PROFI

Project number:	FP6-511572
Project acronym:	PROFI
Title:	Perceptually-Relevant Retrieval of Figurative Images

Deliverable No: D11.3: Workshop proceedings

Short description:

This deliverable contains the papers of two special sessions we organized on logo image retrieval:

- Special session "Trademark Image Retrieval" at the ACM International Conference on Image and Video Retrieval, CIVR'07, 9-11 July 2007, Amsterdam.
- Special session "Perceptually-relevant Retrieval of Figurative Images" at the 5th conference on Signal Processing, Pattern Recognition, and Applications, SPPRA'08, 13-15 February 2008, Innsbruck.

Due month:	M30
Delivery month:	M30, updated M38
Partners owning:	all
Partners contributed:	all
Classification:	PU

Information Society Technologies

Project funded by the European Community under the "Information Society Technologies" Programme

1. Special sessions description

Despite over a decade of research into content-based image retrieval (CBIR), the task of finding a desired image in a large collection remains problematic. Even in application areas where there is a clear need for effective image retrieval, such as medical diagnosis and trademark registration, current technology fails to meet user needs. Much existing research has concentrated on retrieval techniques for natural images (typically photographs of natural scenes or objects), using various combinations of extracted colour, texture and layout feature. Techniques for the retrieval of trademark images, and other artificially-produced images such as icons, logos, coats of arms, and clip-art images, have received less attention, even though there is evidence, that these images require different techniques for effective retrieval.

All these artificially-produced images are designed to have visual impact, and consisting of multiple homogeneous elements, which may be closed regions, lines, or areas of texture. They may represent a given type of object (such as a dog or car) in stylised form, or consist purely of abstract patterns. They may be coloured or monochrome. A comprehensive investigation of retrieval techniques for such images is in our view long overdue, for the following reasons:

- Current techniques for the retrieval of such images are demonstrably inadequate.
- Figurative images such as trademark images, logos, clip art, coats of arms, and icons do not readily lend themselves to retrieval on the basis of name.
- Accurate retrieval and management of such images is of major economic importance.
- Figurative images provide an ideal vehicle for the development of improved shape retrieval techniques, which could be applicable to a much wider domain of images.

Shape is probably the single most important feature used by human observers to characterize an image - psychological studies show that a whole range of familiar objects can be recognized as readily from stylised line drawings as from full-colour natural images. However, the process of automatically extracting image features that characterize these elements has proved remarkably difficult, as illustrated in Fig 1. Professional trademark examiners judge all of the following four images to be similar, because all can be perceived as a triangle enclosing a circle - even though they differ in such basic physical characteristics such as the number of components they contain, and not all of them explicitly contain a triangle and a circle.

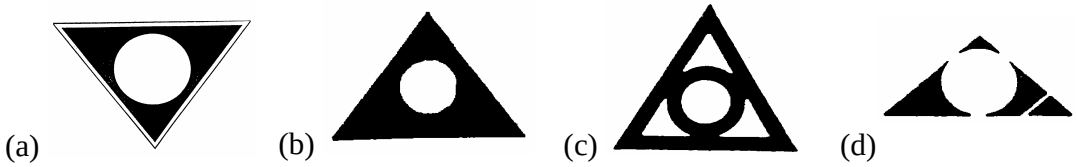


Fig. 1. Example of four figurative images judged by professional trademark examiners to be perceptually similar.

Other aspects can also be important when judging similarity, including image **structure**, the layout of individual image elements (Fig 2). Here, the triangular layout of image (b) makes it appear more similar to query image (a) than does (c), despite the similarity in the shape of individual components. For images that can be interpreted as natural or man-made objects, such as trees or ships (in contrast to abstract shapes illustrated here), there is a further complication: their **semantic** interpretation needs to be considered as well. As discussed below, this is a particularly intractable problem, with no easy solution in sight.

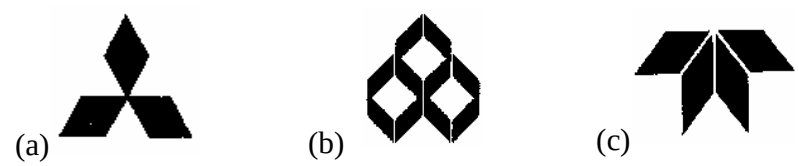


Fig. 2. A typical trademark image (a), together with an image judged to have perceptually similar aspects (b), and one judged to have little perceptual similarity (c).

The decision on what constitutes an image element can often be quite subjective (see Fig 1(d)), and is frequently subject to significant individual variation. The task of devising techniques that can accurately retrieve such images from a database of hundreds of thousands of images is extremely challenging. This is particularly true of trademark image retrieval, where the nature of the application demands virtually 100% recall.

Several further problems are holding back the development of successful retrieval techniques in this area. Partial matching of shapes (see Fig 3(a) and (b) below) is problematic because commonly used feature-based approaches, which generate global feature vectors, do not apply. Developing efficient indexing techniques is crucial when databases can contain literally millions of shapes. However, this is difficult because the ordinary 'point access methods' for feature vectors lose efficiency in high-dimensional search space, and there is a need for new techniques for indexing their relative spatial layout. This is true not only for proprietary databases, but also the collection of trademark images on the web.

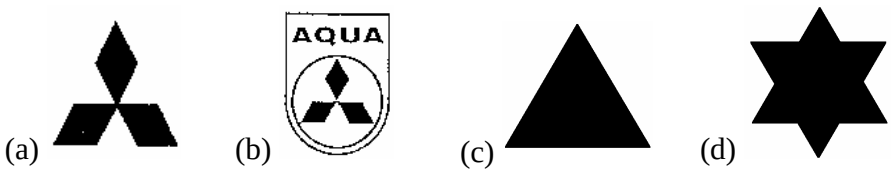


Fig 3. Examples of inadequacy of whole image based measures. Trademark examiners judge that image (a) should retrieve (b), though its global shape is very different. In contrast, (c) should not retrieve (d), even though their edge direction histograms are virtually identical.

The special sessions therefore contains presentations addressing the problems and challenges in trademark image retrieval, the matching of shapes in trademark images, the indexing of spatial layout in trademark images, the trademark image retrieval on the web and the relevance feedback by the user.

The following presentations were given.

Special session “Trademark Image Retrieval” at the ACM International Conference on Image and Video Retrieval, CIVR'07, 9-11 July 2007, Amsterdam:

- John Eakins, Jan Schietse, Remco Veltkamp. Practice and Challenges in Trademark Image Retrieval.
- Victoria J. Hodge, John Eakins, Jim Austin. Inducing a Perceptual Relevance Shape Classifier.
- Helmut Alt, Ludmila Scharf, Sven Scholz. Probabilistic Matching and Resemblance Evaluation of Shapes in Trademark Images.
- Reinier van Leuken, Fatih Demirci, Victoria Hodge, Jim Austin. Lay-out indexing of trademark images.
- Euripides G. M. Petrakis, Epimenides Voutsakis, Evangelos E. Milios. Searching for logo and trademark images on the web.

Special session “Perceptually-relevant Retrieval of Figurative Images” at the 5th conference on Signal Processing, Pattern Recognition, and Applications, SPPRA'08, 13-15 February 2008, Innsbruck:

- John Eakins, Jan Schietse, Remco Veltkamp. Practice and Challenges in Trademark Image Retrieval.
- Shuang Liang and Zhengxing Sun, Active BSVM Learning for Relevance Feedback in Content-Based Sketch Retrieval.
- Victoria J. Hodge, Garry Hollier, Jim Austin, John Eakins. Identifying Perceptual Structures In Trademark Images.
- Sven Scholz, Similarity Evaluation based on Image Primitives.
- Reinier H. van Leuken, Olga Symonova, Remco C. Veltkamp. Topological and directional logo layout indexing using Hermitian spectra.

2. Deviations from plan

In order to reach a larger public than would have been possible whit a dedicated workshop, we have decided to organize special sessions at larger conferences. This was approved by the reviewers.

Appendix

This appendix contains the papers of the two special sessions that have been published in the proceedings.

Practice and Challenges in Trademark Image Retrieval

Jan Schietse
AKTOR Knowledge Technology NV
St-Pietersvliet 7
B-2000 Antwerp, Belgium
+32 3 220 73 67
jan.schietse@thomson.com

John P. Eakins
Department of Computer Science
University of York
Heslington
York YO10 5DD, UK
eakins@cs.york.ac.uk

Remco C. Veltkamp
Department of Computer Science
Utrecht University
3584 CH Utrecht
The Netherlands
Remco.Veltkamp@cs.uu.nl

ABSTRACT
In this paper, we outline some of the main challenges facing trademark searchers today, and discuss the extent to which current automated systems are meeting those challenges.

Categories and Subject Descriptors
H.3.3 Information Search and Retrieval: ***Search process, Selection process;*** H.3.5 Online Information Services: ***Commercial services;*** I.4.0 Image processing and computer vision (general): *image processing software*

General Terms
Design, Economics, Human Factors, Legal Aspects, Management, Performance.

Keywords
Trademark similarity, Content-Based Image Retrieval, Pattern Matching.

1. INTRODUCTION
Despite over a decade of research into content-based image retrieval (CBIR), the task of finding a desired image in a large collection remains problematic. Even in application areas where there is a clear need for effective image retrieval, such as medical diagnosis and trademark registration, current technology fails to meet user needs. Much existing research has concentrated on retrieval techniques for natural images (typically photographs of natural scenes or objects), using various combinations of extracted colour, texture and layout feature. Techniques for the retrieval of trademark images, and other artificially-produced images such as icons, logos, coats of arms, and clip-art images, have received less attention, even though there is evidence, that these images require different techniques for effective retrieval.

All these artificially-produced images are designed to have visual impact, and consisting of multiple homogeneous elements, which may be closed regions, lines, or areas of texture. They may represent a given type of object (such as a dog or car) in stylised form, or consist purely of abstract patterns. They may be coloured or monochrome. A comprehensive investigation of retrieval techniques for such images is in our view long overdue, for the following reasons:

- Current techniques for the retrieval of such images are demonstrably inadequate.
- Figurative images such as trademark images, logos, clip art, coats of arms, and icons do not readily lend themselves to retrieval on the basis of name.
- Accurate retrieval and management of such images is of major economic importance.
- Figurative images provide an ideal vehicle for the development of improved shape retrieval techniques, which could be applicable to a much wider domain of images.

Shape is probably the single most important feature used by human observers to characterize an image - psychological studies show that a whole range of familiar objects can be recognized as readily from stylised line drawings as from full-colour natural images. However, the process of automatically extracting image features that characterize these elements has proved remarkably difficult, as illustrated in Fig 1*. Professional trademark examiners judge all of the following four images to be similar, because all can be perceived as a triangle enclosing a circle - even though they differ in such basic physical characteristics such as the number of components they contain, and not all of them explicitly contain a triangle and a circle.

Other aspects can also be important when judging similarity, including image **structure**, the layout of individual image elements (Fig 2). Here, the triangular layout of image (b) makes it appear more similar to query image (a) than does (c), despite the similarity in the shape of individual components. For images that can be interpreted as natural or man-made objects, such as trees or ships (in contrast to abstract shapes illustrated here), there is a further complication: their **semantic** interpretation needs to be considered as well. As discussed below, this is a particularly intractable problem, with no easy solution in sight.

The decision on what constitutes an image element can often be quite subjective (see Fig 1(d)), and is frequently subject to significant individual variation. The task of devising techniques that can accurately retrieve such images from a database of hundreds of thousands of images is extremely challenging. This is particularly true of trademark image retrieval, where the nature of the application demands virtually 100% recall.

CIVR'07, July 9-11, 2007, Amsterdam, The Netherlands.

* All trademark images illustrated in this article are UK crown copyright

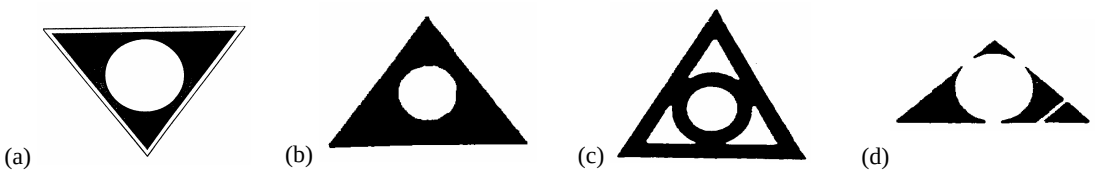


Fig. 1. Example of four figurative images judged by professional trademark examiners to be perceptually similar.

Several further problems are holding back the development of successful retrieval techniques in this area. Partial matching of shapes (see Fig 3(a) and (b) below) is problematic because commonly used feature-based approaches, which generate global feature vectors, do not apply. Developing efficient indexing techniques is crucial when databases can contain literally millions of shapes. However, this is difficult because the ordinary 'point access methods' for feature vectors lose efficiency in high-dimensional search space, and there is a need for new techniques for indexing their relative spatial layout. This is true not only for proprietary databases, but also the collection of trademark images on the web.

2. Trademark Image Retrieval

One of the major issues in the Intellectual Property field is trademark infringement. It is very important for a firm to know if there are other firms which are using "confusingly similar" trademark logos with respect to their newly designed trademark logo since this can lead to a (costly) legal battle; this is a search task. Besides that, firms with "strong" trademarks want to monitor all new registrations of trademarks since they do not want to admit trademarks which are similar to their own trademark on the market; this is a watch task.

Thomson Compu-Mark (TCM) is the market leader of trademark research with offices in Antwerp, Milan, Stockholm, Paris, London, Boston and Tokyo. They are offering both search and watch products for textual and graphic trademarks. We will describe in some more detail the 2 basic trademark research products, i.e. the search and watch product.

Typically a search is performed when one wants to launch a new product or service in the market. A trademark (candidate) is created by a name creation team, but before registration they want to perform a check if there are registered trademarks which are similar to their trademark candidate because the marketing campaign will fail when an existing trademark successfully opposes the registration of the new trademark. To check potential infringement one wants to compare the trademark candidate to all registered trademarks in a set of registers and classes defining both the geographic region and the goods and the services attached to the new product/service.

Trademarks are registered in individual countries (by the Trademark Offices) and by international organisations like OHIM (Office of Harmonization for the Internal Market) for European trademarks, or WIPO (World Intellectual Property Organization) for international trademarks.

In table 1a we list the sizes of the International register (INTE), the Community Trademark register (CTM), the Benelux register (BENE), the French register (FRAN) and the UK register (GBRI). As can be observed, about 30% of all registered trademarks contain next to the textual information graphical elements.

Trademarks are registered for a certain product/service class. This classification (there are 45 different product categories) defines the goods or services you can use your trademark for.

Fig. 2. A typical trademark image (a), together with an image judged to have perceptually similar aspects (b), and one judged to have little perceptual similarity (c).

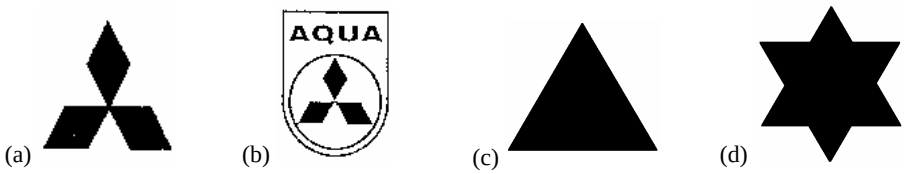


Fig 3. Examples of inadequacy of whole image based measures. Trademark examiners judge that image (a) should retrieve (b), though its global shape is very different. In contrast, (c) should not retrieve (d), even though their edge direction histograms are virtually identical

Table 1a. Trademark Database sizes

Trademark register	Number of trademarks	Nr. of trademarks with logo
INTE	667659	209700
CTM	447421	151398
BENE	440481	140686
FRAN	961355	298665
GBRI	816807	223125

Examples are:

- Class 32: Beers; mineral and aerated waters and other non-alcoholic drinks; fruit drinks and fruit juices; syrups and other preparations for making beverages.
- Class 13: Firearms; ammunition and projectiles; explosives; fireworks.
- Class 42: Scientific and technological services and research and design relating thereto; industrial analysis and research services; design and development of computer hardware and software; legal services.

The result of a search done in a set of registers and classes is a report containing a list of registered trademarks similar to the "candidate trademark". Legal experts will use this report to form an opinion about whether it is safe to register (to avoid claims).

The situation is different when you own a registered trademark. To protect your trademark from infringement it is useful to perform a watch, because if you use as trademark logo a triangle with a hand inside to sell hand cream you will want to oppose another producer which registers a logo also containing triangles with a hand for facial cream since an average consumer might be confused about this.

Table 1b. Number of new Trms in 2006

Trademark Register	# new trademarks in 2006
INTE	44.727
CTM	66.653
BENE	27.327
FRAN	69.706
GBRI	32.383

If a watch is performed, every day one compares the watched trademark with all new trademarks published that day. Similar trademarks are reported on a daily basis to the watch client and again legal experts will evaluate these possible infringements and decide if it is appropriate to start a legal action. This is called opposition.

In table 1b we list the number of new trademarks in the year 2006 in the same registers as in table 1a. As you can see for each register you have several hundreds of new trademarks per day.

3. EXISTING TRADEMARK SEARCH TECHNOLOGY

Until now, the principal means of organizing service- and trademark image collections for retrieval has been to use manually assigned classification codes to reflect image content. The most widely used system is the Vienna classification developed by the World Intellectual Property Organization. In principle, this solves the problem of retrieving all images similar to a given logo by ensuring that similar images will receive identical classification codes.

Table 2. Extract from Vienna Codes

3.5.1	Rabbits, hares
3.5.3	Squirrels
3.5.5	Beavers, marmots, badgers, martens, mink
3.5.7	Rats, mice, moles
3.5.9	Hedgehogs, porcupines
3.5.11	Pangolins, anteaters
3.5.15	Kangaroos, koalas
…	**…**
26.3	**TRIANGLES, LINES FORMING AN ANGLE**
26.3.1	One triangle
26.3.2	Two triangles, one inside the other
26.3.3	More than two triangles, inside one another
26.3.4	Several triangles, juxtaposed, joined or intersecting
26.3.10	Triangles containing one or more circles, ellipses or polygons (except 26.3.11)
26.3.11	Triangles containing one or more quadrilaterals
26.3.12	Triangles containing one or more other geometrical figures
26.3.23	Lines or bands forming an angle

An extract of the codes can be found in Table2.

Practically, it goes as follows. Every new registered/published trademark logo will be analysed and will be attributed one or more Vienna codes. These codes will be added as indexes in the database. When a logo search has to be carried out, one determines which Vienna codes could be attached to the order (i.e. query) trademark logo. These codes will then be queried and the human expert will be presented a list of trademark logos which will have to be verified visually one by one, and the human expert has to decide whether or not it will be put in the search report as being similar, or at least relevant for the client.

The watch is organized in a similar way. One compares for all device (i.e. drawing) watch orders the attached Vienna codes to the codes of the newly registered and when there is a match, the resulting query logo is compared with the newly registered logo. It is again a human expert who does the final evaluation.

However, this approach suffers from two major drawbacks, both inherent in any retrieval system based on manual classification

codes. Manual classification of images is time-consuming and potentially error-prone, and classification codes are not always helpful for retrieval, particularly for abstract images. Similarity judgments may be based on a number of criteria, including overall shape, the shapes of image components, the spatial configuration of components and the presence of particular types of object (image semantics). No current classification scheme can reflect the full range of such criteria.

As a result, both in search and watch, the human expert is confronted with large sets of logos to inspect. The ranking of the query result is also quasi random. Only by using some combination matching one can influence the ranking of the retrieved trademarks.

For example, for a search order containing a hand and a triangle, one typically queries first the logos containing a hand AND a triangle, and next one would query all logos with a hand and all logos with a triangle. The first category "should" contain the most similar trademarks. The other queries can also contain similar images because the trademark knowledge logic can lead to the conclusion that trademarks with a very dominant hand in their logo are confusingly similar.

Since TCM is confronted with faster and faster delivery times and higher quality constraints together with the fact that the number of trademark logos grows rapidly, it becomes necessary to investigate the possibilities of a system or decision support tool based on content based image retrieval (CBIR) techniques to streamline their device searching ensuring consistency and an acceptable degree of precision and recall. It is especially crucial that no confusingly similar trademarks are missed while doing a trademark search.

4. POSSIBLE FUTURE TECHNOLOGY

By investigating the field for device mark comparison in detail, it is clear that a high level of sophistication is needed to provide a refined similarity comparison and ranking system. In contrast to spotting identical or near-identical images, the challenge for providing refined similarity measurement and judgement comparable is much bigger. The human decision taking in the current device watch and search product lines of TCM is based on *refined image understanding* (decomposing images, recognizing explicit but also more implicit image components and configurations), *refined comparison* (invariance for rotation, scaling, transformation, occlusion and noise), and last but not least on (*trade)mark knowledge* (judging the strength or weakness of used image elements, judging the relative importance of elements, etc). In these judgements, human experts perform an image interpretation based on recognition of shapes, regions, texture, text and spatial configuration.

On a high level an image retrieval system suited for comparing trademark images should fulfil the following constraints:

- One should take into account every possible interpretation of a trademark image.
- It should be possible to search in big sets of images with an acceptable speed (relatively short delivery times).
- Very similar (to the query image) images in the database can not be missed (zero tolerance).
- Trademark images should be compared in great detail (such as shape, contour, and structure) taking into account all sorts of transformations (such as rotation, scaling, inversion, and blurring).

4.1 Scope

Before going into detailed characteristics of a trademark image retrieval systems it is important to elaborate on the scope of an "ideal" trademark image system.

First of all we have to take care of the semantic gap problem. For a search of a logo with a lion, the client will want to receive in his search result all logos containing a lion, even if the "image characteristics" of both lions are completely different. In that case we are dealing with a semantic search and as a result this kind of orders will not be solved by a "traditional" content based image retrieval system which compares contours, shapes, lines or structures. Fortunately, it is easier to perform a search with natural objects than for more geometric order queries (the number of logos to inspect are smaller and the decision is easier).

The added value for abstract orders containing mainly geometric shapes will be higher, since currently with the text retrieval system, for this kind of orders the human expert is confronted with very large collections of logos which have to be inspected. This is simply due to the fact that a very big part of all registered trademark logos are abstract and the fact that in order to retrieve all potential similar trademarks using the Vienna codes, one should enter general/broad categories.

4.2 System Features

The main characteristics of a possible solution for a trademark logo retrieval system are the following:

Order Query Specification

In a 'Query by example' environment the order image is taken as a starting point, image understanding is performed by the system, and the analyst is able to provide additional information. Codification is no longer needed, except for image components with clear semantic meaning (natural objects like pelicans and known artefacts like the statue of liberty). It is clear that segmentation should be an important module in the image analysis component.

Since the human expert can indicate the relative importance of certain elements/shapes, add tags to natural objects, and correct the segmentation results, we will start from an analyzed and enriched order image.

Analysis of target images

The system should provide a (semi-)automatic analysis of new target images. This ensures the incorporation of new (trade)marks for device watching, and also incorporation of existing trademark databases for device searching. Coding should no longer be needed, except for image components with clear semantic meaning (natural objects and known artefacts). As in the case of the order query, it seems likely that the images are segmented.

Robust to noise

The system should be robust to noise in both order image and trademark logo images. Trademark logos with a noise level too

high to do an automatic segmentation should be (semi) automatically cleaned.

Advanced image interpretation

The system should provide image interpretation, and should be able to detect image components like shapes, regions, texture, colour and text components. Also the spatial configuration of all components should be detected. The system should use human-perception-based segmentation to also identify more implicit shapes in the image or partially occluded shapes.

Advanced image comparison

The system should be able to compare device mark images by comparing all elements resulting from the image understanding step, and taking into account their spatial configuration. A query image with a circle in a triangle is more similar to a logo with both shapes present in a deformed way but in the same configuration than to a logo with a circle and triangle in a completely different configuration.

This comparison of the shapes should remain effective under variations like rotation, scaling, transformation, partial occlusion and noise. It is essential for this application that partial matching is supported and on top of that the matching algorithm should reflect in detail how good the partial matching is and which parts of the images are matching.

The fact that text is present on a trademark logo is important. It is not needed to take into account the individual letters of the text, but the fact that a text field is present in both query and target logo in a comparable spatial configuration contributes to the similarity measure.

Colour is also recorded as image element attributes, and can be used as a feature in the comparison.

(Trade)Mark Knowledge Layer

The image comparison results are combined in (trade)mark knowledge rules to provide the similarity judgement and ranking. This provides the flexibility to tune/refine the system based on human expertise. One of the most important trademark features is dominance. The concept of dominance is influenced by the size or other characteristics of a shape or object but also the frequency of occurrence of a certain object can influence the fact that it is dominant. For example: if there are only a very limited number of trademarks which use a certain shape in their logo, then this shape is very distinctive and therefore dominant. Even if there are stars, triangles or circles added to the distinctive shape (for example a swoosh) it will be important to retrieve all trademarks containing this distinctive shape and to rank them high. Trademarks containing stars, triangles or circles can lead also to a similar trademark but the probability is much lower.

It should be possible to tune the system in order to solve quality issues from clients or internal quality checks. Therefore the system should represent all information in an image content graph. Based on comparison results from both graphs, a *tuneable (trade)mark logic layer* decides on the overall similarity between order and trademark logo. This knowledge representation approach will enable quality updates and complaint solving.

Acceptance

The similar device marks are presented in an acceptance environment, that provides ranking, and logical groupings. The analyst acceptance is used to refine the proposal even more by using relevance-feedback. The human expert should also be provided with tools supporting consistent selection.

Indexing

The fast delivery times are implying that a trademark image retrieval solution also includes advanced indexing schemes that provide a fast response despite the complexity of the underlying computations.

4.3 Benefits

The benefits of a system such as the one described above would be quite considerable. Such a system should make it possible to deliver in a consistent way high volume logo searches and watches with quality assurance and controllable cost.

5. TECHNOLOGICAL CHALLENGES

While trademark image retrieval has been the subject of considerable research over the last fifteen years [1], no system described in the literature is yet capable of meeting all the criteria set out above. A brief outline of previous research in the field is given below.

5.1 Previous research

One strand of research has concentrated on extracting and comparing features from trademark images *taken as a whole*; The earliest example of the first approach was Kato's TRADEMARK system [2]. It maps normalized trademark images to an 8×8 pixel grid, and calculated a *GF-vector* for each image from frequency distributions of black and edge pixels appearing in each cell of the grid. Query and stored images could then be matched by comparing GF-vectors. Other work following this approach include the following.

- Jain & Vailaya [3] use a two-stage process comprising rapid screening using edge direction histograms and moment invariants followed by template matching;
- Kim & Kim [4] calculate all Zernike moments up to order 17 for each stored and query image, and then select and use the moment with greatest discriminating power for matching;
- Ravela & Manmatha [5] use multi-resolution matching based on histograms of local curvature ratios and gradient orientations computed from Gaussian derivatives.

