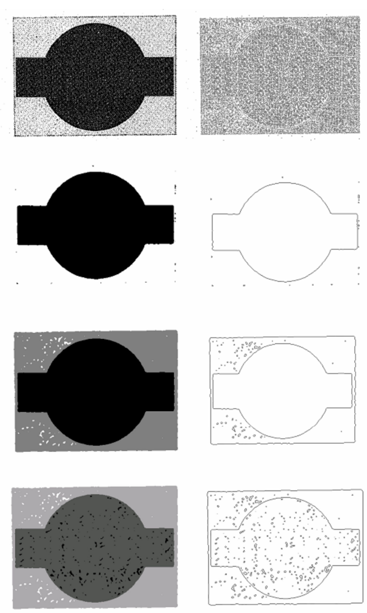



**Fig. 6. The original image (a) is processed to produce a series of image views (b, c and d). The edges found are shown in e, f, g, and h**

Our results are not perfect. For example, in fig 6, results b and c are good. View (d) is probably superfluous here but the energy level and pixel intensity minimum have to be set globally so this may result in an occasional superfluous output for some images. The edges shown in f, g and h demonstrate that we have found the image structures to allow image matching. Although there is a tiny amount of noise remaining, it comprises very small blobs which could easily be removed using a suitable image processing technique. In contrast, image (e) shows the (1000+) edges detected in the original image and no discernible structures.

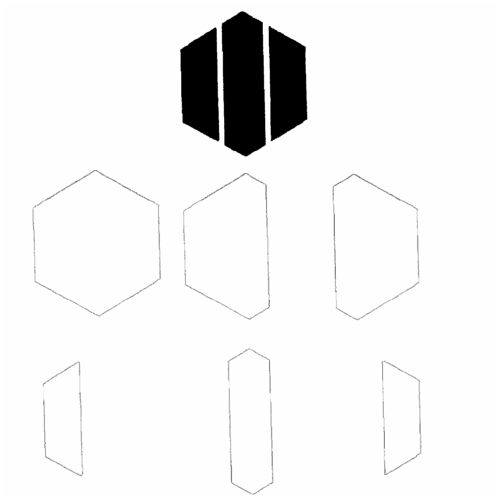


**Fig. 7. The six perceptual shapes found by the shape identifier from the trademark image view in the top row.**

In fig 7, the shape identifier has found the set of perceptual shapes we may expect a human to identify [11] in the trademark image view. This set of shapes may be used for perceptual image matching and retrieval.

## 4 Conclusion

We have developed and demonstrated a figurative image processing pathway comprising a suite of methods to find perceptual shapes (structures) within images. Each image will produce a number of views and each view will produce a number of perceptual shapes. The set of shapes found for each view may be matched and thus used for image matching and retrieval.

No single shape finding method works for all images so, by systematically combining different methods and using image information to guide the processing we have identified perceptual structures. The method follows the Gestalt principles (such as proximity, continuity and similarity) and has been designed using results from human image analysis experiments.

The method has been developed within the EU PROFI project to extract the perceptual structures from trademark images to be stored in a trademark database for trademark image retrieval.

## Acknowledgments

This work was supported by E.U. FP6 IST **Project Reference:** 511572 - **PROFI**.

## References

[1] J.P. Eakins, Trademark image retrieval - a survey, *Multimedia Storage and Retrieval Techniques - State of the Art* (Berlin: Springer-Verlag, 2000).
[2] E. Goldmeier. Similarity in Visually Perceived Forms, *Psychological Issues, 8*(1), 1972.
[3] J. Ashley, et al, Automatic and Semiautomatic Methods for Image Annotation and Retrieval in QBIC, *Proc Storage and Retrieval for Image and Video Databases Conf*, 1995.
[4] J. French, et al, An Exogenous Approach for Adding Multiple Image Representations to Content-Based Image Retrieval Systems, *Proc 7th Int'l Symposium on Signal Processing and its Applications*, Paris, 2003.
[5] T. Lindeberg, Feature Detection with Automatic Scale Selection, *Int'l Journal of Computer Vision*, *30*(2), 1998.
[6] P.J. Burt & E.H. Adelson. The Laplacian Pyramid as a compact image code, *IEEE Trans on Communications, 31*(4), 1983, 532-540.
[7] M. Unser, A. Aldroubi & M. Eden, B-Spline Signal Processing, *IEEE Trans on Signal Processing, 41*(2) 1993, 821-833 (part I) & 834-848 (part II).
[8] C-S. Lu & P-C. Chung, Wold Features for Unsupervised Texture Segmentation, *Proc 14th Int'l Conf on Pattern Recognition (ICPR'98),* 1998.
[9] J. Mao & A.K. Jain, Texture classification and segmentation using multiresolution simultaneous autoregressive models, *Pattern Recognition, 25*, 1992, 173-188.
[10] R. van Leuken, F. Demirci, V.J. Hodge et al, Layout Indexing of Trademark Images, *Proc ACM Int'l Conf. on Image and Video Retrieval (CIVR07)*, Amsterdam, 2007.
[11] V.J. Hodge, et al, Eliciting Perceptual Ground Truth for Image Segmentation, *Proc Int'l Conf on Image and Video Retrieval (CIVR06),* Tempe, AZ, 2006.
[12] M. Ren, J.P. Eakins & P. Briggs, Human perception of trademark images: implications for retrieval system design, *Journal of Electronic Imaging*, *9*(4), 2000, 564-575.
[13] E. Saund, Finding Perceptually Closed Paths in Sketches and Drawings, *IEEE Trans. Pattern Analysis and Machine Intelligence, 25*(4), 2003, 475-491.
[14] D.M. Wuescher & K.L. Boyer, Robust contour decomposition using a constant curvature criterion, *IEEE Trans. Pattern Analysis and Machine Intelligence, 13*(1), 1991, 41-51.
[15] V.J. Hodge, J. Eakins & J. Austin, Inducing a Perceptual Relevance Shape Classifier, *Proc ACM Int'l Conf. on Image and Video Retrieval, (CIVR07)*, Amsterdam, 2007.