The second approach regards trademark images as a set of discrete components which are best matched on an individual basis. Overall image similarity can then be computed in a variety of ways from component similarities. The earliest example of this method was the STAR system developed by Wu et al. [6]. This system is based on the principles that perceived trademark similarity is a function of shape, structural and semantic similarity, and that human intervention is essential to achieve acceptable results. The first stage of processing thus involves human indexers, who segment trademark images into perceptually meaningful components. A mixture of human and automated labelling can then be performed, assigning shape features such as

Fourier descriptors and moment invariants, structural features such as the presence of regular patterns of shapes and semantic features such as the presence of particular types of object. The overall similarity between trademarks can then be computed from component feature similarities.

The ARTISAN[1] system developed by Eakins et al. [7] is based on similar principles, though with the important difference that all segmentation and feature extraction is performed automatically. Gestalt principles are used to derive rules allowing individual image components to be grouped into perceptually significant families. Similarity matching can be performed at three levels: whole images, component families or individual image components. More recent versions of the system [8] have incorporated multiresolution analysis to remove texture and group low-level components into higher-level regions, as well as a wider range of shape and structural features.

ARTISAN's use of Gestalt principles has been taken one step further by Alwis and Austin [9], who aim to identify all significant line segments in an image and then cluster these into perceptually significant units according to Gestalt rules. Rather than using conventional similarity matching, their system uses an evidence counting method based on feature values extracted from closed contours in both raw images and "Gestalt" images.

Another technique based on differing views of an image is that of Leung and Chen [10]. They characterize regions as either solid or line-like, extracting boundary contours for the former and skeletons for the latter. After extracting features from line segments derived from both types of representation, overall image similarity is computed by performing a best match between line segments in query and stored images.

5.2 Limitations of current systems

Despite considerable ingenuity by researchers into trademark matching, it is clear that a significant gap remains between the needs of users and the capabilities of current technology. Indeed, it is not immediately apparent that researchers have always been aware of user needs, suggesting that much research may not even have tried to tackle the most pressing problems. Taking the criteria from Section 3 in turn:

1. ***One should take into account every possible interpretation of a trademark image.*** Studies of the ways by which humans perceive and interpret images confirm that it is a complex process [11], and one that is not straightforward to model in software [12]. However, most research to date has concentrated on matching trademarks purely on the basis of shape or other low-level features. Some researchers have looked at shape features derived from the image as a whole, while others have compared shape features derived from components of segmented images. Some have used regions as the basis for shape comparison, others have used line segments. But with the exception of Alwis & Austin [9] and to a lesser extent Eakins et al. [7], there have so far been few attempts to base image matching on *multiple* views of an image.

2. ***It should be possible to search in big sets of images with an acceptable speed (relatively short delivery times).*** Most research to date has been conducted on relatively small sets of trademark images, many researchers using collections of little more than a thousand images. Search efficiency has therefore not been a high priority, though the two-stage approach of Jain & Vailaya [3] demonstrates the feasibility of one approach to the problem. In the future, significant improvements in search efficiency will still be needed before any system becomes usable in a commercial environment.

3. ***Very similar (to the query image) images in the database can not be missed (zero tolerance).*** This is a fundamental requirement of trademark searching, though not necessarily true of image searching in general. For many applications such a journalism and fashion, it does not matter if some relevant images are missed as long as the ones retrieved are acceptable to users. In this context it is important that the retrieval effectiveness of prototype systems should be exhaustively investigated.

4. ***Trademark images should be compared in great detail (such as shape, contour, and structure) taking into account all sorts of transformations (such as rotation, scaling, inversion, and blurring).*** This requirement is in fact relatively easy for current image matching technology to fulfil. Most, if not all, current feature matching and shape comparison techniques are either inherently invariant to transformations, or can be made so. Multi-resolution matching can handle images at varying levels of detail and blurring. However, this kind of processing is extremely computationally expensive. Hence the more exhaustively query and stored images have to be analysed, the slower the system. Even with the most powerful modern computers, there still needs to be a tradeoff between speed and effectiveness.

5.3 Challenges and prospects for future progress

Perhaps the most serious limitation of current automated systems lies in the area of initial image analysis. Unless all crucial features of target images have been effectively computed and stored, subsequent matching is unlikely to identify all relevant similarities. As indicated above, an ideal system should be able to recognize similarities of shape, structure, and semantics, and to be able to handle (possibly stylised) text – a challenge well beyond the capability of current technology. Even at the level of retrieval by shape or structure, considerable advances will need to be made in modelling human image perception.

The importance of providing alternative representations based on different views of an image has already been mentioned. One possible way to achieve this is follows:

- ***Line-based views*** of an image can be generated by taking the output from a suitable edge detector and aggregating it into perceptually significant groupings according to Gestalt principles, following the approach pioneered by Alwis and Austin [9].
- ***Region-based views*** can be generated by multiresolution analysis using techniques derived from those already developed by Eakins et al [8], augmented by texture

[1] *Automatic Retrieval of Trademark Images by Shape ANalysis*

classification and possibly by splitting and merging regions (following rules similar to those proposed by Hoffmann and Richards [13]) to form more perceptually significant groupings.

- ***Concept-based views*** can be generated by identifying and characterizing familiar (i.e. named) visual concepts within an image. These could include shapes such as circles, triangles, squares, and hexagons, as well as more abstract concepts such as crossover, linear repetition and symmetry, which appear from previous studies [8] to play a crucial role in similarity determination in some contexts.

Developing a whole series of views in this way runs the risk that many of them will represent a nonsense interpretation of the image. This can be avoided by using AI techniques to select those views of a given image most likely to make perceptual sense, or identify the most effective combination of processing methods. However, it may not be possible to train up a machine to perform this task to the exacting standards required by trademark searchers. Hence a hybrid system may be necessary in which human indexers review and if necessary correct machine interpretations of images added to a trademark database. Such indexers could also assign semantic terms to the images (a task which even the best machine learning systems are still incapable of performing reliably), thus bringing such a system one step closer to commercial acceptability.

Conventional image matching techniques, based on 1-to-1 comparison of pairs of image feature vectors, are for the most part far too slow to be acceptable with databases containing up to a million images. Methods based on searching and matching ***groups*** of lines, regions or feature vectors may be needed before acceptable performance is achieved. Further improvements in search efficiency may be gained by using multidimensional indexes such as the X-tree [14] to organize feature vectors, and Vantage Objects [15] for indexing object space.

Interfacing, both at the query specification and results display stage, is another area that has been relatively neglected by researchers to date. Better methods of search formulation are needed, allowing users to specify:

- whether the search should be based on a complete image, specified parts of an image or a sketch, and
- the most appropriate search parameters for a given image – for example, giving shape and structural features different weights.

Potentially useful improvements at the display stage include:

- two- or even three-dimensional display of retrieved images, allowing searchers to view similarities between them, and
- relevance feedback [16], allowing users to improve system effectiveness by indicating which retrieved images are genuinely relevant to the query.

Many further approaches remain to be explored, and prospects for long-term progress remain good. But the difficulty of finding solutions to the trademark matching problem which are sufficiently robust for commercial use should not be underestimated.

6. ACKNOWLEDGMENTS

Part of this work has been supported by the European Commission under the Sixth Framework Programme project 511572-2 PROFI (Perceptually-relevant Retrieval Of Figurative Images).

7. REFERENCES

[1] Eakins, J. P. Trademark image retrieval. Ch 13 in *Principles of Visual Information Retrieval* (Lew, M, ed). Springer-Verlag, Berlin, 2001

[2] Kato, T. Database architecture for content-based image retrieval. *Image Storage and Retrieval Systems,* Proc SPIE 2185, 112-123, 1992

[3] Jain, A.K., & Vailaya, A. Shape-based retrieval: a case study with trademark image databases. *Pattern Recognition,* **31**(9) 1369-1390, 1998.

[4] Kim, Y.S., & Kim, W.Y. Content-based trademark retrieval system using a visually salient feature. *Image and Vision Computing*, **16**, 931-939, 1998.

[5] Ravela, S., & Manmatha, R. Multi-modal retrieval of trademark images using global similarity. Internal Report, University of Massachusetts at Amherst, 1999.

[6] Wu, J.K. et al. Content-based retrieval for trademark registration. *Multimedia Tools and Appl.* **3**, 245-267, 1996.

[7] Eakins, J.P., Boardman, J.M., & Graham, M.E. Similarity Retrieval of Trademark Images. *IEEE Multimedia*, **5**(2), 53-63, 1998.

[8] Eakins, J.P., Edwards, J.D., Riley, J., & Rosin, P.L. A comparison of the effectiveness of alternative feature sets in shape retrieval of multi-component images. *Storage and Retrieval for Media Databases*, Proc SPIE 4315, 196-207, 2001.

[9] Alwis, S. and Austin, J. Trademark image retrieval using multiple features. Presented at *CIR-99: The Challenge of Image Retrieval*, Newcastle-upon-Tyne, U.K., Feb. 1999.

[10] Leung, W.H. & Chen, T. Trademark retrieval using contour-skeleton classification. *Proc. IEEE Intl. Conf. on Multimed. and Expo (ICME 2002)*, Lausanne, Switzerland, Aug. 2002.

[11] Goldmeier, E. "Similarity in visually perceived forms" *Psychological Issues* **8**(1), 1-135, 1972.

[12] Ren, M., Eakins, J. P. &d Briggs, P. Human perception of trademark images: implications for retrieval system design. *Journal of Electronic Imaging*, **9**(4) 564-575, 2000.

[13] Hoffmann, D. D. & Richards, W. A. Parts of recognition. *Cognition* **18,** 65-96, 1985.

[14] Berchtold, S., Keim, D.A. & Kriegel, H. P. The X-tree: an index structure for high-dimensional data. *Proceedings of the 22nd Conference on Very Large Databases*, 1996

[15] Vleugels, J. & Veltkamp, R. C. Efficient Image Retrieval through Vantage Objects, *Pattern Recognition* **35**(1), 69-80, 2002.

[16] Zhou, X. S. & Huang, T. S. Relevance feedback in image retrieval: a comprehensive review. *Multimedia Systems* **8**, 536-544, 2003.

Inducing a Perceptual Relevance Shape Classifier

Victoria J. Hodge
Dept of Computer Science
University of York
York, UK
+44 1904 433067
vicky@cs.york.ac.uk

John Eakins
Dept of Computer Science
University of York
York, UK
eakins@cs.york.ac.uk

James Austin
Dept of Computer Science
University of York
York, UK
+44 1904 432734
austin@cs.york.ac.uk

ABSTRACT
In this paper, we develop a system to classify the outputs of image segmentation algorithms as perceptually relevant or perceptually irrelevant with respect to human perception. The work is aimed at figurative images. We previously investigated human visual perception of trademark images and established a body of ground truth data in the form of trademark images and their respective human segmentations. The work indicated that there is a core set of segmentations for each image that people perceive. Here we use this core set of segmentations to train a classifier to classify closed shapes output from an image segmentation algorithm so that the method returns the image segments that match those produced by people. We demonstrate that a perceptual relevance classifier is attainable and identify a good methodology to achieve this. The paper compares MLP, SVM, Bayes and regression classifiers for classifying shapes. MLPs perform best with an overall accuracy of 96.4%.

Categories and Subject Descriptors
I.5.4 [**Pattern Recognition**] Applications - *Computer vision* I.5.1 [**Pattern Recognition**] Models - *Neural nets, Statistical* I.2.10 [**Artificial Intelligence**] Vision and Scene Understanding - *Perceptual reasoning, Representations, data structures, and transforms, Shape* I.4.6 [**Image Processing And Computer Vision**] Segmentation - *Edge and feature detection* I.4.7 [**Image Processing And Computer Vision**] Feature Measurement -- *Feature representation, Invariants, Moments, Size and shape.*

General Terms
Performance, Experimentation, Human Factors, Verification.

Keywords
Perceptual relevance, classification, image segmentation, perceptual classifier, human image segmentation.

1. INTRODUCTION
There has recently been tremendous growth in the storage of digital imagery producing a need for accurate and fast indexing and retrieval systems. Examples of applications include archiving images or photographs, medical image analysis and trademark retrieval. In Content-based Image Retrieval (CBIR) the aim is to retrieve images form an image database that are similar to a query image. This process may be performed by matching the whole image as a single entity or matching components within each image [8]. In this work, we focus on component-based similarity matching of trademark images.

Our work forms part of the PROFI (Perceptually-Relevant Retrieval of Figurative Images) project [See section 6]. In PROFI, we aim to develop new techniques for the retrieval of figurative images (i.e. abstract trademarks and logos) from large databases and, in particular, aim to reproduce the matches that people find by manual methods on this task. The techniques are based on the extraction of perceptually relevant shape features and the matching of these features in the target image against features in the stored images. The first stage of this procedure is to identify the components present within an image. As our aim is to return the images from the automated system that people would say were similar, we believe that this segmentation process should reflect human perception and segmentation. The principal difficulty for image segmentation algorithms in the context of our work is the selection of parts that accurately reflect the image's appearance to a human observer.

To obtain a base line for the human performance on the task, we have previously conducted a set of experiments investigating human segmentation of trademark images [10,11]. The experimental results detailed in the two papers and outlined in section 2 concur with previous investigations such as [17] in that human image segmentation appears to follow a set of perceptual principles analogous to the Gestalt laws [15,25]. The experiments and analyses show that these Gestalt laws interact and possibly conflict as noted by [6]. The experiments also indicate that there are a core set of segmentations for each image perceived by two or more people along with a set of segmentations seen only by individuals. This core set of segmentations forms the ground truth for our evaluations into inducing a perceptual relevance classifier. It is vital for any computerised image segmentation algorithm to include a perceptual relevance classifier, effectively a global goodness score. This allows the algorithm to reduce the number of segmentations output and to focus on perceptually relevant shapes whilst, hopefully, discarding irrelevant segmentations.

The first stage is to identify the shapes present in an image. To do this we require a shape identification algorithm. In practice any closed shape identifier could underpin the procedure, such as region growing [28], watershed [2] or closed shape identification [1] provided the result of the algorithm may be represented by a

CIVR'07, July 9-11, 2007, Amsterdam, Netherlands.

list of boundary points to calculate the attributes used here. It is the classification process where we focus our research here, not the underlying shape identification algorithm. We use Saund's closed shape identification algorithm [22] here. It was developed for the sketch retrieval domain to identify shapes within human sketches but is equally applicable to the trademark retrieval domain. It aims to find closed shapes satisfying global criteria and has similarities to our aim of classifying perceptual relevance.

We aim to use our previous empirical evaluations of human perception to induce a classifier that classifies the closed shapes output by a closed shape identification algorithm as perceptually relevant (to keep) or perceptually irrelevant (to discard). This would effectively replace the global goodness measure used in Saund's method. To do this we require a set of attributes to represent each shape as a vector and a classifier to classify the shapes output for relevance.

To decide which attributes to use, this work takes its cue from the attributes elicited by Alwis [1], Chan & King [4], ARTISAN [8] and QBIC [9]. Alwis [1] has produced a trademark retrieval system, with many similarities to the work here, so the features he used are particularly relevant for the work here: circularity, aspect ratio, stuffedness, "right angledness", sharpness, complexity, directness and straightness. Chan & King [4] propose a method for feature weight assignment in a trademark system. They use invariant moments, Euler number, eccentricity, and circularity in their evaluations. ARTISAN [8] is "regarded as one of the most comprehensive trademark retrieval systems in the current literature" [13]. ARTISAN uses component-based matching so the features used by ARTISAN should be relevant: aspect ratio, circularity, convexity, the Fourier descriptors, the shape's area and the three 'natural' shape measures defined by Rosin [18]: rectangularity, triangularity and ellipticity for trademark retrieval. The IBM QBIC [9] system is one of the most ubiquitous image retrieval systems developed and has been used widely so the features used for matching should be relevant to our developmental system. The shape features used in QBIC consist of shape area, circularity, eccentricity and a set of algebraic moment invariants.

In this work, we analyse a series of common classifiers to verify that it is possible to classify perceptual relevance using human classifications and to pinpoint the classifier that achieves the highest recall accuracy while maintaining recall consistency. We assess four supervised learning classifiers: Naïve Bayes [14], Multi-Layer Perceptron (MLP) [19], Support Vector Machine (SVM) [24] and Regression [20]. Naïve Bayes is a simple statistical model linear classifier that often outperforms more sophisticated classifiers [26]. Standard regression is a statistical model linear classifier aimed at classification with numeric attributes such as we use here. Non-linear statistical model classifiers such as the MLP or SVM can model non-linear class boundaries and are usually robust to outliers in the data. These four classifiers together thus provide a broad cross-section of classifier technology.

In the remainder of this paper, we describe: our previous human segmentation experiments, the underlying closed shape identification algorithm that we used for our analyses, the 23 attributes used to represent each closed shape, the data used to perform our classification analyses, the four classifiers we have evaluated for recall accuracy, the methodology we use for our evaluations, the results, analyses and conclusion inferred.

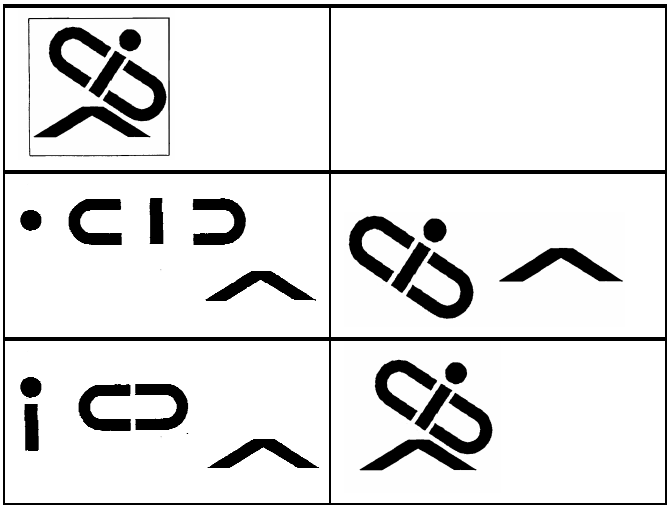
2. HUMAN PERCEPTION ANALYSES

To test the system, it has been necessary to collect ground truth data from human subjects on how individuals segment images – thus asking the question: "what are the human segmentation preferences?" The following explains how we collected this data and summarises the work published in [10,11].

In our human perception experiments, ***53*** subjects each received **32** trademark like images in a booklet. The subjects were requested to draw (using pen or pencil) their perceived segmentations of each image in turn on to the booklet. We collated the segmentations drawn by the subjects and produced a listing of all segmentations for each image in turn. For our work here, we only consider segmentations seen by 2 or more people which represent our core set of segmentations that the trademark system should output to represent each image.

Table 1 shows an example image and the human segmentations perceived for that image. The human subjects perceived four different segmentations – they comprised the following number of components (shapes): 5, 2, 3 and 1 components respectively. We identify these as the perceptually relevant components (shapes) for this image which the closed shape identification algorithm should ideally identify.

Table 1 Table showing an image (top row) and the four segmentations seen by 2 or more people for that image.



3. CLOSED SHAPE IDENTIFICATION.

To identify the closed shapes in the image, we use Saund's method as pointed out above. This method requires an underlying algorithm to identify line segments in an image and the relationships between those line segments. Therefore, we initially find the edges in an image and subdivide these into constant curvature segments using the Sarkar & Boyer [21] edge detection algorithm and the Wuescher & Boyer [26] curve segmentation algorithm. These methods were selected as they had successfully been used in the trademark system developed by Alwis [1]. The Sarkar & Boyer method finds the edge lines in an image and splits

these lines into primitives. Wuescher & Boyer performs some aggregation of these primitives into more perceptually-oriented constant curvature segments and outputs these as a list of constant curvature segments. These segments provide the building blocks for our closed shape identifier. Our aim is to group these constant curvature segments using Gestalt like methods to produce a graph of segment relations which will underpin the Saund closed shape identification algorithm. To produce this graph we use the following methods. Each constant curvature segment 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 use Lowe's method [16] to extract endpoint proximity by comparing two lines with lengths l_1 and l_2 or curves with perimeter lengths l_1 and l_2. In the following, ($l=l_1$ if $l_1<l_2$ else $l=l_2$). The distance between their endpoints is r. The inverse significance of endpoint proximity between them is $\frac{r^2\rho}{l^2}$. The parameter ρ is a unit-less constant and may effectively be ignored (i.e. set to 1). So if: $\frac{r^2}{l^2}<\text{threshold}$ where threshold = 0.01 then the two endpoints are joined. This effectively joins the graph by linking the proximal end-points.

The Saund algorithm overlays this and focuses on managing the search of possible path continuations through the graph particularly where the graph nodes represent junctions (crossroads, t-junctions etc) of lines in the original image. The search is managed through the use of local criteria for prioritising the order in which paths are pursued. Saund has identified criteria (scores) for ranking possible paths through junctions based on observations. Path scores accumulate by multiplying junction preference scores as the path progresses.

The closed path search commences from each end (first and last) of each node (line segment) identified by the underlying Wuescher & Boyer algorithm. 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 paths of candidate shapes. As each leaf node in the tree is expanded, any 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 has been identified and its nodes and pixels are added to the list of candidate paths. All closed paths exceeding a threshold score are thus stored as candidate paths. Saund terminates searching when a closed path score exceeds a pre-specified threshold. Saund accepts a closed path as a candidate if its cumulative junction score exceeds 0.6 or accepts the closed path and terminates search from the particular root node if the score exceeds 0.9. We do not terminate search if the score exceeds this threshold as we feel potential closed paths may be missed due to higher scoring and shorter paths terminating the search prematurely. Hence, all paths that exceed 0.6 are accepted as candidates but search continues.

Saund discards closed paths that are subsumed by other closed paths with higher scores. Hence, each new closed path is compared to all existing stored paths. If the new path is a subset (including the equivalence set) of an existing path but has lower score then the new path is discarded. If the new path has higher score than the existing saved path then the saved path is discarded.

3.1 Determining Good Shapes

In Saund's methodology, all paths accepted as candidates are then assessed for global figure goodness by awarding a score. In Saund's approach the score for each closed path is produced multiplicatively (C*N*E) where C, N and E are:

Compactness (C) - the ratio of [figure area: area of convex hull].

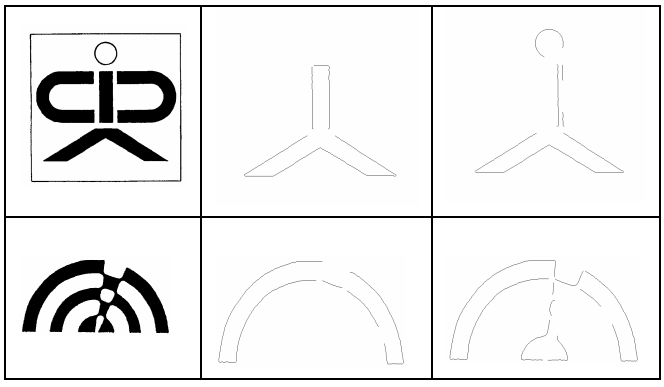
End-point distance (E) - calculated using $1 - d_e/p$ where d_e is the distance between endpoints of the path and p the path length.

Non-end-nearest-approach (N) which penalises paths where an endpoint terminates near the body of the path.

This method does not produce the perceptually relevant closed figures identified by our experiments. This is where our work has changed the method: by adding a perceptual classifier taught using the data from human experiments. Through a brief comparison, we identified that, of the three Saund attributes, only C (which we call areaScore) matches to some extent the human preferences from our experiments.

The output of our implementation of the Saund algorithm is therefore a list of candidate closed shapes found in the image. These closed shapes are the candidate shapes whose cumulative junction score exceeds 0.6. Each candidate shape is classified as relevant or irrelevant using our perceptual classifier and only shapes classified as relevant will be retained for further processing. The candidate closed shapes are represented by a list of x, y coordinates representing each point on the shape's boundary (in order with no gaps). Two example images with one relevant shape and one irrelevant shape identified by our implementation are shown in Table 2. The classifier should classify the relevant shapes as relevant and the irrelevant shapes as irrelevant thus slowing us to discard the irrelevant shapes from any further processing.

Table 2 Table listing 2 images (leftmost column) and two paths identified by the Saund algorithm for each image (one perceptually relevant (middle column) and one perceptually irrelevant (right column)).



3.2 Attributes

As outlined above, we use a classifier to determine which closed figures output from our implementation of the Saund algorithm are perceptually relevant. The classifier has to be trained on the data collected from our ground truth experiments described in Section 2. The selection of the attributes for the classifier is considered as follows.

Each output is a list of boundary points (x, y coordinates) of each closed shape. We produce various attributes from the boundary points thus representing each closed shape as a vector of 24 attributes: 23 attributes calculated from the list of boundary points (x, y coordinates) plus the class (perceptually relevant or irrelevant). Note that the 23 attributes are not independent of each other; many are closely related such as AreaRatio and Roughness; it is the job of the classifier to determine the optimum set of attributes. The following attributes are calculated:

Roughness = Perimeter / Convex Hull Perimeter

AspectRatio = Perimeter / Min. Area Bounding Box Perimeter

Stuffedness = Area / Min. Bounding Box Area

AreaRatio = Area / Convex Hull Area

GapScore = Max. Gap in Perimeter / Perimeter

Circularity = 4π* Area / Perimeter^2

Eccentricity = $\frac{(M_{20}-M_{02})^2+4M_{11}^2}{(M_{20}+M_{02})^2}$ where M is calculated using the centroid thus $M_{pq} = \sum_{i=0}^{N-1} (x_i - C_x)^p \times (y_i - C_y)^q$

Ellipticity = $\begin{cases} 16\pi^2 I_1 & \text{if } I_1 \le \frac{1}{16\pi^2} \\ \frac{1}{16\pi^2 I_1} & \text{otherwise} \end{cases}$ where $I_1 = \frac{\mu_{20}\mu_{02}-\mu_{11}^2}{\mu_{00}^4}$

and $\mu_{pq} = \sum_x \sum_y (x_i - C_x)^p \times (y_i - C_y)^q \times f(x_i, y_i)$

Triangularity = $\begin{cases} 108 I_1 & \text{if } I_1 \le \frac{1}{108} \\ \frac{1}{108 I_1} & \text{otherwise} \end{cases}$

Hu Moments [12] (from boundary points).

Fourier coefficients (from boundary points): The Fourier coefficients are the amplitude of the Fourier expansion of the cumulative angular bend around the shape's boundary points with $0 \le k \le 6$ here.

$$c[k] = \frac{\sqrt{\left(\sum_0^{n-1}\left(\theta_j \cos\left(\frac{-2\pi jk}{n}\right)\right)\right)^2 + \left(\sum_0^{n-1}\left(\theta_j \sin\left(\frac{-2\pi jk}{n}\right)\right)\right)^2}}{n}$$

3.3 Data Preparation

To allow the system to classify the data from these attributes, we first collected a set of data from the ground truth images. All images from the experimental set of 84 images [10,11] which contained texture were discarded as texture confuses shape identifiers and produces very poor segmentation results. The underlying line segmentation algorithm finds a large number of edges in texture data. To do this we discarded all images that produced more than 500 shapes as this is too many to process by hand. This left 48 images.

Shape | *Image 1* *Shape is irrelevant* | *Image 2* *Shape is relevant*

Figure 1. The leftmost shape (two interlocking loops) is relevant for the rightmost image but irrelevant for the middle image.

Our experiments indicated that there are a core set of segmentations for each image perceived by 2 or more people. From the chosen 48 images, we ran the closed shape identifier (described in section 3) and selected (by hand) shapes output by this algorithm that were perceptually relevant (matched shapes within the segmentation drawn by 2 or more human subjects) and shapes that were perceptually irrelevant (very dissimilar from the shapes drawn by the human subjects). We tried to balance the number of relevant with the number of irrelevant examples from each image although this is not always possible.

We note that for our analyses here, it is important to choose relevance/irrelevance carefully. We are representing the global picture; one shape that is relevant for one image may in fact be irrelevant for another similar image containing that shape as shown in figure 2. We took this into account when preparing our training/test sets for the classifiers and only selected shapes that were perceptually irrelevant across the board. The final classifier is a global classifier; it is not trained on a per image basis so we need the global relevance picture which needs careful consideration.

From the 48 images available, we used 29 images to produce an original training set comprising 435 records and 19 images to produce an original test set comprising 306 records giving a total data set size of 741 records. This represents all of the data we had available. We labelled these two data sets: **set1** and **set2** respectively. We then extracted the top half of the training set1 and the top half of the test set2 to produce **set3** comprising 371 records. When splitting the data sets in half, we ensured that all records from a particular image were kept together in one half or the other. **Set4** which contains 370 records is the bottom half of the training set1 and the bottom half of the test set2. **Set5** comprises 365 records and is the top half of the training set1 and the bottom half of the test set2 and finally we merged the bottom half of the training set1 and the top half of the test set2 to produce **set6** with 376 records. This subdivision allows us 6 runs of each classifier with a training set and a test set. In these analyses, standard x-fold cross validation is not feasible as the data set contains images that are variants of other images (altered according to Gestalt principles) so the constituents of the test and training sets must be considered carefully to prevent biasing and instability and we tried to prevent this by splitting the sets carefully. Also, we cannot split the records for each image, for example if image 1 produced 10 relevant and 10 irrelevant shapes then these must all be kept together in one set to prevent biasing

of the data. We are searching for a classifier that generalises well so all equivalent data must be together but image variants may be split in a considered way.

3.4 Classifiers

In the work we assess four classifiers to select the perceptually relevant and irrelevant shapes from the images, these are Naïve Bayes [14], MLP [19], SVM [24] and Regression [20]. These were selected as the most common methods used currently. The work aims to pinpoint the best (highest recall coupled with highest recall consistency) classifier for classifying the outputs of the segmentation algorithm. The Naïve Bayes does not require parameter setting so we do not tune that algorithm. We ran various configurations (as outlined below) of the MLP and SVM algorithm on all 6 train/test set combinations and selected the configuration for each classifier with the highest recall. All classifiers use identical sets and are free to choose any attributes from each training set in turn. The relatively small size of the data prevents us using train and validation sets prior to classifying a blind test set. The regression algorithm does allow tuning but is slow to run (up to 1 day) with some configurations. For this algorithm, we ran some initial analyses and selected the best performing (highest recall accuracy) configuration when classifying the train set (set1) only.

Naïve Bayes assumes that the attributes $\boldsymbol{X}= \{x_1, x_2, x_3, ..., x_d\}$ are independent to simplify the classification task by allowing the class conditional densities $p(x_k \mid C_j)$ to be calculated separately for each attribute. This assumption appears not to affect the posterior probabilities greatly, especially in areas near decision boundaries, thus, leaving the classification task unaffected. We use the Naïve Bayes C source code available from [3] running under Linux.

The MLP neural network is a feed forward topology with a single hidden layer comprising 23 input neurons, a hidden layer of neurons and a single output neuron. We have 23 neurons in the input as there are 23 attributes in the input data and a single output neuron to represent perceptual relevance for these analyses. Selecting the number of hidden neurons is important. We tried various settings to choose the optimal configuration of the MLP. We selected between 3 to 12 hidden neurons and ran each MLP on each of the 6 train/test set combinations (60 runs in total). An MLP with 4 hidden neurons produced the highest recall. We then ran the 4 hidden neurons MLP for between 1000 and 7000 epochs. The MLP recall percentage increases from 1000 to 2000 to 3000 training epochs. However, with 4000 epochs, recall accuracy degrades markedly and remains worse for both 5000 and 7000 epochs which indicates overtraining. Thus, 3000 epochs produced the best results coupled with 4 neurons in the middle (hidden) layer. Multi-layer networks use a variety of learning techniques; we use back-propagation where the output values are compared with the correct answer during network training to compute the value of the error-function. In our analyses here, we use the MLP C source code available from [3] running under Linux.

For the SVM, we use C++ source code (LibSVM) available from [5] running under Linux. We use the nu-SVC SVM type (where nu is related to the ratio of support vectors and the ratio of the training error) with radial basis function kernels ($exp(-\gamma * | x_j - z |^2)$). All data attributes are scaled in the range *[0, 1]*. We used the script available with libSVM (grid.py) to select values for γ in the kernel function. This recommended 1.0, 0.125 and 0.0625. We then ran the SVM on the 6 train/test set combinations with each of these three γ settings along with 0.5 and 0.25. A setting of 0.125 produced the highest overall recall. With γ set to 0.125 we tried various nu-values (0.1, 0.3, 0.4, 0.5 (default) and 0.6). 0.4 produced the highest recall figure.

For the regression analyses, we use the Sagata regression program available from [20] which provides proprietary regression algorithms. It runs under MS Windows XP and sits on top of MS Excel. Our preliminary analyses which involved generating the regression equation using the training set1 and then classifying the same training set1 indicated that the combination of selecting an initial set of attributes using the MinPress regression algorithm with default settings then using standard stepwise with order up to 2 attributes followed by Least Squares regression to select the equation coefficients produced the highest recall.

MinPress is similar to stepwise regression except that attributes are selected based on improvements in the Press statistic defined as:

$PRESS = \Sigma_{i=1,...,n} w_i[y_i - y^{(i)} \bullet est(\boldsymbol{x}_i)]^2$ where $y^{(i)} \bullet est(\boldsymbol{x}_i)$ is the prediction at the data point x_i. Inputs, classes, and weights for the x_i-th record are omitted. The same model is fitted to the data minus the x_i-th record. This fitted model is used to make a prediction for x_i. This is $y^{(i)} \bullet est(\boldsymbol{x}_i)$.

Once we have used MinPress to select an initial set $\{S_1\}$, we supplement this set with a set of 2nd order attributes $\{S_2\}$ selected using standard stepwise regression [20].

We merge $\{S_1\}$ and $\{S_2\}$ giving the selected attributes $\{S\}$. We use Least Squares estimation (LSE) to select the regression equation coefficients: LSE derives the regression equation coefficients that minimize the sum of squared differences (residuals) between the regression equation predictions and the corresponding actual response (class) values (0 or 1 here).

3.5 Method

The SVM and Naïve Bayes are discrete classifiers; each record is classified as relevant or irrelevant so we use the classes *{0, 1}*. In contrast, the regression algorithm and MLP produce continuous classifications in the range *[0, 1]*. For classification (testing), we use a threshold value of 0.5 for the regression and MLP outputs. If the predicted output class score is >0.5 then the record is classified as relevant but if the output score value is <= 0.5 then we classify as irrelevant.

Each classifier is trained and tested with one pair of sets in turn and the outputs stored for recall accuracy calculation. Each classifier produces 6 separate equations/models for the data.

To measure success we recorded overall recall accuracy, (i.e., the number of perceptually relevant examples classified as perceptually relevant plus the number of perceptually irrelevant examples classifies as perceptually irrelevant) and the recall accuracy for the perceptually relevant examples. False positives (irrelevant shapes classified as relevant) increase the amount of data to be processed which is a nuisance factor but less serious than false negatives (relevant shapes classified as irrelevant) which indicate missing perceptually relevant shapes.

For example, for the pair "Train using set1 + test using set2", we train the classifier with the 435 records in set1 to produce a classifier model. For each of the 306 records in set2, we apply this classifier model to the record to produce a class prediction (perceptually relevant or perceptually irrelevant). We can then calculate the recall accuracy by counting the number of correct predictions and dividing this figure by the number of records in the test set. The set pairs are {train, test}: {set1, set2}, {set2 set1}, {set3, set4}, {set4, set3}, {set5, set6}, {set6 set5}

4. RESULTS & ANALYSIS

The recall accuracy for the four classifiers when run on each of the 6 train/test set combinations is listed in Table 3.

From Table 3, the MLP algorithm has the highest overall recall coupled with the highest recall accuracy for perceptually relevant and perceptually irrelevant shapes by a considerable margin. It also has consistently high recall. The regression algorithm produces the second highest recall figures with the SVM third highest overall. The Naïve Bayes performs worst except for correctly classifying the perceptually irrelevant shapes where it is third best.

It is noted that the size of the training set can have an adverse affect on classifier recall accuracy. The MLP performs worst on the smallest set2 for training with the largest set1 for testing combination and conversely performs best when training on the largest set1 and testing with the smallest set2. The overall recall drops from 98% to 94% so the MLP may be adversely affected by training set size. When the SVM trains on the larger set1 and classifies the smaller set2 it produces 93% recall accuracy. Conversely, when the SVM trains on the smaller set2 and classifies the larger set1 the SVM produces 80% recall accuracy. However, the SVM suffers its worst performance when training with set5 and testing with set6 so we cannot say conclusively whether it is adversely affected by training size at this stage. The Naïve Bayes also suffers a performance drop when using the smallest training set2 compared with the largest training set1 but similarly, Naïve Bayes suffers its worst performance when training with set6 and testing with set5 so again, we cannot say conclusively whether it is adversely affected by training size at this stage. The regression algorithm does not suffer a significant overall performance drop when comparing the largest and smallest training sets.

It is possible to look at the weights in an MLP to see which attributes are being used to classify the shapes. Roughness is weighted consistently highly (either +ve or –ve). AspectRatio, Stuffedness, AreaRatio, GapScore, Circularity, Eccentricity, Ellipticity & Triangularity are generally weighted high. The Fourier descriptors are occasionally weighted highly and the Hu moments are generally weighted low. Roughness indicates how convex a shape is. A high score indicates that the shape fills its convex hull and thus the shape is convex. Conversely, a low score indicates a concave shape. This corresponds with visual observations of the results of our experiments: for images comprising flood-filled regions in particular, convex shapes tend to be perceptually relevant. Where concave shapes are relevant (generally more so for line-based images or thin regions from our experiments) the MLP may use a combination of the other attributes to achieve the correct classification. AreaRatio is very similar to Roughness so we would not expect both to score highly. Circularity, Ellipticity, Triangularity and Stuffedness (with AspectRatio closely related to Stuffedness) all define specific shapes (circle, ellipse, triangle, and rectangle) and hence their applicability varies. GapScore is often weighted highly indicating that shapes with gaps vary in perceptual relevance from shapes without gaps in their perimeter. Eccentricity measures the regularity of a shape and we would expect regular shapes to be more perceptually relevant than irregular shapes. This hypothesis is borne out with the attribute's relatively high weighting.

It is interesting to consider the speed of training, as this indicates the utility of the method for practical applications. For the implementations of the four classifiers used here, the Naïve Bayes trains fastest, followed by the SVM and the MLP all of which train much faster than the regression program. For the data set combination set6 training and set5 testing, the MLP trains in 2.0 seconds, the SVM in 0.4 seconds and the Naïve Bayes trains in 0.3 seconds all running on a 3.4GHz Pentium PC with 2GB RAM running Linux. The regression program takes 40m 24s (2424 seconds) to complete the three steps of regression training on the same data set pair running on a dual 2.8GHz AMD Athlon PC with 3GB RAM running MS Windows XP with the regression program running on top of MS Excel. Obviously, we are comparing slightly different machines and different operating systems (3 C++ algorithms running under Linux on 3.4GHz Pentium PC and one Windows application running on dual 2.8GHz AMD Athlon PC with MS Windows XP) but the time difference between the C++ algorithms and the regression algorithm is still significant if speed is the overriding criterion for the user.

5. CONCLUSION

The work has shown that it is possible to train a classifier to select perceptually relevant closed figures from a segmented image, effectively capturing the segments that humans see in images. Our work has shown that the MLP network can be trained to achieve this with 96.4% recall accuracy overall. The MLP has the highest recall for the important category: the perceptually relevant examples, where it achieves 97.4% accuracy. It is noted that the training time for the MLP is 2 seconds compared to 0.3 seconds for the Naïve Bayes which trains fastest. Although the MLP is slower, the training time is still fast. Therefore, we have identified that an MLP with 4 hidden neurons and a single output neuron running as the optimum perceptual relevance classifier for the perceptual classifier task described in this paper.

We feel the approach described is very flexible and attained using actual human perceptual data. It is a universal approach providing a score of perceptual relevance (global goodness) across all shapes regardless of how they are derived. The approach reduces the number of shapes output by the closed shape identification algorithm and is a precursor to the matching phase of image retrieval. Classifying the closed shapes and discarding perceptually irrelevant shapes reduces the search space during image matching and retrieval. Each image is only represented by a sub-section of the candidate shapes output by the closed shape algorithm; the shapes classified as perceptually irrelevant are removed from the search space. Reducing the search space focuses on human-oriented shapes, speeds further processing during image matching and retrieval as fewer shapes need to be

processed and reduces the memory overhead of any further processing.

We intend to use the classifier within the PROFI project to generate perceptually relevant views of each image. The closed shape identifier produces a set of candidate shapes for each image. The classifier then reduces the set of candidates to the set of perceptually relevant shapes for that image. For all images combined, these reduced sets of shapes represent the database of perceptually relevant shapes for all images. Using shape attributes (such as the 23 attributes detailed in section 3.2, topology attributes such as touching and overlap relations and position attributes such as the centroid coordinates) to represent the shapes as a vector of attributes, the set of shapes for each image may be represented as a similarity graph for the image. In this similarity graph, the nodes are shapes and the arcs in the graph are the relations (similarity) between the shapes calculated using vector distances. Images (trademarks) may then be matched using graph isomorphism matching and attribute matching (vector distance) calculation. The more similar the graphs representing two images, the more similar those two images will be. Thus we can calculate the set of trademarks that are most similar to a query trademark using graph isomorphism and vector distance calculations on the shapes within the images. Graph isomorphism calculations are computationally expensive so by reducing the set of shapes representing each trademark by using our perceptual relevance classifier, we are minimising the graph sizes and minimising the calculation required. We are also eliminating noise (perceptually irrelevant shapes) from the calculation which may adversely affect accuracy.

The methods we have described and the resulting classifier models or regression equation are equally applicable to any underlying shape identifier algorithm such as region growing [11], watershed [12] or closed shape identification [13] providing the result of the algorithm may be represented by a list of boundary points to calculate the attributes used here. Obviously, other attributes could be incorporated or the attribute set changed if, for example fill points were available to allow fill point attributes to be used.

6. ACKNOWLEDGMENTS

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

7. REFERENCES

[1] Alwis, S. Content-Based Retrieval of Trademark Images, PhD Thesis, Dept. of Computer Science, University of York, UK, 1999

[2] Beucher, S. Watersheds of Functions and Picture Segmentation, Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'82. pp. 1928-1931, 1982.

[3] Borgelt, C. Machine Learning Algorithms Implemented in C, 2006, Note: Software available at http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html

[4] Chan, D. Y-M. and King, I. Genetic Algorithm for Weights Assignment in Dissimilarity Function for Trademark Retrieval. In 3rd International Conf. on Visual Information and Information Systems (VISUAL'99), LNCS vol 614, Amsterdam, The Netherlands, 1999. Springer Verlag.

[5] Chang, C.-C. and Lin, C.-J. LIBSVM: a library for support vector machines, 2001, Note: Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[6] Desolneux, A., Moisan, L., and Morel, J.-M. A theory of digital image analysis. 2004. Book in preparation

[7] Diederich, J. Explanation and artificial neural networks, International Journal of Man-Machine Studies: 37: 335-355, 1992.

[8] Eakins, J. P. Riley, K. J. and Edwards, J. D. Shape Feature Matching for Trademark Image Retrieval, In, Image and Video Retrieval: Second International Conference, CIVR 2003. LNCS, vol. 2728, Jan 2003, Pages 28 – 38.

[9] Flickner, M. Sawhney, H. Niblack, et al.. Query by Image and Video Content: The QBIC System, Computer, vol. 28, no. 9, pp. 23-32, Sept., 1995.

[10] Hodge, V.J., Eakins, J. & Austin, J. Eliciting Perceptual Ground Truth for Image Segmentation. Technical Report YCS 401(2006), Department of Computer Science, University of York.

[11] Hodge, V.J., Hollier, G., Eakins, J. & Austin, J. Eliciting Perceptual Ground Truth for Image Segmentation. In Proceedings International Conference on Image and Video Retrieval (CIVR2006). Tempe, Arizona, July 13-15, 2006.

[12] Hu, M.-K. Visual Pattern Recognition by Moment Invariants. IRE Transactions on Information Theory, IT 8:179-187, 1962.

[13] Jiang, H., Ngo, C.-W. & Tan, H.-K. Gestalt-based feature similarity measure in trademark database, Pattern Recognition, 39: pp. 988 – 1001, 2006.

[14] John, G.H. and Langley, P. Estimating Continuous Distributions in Bayesian Classifiers. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. pp. 338-345. Morgan Kaufmann, San Mateo, 1995.

[15] Koffka, K.. Principles of Gestalt Psychology. Harcourt Brace. New York, 1963.

[16] Lowe, D. Three Dimensional Object Recognition from Simple Two Dimensional Images. Artificial Intelligence: 31(3):355-395, 1987.

[17] Ren, M., Eakins, J. P. and Briggs, P. Human perception of trademark images: implications for retrieval system design. Journal of Electronic Imaging, 9 (4):564-575, 2000.

[18] Rosin, P. L. Measuring Shape: Ellipticity, Rectangularity, and Triangularity, In Proceedings of 15th International Conference on Pattern Recognition (ICPR'00) - Volume 1, 2000.

[19] Rumelhart, D. E. and McClelland, J. L. Parallel distributed processing: explorations in the microstructure of cognition (MIT Press, Cambridge, Massachusetts, 1986).

[20] Sagata Regression v1.0 Copyright © 2002-2003 Sagata, Ltd. www.sagata.com

[21] Sarkar, S. and Boyer, K.L. On optimal infinite impulse response edge detection filters IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI): 13(11): 1154-71 (1991).

[22] Saund, E. Finding Perceptually Closed Paths in Sketches and Drawings. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI): 25(4): 475-491, April 2003,

[23] Tzanetakis, G., Traka, M. and Tziritas, G. Motion estimation based on affine moment invariants. In Proc. European Signal Processing Conference (Euspico), Rhodes, Greece, 1998

[24] Vapnik, V.N. The Nature of Statistical Learning Theory. Springer, 1995.

[25] Wertheimer, M. Laws of Organization in Perceptual Forms (1923). In, Ellis (ed) A Source Book of Gestalt Psychology, Routledge & Kegan Paul, London 1938.

[26] Witten, I. and Frank, E. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations (2000), ISBN 1-55860-552-5

[27] Wuescher, D.M. and Boyer, K.L. Robust contour decomposition using a constant curvature criterion. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI): 13(1): 41-51, 1991.

[28] Zucker, S.W. Region Growing: Childhood and Adolescence, Computer Graphics & Image Processing: 5:382-399, 1976

Table 3 Table listing the recall scores for the four classifiers on each of the six train/test set combinations. The highest recall score for each row is indicated in bold font. The maximum column indicates the number of records, number of perceptually relevant (1) and perceptually irrelevant (0) records in the respective test sets.

Train/Test		Max.	Bayes	MLP	SVM	Reg.
Set1/set2	Correct	*306*	276	**300**	285	257
	1s correct	*130*	112	**126**	118	122
	0s correct	*176*	164	**174**	167	135
Set2/set1	Correct	*435*	356	**408**	349	359
	1s correct	*240*	172	**218**	179	175
	0s correct	*195*	184	**190**	170	184
Set3/set4	Correct	*370*	307	**362**	329	313
	1s correct	*169*	147	**161**	145	159
	0s correct	*201*	160	**201**	184	154
Set4/set3	Correct	*371*	306	**356**	314	339
	1s correct	*169*	148	**192**	166	145
	0s correct	*202*	158	164	148	**194**
Set5/set6	Correct	*376*	308	**362**	300	321
	1s correct	*171*	123	**161**	128	126
	0s correct	*205*	185	**201**	172	195
Set6/set5	Correct	*365*	286	**355**	304	329
	1s correct	*199*	130	**192**	155	169
	0s correct	*166*	156	**163**	149	160
Overall	Correct	*2223*	1839	**2143**	1881	1918
	1s correct	*1078*	832	**1050**	891	896
	0s correct	*1145*	1007	**1093**	990	1022
Overall %ge	Correct		82.73%	**96.40%**	84.62%	86.28%
	1s correct		77.18%	**97.40%**	82.65%	83.12%
	0s correct		87.95%	**95.46%**	86.46%	89.26%

Probabilistic Matching and Resemblance Evaluation of Shapes in Trademark Images*

Helmut Alt
Institute of Computer Science
Freie Universität Berlin
alt@mi.fu-berlin.de

Ludmila Scharf
Institute of Computer Science
Freie Universität Berlin
scharf@mi.fu-berlin.de

Sven Scholz
Institute of Computer Science
Freie Universität Berlin
scholz@mi.fu-berlin.de

ABSTRACT

We present a novel matching and similarity evaluation method for planar geometric shapes represented by sets of polygonal curves. Given two shapes, the matching algorithm randomly generates a point sample from each shape and records a vote for a transformation which maps one sample to the other. The experiment is repeated many times. Clusters of votes in the transformation space indicate good candidate transformations for matching the two shapes. Unlike most voting schemes, though, the samples taken in one random experiment are extended as much as possible and the vote is weighted depending on the samples. The best clusters are those with a large total weight. The second part of the method is a resemblance evaluation of the two matched shapes. The definition of our resemblance function incorporates the proximity of line segments as well as the similarity of their slopes. The system is evaluated using the MPEG-7 shape silhouette database and a collection of 10 745 trade mark images. The experiments demonstrate a high performance of our algorithms for contour shapes as well as for trademark images.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Probabilistic algorithms; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling; I.5.3 [**Pattern Recognition**]: Clustering

General Terms

Algorithms, Experimentation

Keywords

Trademark image retrieval, Shape matching, Probabilistic algorithms, Shape similarity

*This research was supported by the European Union under contract No. IST-511572-2, Project Perceptually-Relevant Retrieval of Figurative Images (PROFI).

CIVR'07, July 9–11, 2007, Amsterdam, The Netherlands.

1. INTRODUCTION

Motivated by the task of automated retrieval of figurative images such as trademark images we developed an algorithm for the evaluation of shape similarity. It consists of two main phases: matching and evaluation of the resemblance of the matched shapes.

The approach we introduce for matching two geometric shapes S_1 and S_2 modeled by sets of plane polygonal curves, is close to an intuitive notion of "matching", i.e., find one or more candidates for the best transformation, that when applied to the shape S_1 maps the most similar parts of the two shapes to each other. As allowable classes of transformations we will consider *translations*, *homotheties* (scaling and translation), *rigid motions* (rotation and translation), *similarities* (rotation, scaling, and translation), and general *affine maps*.

The matching step of our algorithm is a voting scheme. Unlike in most well known approaches including geometric hashing [16], pose clustering (generalized Hough transform) [2, 11, 12, 14], and the random sample consensus (RANSAC) [10], we do not use a minimum sample of features to compute the model parameters (the matching transformations in this case), but we find large consistent sequences of the corresponding points of two shapes voting for the same transformation. This transformation gets a vote which is weighted depending on the size of the sample and the quality of the match. Thus, we get a distribution of weighted votes in the transformation parameter space. Similar as in the pose clustering approach, we then take the largest clusters as candidate transformations, where largest clusters are those with the highest total weight.

After several candidate transformations of one shape have been identified by the matching algorithm, each of these transformations t_i is applied to the shape S_1 and the similarity of the shape S_2 and the transformed shape $t_i(S_1)$ is validated using the resemblance function described in section 3. The proposed resemblance function incorporates two perceptual factors: proximity and parallelism (or factor of direction), that is, the resemblance value is high if the distances between the points of two shapes are small and the line segments contained in the shapes are nearly parallel. The transformation with the highest similarity value is then selected as the best match.

We address the problem of matching the complete shape S_1 to the complete shape S_2, called *complete-complete matching* (CCM). In addition, we consider the problem of *complete-partial matching* (CPM), i.e., matching S_1 completely as good as possible to some part of S_2, and *partial-partial*

matching (PPM), i.e., matching some part of S_1 as good as possible to some part of S_2. Clearly, both partial matching problems CPM and PPM are not uniquely specified since there is a trade-off between the quality of the match and the size of the matched parts. Which of these two criteria is more important, depends on the application. We address this problem by introducing a parameter which regulates the influence of the quality of match and the matched size on the similarity value.

Both the matching procedure and the resemblance function are designed to be robust with respect to noise and to differences in the representations of the the shapes. The data used in the experiments described in section 4 show the importance of that robustness.

2. THE MATCHING ALGORITHM

Given two sets S_1, S_2 of planar polygonal curves, a transformation t is searched for, that best maps S_1 to S_2. The classes of allowable transformations considered here are translations, homotheties (scaling and translation), rigid motions (rotation and translation), similarities preserving the direction of rotation (rotation, scaling and translation – without reflection), and general *affine maps*.

The intuition behind that matching is that features of the first set should be mapped into the proximity of corresponding features of the second set. If the images are not similar as a whole but do contain several independent subsets of corresponding features, there may not exist a single transformation but several transformations, each one matching only a subset of corresponding features. The course of the polygonal curves may be very helpful in identifying corresponding features. On the other hand different segmentations at crossings or different appearances of discontinuities may make this identification more difficult. Therefore a probabilistic voting scheme is applied here that uses votes of different weight and gathers them to form a set of candidate transformations.

In [3] we described a probabilistic matching approach related to the generalized Hough transform and briefly presented the idea of the new voting scheme which is described in detail here: During one random experiment a sample is a set of pairs of corresponding points from the two shapes. The quality of the match between two finite ordered point sets is measured by the weighted sum of quadratic distances between the corresponding points. The set of corresponding points is iteratively extended until no further data are available or the samples are no longer consistent. The resulting preliminary transformation is weighted with the quality of the match and the size of the matched point sets. Then we get a weighted sample of the transformation space, where the neighborhoods with large weight are likely to contain candidates for transformations resulting in a good match for the shapes. The idea behind this, is that a transformation which gives a good match for the shapes, would give a good match for larger sets of points on these shapes. The details on selecting the point sets and computing the candidate transformations are presented in the following.

The problem of computing preliminary transformations consists of two subproblems: one is to find correspondences between features, the other is to find a transformation that maps the corresponding features to each other.

2.1 Finding correspondences

Conspicuous features of curves arise from regions of high curvature [5] – regarding polylines these regions are the vertices. But not every vertex, even though its turning angle may be large, yields a feature recognizable by a human observer. For this reason, the mapping algorithm tries to find corresponding vertices but also may match a vertex without corresponding peer to a point lying on a line segment.

The initial part of every vote is the random choice of a single vertex in each of the two sets of polylines, and the direction for traversing the list of subsequent vertices (also both possible directions may be processed). The pairs of points are added to the set one by one. In each iteration step the matching transformation is updated. The final transformation for the set is weighted and handed over to the clustering algorithm.

Let p_0 be the randomly chosen vertex from the first set S_1 of planar polylines and let $p_1, \ldots, p_k$ be the subsequent vertices with respect to the randomly chosen direction. Analogous let q_0 be the randomly chosen vertex from the second set S_2 of planar polylines and let $q_1, \ldots, q_l$ be the subsequent vertices. The pair (p_0, q_0) is added to the – so far empty – sample set S.

In each iteration step the distances from the last added pair of points to the next vertices on the corresponding polylines are computed. If the two computed distances are nearly equal, the next two vertices are taken as a corresponding pair which is added to the sample set S. Otherwise a vertex surrogate is created for the polyline with a larger distance. A surrogate is a point lying on an edge of the polyline, but nevertheless is treated like a vertex. It is chosen to have the same distance to its predecessor as the corresponding two vertices of the other polyline have.

When the end of a polyline is reached, then starting from the initial pair the traversal is performed in the other direction.

2.2 Calculating the transformations

For every new pair of vertices or vertex surrogates added to the sample set S a transformation is computed based on a least squares approach. The easiest way would be to compute the transformation that minimizes the sum of the squared distances of the vertices. This would favor parts with many vertices over parts with less vertices regardless of the extent and the expressiveness. In order to avoid this, we compute the transformation $t \in T$ that minimizes the sum of the weighted squared distances $\varepsilon(t) = \sum_{(p_i,q_i)\in S} w(p_i, q_i) \left\| q_i - t(p_i) \right\|^2$ with $w(p_i, q_i)$ being half the length of the edges incident to p_i and q_j.

If the class of the allowed transformations does not include scaling, which is the case for translations and rigid motions, we can determine vertex correspondences as described above and then compute the transformation minimizing the sum of weighted squared distances. However, if scaling is allowed, then the process of finding correspondences is no longer independent from the transformation, since the transformation could change distances between the vertices. We cope with this problem by using a prescaling factor s, which is randomly chosen such that $\mathrm{ld}(s)$ is normally distributed with mean value $\mathrm{ld}(\bar{s})$, where $\bar{s}$ is the prescaling factor of the transformation rated best so far with the initial value of $\bar{s}$ set to 1. S_1^s denotes the set S_1 scaled by s. For the rest of the vote, every operation concerning the first set S_1 (e.g.

computing the distance between two vertices) is performed on S_1^s.

Detailed analysis shows, that for all the classes of transformations considered here, i.e., translations, homotheties, rigid motions, similarities, and affine maps, having computed the transformation minimizing the weighted sum of squared distances for a sample set S, we can compute in constant time the optimal transformation for the set $S \cup \{(p, q)\}$, which is S extended by a new pair of points p and q. This fact is important since we iteratively add new pairs of corresponding points to our sample set and, thus, compute a sequence of transformations until the sample is no longer consistent or all data points are added to the set.

2.3 Checking Consistency

In each iteration step it is checked whether the error introduced by the new added pair is still within tolerance bounds. We define the maximum tolerated error for a sample as a linear function of the perimeter of the bounding box containing the sample. The perimeter of the bounding box is multiplied with a constant parameter called relative error threshold. If the error introduced by the last added pair is too big, the traversal of the polylines is ceased, as illustrated in Figures 1 and 2: the bold polylines are traversed up to the end of the dashed parts. The transformation for which the error was farthest from the tolerance bound is weighted and handed over to the clustering algorithm (the transformation calculated for the bold polylines up to the beginning of the dashed line in the example).

This definition of the stop criterion and the choice of the best index are invariant under scalings and can be done in constant time. To achieve invariance under rotations also, the bounding box had to be replaced by the minimum enclosing circle.

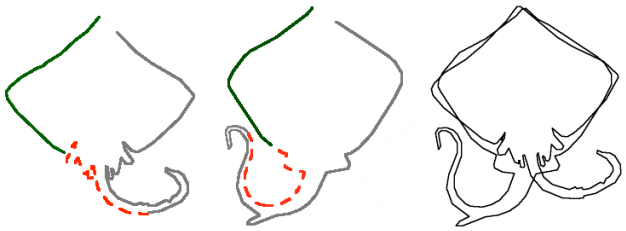


Figure 1: Two instances of the MPEG-7 shape B data set (ray-7, ray-20 and both mapped).

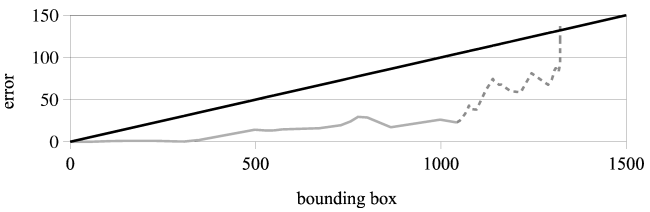


Figure 2: Perimeter of the bounding box vs. error of last added pair of points. The part of the polyline defining the transformation for this vote is plotted as a solid light grey line, the part skipped is plotted dashed. The solid dark line is the maximum tolerated error bound.

2.4 Weighting the transformations

The two factors that have to be considered for weighting a transformation t are the expressiveness of the sample and the quality of the match. Let ε be the sum of weighted squared distances, let $w(S)$ be the sum of all weights of the pairs of points in the sample set S, and let D_{bb} be the diameter of the bounding-box containing the covered part of the polyline. Defining the relative root mean square error $e = \sqrt{\varepsilon / w(S)} / D_{bb}$ yields a value representing the quality of the match which is invariant under scalings.

The match score or weight $W(t)$ of a transformation t is then defined as $W(t) = l/(1+\gamma \cdot e)$ with γ being an arbitrarily chosen constant for balancing out the impact of the length l and the error e.

The most common technique for the clustering of transformations (often referred to as pose clustering) – histogramming the transformations in the multidimensional transformation space (see [12]) – discards the effects on the transformed shapes. Two rotations may yield nearly the same results if applied to a shape with its center being the origin, or totally different results if the shape's center is far away from the origin. To avoid this imbalance, a distance measure for transformations is used here, which considers the shapes' properties. Let t_1 and t_2 be two arbitrary transformations and let S_1 be the transformed shape. The distance measure $d_{S_1}(t_1, t_2) = \max_{p \in S'} \|t_1(p) - t_2(p)\|$, with S' being the set of the vertices of the bounding box of S_1 forms a metric space for affine maps, under the assumption that the four points of S' are pairwise different. The distance between two transformations depends, thus, on the shape to be transformed and reflects the difference in the image of the shape under the considered transformations.

2.5 Clustering

A cluster in our sense is a region of limited diameter, which subsumes a considerable amount of weight of the enclosed input points (transformations). In the clustering process we want to find all clusters with large weight because they give evidence of good matching transformations.

Let T_n be the set of n transformations generated by n random experiments and W_i be the weight of a transformation $t_i \in T_n$. For a fixed cluster radius r_c a cluster C_t with center $t \in T_n$ is defined as the set $\{t_i \in T_n | d(t_i, t) < r_c\}$ that is the set of transformations with distance less than r_c to the center. The weight of a cluster is defined as the sum of the weights of its elements. This definition is related to what is called *naive density estimator* in statistics.

The transformations that are considered as center of a cluster are identified as follows: $t_i \in T_n$ is called *dominator* of $t_j \in T_n$ if and only if $d(t_i, t_j) < r_c$, $W_i > W_j$, and no other transformation is dominator of t_i. Each transformation $t \in T_n$ that has no dominator is the center of a cluster C_t. In other words: a transformation t either is the center of a cluster or it is contained in at least one cluster of its dominators. This definition allows for a fast computation of all clusters and their weights.

The clusters may be determined by iteratively taking the transformation with highest weight as center of a cluster, removing the cluster's members from the set of potential centers and continuing with the reduced set. A naive algorithm would need time quadratic in the number of transformation. This can be decreased by partitioning the transformation space and organizing it in a tree structure.

Every node of the tree may store a cluster which stores the transformations belonging to it. A node u_i^j on level i with index j represents a ball with some radius r_i around its center (its cluster's center). It may have arbitrarily many children, each one representing a ball with radius $r_{i+1} = r_i/2$ and a center that lies inside the ball represented by u_i^j, see Figure 3 for a schematic illustration. The root node represents a ball with radius r_0 containing all sample transformations. The smallest radius in the tree is the given cluster radius r_c. The children of a node are ordered, each one only responsible for the part of the space not covered by its preceding siblings.

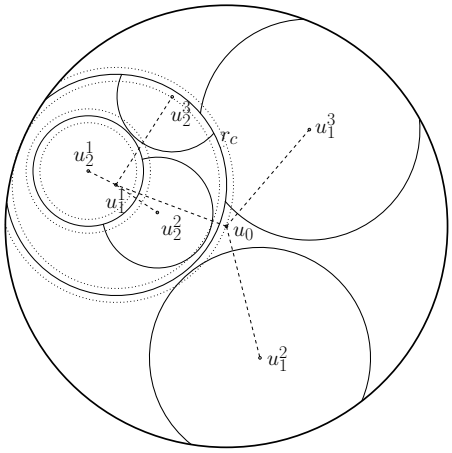


Figure 3: Partition of the transformation space.

The center c_0 and the radius r_0 of the ball represented by the root node may be easily computed for the classes of transformations that do not allow scalings: Let c_{bb1}, c_{bb2} denote the centers and D_{bb1}, D_{bb2} denote the diameter of the bounding boxes of the shapes S_1 and S_2 respectively. Then c_0 is chosen to be the translation defined by $c_{bb2} - c_{bb1}$. Any transformation t generated by the random experiments will fulfill the condition that the transformed bounding box of S_1 at least touches the bounding box of S_2. Therefore $d_{S_1}(t, c_0) \leq D_{bb1}/2 + D_{bb2}/2 + D_{bb1}$.

For the classes of transformations that do allow scalings the space is not bounded in such a natural way. However, if the application provides a bound on the maximum scaling factor s_{max} the distance between any transformation t (homothety or similarity) and c_0 can be bounded in a similar way: $d_{S_1}(t, c_0) \leq D_{bb1}/2 + D_{bb2}/2 + s_{max} D_{bb1}$. If no such bound is given, the root node and its radius can be updated during the construction of the tree.

The clustering is performed as follows: The sample transformations are sorted according to their weight and they are processed in descending order. Beginning from the empty tree with root node (c_0, r_0) in each iteration step a transformation t is added to the tree. First, the tree is searched for all clusters such that t is contained in a ball of radius r_c around the center of the cluster. If such clusters exist, then t is added to all these clusters. If no cluster containing t is found, a new cluster C with center t is created and inserted into the tree.

When a node with center t_u and radius r is searched for clusters neighboring a transformation t, the distance $d(t_u, t)$ is computed. If $d(t_u, t) < r - r_c$, all the clusters worth considering have to lie inside the node's ball and the subsequent siblings of the node may be discarded. If $d(t_u, t) > r + r_c$ the clusters have to lie outside and the node u may be discarded. In the other cases the node and the subsequent siblings have to be considered. The search is then performed recursively in all nodes, that may contain t.

For a cluster C being inserted into a node representing a ball with radius r, if there exists at least one child node representing a ball containing the center of C, C is recursively inserted into the first such child. Otherwise, a new child holding C with radius $r/2$ and a center corresponding to the center of C is created and appended to the list of child nodes.

After having processed all transformations, the clusters are sorted according to their weights. The clusters with the highest weights provide the candidate transformations. The number of candidate transformations may be chosen as a constant or we may consider the clusters with weight up to a certain fraction of the maximum weight.

Properties of the tree. Since the number n of samples is finite, the considered transformation space T_i is bounded, i.e. $\exists r_i \in \mathbb{R} : \forall x, y \in T_i : d(x, y) < r_i$. A subset $C_i \subset T_i$ is called an ε-packing if and only if $\forall x, y \in C_i : d(x, y) > 2\varepsilon$. The size of the largest ε-packing is called the packing number $P(T_i, \varepsilon)$. For $\varepsilon \to 0$, $P(T_i, \varepsilon) = O\left(\left(\frac{r_i}{\varepsilon}\right)^{\mathcal{D}}\right)$ for $\mathcal{D}$ being the dimension of the space [6]. Therefore the maximum number of balls of radius $r_i/2$ with centers in T_i such that no center is contained in another ball, is in $O(2^{\mathcal{D}})$. This means that the number of children a node of the tree may have is bounded by a constant only depending on the dimension of the transformation space. The depth of tree is at most $\lceil \lg(r_0/r_c) \rceil$.

3. THE SIMILARITY FUNCTION

After some candidate transformations have been found, a distance or similarity measure has to be applied to rate the similarity of the two matched shapes. Most of the existing distance measures are either not applicable to the sets of polygonal curves (like Fréchet distance or turning angle function) or are maximum based distances (like Hausdorff distance) and therefore are too sensitive to noise. We describe a new similarity measure which averages over the whole set of polylines, so it is not sensitive to noise, but looses the property of being a metric. It takes into account special properties of line segments, and is invariant to different parameterizations or splitting of polylines.

The resemblance function is defined for every point of the polylines and stands for how good the point is represented by the other set. It is composed of the point's distance to the points of the other shape and of the similarity of slopes.

Let h be a straight line segment of the first set S_1 with endpoints p_0 and $p_0 + v$, and g a segment of the second set S_2 with endpoints q_1, q_2. Let h' and g' denote the supporting lines of the segments h and g respectively. For a point $p \in h$ we define the distance to g as the distance to a point q on g, such that the orthogonal projection of q on h is exactly p, if such a point q exists. Otherwise, if p' is the nearest orthogonal projection of an endpoint of g on h', the the distance from p to g is defined as the distance from p to p' plus the distance from p' to the corresponding endpoint

of g. Formally, consider a parameterization of h': $p(\lambda) = p_0 + \lambda \cdot v$, $\lambda \in \mathbb{R}$. Let $q(\lambda)$ denote a point on g', such that $p(\lambda)$ is an orthogonal projection to h' of $q(\lambda)$. Further, let $p_1 = p_0 + \lambda_1 \cdot v$ and $p_2 = p_0 + \lambda_2 \cdot v$ denote projections of the endpoints q_1 and q_2 on h', and w.l.o.g., let $\lambda_1 < \lambda_2$, see Figure 4 for an illustration. The distance function is then defined as

$$\delta_{h,g}(\lambda) = \begin{cases} \|p_1 - q_1\| + \|p(\lambda) - p_1\|, & 0 \le \lambda < \lambda_1 \\ \|p(\lambda) - q(\lambda)\|, & \lambda_1 \le \lambda \le \lambda_2 \\ \|p_2 - q_2\| + \|p(\lambda) - p_2\|, & \lambda_2 < \lambda \le 1 \end{cases}$$

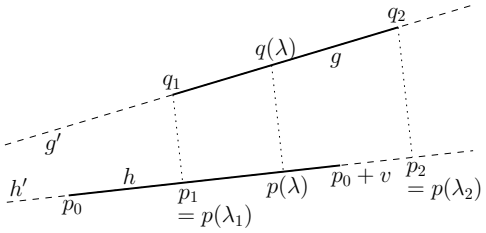


Figure 4: Definition of the distance between two line segments

This definition of the distance (unlike the Euclidean distance) ensures that the function $\delta_{h,g}(\lambda)$ is piecewise linear, which allows for a fast computation of the resemblance function.

If the distance of a point $p \in h$ and the segment g equals zero, that is, p lies on g, then we say that p is exactly represented by g – the degree of being represented, therefore, is 1. The greater the distance gets, the lesser p is represented by g. The decrease in similarity is weighted with the size of the shape S_1. In [13] an *inverse distance function* is used in a similar context for the rating of transformations in an optimization problem. For their task they chose a function that exponentially decreases with higher Euclidean distance to value the correspondence of features (see Figure 5(a)).

In the present case the goal is not to find an optimum, but to rate a given configuration. Small deviations in the position of the features of the two sets should not result in an excessive decrease in similarity function. Therefore an inversion function with a high (negative) slope around the y-axis is inapplicable. The function $\alpha'_{h,g}(\lambda) = \exp(25(\delta_{h,g}(\lambda)/D_{S_1})^2)$, where D_{S_1} denotes the diameter of the shape S_1, seems more promising. It rates pairs with a distance less than 5 % of the diameter very high (over 0.9) and with a distance of more than 25 % of the diameter very low – around 0.2 (see Figure 5(b)). To make the computation easier, the piecewise quadratic function

$$\alpha_{h,g}(\lambda) = \max(1 - 25\left(\frac{\delta_{h,g}(\lambda)}{D_{S_1}}\right)^2, 0)$$

is chosen. It has the same characteristics for small distances (up to 10 %) but decreases faster for greater distances (see Figure 5(c)).

The resemblance of two line segments also depends on their *slopes*. Line segments with similar slopes should get a higher resemblance value, so a slope factor $\beta_{h,g}$ is defined as

$$\beta_{h,g} = \cos\left(\angle(h,g)\right)^4$$

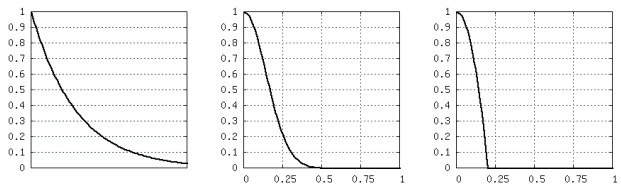


Figure 5: (a) exponentially decreasing inverse distance; (b) inversion function α'; (c) inverse distance function α

It rates pairs with a difference in slopes of less than 10° very high (over 0.9) and with a difference of more than 45° very low (below 0.25). The exponent 4 was chosen experimentally.

The *resemblance function* ϕ_h for a line segment h is defined as a combination of inverse distance function and slope rate:

$$\phi_h(\lambda) = \max_{g \in S_2} (\alpha_{h,g}(\lambda) \cdot \beta_{h,g}) \quad (1)$$

In applications that follow human perception, parts with many parallel line segments have to be prevented from dominating over parts with solitary line segments. Therefore a weight function ω is defined analogously to the resemblance function. It rates the density of similar line segments of an image. For a line segment $h \in S_1$ it is defined as

$$\omega_h(\lambda) = \frac{1}{\sum_{g \in S_1} (\alpha_{h,g}(\lambda) \cdot \beta_{h,g})} \quad (2)$$

Note that the weight function rates the similarity of a segment h to the other segments in the same set.

The directed resemblance measure for two sets of line segments $\Phi_{\rightarrow}(S_1, S_2)$ is defined as a weighted mean over all points of the shape S_1:

$$\Phi_{\rightarrow}(S_1, S_2) = \frac{\sum_{h \in S_1} \left(\int_{\lambda=0}^{1} \phi_h(\lambda) \cdot \omega_h(\lambda)\, d\lambda \cdot l_h \right)}{\Omega(S_1)}, \quad (3)$$

with l_h being the length of h and $\Omega(S_1)$ being the total weight of S_1: $\Omega(S_1) = \sum_{h \in S_1} \left(\int_{\lambda=0}^{1} \omega_h(\lambda)\, d\lambda \cdot l_h \right)$.

The undirected resemblance measure $\Phi(S_1, S_2)$ is defined as the weighted arithmetic mean:

$$\Phi(S_1, S_2) = \frac{\Phi_{\rightarrow}(S_1, S_2) \cdot \Omega(S_1) + \Phi_{\rightarrow}(S_2, S_1) \cdot \Omega(S_2)}{\Omega(S_1) + \Omega(S_2)}. \quad (4)$$

From this resemblance measure a deviation or distance measure may be derived, but of course this will never be a metric as the triangle inequality does not hold.

Computational complexity. The resemblance value is computed evaluating the integrals of a combination of the resemblance function and the weighting function for every line segment. For two sets with n line segments each, the resemblance function – as defined in Equation (1) – for a single line segment is the upper envelope of at most $4 \cdot n + 1$ regular (partially defined) functions. Using quadratic functions, each pair intersects at most 2 times (unless equal). According to the upper bound on the length of Davenport-Schinzel sequences [1] the complexity of the upper envelope of the

$4 \cdot n + 1$ functions is bounded by $O(n \cdot 2^{\alpha(n)})$ with α being the inverse Ackermann function.

The weighting function for a single line segment – as defined in Equation (2) – is the sum of n functions, each one split into at most 4 regular pieces. The number of intervals for the sum is at most $3n+1$. So the overall complexity for all the line segments is bounded by $O(n^2 \cdot 2^{\alpha(n)})$.

3.1 Partial similarity function

For the *complete-complete matching* resemblance function as defined in Equation (4) we took a weighted combination of two one-sided resemblance values. Note that the definition of the directed resemblance function (Equation (3)) applied to the complete shapes S_1 and S_2 gives us a score of how good the complete shape S_1 is matched to shape S_2 and, therefore, a score for the *complete-partial matching*.

For *partial-partial matching* we keep for each cluster a record of which parts of the shapes contribute to the transformations contained in this cluster. Let C be a cluster and $S_1^C \subset S_1$ and $S_2^C \subset S_2$ are the parts of the shapes that contributed to the transformations in C. Then, we compute the resemblances for the matched parts: $s_1 = \Phi_{\rightarrow}(S_1^C, S_2^C)$ and $s_2 = \Phi_{\rightarrow}(S_2^C, S_1^C)$. These values waive the remaining parts S_1
S_1^C and S_2
S_2^C of the shapes completely, so in general the highest values would be achieved by clusters matching very small parts perfectly.

The size of the matched parts also has to affect the value of partial similarity. Therefore, we compute a ratio of the matched parts as $\rho_1 = \frac{|S_1^C|}{|S_1|}$ and $\rho_2 = \frac{|S_2^C|}{|S_2|}$ respectively where $|S_i|$ denotes the total length of the polylines of the shape S_i and $\left|S_i^C\right|$ denotes the length of the parts matched contributing to the cluster C. A weight factor $f_i = 1 - (1 - \rho_i)^k$ (see Figure 6) if defined, where k is a (user-defined) parameter; the default value of k in our implementation is 3. The maximum of two weight factors $f^* = \max(f_1, f_2)$ is then used to adjust the resemblance value: $s^* = f^* \frac{s_1+s_2}{2}$.

The choice of the factor function was motivated by the following consideration: if large parts of at least one shape are matched, we want to leave the resemblance value almost unchanged, and give larger penalties the smaller the matched parts get. With the parameter k the user can control these penalties. If k is large, the resemblance value stays almost unchanged even for small parts, whereas for small values of k the quality of match decreases with the relative size of matched parts.

4. EXPERIMENTAL RESULTS

We implemented the matching algorithms and the resemblance measure as an automated application that finds the best resemblance value for every pair of shapes from a given set of shapes. The "CE-Shape-1" part B dataset from the MPEG-7 shape silhouette database was used as test data. It consists of 1400 (mostly) silhouette images, subdivided into 70 classes containing 20 related images each. From the images the outer closed contours were extracted. The polylines for which every vertex corresponds to a pixel, were then simplified using the Douglas-Peucker algorithm [7].

The resemblance of the shapes was tested under similarity transformations including reflections. To avoid unnecessarily many unsuccessful attempts, the shapes were scaled in advance so that their bounding boxes had the same diameter. The whole comparing process was done repeatedly, 3 times with the original (pre-scaled) shape and 3 times with one shape flipped to incorporate reflections. As result of the comparison of two shapes the highest resemblance value encountered for any of the candidate transformations and for any of the iterations was taken.

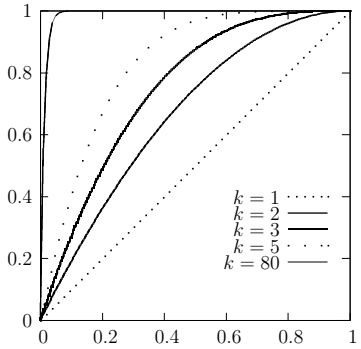


Figure 6: Weight factor function parameterized by k.

Every shape was compared to all the 1400 shapes of the set, including itself, and the nearest neighbors, that is shapes with highest resemblance value, were determined. The performance was rated based on three values:
True Positives as Nearest Neighbors: the ratio of shapes from the same class found as consecutive first nearest neighbors. The average for all the 1400 shapes was 67.78%.
True Positives in Class Size: the ratio of shapes from the same class found among the 20 nearest neighbors. The average for all the 1400 shapes was 76.89%.
True Positives in Double the Class Size: the ratio of shapes from the same class found among the 40 nearest neighbors, the so called bull's eye performance. The average for all the 1400 shapes was 84.28%. The best bull's eye performance of 84.33% on the MPEG-7 shape silhouette database was reported by Attalla and Siy in [4].
Our algorithm is also integrated into SIDESTEP, a system for evaluation of shape-based retrieval algorithms, which is described in [15].

We also evaluated our system using a collection of 10 745 abstract images from the UK Trade Marks Registry and a set of 24 image queries. This is the same test set as was used for the evaluation of the Artisan system as reported in [8]. The set of relevant images for each query was selected by experienced trademark examiners and was used as a benchmark for the system evaluation.

We evaluated the performance of our system on the trademark image set according to the performance measures used in [8]: *normalized recall* R_n, *normalized precision* P_n and *normalized last place* L_n, which are defines as

$$R_n = 1 - \frac{\sum_{i=1}^{n} R_i - \sum_{i=1}^{n} i}{n(N-n)}$$

$$P_n = 1 - \frac{\sum_{i=1}^{n} (\log R_i) - \sum_{i=1}^{n} (\log i)}{\log\left(\frac{N!}{(N-n)!n!}\right)}$$

$$L_n = 1 - \frac{R_l - n}{N - n},$$

where n is the total number of the relevant images, N is the size of the whole collection, R_i is the rank at which relevant image i is actually retrieved, and R_l is the rank at which the last relevant image is retrieved. All three measures rank a system's retrieval performance in response to a query from 0 to 1, with 1 meaning perfect retrieval. The major difference between normalized recall and precision is that normalized recall gives higher weighting to success in retrieving the first few items, while normalized precision gives equal weighting to all retrievals. The last place ranking indicates the number of retrieved items a user has to search in order to have a reasonable expectation of finding all relevant items. This measure is useful for applications requiring an exhaustive search, for example trademark retrieval.

The performance achieved by our system on the trademark test set is: normalized recall of 0.93, normalized precision of 0.71, normalized last place of 0.68. The early implementation of the Artisan system (which is regarded as one of the most comprehensive trademark retrieval systems in the current literature [9]) had the values of 0.90, 0.63, and 0.56, respectively.

The experiments show that the algorithm is robust with respect to noise and to differences in the representation of the shapes. However, apart from the good results achieved, we recognized some cases – especially among the trademark images – that are problematic for our approach:

- frames
  If the important part of a trademark image is surrounded by some kind of a simple frame, most humans do not pay much attention to that frame. The similarity measure however is influenced by it, because the frames naturally are larger than the part contained in it. To tackle this problem by using the partial-partial matching variant may result in high similarity values for completely different logos just because the frames are identical.

- spatially independent parts
  Comparing two images, that consist of two or more spatially independent parts and the corresponding parts are similar but arranged in slightly different ways, most humans do not care about the differences. However, there exists no affine map that aligns all parts properly at the same time.

To overcome these problems will be part of our future work.

We think the results achieved on both test sets are highly encouraging. They indicate that our method is a general purpose matching technique, not limited to contour shapes, but also performs well on complex shapes within the context of the non-trivial task of trademark image retrieval.

5. CONCLUSIONS

In this paper we presented a shape matching algorithm that randomly selects a point sample in each shape and gives a vote to a transformation which maps one random sample to the other minimizing the squared distances between the corresponding points. Instead of selecting a minimum size sample for the given class of transformations, as is usual in voting based methods, we extend the samples until the whole data is incorporated or the samples are no longer consistent. The transformation matching the sample sequences is then weighted according to the quality of match and the size of the samples. After sufficient number of random experiments the weighted votes in transformation space are clustered and the clusters with high total weight are taken as candidate transformations.

The second part of our method is similarity evaluation. Each candidate transformation is applied to one shape and the resemblance of the two shapes is rated according to the distance between the points of two shapes and to the similarity in slopes of the straight line segments contained in the shapes. We also define complete-partial similarity variant of our resemblance function, which reflects how similar the complete shape S_1 is to some parts of the shape S_2, and a partial-partial similarity variant, i.e., how good a part of shape S_1 matches some part of shape S_2.

We applied the implementation of our algorithms to the "CE-Shape-1" part B dataset from the MPEG-7 shape silhouette database, and to a test collection of 10 745 trade mark images provided by the UK Trade Marks Registry with a set of 24 image queries, both with convincing results.

The challenges mentioned in section 4 may be tackled by dividing the images into meaningful parts, weighting them, and applying the matching and similarity evaluation as presented in this paper to this parts.

6. REFERENCES

[1] P. K. Agarwal and M. Sharir. Davenport-Schinzel sequences and their geometric applications. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 1–47. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

[2] A. S. Aguado, E. Montiel, and M. S. Nixon. Invariant characterisation of the hough transform for pose estimation of arbitrary shapes. *Pattern Recognition*, 35:1083–1097, 2002.

[3] H. Alt, L. Scharf, and S. Scholz. Probabilistic matching of sets of polygonal curves. In *Proceedings of the 22nd European Workshop on Computational Geometry (EWCG)*, pages 107–110, Delphi, Greece, March 2006.

[4] E. Attalla and P. Siy. Robust shape similarity retrieval based on contour segmentation polygonal multiresolution and elastic matching. *Pattern Recognition*, 38(12):2229–2241, December 2005.

[5] F. Attneave. Some informational aspects of visual perception. *Psychological Review*, 61(3):183–193, 1954.

[6] K. L. Clarkson. Nearest-neighbor searching and metric space dimensions. In G. Shakhnarovich, T. Darrell, and P. Indyk, editors, *Nearest-Neighbor Methods for Learning and Vision: Theory and Practice*, pages 15–59. MIT Press, 2006.

[7] D. Douglas and T. Peuker. Algorithms for the reduction of the number of points required to represent a digitised line or its caricature. In *The Canadian Cartographer*, volume 10, pages 112–122, 1973.

[8] J. P. Eakins, J. M. Boardman, and M. E. Graham. Similarity retrieval of trademark images. *IEEE MultiMedia*, 5(2):53–63, 1998.

[9] H. Jiang, C.-W. Ngo, and H.-K. Tan. Gestalt-based feature similarity measure in trademark database. *Pattern Recognition*, 39(5):988–1001, 2006.

[10] R. C. B. Martin A. Fischler. Random sample

consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981.

[11] S. Moss and E. R. Hancock. Pose clustering with density estimation and structural constraints. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2085–2091, 1999.

[12] C. F. Olson. Efficient pose clustering using a randomized algorithm. *Int. J. Comput. Vision*, 23(2):131–147, 1997.

[13] A. Pinz, M. Prantl, and H. Ganster. A robust affine matching algorithm using an exponentially decreasing distance function. *Journal of Universal Computer Science*, 1(8):614–631, 1995.

[14] G. Stockman. Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing*, 40:361–387, 1987.

[15] R. C. Veltkamp. Multimedia retrieval algorithmics. In *SOFSEM2007: Theory and Practice of Computer Science, LNCS 4362*, pages 138–154, 2007.

[16] H. J. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *IEEE Computational Science and Engineering*, 04(4):10–21, 1997.

Layout Indexing of Trademark Images

Reinier H. van Leuken
Computer Science
Universiteit Utrecht
reinier@cs.uu.nl

M. Fatih Demirci
Computer Science
Universiteit Utrecht
mdemirci@cs.uu.nl

Victoria J. Hodge
Dept. of Computer Science
University of York
vicky@cs.york.ac.uk

Jim Austin
Dept. of Computer Science
University of York
austin@cs.york.ac.uk

Remco C. Veltkamp
Computer Science
Universiteit Utrecht
remco.veltkamp@cs.uu.nl

ABSTRACT

Ensuring the uniqueness of trademark images and protecting their identities are the most important objectives for the trademark registration process. To prevent trademark infringement, each new trademark must be compared to a database of existing trademarks. Given a newly designed trademark image, trademark retrieval systems are not only concerned with finding images with similar shapes but also locating images with similar layouts. Performing a linear-search, i.e., computing the similarity between the query and each database entry and selecting the closest one, is inefficient for large database systems. An effective and efficient indexing mechanism is, therefore, essential to select a small collection of candidates. This paper proposes a framework in which a graph-based indexing schema will be applied to facilitate efficient trademark retrieval based on spatial relations between image components, regardless of mutual shape similarity.

Our framework starts by segmenting trademark images into distinct shapes using a shape identification algorithm. Identified shapes are then encoded automatically into an attributed graph whose vertices represent shapes and whose edges show spatial relations (both directional and topological) between the shapes. Using a graph-based indexing schema, the topological structure of the graph as well as that of its subgraphs are represented as vectors in which the components correspond to the sorted Laplacian eigenvalues of the graph or subgraphs. Having established the signatures, the indexing amounts to a nearest neighbour search in a model database. For a query graph and a large graph data set, the indexing problem is reformulated as that of fast selection of candidate graphs whose signatures are close to the query signature in the vector space. An extensive set of recognition trials, including a comparison with manually constructed graphs, show the efficacy of both the automatic graph construction process and the indexing schema.

CIVR '07 Amsterdam, The Netherlands

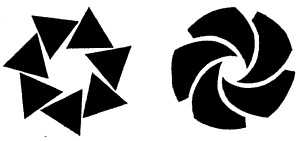


Figure 1: Two trademarks resemble each other based on the layout of their shapes despite the dissimilarity between their individual component shapes.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Content-based Image Retrieval, Trademark Retrieval, Indexing, Laplacian spectrum

1. INTRODUCTION

One of the highly active research areas within the broad field of shape matching and Content-based Image Retrieval (CBIR) is trademark retrieval. Trademarks (or, logos) [1] come in different forms, with varying kinds of unique properties. Textual information, shape, layout, and in some cases colour are probably the most important ones. Ensuring the uniqueness of trademarks and protecting their identities are the most important objectives for the trademark registration process. To prevent trademark infringement, each new trademark must be compared to the database of existing trademarks. Traditionally, this process is done by assigning keywords to shapes using predetermined vocabulary such as the Vienna classification and searching trademarks based on the keywords [21]. Since these kinds of methods involve heavy human interference, automatic trademark retrieval is of great importance.

Given a query image, most automatic trademark retrieval systems aim to find images with similar shapes without taking into account the spatial layout of the shapes. Although retrieving images containing similar shapes may seem as the

[1]defined by the UK patent office as *a sign which can distinguish the goods and services of one trader from those of another, and be represented graphically*

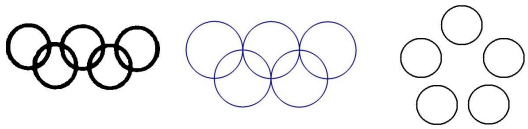


Figure 2: Three different configurations of 5 circles. Suppose the leftmost image, the Olympic logo, is used as a query. Because of a similarity in layout,the middle image should receive a higher vote than the rightmost image, despite the fact that pure shape similarity on components is the same.

primary goal, there are many cases where the layout similarity plays a more important role for ensuring uniqueness. An example of this scenario is given in Figure 1 in which the layout of the shapes reveals a strong figure in itself. The two trademarks resemble each other despite the dissimilarity between their individual shapes. In case these two trademarks are to be registered in the same or in a closely related product group or service category, a conflict of uniqueness arises.

Layout similarity between trademarks is also used to improve the quality of matching based on shape similarities. Consider Figure 2, where two candidates are returned with the same similarity scores against a given query. Although they both contain the same shapes, the middle candidate should be assigned a stronger similarity value since its shapes are in a configuration similar to that of the query. Hence, one may observe that applying layout similarity improves the overall quality of a trademark retrieval system.

The work presented in this paper proposes a framework in which a graph-based indexing schema will be applied to facilitate efficient trademark retrieval based on spatial relations between image shapes regardless of their mutual shape similarities. Our framework begins by segmenting trademark images into distinct shapes using a closed shape identification algorithm. A simple edge detector would not be sufficient as many images in our test set are noisy and this noise causes small gaps in the shape boundaries so these gaps need to be closed. In practice any closed shape identifier could be used here, such as region growing [29] or watershed [3].

We chose to refine and adapt Saund's closed shape identification algorithm [25] within the PROFI project [15]. Saund's approach was developed for the sketch retrieval domain but is equally applicable to the trademark retrieval domain. Our adapted algorithm integrates seamlessly with our other project software and provides two complementary versions. Using two complementary versions of the same technique ensures consistency which is essential in image retrieval. For this evaluation, we use a simplified version of our algorithm [15]. The simplified version aims to find just the basic shapes present in an image as the graph layout matching requires only the basic shapes compared to the more perceptual shapes perceived using Gestalt[2] principles [27, 16] discovered by our complementary version. In figure 3, we are interested in the three small triangles (to the left in figure 4) for our evaluation here but not the larger perceptual triangle formed by the three smaller triangles (to the right

[2]The Gestalt principles refer to the shape-forming capability of human vision. In particular, they refer to the visual recognition of figures and whole shapes rather than just 'seeing' a simple collection of lines and curves.

Figure 3: A sample trademark image.

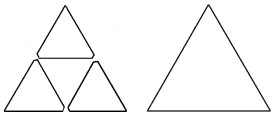


Figure 4: Possible shapes identified in figure 3.

in figure 4). The fact that the three small triangles form a larger triangle will be discerned by the layout indexing so we do not need to find it here. The two approaches may therefore be viewed as complementary. The simple closed shape approach used here to find the basic shapes which feed into a layout indexing algorithm where the layout indexing infers the perceptual relations. The more perceptual closed shape approach described in [15] finds the perceptual shapes which may be used for shape matching where similarity is determined by "shape" and which requires higher level (perceptual) shapes for matching.

Given this set of shapes identified within a trademark, its layout is then encoded automatically into an attributed graph whose vertices represent shapes and whose edges show spatial relations (both directional and topological) between the shapes. Using a new graph-based indexing schema, the topological structure of the graph as well as that of its subgraphs are represented as vectors in which the components correspond to the sorted Laplacian eigenvalues of the graph or subgraphs. Having established the signatures, the indexing now amounts to a nearest neighbour search in a model database. For a query graph and a large graph data set, the indexing problem is reformulated as that of fast selection of candidate graphs whose signatures are close to the query signature in the vector space.

The rest of the paper is organized as follows. After giving an overview of the related work in Section 2, we review the basics for our shape identification algorithm in Section 3. Section 4 presents a method to encode shape layouts in a graph. We give the details for our graph-based indexing algorithm in Section 5. Our framework is evaluated in the domain of trademark retrieval in Section 6. We close the paper with our conclusions and future work in Section 7.

2. RELATED WORK

Since our framework consists of encoding layout of a trademark in a graph and graph-based indexing, we will separately review the related work in these two concepts.

2.1 Encoding layout

The spatial relations between two objects in an image can be divided into topological and directional relations. Egenhofer [8] describes 8 basic topological relations: *disjoint, contains, inside, meet, equal, covers, covered-by* and *overlap*. Directional relations are usually represented by the four primary directions (North, South, East and West) and the four secondary directions (NW, SE, SW and SE). An alter-

native for this representation is the angle between the line connecting the two centres of mass and the horizontal line. For instance, the latter method was used by El-Kwae et al. [9] and Gudivada et al. [13].

A method for encoding layout taking into account only directional information was proposed by Chang et al. [5] and is called 2D-Strings. To produce a 2D-string representation, the centre of mass of each object in the image is projected on the x and y axes. By taking the objects from left to right and from below to above, two one-dimensional strings are obtained, in which the objects are represented by a class identifier. The shape matching problem is now transformed into string matching. Various extensions have been proposed such as the 2D G-String [4], 2D C-String [17]. These extensions deal mainly with overlapping objects with complex shapes.

Petrakis and Orphanoudakis [23] propose an indexing scheme based on 2D-Strings. For each image, all possible subsets of size 2 up to a predefined number K_{max} are created. These subsets are represented by a string taking into account both layout information and object specific information: the order (as in a 2D-String), inclusion properties, object size, roundness and orientation.

A major drawback of these *symbolic projection methods* such as 2D-strings is that in general they are not rotation invariant. Therefore, El-Kwae et al. [9] propose a robust Framework for Retrieving Images by Spatial Similarity (FRISS). It can handle translation, scaling, perfect rotation (all objects in the image are rotated around a reference point with the same angle), multiple rotation (objects are rotated around a reference point with different angles). Furthermore, it takes into account topological relations between the objects and shape-based similarities.

A popular alternative that has also been applied in this paper is the graph representation. Gudivada and Raghavan [13] propose spatial orientation graphs (SOG's), in which each vertex represents an object and the edges between them are weighted with the slope of the line connecting the two centres of mass. The distance between two graphs is calculated by finding the angle between each pair of corresponding edges. ImageMap [22] proposed by Petrakis and Faloutsos extends this idea. The images are represented by attributed relational graphs (ARG's), storing object size, orientation, and roundness in the nodes and distance, angle, and contains-relationships in the edges. This approach first computes an $n \times n$ distance matrix, where each entry corresponds to the graph-edit distance between its corresponding graph pairs. The graphs are then embedded into an f-dimensional space (target space) using Fastmap [10] such that the distances in the target space are approximately equal to those in the original graph space. The method formulates the image retrieval problem as that of range search in the target space. The embedding process in this method does not preserve the distances exactly, but the distances are distorted up to a certain degree. Although powerful, the method suffers from the limitations of the graph-edit distance approach. Specifically, if the graphs are not trees then the graph distances cannot be computed in polynomial-time using this approach. In addition, due to the fact that the graph-edit distance does not deal well with the occlusion, it is not clear how this indexing schema performs against noise and occlusions.

2.2 Graph indexing and spectral methods

Our method deploys a graph representation for encoding a trademark's layout. The problem of retrieving similar graphs to a given query may be solved by finding graphs that are isomorphic to the query or one of its subgraphs. One important indexing method solving this problem is a decision tree approach. Here, the goal is to hierarchically partition the database so that the query is first matched to the root. Depending on the result of this match, the query is then matched to either the right or the left child of the root. This process is repeated recursively until a match is found at an internal node (or leaf), or it exits with a failure indicating that no database graphs are isomorphic to the query. Messmer and Bunke [19] use this approach to organise the set of all permutations of the adjacency matrix of database graphs in a decision tree. At run time, the (sub)graph isomorphisms from the query to the database graphs are found by a decision tree traversal. A significant drawback of this method is its space requirement. All permutations of the adjacency matrix have to be encoded in decision trees, whose sizes grow exponentially with the size of the database graph. A set of pruning techniques is discussed to cut down the space complexity.

Although indexing methods with (sub)graph isomorphism detection algorithms are effective, due to noise, occlusion, or segmentation errors, no (sub)graph isomorphism may exist between the query and the database. Furthermore, only a certain degree of similarity between two graphs may be present. The indexing problem, therefore, is reformulated as efficiently retrieving database graphs whose (sub)structure is similar to the query. Although considerable research has been devoted to the problem of inexact (or error-tolerant) graph matching, rather less attention has been paid to this type of indexing based on graph structures.

An indexing framework related to the approach reported in this paper is that of Shokoufandeh et al. [26]. This framework is designed especially for tree structures in which the sum of the largest eigenvalues of the adjacency matrix for each subtree of the root form the component of its $\delta-$dimensional vector, where δ is the root degree. To account for occlusion and local deformation, these vectors are also computed for the root of each subtree. At indexing time, each non-leaf node of the query is represented as such a vector, and a nearest neighbour search is performed for each vector. Although effective, by summing up the largest eigenvalues one loses uniqueness, resulting in less representative graphs in the vector space.

3. CLOSED SHAPE IDENTIFICATION

Prior to the encoding of layout in graphs, we require a shape identification algorithm to segment the trademark into separate closed shapes. For this evaluation, we use a simplified version of our adapted algorithm [15]. The simplified version aims to find just the basic shapes present in an image as the graph layout matching requires only the basic shapes.

Our closed shape algorithm requires an underlying technique to identify the line segments within an image and to detect the relationships between those line segments. The closed shape identifier then uses this output to identify the closed shapes. Therefore, we initially find the edges in an image and subdivide these into constant curvature segments

using the Sarkar & Boyer [24] edge detection algorithm and the Wuescher & Boyer [28] curve segmentation algorithm. These methods are used as they have been successfully used in the trademark system developed by Alwis [1]. The Sarkar & Boyer method finds the edge lines in an image and splits these lines into primitives. Wuescher & Boyer aggregates these primitives into more perceptually-oriented constant curvature segments. These segments thus provide the building blocks for our closed shape identifier. From these constant curvature segments, we produce a graph of segment relations. Each constant curvature segment 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)). In our simplified implementation here, we find all segments that are end-point proximal within two pixels length. This effectively joins the graph by linking the proximal end-points. The resulting graph underpins the closed shape identification algorithm.

Our closed shape algorithm overlays this graph. Saund's approach focuses on managing the search of possible path continuations through the graph, particularly where the graph nodes represent junctions (crossroads, T-junctions etc) of lines in the original image. We use the same technique here. The closed path search commences from each end (first and last) of each node (line segment) identified by the underlying Wuescher & Boyer algorithm. 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 paths of candidate shapes. The search is managed through the use of Saund's local criteria [25] (scores) for ranking possible paths through junctions. Saund derived the scores from observations. These scores prioritise which node to expand next. As each leaf node in the tree is expanded, any 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 has been identified and its nodes and 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. However, closed paths that are subsumed by other closed paths with higher scores are discarded. Hence, each new closed path is compared to all existing stored paths. If the new path is equivalent to an existing path but has lower score then the new path is discarded. If the new path has higher score than the existing saved path then the saved path is discarded.

In our simplified approach used in this paper, we have included three changes from our usual method. We measure end-point proximity as within a 2 pixels length, normally we use Lowe's method [18] to extract endpoint proximity which is more perceptually plausible as it uses the ratio of line length to gap length [15] to decide when two lines are end-point proximal. We keep all candidate paths that are not subsumed, normally we use a minimum score threshold to identify plausible shapes as our previous approach [15] is aimed at identifying shapes perceived using Gestalt principles. Finally, we keep all paths as our set of shapes for the image, normally we use a global goodness score to assess perceptual relevance and discard perceptually irrelevant shapes (see [15]).

4. ENCODING LAYOUT IN GRAPHS

Our indexing schema can handle graphs carrying different kinds of layout information. In the experiments conducted for this paper, the vertices in the graphs correspond to the shapes in the trademark image, and the edges between them carry relations between them. Foremost, we encode the directional information (in the form of both primary and secondary directions). Rotational invariance can be achieved on demand, by neglecting for instance the difference between a south-north edge and a east-west edge. Furthermore, we are interested in detecting certain basic layout configurations that often occur in trademarks, such as triangular, circular or square configurations. If one or more of these types of layout are present in a trademark, they are encoded in the appropriate edges too.

The trademarks are segmented into individual shapes using the method described in the previous section. After this stage, their centroids are used as shape representatives to calculate the appropriate information and determine the edge labels. Each shape is connected to its n nearest neighbors, where n is a user defined parameter. The first step is to calculate the angle between the horizontal axis and a line connecting two centroids to determine the directional label for this edge. In principle there are eight possible directions (4 primary and 4 secondary), but the edges are undirected so there are four possible directions for each edge.

The next step is to detect the special pattern 'square'. This is done by performing a template match on the directional graph with a template representing a configuration of four shapes in a 2×2 square. Whenever this template is found, the edge labels are updated accordingly from the directional information to the special edge type square. Note that the square needs to be isolated to a certain extent; e.g. a grid is not a large collection of squares according to this definition. The same kind of template matching is performed for triangular and circular configurations. The decision of triangularity depends on the angles between the possibly triangular edges. Since every triplet of objects forms a triangle by definition, only the edges of a perfect triangle (or close to a perfect triangle) are labeled with the special triangle edge type. To detect circular configurations, the following circularity criterion is evaluated on the convex hull of the shape centroids: $4\pi A / \rho^2$, where A is the area and ρ is the perimeter of the convex hull. A threshold is set on the outcome of this circularity criterion to determine whether the edges on the convex hull need to be labeled with the special circular type or not.

In the experiments, the trademarks in our dataset are classified based on their layouts, not on the shapes they consist of. This classification was used to measure the retrieval performance. Since trademark retrieval and similarity are complex issues involving specific knowledge of perception and trademark logic, we presented our classification to a group of experts at Aktor Knowledge Technology who examine trademark similarity in commercial surroundings on a daily basis. It was only after their concise inspection of our dataset and classification, that we could be sure conducting our experiments and measuring performance are in correspondence with the real trademark similarities.

5. INDEXING VIA LAPLACIAN SPECTRA

Given a query graph and a large database, the objective of an indexing algorithm is to efficiently retrieve a small set of candidates, which share topological similarity with the query or one of its subgraphs. In our framework, we encode the topology of a graph through its laplacian spec-

trum. The laplacian matrix $L(G)$ of graph G is computed as $L(G) = D(G) - A(G)$, where $D(G)$ is the degree matrix and $A(G)$ is the adjacency matrix for G. The spectrum of a graph's laplacian matrix is obtained from its eigendecomposition. More formally, the eigendecomposition of a laplacian matrix is $L(G) = P\Lambda P^T$, where $\Lambda = \text{diag}\ (\lambda_1, \lambda_2, \ldots, \lambda_{|V|})$ is the diagonal matrix with the eigenvalues in increasing order and $P = (p_1|p_2|\ldots|p_{|V|})$ is the matrix with the ordered eigenvectors as columns. The laplacian spectrum is the set of eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_{|V|}\}$.

Our main motivation for encoding the topology of a graph using the lapcacian rather than the adjacency matrix as done by earlier work [26] comes from the fact that laplacian matrices are more natural, more important, and more informative about the input graphs [20]. Previously, Godsil and McKay [11] and more recently Haemers and Spence [14] have also shown that the laplacian matrix has more representational power than the adjacency matrix, i.e., it results in less number of cospectral graphs. Recall that two graphs are called cospectral (or, isospectral) if they have the same eigenvalues.

In our framework, we define the signature of a graph as the sorted eigenvalues of its laplacian matrix. To compute the similarity between two graphs, we compute the Euclidean distance between their signatures, which is inversely proportional to the structural similarity of the graphs. For a given query, retrieving its similar graphs, therefore, can be reduced to a nearest neighbor search among a set of points.

Unfortunately, this formulation cannot deal with occlusion or segmentation errors as two graphs may share similar structures up to only some level. Although adding or removing edges changes the laplacian spectrum, the spectrum of the subgraphs that survive such alteration will not be affected. Our indexing mechanism, therefore, cannot depend on the signature of the whole graph only. Instead, we will combine the signatures of the subgraphs in the framework.

Let $G = (V, E)$ be a graph and let G' be a graph obtained from G by adding a new edge e' such that $e' \notin E$. Then the following theorem, known as the interlacing theorem, relates the laplacian spectrum of both graphs [3].

THEOREM 1. *The eigenvalues of G and G' interlace:*

$$0 = \lambda_1(G) = \lambda_1(G') \leq \lambda_2(G) \leq \lambda_2(G') \leq$$

$$\ldots \leq \lambda_n(G) \leq \lambda_n(G').$$

In addition, it is known that $\sum_{i=1}^{n}(\lambda_i(G') - \lambda_i(G)) = 2$ [2]. Therefore, at least one inequality is strict. Overall this theorem implies the following. Assume that we are given a pair of isomorphic graphs g_1 and g_2. If we construct G_1 and G_2 out of g_1 and g_2 by adding different edges to each of them, one at a time, the laplacian spectra of G_1 and G_2 become proportionally less similar. As a result, the similarity between the signatures of G_1 and G_2 may not reflect the similarity between the signatures of their subgraphs g_1 and g_2. This shows that constructing an indexing mechanism based on graph signatures alone is too weak. An ideal indexing framework should, in fact, select candidate database elements based on both local and global similarities. To account for both local and global information, we will adopt

[3]This theorem is obtained by Courant-Weyl ([6], Theorem 2.1). The reader may also refer to [12].

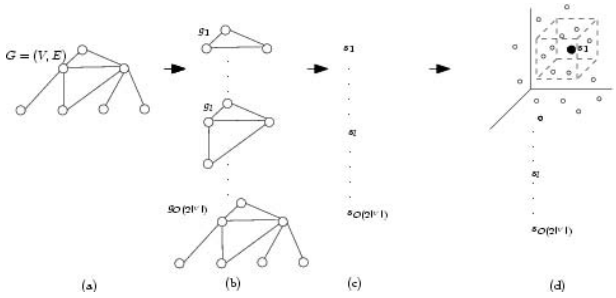


Figure 5: Retrieving similar graphs. For graphs given in Part (a), its subgraphs are constructed in Part (b). A signature is computed for each subgraph in Part (c). Given a signature, retrieving its similar graphs from a large database is formulated as a nearest neighbor search as shown in Part (d).

the following method analogous to that used in the decision tree approach [19].

For a given database graph $G = (V, E)$, rather than storing its signature in the system only, we compute the signatures of each subgraph of G in our algorithm. In this process, we gradually increase the size of the subgraphs. Since the sorted eigenvalues are invariant under consistent re-orderings of the graph's vertices, it is sufficient to compute the spectrum of permutation-similar matrices once. This property avoids the need for a high-load compilation process described for adjacency matrices in the decision tree approach.

Associated with each signature in the system is a pointer to the corresponding graph or subgraph in the database. At runtime, we first generate the signature of each subgraph of the query. Given a query signature s_q, we retrieve its nearest neighbors of the same size from the database through a nearest neighbor search (see Figure 5). Each neighbor of s_q retrieved from the database gets a vote whose value is inversely proportional to the distance from s_q. Thus, as a result, each signature of the query generates a set of votes. Moreover, we weigh the votes according to the size of the subgraphs corresponding to the signatures, i.e., the bigger the size, the more weight the vote receives.

Our encoding of a graph's structure captures its local topology, thus allowing for its use in the case of occlusion and segmentation errors. Furthermore, the signature of a graph is invariant under the reorderings of its vertices. This, in turn, allows us to compare the signatures of a large number of graphs without solving the computationally expensive correspondence problem between their vertices. In addition, based on Theorem 1, not only do isomorphic graphs share the same signature, non-isomorphic but similar graphs or subgraphs have close signatures in the vector space. The database, therefore, can be pruned without losing structurally similar graphs to the query.

6. EXPERIMENTS

In this section we evaluate our framework in the context of a trademark retrieval experiment. We use a set of 450 trademark images from the UK PTO dataset used in the Artisan project [7]. Figure 6 shows some trademark images used in the experiments. We begin by representing the layout of

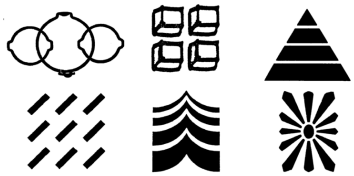


Figure 6: Some trademark images used in the experiments.

each image in the database as a graph. Given a graph, we compute the signatures for each of its subgraphs and populate the resulting signatures in the vector space. We applied the following leave-one-out procedure to the datasets to evaluate the framework in the experiments. We initially remove the first graph from the database and use it as a query for the remaining database graphs. The graph is then put back in the database and the procedure is repeated with the second graph from the database, etc., until all database graphs have been used as a query.

To check our segmentation and automatic graph construction procedures, the graph representation process was also performed manually in our experimental setup. Specifically, we manually selected shapes for each input trademark image, created a vertex for each shape, and connected two vertices by an edge if the layout of the corresponding shapes should be encoded in the graph based on the human perception. As a result, one manually and one automatically constructed graph datasets have been generated. The performance of the proposed indexing algorithm was evaluated for each dataset.

Precision and recall are two well-known performance measures to compute the quality of an indexing mechanism. In high precision, relevant items are in the top of the ranking, whereas in high recall, false negatives are avoided and the returned result contains all relevant objects. A good indexing system should, in fact, perform well according to both of these two measures. We conducted two sets of experiments to cover both scenarios. In the first experiment, our goal is to quickly determine the class of the query. In the second experiment, the objective is to return a small candidate set, which contains all the objects belonging to the query class. In both experiments, the indexing system ranks the database graphs in decreasing order of similarity from each query graph. According to the results, in 98.4% and 89.1% of the cases, the most similar database graph belongs to the correct shape class for manually constructed and automatically constructed datasets, respectively (nearest-neighbor rates). In addition, the worst position of the closest matching graph is 5 for manual graphs, while this number is 9 for automatic graphs. These numbers show that 98% of the datasets can be pruned by the indexing mechanism to determine the correct layout class for a query. In the second experiment, the system's performance was evaluated by computing the total number of retrieved images that is necessary to retrieve the entire query class (maximum minimal scope). Our results show that the first 71 of the candidate return set always contains all the graphs belonging to the query class for manual graphs; this number is 80 for automatic graphs. This indicates that for this task our framework prunes more than 84% and 82% of the manual and automatic graph datasets, respectively. In other words, the recall in each dataset is 100% if the scope is set to the first 16% and 18% of the sorted candidate models for manual and automatic graphs respectively. We also computed how many of the models in the query's class appear within the top $K-1$ matches, where K is the size of the query class (first tier). This number was 91.2% for manual graphs and 86.3% for automatic graphs. Repeating the same experiment but considering the top $2 \times K - 1$ matches (second tier) covers 98.1% and 91.7% members of the layout classes for manual and automatic graph datasets, respectively.

In Table 1, we have presented the matching results for a small subset of trademark images whose graphs were generated automatically using our approach. The first column of each row represents the query image; the remaining elements of each row show the top 10 closest database trademarks retrieved by our indexing algorithm. Squares are drawn around the wrong matches. In all but once case (row 5) the closest trademark image belongs to the same layout class as the query. Although the closest match for the query in row 5 was classified as a mismatch, one may notice that the query consists of three sets of small squares on top of each other and each set has the same layout as the mismatch. As another example, consider the query in row 8 of the left-right class and its first mismatch of the triangle class. Notice that three small triangles in the mismatch have the same layout as the query. Overall, rather than focusing on the mismatches that occur because of the result of a partial match, we observed that the wrong selections happen mainly due to the poor segmentation of the trademark. If different shapes in a trademark are connected, for instance, our segmentation algorithm detects them as one shape. Thus, the layout within these shapes are not encoded in the graphs. We will extend our segmentation technique to region-based and will use it within our framework in the future.

7. CONCLUDING REMARKS

In this paper we have presented a framework for retrieving trademark images based on spatial layout of the shapes. Besides pure shape similarity between trademark images, similarities in configuration of the shapes may also give rise to a conflict of uniqueness. The process of content based trademark retrieval, therefore, can be significantly improved by taking into account these layout features, enabling a stronger prevention of trademark infringement.

In our framework, trademark images are first segmented into closed, distinct shapes. This segmentation is line based; after an initial edge detection step, the shape boundaries are subdivided into constant curvature segments. These segments are then aggregated to more perceptually relevant primitives, which form the input blocks for the closed shape identifier. By searching for closed paths in the primitives, the shapes are returned and passed on to the next layer of our framework, the construction of a layout graph.

The centroids of the shapes are taken as shape representatives while constructing a graph that reflects the layout of the trademark. Each shape is represented by a vertex, and connected to a predefined number of nearest neighbors, and layout information is stored in the edges that connect these vertices. After the graph construction, the laplacian matrix is taken (by subtracting the adjacency matrix from the degree matrix) and its spectrum is computed. Every trademark is then stored in a database, by populating a vector space with the laplacian spectra. The laplacian spectrum

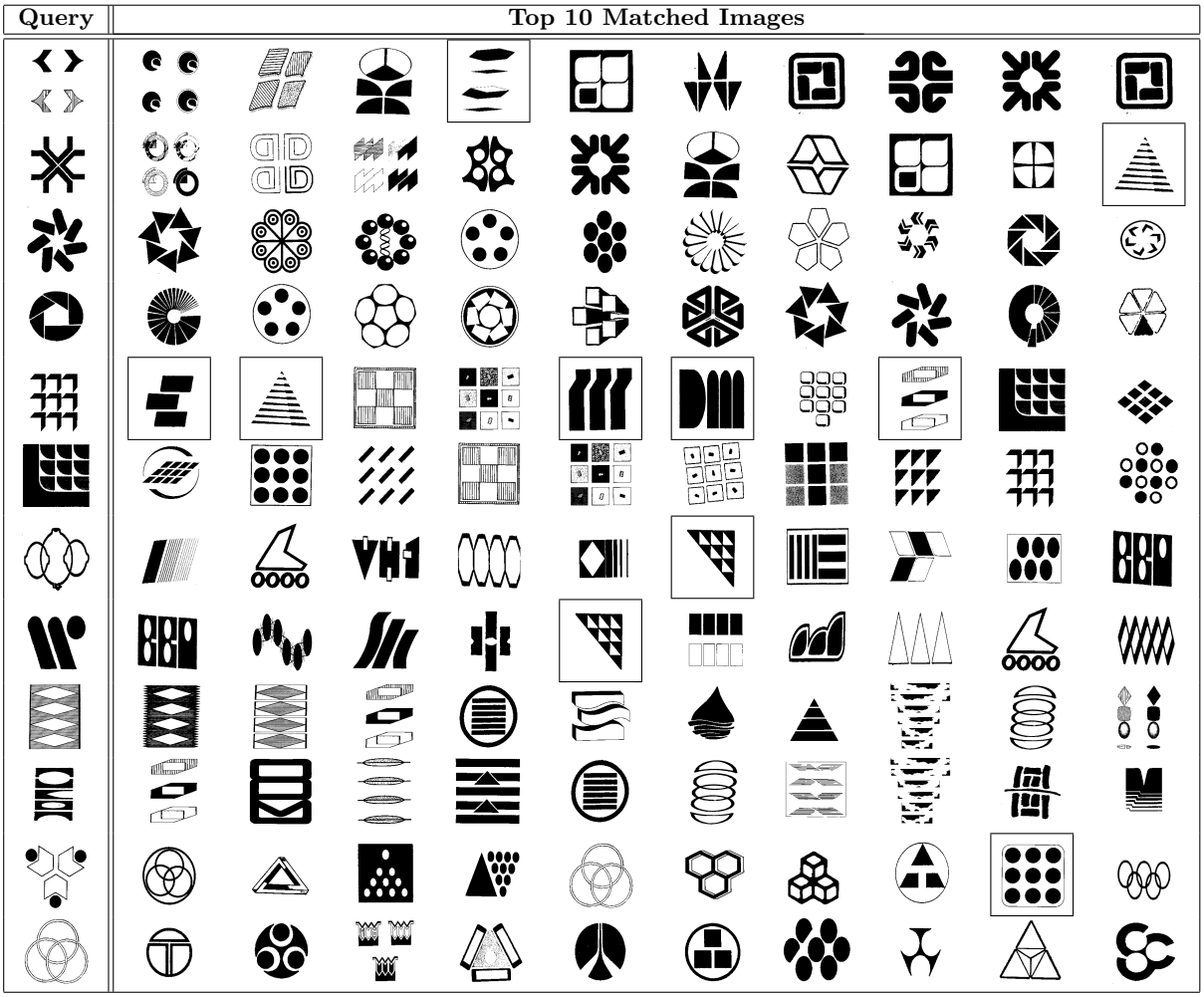


Table 1: Top matched models are sorted by the similarity to the query.

reflects important properties of the graph and its topology. Besides computing the laplacian spectrum of the complete graph, we also store this feature vector for every possible subgraph to perform partial matching.

We evaluated our framework on a test collection of 450 real trademark images and the results are promising. First and second tier results, averaged over all possible queries, were 86.3% and 81.7% respectively. It is one of our future works to extend the test collection and perform a comparison with other, known layout indexing techniques. Furthermore, we want to take into account topological information as well, besides the directional information and special configurations that are encoded now. Finally, we will use a region-based segmentation algorithm within our framework to reduce the number of mismatches that occured because of the current line-based segmentation procedure.

Acknowledgements
The authors would like to thank Aktor Knowledge Technology for their help with establishing the ground truth. This research was supported by the FP6 IST project 511572-2 PROFI.

8. REFERENCES

[1] S. Alwis. *Content-Based Retrieval of Trademark Images*. PhD thesis, Dept. of Computer Science, University of York, UK, 1999.

[2] W. N. Anderson and T. D. Morley. Eigenvalues of the laplacian of a graph. *Linear and Multilinear Algebra*, 18:141–145, 1985.

[3] S. Beucher. Watersheds of functions and picture segmentation, acoustics, speech, and signal processing. In *IEEE International Conference on ICASSP'82*, pages 1928–1931, 1982.

[4] S. Chang, E. Jungert, and Y. Li. Representation and retrieval of symbolic pictures using generalized 2D strings. In *SPIE Conference on Visual Communications and Image Processing*, volume 3, pages 1360–1372, November 1989.

[5] S. K. Chang, Q. Y. Shi, and C. W. Yan. Iconic indexing by 2-d strings. *IEEE Transansctions on Pattern Analysis and Machiche Intelligence*, 9(3):413–428, 1987.

[6] D. Cvetković, M. Doob, and H. Sachs. *Spectra of Graphs: Theory and Application*. VEB Deutscher Verlag der Wissenschaften, Berlin, 2nd edition, 1982.

[7] J. P. Eakins, K. Shields, and J. M. Boardman. Artisan: A shape retrieval system based on boundary family indexing. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 17–28, 1996.

[8] M. Egenhofer and R. Franzosa. Point Set Topological Relations. *International Journal of Geographical Information Systems*, 5(2):161–174, 1991.

[9] E. El-Kwae and M. Kabuka. A Robust Framework for Content-Based Retrieval by Spatial Similarity in Image Databases. *ACM Transactions on Information Systems*, 17(2):174–198, April 1999.

[10] C. Faloutsos and K. Lin. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In M. J. Carey and D. A. Schneider, editors, *Prooceedings of ACM SIGMOD '95*, pages 163–174, San Jose, California, 22–25 1995.

[11] C. Godsil and B. McKay. Constructing cospectral graphs. In *Aequationes Mathematicae*, pages 257– 268, 1982.

[12] R. Grone, R. Merris, and V. S. Sunder. The laplacian spectrum of a graph. *SIAM Journal on Matrix Analysis and Applications*, 11:218–238, 1990.

[13] V. N. Gudivada and V. V. Raghavan. Design and Evaluation of Algorithms for Image Retrieval by Spatial Similarity. *ACM Transactions on Information Systems*, 13(2):115–144, April 1995.

[14] W. H. Haemers and E. Spence. Enumeration of cospectral graphs. *Eur. J. Comb.*, 25(2):199–211, 2004.

[15] V. Hodge, J. Eakins, and J. Austin. Inducing a perceptual relevance shape classifier. In *ACM International Conference on Image and Video Retrieval, (CIVR07). July 9-11 2007*, 2007.

[16] K. Koffka. *Principles of Gestalt Psychology*. Harcourt Brace. New York, 1963.

[17] S. Lee and F. Hsu. 2d c-string: a new spatial knowledge representation for image database systems. *Pattern Recognition*, 23(10):1077–1087, 1990.

[18] D. Lowe. Three dimensional object recognition from simple two dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

[19] B. Messmer and H. Bunke. A decision tree approach to graph and subgraph isomorphism detection. *Pattern Recognition*, 32(12):1979–1998, 1999.

[20] B. Mohar. The laplacian spectrum of graphs. In *Sixth International Conference on the Theory and Applications of Graphs*, pages 871–898, 1988.

[21] W. I. P. Organisation. *CD-NIVILO ISBN 92-805-1280-7*. WIPO, 2003.

[22] E. Petrakis, C. Faloutsos, and K.-I. Lin. Imagemap: an image indexing method based on spatial similarity. In *IEEE Transactions on Knowledge and Data Engineering*, volume 14, pages 979– 987, 2002.

[23] E. Petrakis and S. Orphanoudakis. A Methology for the Representation, Indexing, and Retrieval of Images by Content. *Image and Vision Computing*, 8(11):504–512, October 1993.

[24] S. Sarkar and K. Boyer. On optimal infinite impulse response edge detection filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(11):1154–1171, 1991.

[25] E. Saund. Finding perceptually closed paths in sketches and drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(4):475–491, 2003.

[26] A. Shokoufandeh, D. Macrini, S. Dickinson, K. Siddiqi, and S. Zucker. Indexing hierarchical structures using graph spectra. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(7), 2005.

[27] M. Wertheimer. Laws of organization in perceptual forms (1923)., 1938.

[28] D. Wuescher and K. Boyer. Robust contour decomposition using a constant curvature criterion. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(1):41–51, 1991.

[29] S. Zucker. Region growing: Childhood and adolescence. *Computer Graphics & Image Processing*, 5:382–399, 1976.

Searching for Logo and Trademark Images on the Web

Euripides G.M. Petrakis† Epimenides Voutsakis† Evangelos E. Milios‡
petrakis@intelligence.tuc.gr pimenas@softnet.tuc.gr eem@cs.dal.ca

† Dept. of Electronic and Comp. Engineering, Technical University of Crete (TUC), Chania, Greece
‡ Faculty of Comp. Science, Dalhousie University, Halifax, Nova Scotia, Canada

ABSTRACT

This work shows that it is possible to exploit text and image content characteristics of logo and trademark images in Web pages for enhancing the performance of retrievals on the Web. Searching for important (authoritative) Web pages and images is a desirable feature of many Web search engines and is also taken into account. State-of-the-art methods for assigning higher ranking to important Web pages, over other Web pages satisfying the query selection criteria, are considered and evaluated. PicASHOW exploits this idea in retrieval of important images on the Web using link information alone. WPicASHOW (Weighted PicASHOW), is a weighted scheme for co-citation analysis incorporating within the link analysis method of PicASHOW the text and image content of the queries and of the Web pages. The experimental results demonstrate that Web search methods utilizing content information (or combination of content and link information) perform significantly better than methods using link information alone.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Abstracting methods, Indexing methods*; I.5.4 [**Pattern Recognition**]: Applications—*Computer Vision*

General Terms

Performance, Experimentation, Algorithms

Keywords

image retrieval, logo, trademark, feature extraction, authority, link analysis

1. INTRODUCTION

The World Wide Web is host to millions of images on every conceivable topic. The images are used to enhance the information content of Web pages, capture the attention of users or to reduce the textual content of Web sites. In scientific, artistic, technical, or corporate Web sites, images comprise the majority of digital content and are characteristic of the content and type of these Web sites.

Searching for effective methods to retrieve images from the Web has been in the center of many scientific efforts during the last few years [9]. The relevant technology evolved rapidly also thanks to prior advances in Web systems technology [1], information retrieval [16] and image database research [20, 17]. Several approaches to the problem of content-based image retrieval on the Web have been proposed and some have been implemented on research prototypes (e.g., PicToSeek [6], ImageRover [24],WebSEEK [21], Diogenis [2]) and commercial systems. The last category of systems, includes general purpose image search engines such as Google Image Search [1], Yahoo [2], Altavista [3], Ditto [4] etc.) as well as systems providing specific services to users such as unauthorized use of images (e.g., CreativePro [5]), Web and e-mail content filters, systems for image authentication (e.g., Dicontas [6]), licensing and advertising (e.g., Corbis [7]).

This work deals with the problem of retrieval of logo and trademark images on the Web. Logos and trademarks in particular are important characteristic signs of corporate Web sites or of products presented there. A recent analysis of Web content [7] reports that logos and trademarks comprise 32,6% of the total number of images on the Web. Therefore, retrieval of logo and trademarks is of significant commercial interest (e.g., Patent Offices provide services on unauthorized uses of logos and trademarks).

The contribution of this work is not only in using existing technology for solving the retrieval problem but also, in showing how to exploit the content characteristics of logo and trademarks for enhancing the performance of retrievals on the Web. Retrieval by image content, in particular, requires integration of text and image based approaches for analyzing the content of Web pages.

Logo and trademark images are easier (than natural images) to describe by low level features (intensity, frequency histograms and features computed on the above types of histograms). Because images on the Web are not properly categorized, filters based on machine learning by decision trees for distinguishing logo and trademark images from images

CIVR'07, July 9–11, 2007, Amsterdam, The Netherlands.

[1] http://www.google.com/imghp
[2] http://images.search.yahoo.com
[3] http://www.altavista.com/image
[4] http://www.ditto.com
[5] http://www.creativepro.com
[6] http://www.dicontas.co.uk
[7] http://www.corbis.com

of other categories (e.g., graphics, photographs, diagrams, landscapes) are designed and implemented. The decision tree demonstrated classification accuracy as high as 85%.

Once logo and trademark images are detected, effective content-based image retrieval on the Web often requires that important (authoritative) images satisfying the query selection criteria are assigned higher ranking over other relevant images. This is achieved by exploiting the results of link analysis for re-ranking the results of retrieval. Classical link analysis methods such as *HITS* [10] and *PageRank* [13] estimate the quality of Web pages and the topic relevance between the Web pages and the query. These methods estimate the importance of Web pages as a whole. PicASHOW [11], in particular, shows how to estimate the importance of images contained within Web pages. However, PicASHOW does not show how to handle image content and queries by image example. This is solved by WPicASHOW [26] (Weighted PicASHOW) a weighted scheme for co-citation analysis that incorporates, within the link analysis method of PicASHOW, the text and image content of the queries and of the Web pages.

Existing approaches for handling logos and trademarks [8, 12] focus entirely on image content analysis and high precision answers to queries by image example on stand-alone data sets. They don't focus on detection (i.e., discrimination between trademark and not trademark images) nor do they show how to retrieve high quality answers from the Web.

The methods referred to above are implemented and evaluated in *IntelliSearch* [8] [25], a complete and fully automated information retrieval system for the Web. It supports fast and accurate responses to queries addressing text and images in Web pages by incorporating state-of-the-art image indexing and retrieval methods by text (e.g., the Vector Space Model) in conjunction with efficient ranking of Web pages and images by importance (authority) such as WPicASHOW. *IntelliSearch* stores a crawl of the Web with more than 1,5 million Web pages with images. It offers an ideal test-bed for experimentation and training and serves as a framework for a realistic evaluation of many Web image retrieval methods. The experimental results demonstrate that giving higher ranking to important images seems to reduce the accuracy of retrievals (the important images are not always the most relevant ones).

The rest of this paper is organized as follows: Extraction of meaningful image descriptions from Web pages and image similarity measures based on the matching of image descriptions are discussed in Section 2. PicASHOW and WPicASHOW, the image authority searching methods considered in this work are presented in Section 3. *IntelliSearch*, a content-based retrieval of logo and trademark images that integrates the above ideas is presented in Section 4. Experimental results are presented and discussed in Section 5 followed by conclusions in Section 6.

2. IMAGE CONTENT REPRESENTATION

Logo and trademark images are easier to describe by low level features (e.g., color histograms, text features). The focus of this work is not on novel image feature extraction but on showing how to search for logo and trademarks on the Web for a given and well established set of features (such as those used in [8, 12]). Logo and trademark images are easier to describe by low level features (e.g., color histograms, text features).

[8]http://www.intelligence.tuc.gr/intellisearch

2.1 Text Description

Typically, images are described by the text surrounding them in the Web pages [19]. The following types of image descriptive text are derived based on the analysis of `html` formatting instructions:

Image Filename: The `URL` entry (with leading directory names removed) in the `src` field of the `img` formatting instruction.

Alternate Text: The text entry of the `alt` field in the `img` formatting instruction. This text is displayed on the browser (in place of the image), if the image fails to load. This attribute is optional (i.e., is not always present).

Page Title: The title of the Web page in which the image is displayed. It is contained between the `TITLE` formatting instructions in the beginning of the document. It is optional.

Image Caption: A sentence that describes the image. It usually follows or precedes the image when it is displayed on the browser. Because it does not correspond to any `html` formatting instruction it is derived either as the text within the same table cell as the image (i.e., between `td` formatting instructions) or within the same paragraph as the image (i.e., between `p` formatting instructions). If neither case applies, the caption is considered to be empty. In either case, the caption is limited to 30 words before or after the reference to the image file.

The following are two examples of `html` code, both with a reference to image *"logo.gif"*.

- *...</td> <td> Our company's logo <img src="logo.gif" alt="software logo">
 ...</td> <td> is registered since 1990 ...</td> <td>.*
- *... Our company's logo logo is registered since 1990 .*

All descriptions are lexically analyzed and reduced into term (noun) vectors. First, all terms are reduced into their morphological roots, using the Porter [15] suffix stripping (stemming) algorithm. Similarly, text queries are also transformed to term vectors and matched against image term vectors according to the vector space model. More specifically, the similarity between the query Q and the image I is computed as a weighted sum of similarities between their corresponding term vectors

$$\begin{aligned} &S_{text}(Q,T) = \\ &S_{file\_name}(Q,I) \quad + \quad S_{alternate\_text}(Q,I) \quad + \\ &S_{page\_title}(Q,I) \quad + \quad S_{image\_caption}(Q,I). \end{aligned} \tag{1}$$

Each S term is computed as a weighted sum of $tf \cdot idf$ terms without normalizing by query term frequencies (it is not required for short queries). All measures above are normalized on [0,1].

2.2 Image Content Descriptions

Image content is described in terms of features computed from raw images. All images are converted to grey scale. For logo and trademark images the following features are computed:

Intensity Histogram: Shows the distribution of intensities over the whole range of intensity values ([0..255] in this work).

Energy Spectrum [22]: Describes the image by its frequency content. It is computed as a histogram showing the distribution of average energy over 256 co-centric rings (with the largest ring fitting the largest inscribed circle of the DFT spectrum).

Moment Invariants [23]: Describes the image by its spatial arrangement of intensities. It is a vector of 7 moment coefficients.

The above representations are used to solve the following two problems:

Logo-Trademark Detection: A five-dimensional vector is formed from each image: Each image is specified by the mean and variance of its Intensity and Energy spectrums plus a count of the number of distinct intensities per image. A set of 1,000 image examples is formed consisting of 500 logo-trademark images and 500 images of other types. Images of other types can belong to more than one class: non-logo graphics, photographs, diagrams etc. Their feature vectors are fed into a decision-tree [27] which is trained to detect logo and trademark images. The estimated classification accuracy by the algorithm is 85%. For each image the decision computes an estimate of its likelihood of being logo or trademark or "Logo-Trademark Probability".

Logo-Trademark Similarity: The similarity between two images Q, I (e.g., query and a Web image) is computed as

$$S_{image}(Q,I) = S_{intensity\_spectrum}(Q,I) + S_{energy\_spectrum}(Q,I) + S_{moment\_invariants}(Q,I). \quad (2)$$

The similarity between histograms is computed by their intersection [5] whereas the similarity between their moment invariant is computed as *1 - Euclidean_vector_distance*.

All measures above are normalized to lie in the interval [0, 1]. To answer queries consisting of both text and example image, the similarity between a query Q and an image I is computed as

$$w = \lambda S_{image}(Q,I) + (1-\lambda) S_{text}(Q,I), \quad (3)$$

where λ denotes the relative significance of image and text descriptions. In this work $\lambda = 0.5$. More appropriate weights may be specified by machine learning.

3. IMAGE LINK ANALYSIS METHODS

Co-citation analysis is proposed as a tool for assigning importance to pages or for estimating the similarity between a query and a Web page. The main idea behind this approach is that a link from page a to page b may be regarded as a reference from the author of a to b.

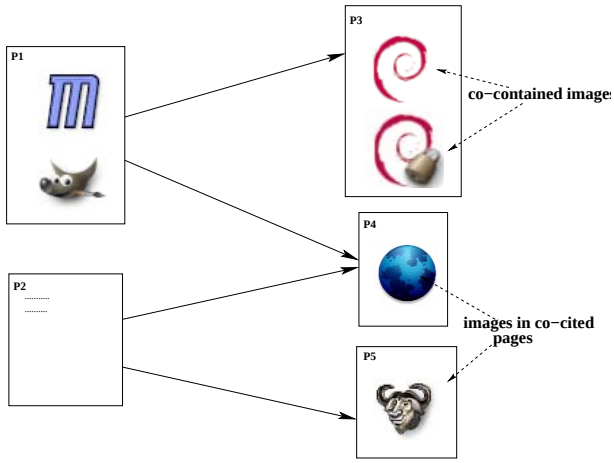


Figure 1: An example of a focused graph with co-contained and co-cited images.

The main idea behind co-citation analysis is that the number and quality of references to a page provide an estimate of the quality of the page and also a suggestion of relevance of its contents with the contents of the pages pointing to it. *HITS* [10] exploits this information to estimate the relevance between a query and a Web page and ranking of this page among other relevant pages. Building upon the same idea, PicASHOW [11] demonstrates how to retrieve high quality Web images on the topic of a keyword-based query. It does not show how to process queries by example image. This is exactly the focus of this work.

PicASHOW relies on the idea that images co-contained or co-cited by Web pages are likely to be related to the same topic. Figure 1 illustrates examples of co-contained and co-cited images. *PicAHOW* computes authority and hub values by link analysis on the *query focused graph* $\mathcal{F}$ (i.e., a set of pages formed by initial query results expanded by backward and forward links). PicASHOW filters out from $\mathcal{F}$ non-informative images such as banners, logo, trademarks and "stop images" (bars, buttons, mail-boxes etc.) from the query focused graph utilizing simple heuristics such as small file size.

PicASHOW introduces the following adjacency matrices defined on the set of pages in the query focused graph:

$\mathcal{W}$: The page to page adjacency matrix (as in HITS) relating each page in $\mathcal{F}$ with the pages it points to. The rows and the columns in $\mathcal{W}$ are indices to pages in $\mathcal{F}$. Then, $w_{ij} = 1$ if page i points to page j; 0 otherwise.

$\mathcal{M}$: The page to image adjacency matrix relating each page in $\mathcal{F}$ with the images it contains. The rows and the columns in $\mathcal{M}$ are indices to pages and images in $\mathcal{F}$ respectively. Then, $m_{ij} = 1$ if page i points to (or contains) image j.

$(\mathcal{W}+\mathcal{I})\mathcal{M}$: The page to image adjacency matrix ($\mathcal{I}$ is the identity matrix) relating each page in $\mathcal{F}$ both, with the images it contains and with the images contained in pages it points to.

Similarly to HITS, PicASHOW defines the so called image *co-citation* $[(\mathcal{W}+\mathcal{I})\mathcal{M}]^T \cdot (\mathcal{M}+\mathcal{I})\mathcal{W}$ and *bibliographic* $(\mathcal{W}+\mathcal{I})\mathcal{M}) \cdot [(\mathcal{W}+\mathcal{I})\mathcal{M}]^T$ matrices respectively. The ij-th entry of the image co-citation matrix is the number of

	P_1	P_2	P_3	P_4	P_5
P_1	0	0	1	1	0
P_2	0	0	0	1	1
P_3	0	0	0	0	0
P_4	0	0	0	0	0
P_5	0	0	0	0	0

						
P_1	0	0	1	1	0	0
P_2	0	0	0	0	0	0
P_3	1	1	0	0	0	0
P_4	0	0	0	0	1	0
P_5	0	0	0	0	0	1

						
P_1	1	1	1	1	1	0
P_2	0	0	0	0	1	1
P_3	1	1	0	0	0	0
P_4	0	0	0	0	1	0
P_5	0	0	0	0	0	1

Figure 2: Adjacency matrices $\mathcal{W}$, $\mathcal{M}$ and $(\mathcal{W}+\mathcal{I})\mathcal{M}$ for the focused graph of Figure 1.

Image						
Authorities	0.492	0.492	0.339	0.339	0.519	0.117

Page	P_1	P_2	P_3	P_4	P_5
Hubs	0.519	0.0001	0.854	0.001	0

Figure 3: Image Authority (top) and Hub values (bottom) computed by WPicASHOW in response to query "Debian trademark".

pages that jointly point to images with indices i and j. The ij-th entry of the image bibliographic matric is the number of images jointly referred to by pages i and j. PicASHOW computes the answers to a query by ranking the elements of the principal eigenvector of the image co-citation matrix by their authority values.

Figure 2 illustrates these matrices for the pages $(P_1, P_2, \ldots P_5)$ and images of Figure 1. Notice that, in PicASHOW all non-zero values in $\mathcal{M}$, $\mathcal{W}$ and $(\mathcal{W}+\mathcal{I})\mathcal{M}$ matrices are 1 (non normalized weights). Figure 3 illustrates authority and hub values computed by PicASHOW in response to query "Debian logo". Notice the high authority scores of pages showing logo or trademark images of "Debian Linux". Notice that Mozila trademark has higher authority value than Debian trademark.

Hub and Authority values of images are computed as the principal eigenvectors of the image-co-citation $[(\mathcal{W}+\mathcal{I})\mathcal{M}]^T \cdot (\mathcal{W}+\mathcal{I})\mathcal{M}$ and bibliographic matrices $(\mathcal{W}+\mathcal{I})\mathcal{M}) \cdot [(\mathcal{W}+\mathcal{I})\mathcal{M}]^T$ respectively. The higher the authority value of an image the higher its likelihood of being relevant to the query.

PicASHOW can answer queries on a given topic but, similarly to $HITS$, it suffers from the following problems [4]:

Mutual reinforcement between hosts: Encountered when a single page on a host points to multiple pages on another host or the reverse (when multiple pages on a host point to a single page on another host).

Topic drift: Encountered when the query focused graph contains pages not relevant to the query. Then, the highest authority and hub pages tend not to be related to the topic of the query.

PicASHOW does not handle mutual reinforcement between nodes (except that it constraints the number of references per image to one by identifying replicated images) and topic drift nor does it handle queries by example. WPicASHOW handles all these issues:

Mutual reinforcement is handled by normalizing the weights of nodes pointing to k by $1/k$. Similarly, the weights of all l pages pointing to the same page are normalized by $1/l$. An additional improvement is to purge all intra-domain links except links from pages to their contained images.

Topic Drift is handled by regulating the influence of nodes by setting weights on links between pages. The links of the page-to-page relation $\mathcal{W}$ are assigned a relevance value computed according to the vector space model as the similarity between the term vector of the query and the term vector of the anchor text on the link between the two pages. The weights of the page-to-image relation matrix $\mathcal{M}$ are computed depending on query type: For text (e.g., keyword) queries the weights are computed according to Equation 1 (as the similarity between the query and the descriptive text of an image). For queries combining text and image example, the weights are computed according to Equation 3 (as the average of similarities between the text and image contents of the query and the image respectively).

Queries may be formulated either by keywords (or phrases) or by a combination of keywords and image example. In both cases of image queries, WPicASHOW starts by formulating the query focused graph as follows:

- An initial set S of images is retrieved. These are images contained or pointed to by pages matching the query keywords according to Equation 1.
- Stop images (banners, buttons, etc.) and images with logo-trademark probability less than 0.5 are ignored. At most T images are retained and this limits the size of the query focused graph ($T = 10,000$ in this work).
- The set S is expanded to include pages pointing to images in S.
- The set S is further expanded to include pages and images that point to pages or images already in S. To limit the influence of very popular sites, for each page in S, at most t ($t = 100$ in this work) new pages are included.
- The last two steps are repeated until S contains T pages and images.

WPicASHOW then builds $\mathcal{M}$, $\mathcal{W}$ and $(\mathcal{W}+\mathcal{I})\mathcal{M}$ matrices for information in S. Figure 4 illustrates these matrices

	P_1	P_2	P_3	P_4	P_5
P_1	0	0	.6	.1	0
P_2	0	0	0	.1	.1
P_3	0	0	0	0	0
P_4	0	0	0	0	0
P_5	0	0	0	0	0

						
P_1	0	0	.1	.1	0	0
P_2	0	0	0	0	0	0
P_3	.8	.7	0	0	0	0
P_4	0	0	0	0	.2	0
P_5	0	0	0	0	0	.15

						
P_1	.48	.42	.1	.1	.02	0
P_2	0	0	0	0	.02	.015
P_3	.8	.7	0	0	0	0
P_4	0	0	0	0	.2	0
P_5	0	0	0	0	0	.15

Figure 4: Adjacency matrices $\mathcal{M}$, $\mathcal{W}$ and $(\mathcal{M}+\mathcal{I})\mathcal{W}$ for the focused graph of Figure 1 corresponding to query "Debian logo".

Image						
Authorities	.751	.657	.0418	.0418	.008	0

Page	P_1	P_2	P_3	P_4	P_5
Hubs	.519	.0001	.854	.001	0

Figure 5: Image Authority (top) and Hub values (bottom) computed by WPicASHOW in response to query "Debian logo".

for the same pages and images of Figure 1 with weights corresponding to query "Debian logo".

Figure 3 illustrates authority and hub values computed by WPicASHOW in response to query "Debian logo". Notice the trademark images of "Debian Linux" are assigned the highest authority values followed by the images of Mozilla Firefox.

4. INTELLISEARCH

IntelliSearch [25] is implemented in Java under Linux. Figure 6 illustrates the architecture of the proposed system. *IntelliSearch* consists of several modules, the most important of them being the following:

Crawler module: Implemented based upon Larbin [9], the crawler assembled locally a collection of 1,5 million pages with images. The crawler started its recursive visit of the Web from a set of 14,000 pages which is assembled from the answers of Google image search[10] to 20 queries on topics related to Linux and Linux

[9]http://larbin.sourceforge.net
[10]http://www.google.com/imghp

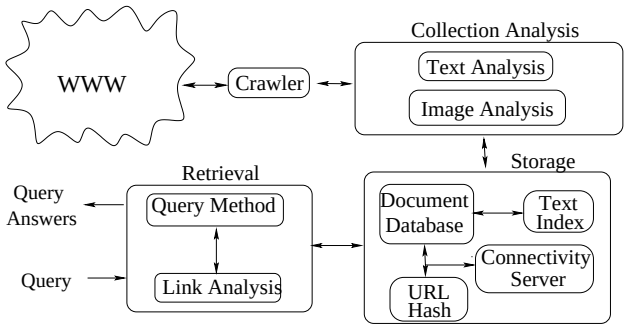


Figure 6: *IntelliSearch* Architecture.

products. The crawler worked recursively in breadth-first order and visited pages up to depth 5 links from each origin.

Collection analysis module: The content of crawled pages is analyzed. Text, images, link information (forward links) and information for pages that belong to the same site is extracted.

Storage module: Implements storage structures and indices providing fast access to Web pages and information extracted from Web pages (i.e., text, image descriptions and links). For each page, except from raw text and images, the following information is stored and indexed: Page URLs, image descriptive text (i.e., alternate text, caption, title, image file name), terms extracted from pages, term inter document frequencies (i.e., term frequencies in the whole collection), term intra document frequencies (i.e., term frequencies in image descriptive text parts), link structure information (i.e., backward and forward links). Image descriptions are also stored.

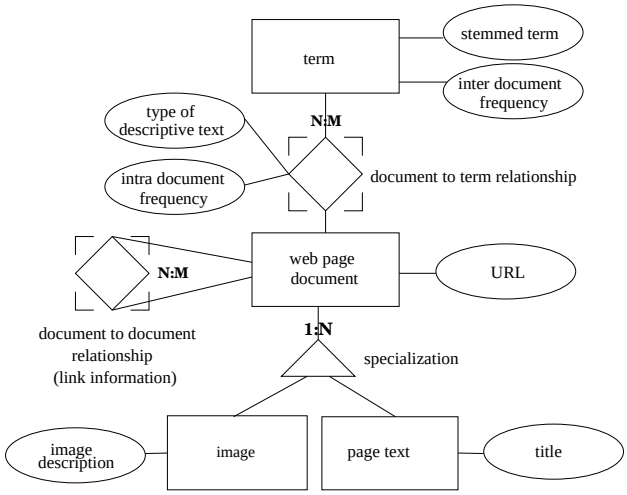


Figure 7: The Entity Relational Diagram (ERD) of the database.

The Entity Relationship Diagram (ERD) of the database in Fig. 7 describes entities (i.e., Web pages)

and relationships between entities. There are many-to-many (denoted as $N : M$) relationships between Web pages implied by the Web link structure (by forward and backward links), one-to-many (denoted as $1 : N$) relationships between Web pages and their constituent text and images and $N : M$ relationships between terms in image descriptive text parts and documents. The ERD also illustrates properties of entities and relationships (i.e., page URLs for documents, titles for page text, image content descriptions for images, stemmed terms, inter and intra document frequencies for terms in image descriptive text parts).

The database schema is implemented in BerkeleyDB [11] Java Edition. BerkeleyDB is an embedded database engine providing a simple Application Programming Interface (API) supporting efficient storage and retrieval of Java objects. The mapping of the ERD of Fig. 7 to database files (Java objects) was implemented using the Java Collections-style interface. Apache Lucene[12] is providing mechanisms (i.e., inverted files) for indexing text and link information. There are Hash tables for URLs and inverted files for terms and link information. Two inverted files implement the connectivity server [3] and provide fast access to linkage information between pages (backward and forward links) and two inverted files associate terms with their intra and inter document frequencies and allow for fast computation of term vectors.

Retrieval module: Queries are issued by keywords (or free text) or by a combination of example image and text. The user is prompted at the user interface to select mode of operation (retrieval of text pages or image retrieval). All methods in Sec. 2.1 are implemented.

5. EXPERIMENTS

Different image retrieval methods are implemented and evaluated. The competitor methods are:

PicASHOW [11]: Ranks Web images by exploiting co-citation information only. It can answer only text queries. Queries by example image (image queries) are not supported.

WPicASHOW (weighted PicASHOW) [26]: Extends PicASHOW to take into account, in addition to link information, the text and image content of both the queries and of the Web pages. It can answer both text and image queries.

Vector Space Model (VSM) [18]: Text queries are transformed to term vectors and matched against term vectors extracted from database images. The similarity between a query and a database image is computed according to Equation 1. To answer queries specifying text and image example, the similarity between a query and a database image is computed according to Equation 3.

For the evaluations, 20 characteristic queries of each type are created on topics related to Linux and Linux products. For each query the top 30 answers are retrieved. All performance results are averages over 20 queries. The evaluation is based on human relevance judgments by a human subject. For each method, the subject inspected the answers of each query and, for each answer, judged if it is similar to the query or not. This is a highly subjective process. Two or more methods may retrieve the same answer for the same query, but the same answer (by mistake) may not be recognized as similar when it is retrieved by different methods. To obtain consistent evaluations a query and a retrieved image are considered as similar if they are taken as similar by at least one method.

To evaluate the effectiveness of each candidate method, the following quantities are computed:

Precision, the percentage of relevant images retrieved with respect to the number of retrieved images.

Recall, the percentage of relevant images retrieved with respect to the total number of relevant images in the database. Due to the large size of the data set, it is practically impossible to compare every query with each database image. To compute recall, for each query, the answers obtained by all candidate methods are merged and this set is considered to contain the total number of correct answers [14].

In the following, a *precision-recall plot* is presented for each experiment. The horizontal axis in such a plot corresponds to the measured recall while the vertical axis corresponds to precision. Each method is represented by a curve. Each query retrieves the best 30 answers (best matches) and each point in a curve is the average over 20 queries. Precision and recall values are computed after each answer (from 1 to 30) and therefore, each curve contains exactly 30 points. The top-left point of a precision/recall curve corresponds to the precision/recall values for the best answer or best match (which has rank 1) while the bottom right point corresponds to the precision/recall values for the entire answer set.

A method is better than another if it achieves better precision and recall. It is possible for two (or more) precision-recall curves to intersect. This means that one of the methods performs better for small answer sets (containing less answers than the answer set at the intersection) while the other performs better for larger answer sets. The method achieving higher precision and recall for large answer sets is considered to be the best method (the typical users retrieve more than 10 or 20 images on the average).

Typically, image queries on the Web are issued through the user interface by specifying keywords or free text queries and the system returns images in Web pages with similar keywords or text as descriptions. The highest complexity of image queries is encountered in the case of queries by image example: The user may specify an example image along with a set of keywords or annotation expressing his information needs.

5.1 Text Queries

In this experiment, each query is specified by a set of keywords. All queries specified the term "logo". An image in the answer is considered similar to the query if they are at the same topic (e.g., query "Linux logo" may retrieve the logo of any Linux distribution (e.g., "Debian Linux").

Figure 8 illustrates the precision/recall diagram of the three candidate retrieval methods for text queries. PicASHOW is obviously the worst method. This result indicates that link information alone is not an effective descriptor for

[11]http://www.sleepycat.com
[12]http://lucene.apache.org

image content. The answers indeed contain a lot of irrelevant images. These are images that coexist within the same high quality pages with other relevant images, or are pointed to by high quality pages (e.g., pages of software companies). WPicASHOW is more effective than PicASHOW achieving up to 20% better recall and 15% better precision. WPicASHOW assigned higher ranking to images whose surrounding text is more relevant to the topic of the query. However, VSM is the most effective method (except from the first 3 answers). This result indicates that the surrounding text is a very effective descriptor of the image itself. This method assigned higher ranking even to images contained or pointed to by very low quality pages such as pages created by individuals or small companies.

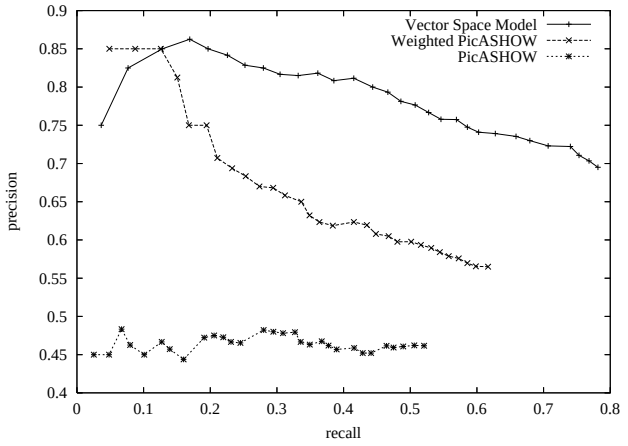


Figure 8: Precision-recall diagram for text queries corresponding to PicASHOW, WPicASHOW and the Vector Space Model.

5.2 Image Queries

Each query specifies a set of keywords along with an example logo image. For each keyword query, an appropriate logo is used as query. An image in the answer is considered similar only if its similar with the query image (e.q., query "Linux logo" with the penguin logo may only retrieve images showing a Linux penguin logo.

PicASHOW cannot answer such queries. Figure 9 illustrates the precision/recall diagram of the remaining two methods. The Vector Space Model is obviously far more effective than WPicASHOW. An important observation is that the performance gap between the two methods is wider than that of keyword queries. A closer look into the answers reveals that link analysis assigned higher ranking to Web pages with more general content on the topic of the query. The reason for this behavior is that Web pages with more general content are more strongly connected than pages with more specific topic. In this experiment, with the addition of logo image, the queries become more specific than before and WPicASHOW assigned higher ranking to more general but irrelevant images although in many cases these images are somehow related to the topic of the query (e.g., the "GNU" head logo with the "FSF" logo).

This behavior is in fact common to any link analysis method. WPicASHOW, as any other link analysis method, assigned higher ranking to higher quality but not necessary relevant pages. High quality pages, on the other hand, may be irrelevant to the content of the query. WPicASHOW attempted to compromise between the two.

The size of the data set is also a problem in both experiments. If the queries are very specific, the set of relevant answers is small and within it, the set of high quality and relevant answers are even smaller. The results may improve with the size of the data set, implying that it is plausible for the method to perform better when applied to the whole Web.

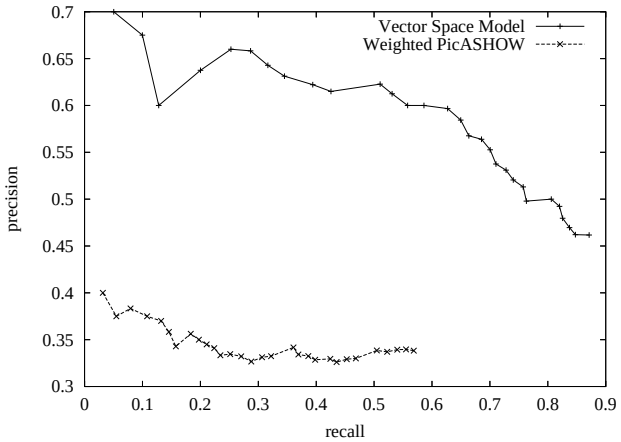


Figure 9: Precision-recall diagram for image queries corresponding to WPicASHOW and the Vector Space Model.

6. CONCLUSIONS AND FUTURE WORK

Existing approaches for handling logos and trademarks (e.g., [8, 12]) focus entirely on image content analysis and high precision answers to queries by image example but, they neither focus on detection (i.e., discrimination between trademark and non-trademark images) nor do they perform retrievals on the Web by image content. This work handles both these issues. Higher quality results are obtained when more important (authoritative) Web pages are assigned higher ranking over less important pages. This work implements PicASHOW and WPicASHOW, two well established methods for image link analysis and retrieval on the Web. Compared with PicASHOW, WPicASHOW allows also for more sophisticated image queries such as queries by example image in addition to text queries.

A complete prototype Web retrieval system for the retrieval of logo and trademark images is also designed and implemented as part of this work. The system stores a crawl of the Web rich in image and text content and offers the framework for a realistic evaluation of many Web image retrieval methods including PicASHOW, WPicASHOW and the Vector Space Model. The experimental results demonstrate that WPicASHOW is far more effective than PicASHOW, which uses link information alone. Link analysis improved the quality of the results but not necessarily their accuracy (at least for data sets smaller than the Web). The analysis revealed that content relevance and searching for authoritative answers can be traded-off against each other: Giving higher ranking to important pages seems to reduce the accuracy of the results.

Future work includes experimentation with larger data sets and image types, more elaborate methods for logo

and trademark detection and matching, and more elaborate crawling methods for fetching pages more relevant to the image type of the application (focused crawling) and the design of Web interface for making the system accessing to the public.

7. REFERENCES

[1] A. Arasu, J. Cho, H. Garcia-Molina, A. Paepke, and S. Raghavan. Searching the Web. *ACM Trans. on Internet Technology*, 1(1):2–43, Aug. 2001.

[2] Y.A. Aslandongan and C.T. Yu. Evaluating Strategies and Systems for Content-Based Indexing of Person Images on the Web. In 8^{th} *Intern. Conf. on Multimedia*, pages 313–321, Marina del Rey, CA, 2000.

[3] K. Bharat, A. Broder, M. R. Henzinger, P. Kumar, and S. Venkatasubramanian. The Connectivity server: Fast access to Linkage Information on the Web. In *Proceedings of the 7th International World Wide Web Conference (WWW-7)*, pages 469–477, Brisbane, Australia, 1998.

[4] K. Bharat and M. R. Henzinger. Improved Algorithms for Topic Distillation in a Hyperlinked Environment. In *Proc. of SIGIR-98*, pages 104–111, Melbourne, 1998.

[5] A.D. Bimbo. *Visual Information Systems*, chapter 2. Morgan Kaufmann, Academic Press, 1999.

[6] T. Gevers and A.W.M. Smeulders. The PicToSeek WWW Image Search Engine. In *IEEE ICMS*, June 1999.

[7] J.-Hu and A. Bagga. Identifying Story and Preview Images in News Web Pages. In 7^{th} *Intern. Conf. on Document Analysis and Recognition (ICDAR'2003)*, pages 640–644, Edinburgh, Scotland, Aug. 2003.

[8] A. K. Jain and A. Vailaya. Shape-Based Retrieval: A Case Study With Trademark Image Databases. *Pattern Recognition*, 31(9):1369–1399, 1998.

[9] M.L. Kherfi, D. Ziou, and A. Bernardi. Image Retrieval from the World Wide Web: Issues, Techniques, and Systems. *ACM Comp. Surveys*, 36(1):35–67, March 2004.

[10] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5):604–632, 1999.

[11] R. Lempel and A. Soffer. PicASHOW: Pictorial Authority Search by Hyperlinks on the Web. *ACM Trans. on Info. Systems*, 20(1):1–24, Jan. 2002.

[12] B. M. Mehtre, M. S. Kankanhalli, and W. F. Lee. Content-Based Image Retrieval using a Composite Color-Shape Approach. *Information Processing and Management*, 34(1):109–120, 1998.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Computer Systems Laboratory, Stanford Univ., CA, 1998.

[14] E.G.M. Petrakis, A. Diplaros, and E. Milios. Matching and Retrieval of Distorted and Occluded Shapes using Dynamic Programming. *IEEE Trans. on Pattern Analysis and Machine Intel.*, 24(11):1501–1516, Nov. 2002.

[15] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[16] Eds R. Baeza-Yates. *Modern Information Retrieval*. Addison Wesley, 1999.

[17] M. Tanase R.C. Veltkamp. Content-Based Image Retrieval Systems: A Survey. Technical Report UU-CS-2000-34, Department of Computing Science, Utrecht University, Oct. 2001. http://www.aa-lab.cs.uu.nl/cbirsurvey/cbir-survey.

[18] G. Salton, A. Wong, and C.-S. Yang. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[19] H.-T. Shen, B.-Chin Ooi, and K.-Lee Tan. Giving Meanings to WWW Images. In 8^{th} *Intern. Conf. on Multimedia*, pages 39–47, Marina del Rey, CA, 2000.

[20] A. W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE Trans. on Pattern Analysis and Machine Intel.*, 22(11):1349–1380, Dec. 2000.

[21] J.R. Smith and S.-Fu Chang. Visually Searching the Web for Content. *IEEE Multimedia*, 4(3):12–20, July-Sept. 1997.

[22] M. Sonka, V. Hlavec, and R. Boyle. *Image Processing Analysis, and Machine Vision*, chapter 14. PWS Publishing, 1999.

[23] M. Sonka, V. Hlavec, and R. Boyle. *Image Processing Analysis, and Machine Vision*, chapter 6. PWS Publishing, 1999.

[24] L. Taycher, M.La Cascia, and S. Sclaroff. Image Digestion and Relevance Feedback in the ImageRover WWW Search Engine. In 2^{nd} *Intern. Conf. on Visual Information Systems*, pages 85–94, San Diego, Dec. 1997.

[25] E. Voutsakis, E. G.M. Petrakis, and E. Milios. IntelliSearch: Intelligent Search for Images and Text on the Web. In 3^{rd} *Intern. Conference on Image Analysis and Recognition (ICIAR 2006)*, pages 697–708, Povoa de Varzim, Portugal, Sept. 2006.

[26] E. Voutsakis, E.G.M. Petrakis, and E. Milios. Weighted link analysis for logo and trademark image retrieval on the web. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI2005)*, pages 581–585, Compiegne, France, Sept. 2005.

[27] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, chapter 4. Morgan Kaufmann, Academic Press, 2000.

Proceedings of the Fifth IASTED International Conference
Signal Processing, Pattern Recognition, and Applications
February 13–15, 2008, Innsbruck, Austria
ISBN Hardcopy: 978-0-88986-717-8 / CD: 978-0-88986-718-5

ACTIVE BSVM LEARNING FOR RELEVANCE FEEDBACK IN CONTENT-BASED SKETCH RETRIEVAL

Shuang Liang and Zhengxing Sun
State Key Lab for Novel Software Technology, Nanjing University, Nanjing 210093, P R China
Department of Computer Science and Technology, Nanjing University, Nanjing 210093, P R China
szx@nju.edu.cn

ABSTRACT
The availability of relevance feedback is held back by the problem of the imbalance and limited size of labeled training data, as well as the real-time requirement of online interaction demands. In this paper, we propose a relevance feedback algorithm called active biased SVM (BSVM) learning, in which biased classification and active learning are employed to address these difficulties. The algorithm is applied to content-based sketch retrieval (CBSR), and the experiments prove both the effectiveness and efficiency of the proposed approach.

KEY WORDS
Content-Based Sketch Retrieval (CBSR), Relevance Feedback, Active Learning, Biased SVM (BSVM)

1. Introduction

As more electronic pen-based devices have and continue to become available for entering and processing information, sketching turns into one of the most popular modalities of digital ink to provide natural and flexible interaction, while not hindering creative thinking [1]. The digital sketching data are appealing because they capture user input interests, and people tend to organize and store the sketches for later retrieval. Hence the demand for finding similar sketches from large databases, or so called content-based sketch retrieval (CBSR), is shared not only by researchers and engineers, but also by ordinary users.

Considering the intrinsic structural characteristic of sketches, researchers tend to facilitate CBSR by shape information [2][3][4] as layout, hierarchy, topology, etc. However, similar to other information retrieval problems, despite exploring the best content representation and matching schemes, only a limited number of relevant items can be retrieved with the initial query. The two major reasons that bring on this problem are the semantic gap between computer and human beings and the dynamic understanding of user perception. So it is almost impossible to find a similarity matching approach that satisfies all user variations, and the ability of on-line associating users' understanding to computer features is required to bridge the semantic gap in retrieval process.

Relevance feedback is the most powerful way to reduce semantic gap and extend pure shape based similarity to more perceptually relevant similarity by introducing human judgments. The concept of relevance feedback was first carried out in text retrieval [5], and it has been shown to provide dramatic performance boost in CBIR systems. The main idea is to ask user to label each result as relevant (positive) or not (negative). This information are then collected and fed back to the system to refine the searching strategies online and get better results. Relevance feedback approaches develop from early heuristic weight adjustment to recently optimal learning methods [6], which formulate relevance feedback as a classification or density estimation problem. Many popular machine learning techniques were employed [7][8], among which the SVM-based techniques are the most attractive ones because of their good generalization ability [9]. Besides of binary SVM[10], one-class SVM (1SVM) [11] and biased SVM (BSVM) [12] are also proposed to deal with the imbalance between positive and negative examples.

However, there are three challenges associated with the specific relevance feedback scenario: First, small sample case. The number of labeled samples users can provide in each feedback loop is small. And the classifiers' performance may not be stable or meaningful when facing insufficient training samples; Second, imbalance in training datasets. The number of negative examples is significantly larger than the positive ones, so binary classification is not suited here; Finally, real-time requirement. The relevance feedback algorithm needs to perform sufficiently fast for online real time interaction. In this paper, we propose a new relevance feedback algorithm called active BSVM learning to attack the above relevance feedback difficulties. Active learning strategy and biased classification technique are combined together to effectively capture user's query interests and refine the searching strategy to further improve the retrieval accuracy. The algorithm is applied to CBSR to show its effectiveness. However, given that the features are represented in the form of vectors, this approach can be well extended to other information retrieval applications, such as trademark image retrieval, and so on.

2. BSVM Classification

Relevance feedback is considered as an online biased learning or classification problem in this paper [12], since the system needs to treat positive and negative data differently. In a biased classification, users are biased towards one certain class while neglect the others, which is exactly the situation we meet in typical relevance feedbacks.

We incorporate BSVM classifier to model the biased classification process in CBSR. The main idea of BSVM [12] is to find an optimal hypersphere which can include most of the positive instances while exclude most of the negative data with the help of user's labeled information. By modeling through a pair of concentric hyperspheres and allocating larger importance or weight to the positive instances than the negative ones, BSVM is more suitable for distinguishing the user's target items from other non-relevant ones. Given the training data set of $(\mathbf{x}_1, y_1),\ldots, (x_n, y_n) \in R^d \times Y, Y = \{-1, +1\}$, where $n \in N$ is the number of training samples, $d \in N$ is the dimension of feature space, x_i is a feature vector and y_i is its class label of the training sample. The decision function can be defined as [12]:

$$\begin{aligned} f(\mathbf{x}) &= \operatorname{sgn}(R^2 - \|\Phi(\mathbf{x}) - \mathbf{c}\|^2) \\ &= \operatorname{sgn}[-k(\mathbf{x},\mathbf{x}) + 2\sum_{i=1}^{n} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) \\ &\quad + R^2 - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)] \end{aligned} \tag{1}$$

Here $\Phi(\mathbf{x}_i)$ is the mapping function, $\mathbf{c}$ and R are the center and radius of the optimal hypersphere, and k corresponds to Mercer kernel $\mathbf{x}$. We use the concise form of the decision function to represent the sketch relevance in this paper by eliminating the constant values as:

$$relevance(\mathbf{x}) = -k(\mathbf{x},\mathbf{x}) + 2\sum_{i=1}^{n} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) \tag{2}$$

Consequently, sketches in the database are all ranked based on their relevance obtained from this evaluation function.

3. Active Learning

As we mentioned before, the small sample case is a main difficulty in achieving stable and meaningful results for relevance feedback. In this paper, we incorporate the idea of active learning to improve the learning performance and generalization ability of classifiers under a limited number of labeled samples.

Active learning refers to the strategies for the learner, e.g. the machine, to actively select the presentation samples to query the user for labels. The goal of active learning is to achieve the maximal information gain, or the minimized uncertainty in decision making [9]. Therefore it is desirable and necessary for the machine itself to select the optimal training samples. Assume the dataset D is made up of the labeled sample set L and unlabeled sample set U, then active learning l has two components (C, S), in which C: $L \rightarrow \{-1, +1\}$ is the classifier trained on the labeled dataset L; S is the selecting strategy which decides the next query subset u in U, given the current labeled set L. The main difference between active learning and traditional passive learning lies in the selecting strategies of how to choose the query samples from the unlabeled dataset. In a passive machine learning process, the learner typically serves as a passive recipient of the input data. While active learning enables the learner to have its own ability to choose the query data and collect user's response information. In this way, the learner could ask for what is best for refining the system.

More specifically, the special form of active learning we are interested in relevance feedback is selective sampling, which aims to reduce the number of training samples by selecting the "most informative" samples for query, thus decreases the training data requirements. Usually, uncertainty sampling [13] is taken to find out these "most informative" samples by choosing the least confident samples in classification to label.

4. Active BSVM Learning Algorithms

An active BSVM learning algorithm for relevance feedback is proposed to improve the retrieval accuracy of CBSR. The algorithm is carried out with BSVM selective sampling, which regards the samples with the highest uncertainty in BSVM classification as the "most informative" ones, and presents them to users to label.

4.1 BSVM Selective Sampling

The "most informative" samples should be the ones whose labels the learner is most uncertain about. Intuitively, these "most informative" samples lie close to the decision boundary. This is because the samples on or near to the decision boundary are the modes that are the hardest to classify, and their classification results are least reliable. In order to improve the accuracy of decision making, it is desirable to reduce this uncertainty area that situates near the decision boundary by maximally narrowing the boundary margins. Therefore, we select the samples with minimal distances to the BSVM decision hypersphere as the "most informative" ones, and query the user for labeling. The information each sample provides can be computed on the basis of the distance value $d(\mathbf{x})$ accordingly as:

$$information(\mathbf{x}) = \begin{cases} 0 & if\ d(\mathbf{x}) \geq threshold \\ 1 - \dfrac{d(\mathbf{x})}{threshold} & otherwise \end{cases} \quad (3)$$

where threshold is introduced from our experiments to normalize the sample information value. This calculation is simple and effective, making all information values fall into the range of [0, 1] while not violating their sequences in the feature space. The samples with the highest information values will be selected and presented to user to label.

4.2 Active BSVM Learning Framework for CBSR

We introduce active BSVM learning algorithm into CBSR system. The framework of active BSVM learning is described as follows: (1) Given a query: A set of ranked sketches are returned to user in sequence of their similarities to the query. (2) For each round of relevance feedback: If this is the first round, ask the user to label the top-k sketches; otherwise, a set of "most informative" sketches is recommended to user to label by BSVM selective sampling. Then the system learns the BSVM classifier with the current labeled data, and returns the k most positive sketches as the refined retrieval results.

The active BSVM learning algorithm can be divided into two stages. In order to find out the relevant sketches as much as possible, the top positive sketches are returned to users as the refined results in the first stage. In this way, users always get the best results in each loop, which implies the effectiveness of the algorithm. The second stage actively selects the most valuable query samples to decrease the requirements of the training data, thus maximally reduce user's burden in labeling process. This is the efficiency of the algorithm. Meanwhile, we employ incremental learning in BSVM training, and it is fast to calculate the sample information after the decision boundary is solved, which makes active BSVM learning suitable for on-line feedback interactions. Moreover, the active BSVM learning algorithm is independent on the specific features. As long as the features can be represented in the form of feature vectors, the algorithm can be employed to model the feedback problem effectively. Therefore, active BSVM learning is well extended to other figurative retrieval environments.

5. Experiments and Evaluation

Our proposed active BSVM learning algorithm is evaluated with CBSR under the environment of Pentium 4 2.0G CPU, 512MB memory, using a sketchy symbol database collected from 55 classes of engineering and electric symbols shown in figure 1. Ten persons are asked to draw 20 sketches for each class respectively, obtaining total 1100 sketches in the database. Spatial relations are employed as the features to represent the sketchy content information, which has been shown the availability of similarity matching for CBSR in our previous work [4].

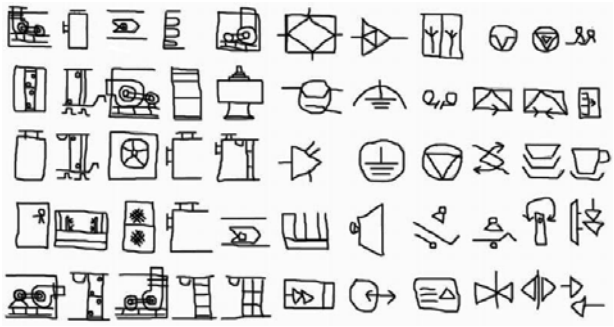


Figure 1. Illustration of 55 classes of sketches

Experiments are designed concerning the small sample case, training data imbalance, and real-time requirements. Each of 3 other users draws 55 query sketches of different classes to retrieve other sketches within the same class, which will be considered as positive results. Other returned candidates are then regarded as negative. Recall and precision values are averaged over all the classes and users. Relevance feedback results are recorded after 3 iterations since most users are not willing to label a large dataset and take too many feedback loops in a retrieval task.

5.1 Biased Vs. Regular

When confronting the imbalance dataset problem in relevance feedback, the biased classification ability of active BSVM learning is compared with other two regular SVM-based feedback algorithms of binary SVM and 1SVM. All these feedback algorithms are performed with the same feature extraction methods. In our experiments, LIBSVM library [14] is introduced to develop the SVM-based algorithms. The same kernel (Radial Basis Function, RBF) and parameters (ν=0.2) are chosen for all the SVM settings to enable an objective measure of performance without bias.

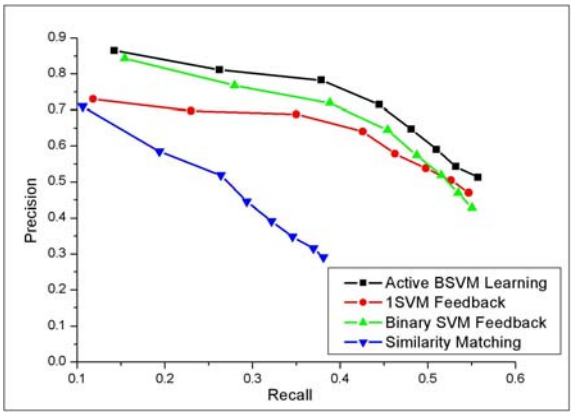


Figure 2. RP Graph of CBSR with relevance feedbacks

Figure 2 shows the RP graph of the SVM-based relevance feedback algorithms in CBSR. Compared with original similarity matching result, the retrieval performance is obviously improved by relevance feedback to a large extent. Among these algorithms, the curve of active BSVM learning is the highest, which indicates it outperforms the other two SVM-based relevance feedbacks by maximally boosting the retrieval accuracy. This is to say, the active BSVM learning algorithm with biased classification ability can retrieve the sketches more accurately than regular classical approaches.

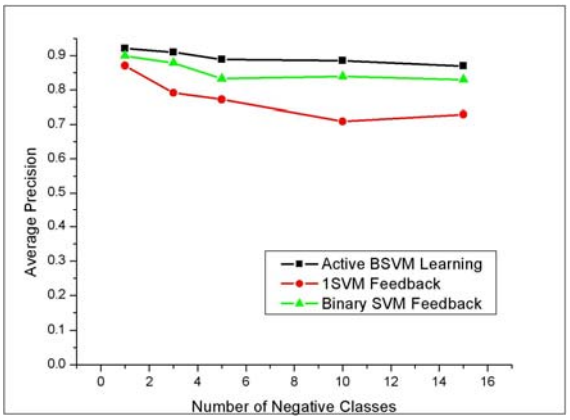


Figure 3. Relevance feedbacks with asymmetric dataset

More specifically, we randomly construct an imbalance database with n+1 classes (n = 1, 3, 5, 10, and 15) of sketches, which includes 1 positive class and n negative classes. As shown in figure 3, the imbalance problem becomes more serious with more negative classes, which results in a drop of the retrieval precision. Binary SVM method is not effective enough without considering the bias of positive data. While 1SVM feedback does not make use of the negative data, which cannot further improve the retrieval accuracy after the enclosed positive region is estimated. However, the curve of active BSVM learning algorithm drops little and almost reaches a horizontal status. This means active BSVM learning is least influenced by the asymmetric dataset and thus can handle the biased classification problem very well.

5.2 Active Vs. Passive

In order to validate the effectiveness of active BSVM learning in alleviating the small sample limitation for relevance feedback, we compare it with traditional BSVM passive learning.

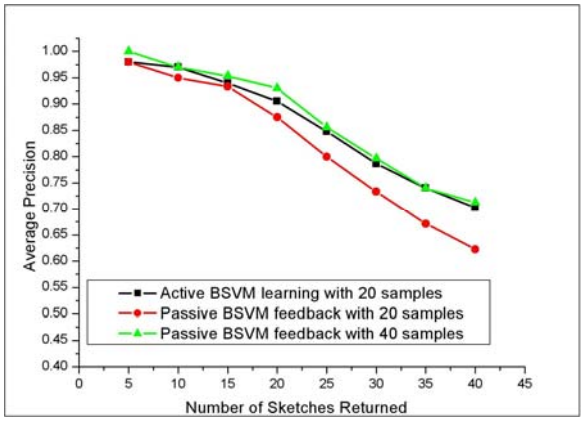


Figure 4. Performance of active and passive learning

Figure 4 depicts the AP curves of active and passive BSVM learning respectively. With the same number (20 samples) of returned samples, active learning algorithm can provide a larger performance boost than the passive one. While with different number of returned samples, by returning 20 samples in active BSVM learning, the retrieval performance can match the passive learning method with 40 returned samples. This is to say, the active BSVM learning algorithm can achieve high performance under the small sample case. And the number of labeling samples needed to refine the classifier is much less (almost a half) than that of traditional passive learning method, which reduces the burden of labeling in relevance feedback effectively.

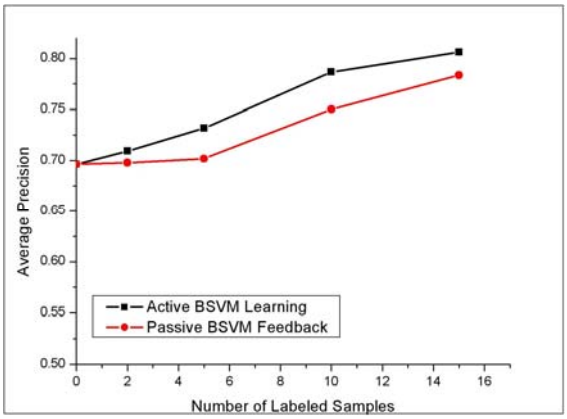


Figure 5. Performance with increasing labeled samples

The relationship between the average precision and the number of labeled samples is given in figure 5. We can notice that both algorithms lead to better results with the increasing of the labeled samples. While active BSVM learning algorithm can provide better performance with the same amount of labeled samples. Therefore, the small sample limitation of relevance feedback can be effectively attacked by active BSVM learning algorithm.

5.3 Real-time Requirement

Moreover, the time cost used in the retrieval process is also a much-concerned factor for evaluating the relevance

feedback performance. Especially, in our real-time interactive CBSR environment, the time cost should be as small as possible. In our experiments, the average response time is about 87.2 milliseconds for content matching and 142.7 milliseconds for feedback. Generally, the time cost is much less than a second, which is sufficiently fast for real-time interaction.

6. Conclusion

Similar to other information retrieval domains, benefits of advances in CBSR cannot be expected without introducing relevance feedback. In this paper, relevance feedback is considered as a small sample biased classification problem, and we propose active BSVM learning to effectively solve the relevance feedback difficulties. We employ a BSVM classifier to describe the data distribution for biased classification. Meanwhile, active learning is incorporated to select the "most informative" samples to label, thus improve the generalization ability of the classifier maximally and reduce user's burden of labeling. We also perform an incremental BSVM learning to reduce the processing time, which is fast enough for online interaction. Moreover, the active BSVM learning algorithm is independent on the feature extraction, given the features can be represented in the form of vectors in a feature space. Therefore it can be well extended to other domains of information retrieval as trademark image retrieval, and so on.

However, how to establish the accurate mapping between user and computer is still a difficulty for retrieval systems. In the future, we wish to further reduce the semantic gap. For example, the historical retrieval information can be organized effectively to carry out long-term feedbacks.

Acknowledgement

This paper is supported by the grants from the National 863 project of China [No. 2007AA01Z334], National Natural Science Foundation of China [No. 69903006 and 60373065], Program for New Century Excellent Talents in University of China [No. NCET-04-0460], and the FP6 IST project 511572-2 PROFI.

References

[1] Y.M. Chee, M. Froumentin, Ink Markup Language. W3C Working Draft. 23-Octbor-2006. Latest version URL: http://www.w3.org/TR/InkML.

[2] H. Leung, Representations, feature extraction, matching and relevance feedback for sketch retrieval, Doctoral Dissertation, Carnegie Mellon University, 2003.

[3] M.J. Fonseca, J.A. Jorge, Towards content-based retrieval of technical drawings through high-dimensional indexing, Computers and Graphics, 27(1), 2003, 61-69.

[4] S. Liang, Z.X. Sun, B. Li, Sketch retrieval based on spatial relations, Proc. 5th IEEE Conf. on Computer Graphics, Imaging and Visualization, Beijing, China, 2005, 24-29.

[5] G. Salton, Automatic text processing (Boston: Addison-Wesley, 1989).

[6] T.S. Huang, X.S. Zhou, Image retrieval with relevance feedback: from heuristic weight adjustment to optimal learning methods, Proc. 2001 IEEE International Conf. on Image Processing, Thessaloniki, Greece, 2001(3), 2-5.

[7] I.J. Cox, M.L. Miller, T.P. Minka, T.V. Papathomas, P.N. Yianilos, The bayesian image retrieval system, PicHunter: Theory, implementation and psychophysical experiments, IEEE Transactions on Image Processing, 9(1), 2000, 20-37.

[8] B. Li, Z.X. Sun, S. Liang, Y.Y. Zhang, Y. Bo, Relevance feedback for sketch retrieval based on linear programming classification, Proc. of Pacific-Rim Conf. on Multimedia 2006, Hangzhou, China, 2006, 201-210.

[9] X.S. Zhou, T.S. Huang, Exploring the nature and variants of relevance feedback, Proc. IEEE Workshop on Content-based Access of Image and Video Libraries, Kauai, USA, 2001, 94-101.

[10] P. Hong, Q. Tian, T.S. Huang, Incorporate support vector machines to content-based image retrieval with relevant feedback, Proc. IEEE International Conf. on Image Processing, Vancouver, Canada, 2000, 750-753.

[11] Y. Chen, X. Zhou, T. S. Huang, One-class svm for learning in image retrieval, Proc. IEEE International Conf. on Image Processing, Thessaloniki, Greece , 2001, 34-37.

[12] C.H. Hoi, C.H. Chan, K.Z. Huang, M.R Lyu, I. King, Biased support vector machine for relevance feedback in image retrieval, Proc. IEEE International Joint Conf. on Neural Networks, Budapest, Hungary, 2004(4), 3189-3194.

[13] D.D. Lewis, W.A. Gale, A sequential algorithm for training text classifiers, Proc. 17th Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval, Dublin, Ireland, 1994, 3-12.

[14] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/.cjlin/libsvm.

Proceedings of the Fifth IASTED International Conference
Signal Processing, Pattern Recognition, and Applications
February 13–15, 2008, Innsbruck, Austria
ISBN Hardcopy: 978-0-88986-717-8 / CD: 978-0-88986-718-5

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 ($\mathbf{I_{xy}} \in \Re^2$) of continuous $f : \Re^2 \to \Re$ where *f (x, y)* is the pixel intensity at *(x, y)* is $L : \Re^2 \times \Re_+ \to \Re$ which is given by the solution of the diffusion eqs 1 and 2.

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

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

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

$g(\mathbf{x},t) = \frac{1}{(2\pi t)^{D/2}} e^{-\mathbf{x}^{\mathrm{T}}\mathbf{x}/(2t)}$ 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 σ^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 $\{\sigma\}$ 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 σ 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.

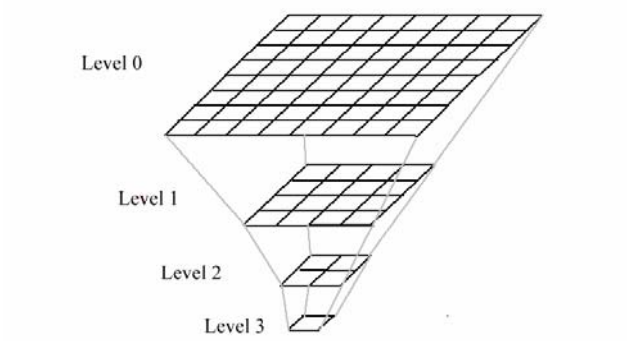


Fig. 1. The multiple levels of the Gaussian pyramid where the filtered image levels effectively form an inverted pyramid structure.

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 $\mathbf{I_{xy}} \in \Re^2$ is the original 512x512 pixel 2-D image then the pyramid is computed as eqs 3 and 4:

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

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

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

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 (σ) is selected and the image is blurred with a Gaussian kernel of size σ

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.

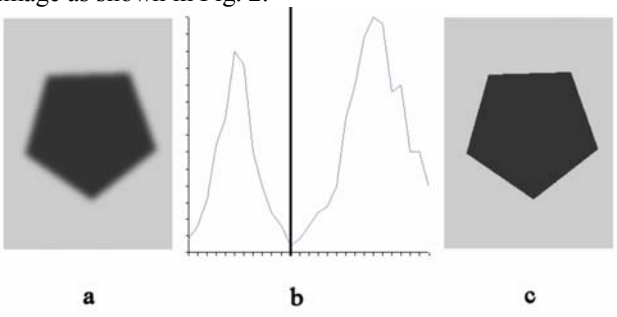


Fig. 2. (a) is the source image. (b) is this image's pixel intensity histogram with the pixel intensity threshold drawn for k=2 categories - the trough in the histogram identifies the threshold (category boundary). (c) shows the result of categorisation.

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 ...* 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 $(G_0\text{-}G_1)$ [6]. This is essentially an edge detection of G_0 and is given in eq 6:

$$L_0(x, y) = G_0(x, y) - RESIZE(G_1(x, y)) \quad (6)$$

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.

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

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:

2.4.1 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.

2.4.2 For texture/noisy images

There is a tendency for $\sigma_0 == \sigma_2$ in textured/noisy images where σ_0 is the scale selected for G_0 and σ_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 (σ_0) with G_0 image, categorise resulting convolved image.

- G_2 – select scale *(kernel width)*, convolve Gaussian (σ_2) with G_2 image, categorise resulting convolved image.

- If ($\sigma_0 == \sigma_2$) then
  - G_0 – select scale *(kernel width)*, convolve Gaussian (σ_0) with G_0 image, categorise resulting convolved image.
  - 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.

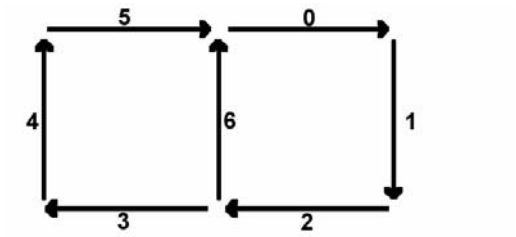


Fig. 3. A set of CCSs (0-6). The arrow heads denote the first end of the line segment and the opposite end of the line segment is hence the last end.

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.

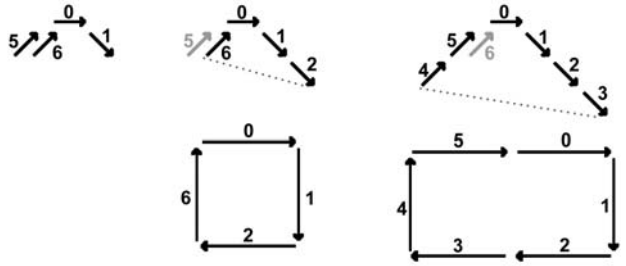


Fig. 4. The search tree for the set of CCSs in Fig. 3. The left tree shows the tree after expanding each end of node 0 (root). The middle tree shows how, when the tree is expanded by node 2, a closed path is found - 0126. When 2 is expanded, although 6 is end-point proximal it is not added as it is already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

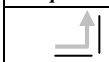
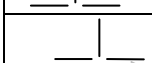
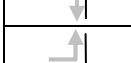
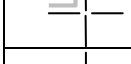
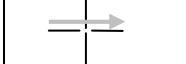
Junction	Turning ACW	Turning CW	Straight ACW	Straight CW
dist(CCS$_1$,CCS$_2$) < 2 pixels	***1.0***	***1.0***	***1.0***	***1.0***
	1.0	0.7	1.0	0.7
	0.7	1.0	0.7	1.0
	1.0	0.5	0.9	0.5
	***1.0***	1.0	1.0	1.0
	1.0	***1.0***	1.0	1.0
	0.5	1.0	0.5	0.9
	1.0	0.5	0.9	0.5
	0.5	1.0	0.5	0.9
	1.0	0.5	0.6	0.5
	***1.0***	***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.

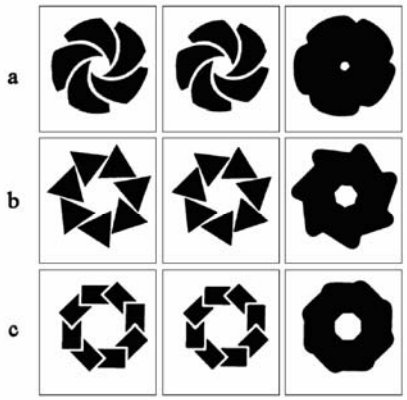


Fig. 5. Three images (a, b and c) and their respective outputs. All images were classified as line/region by the energy-based classifier.

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.

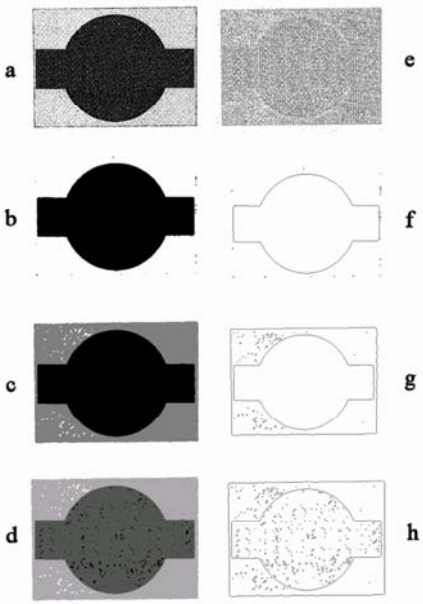


Fig. 6. The original image (a) is processed to produce a series of image views (b, c and d). The edges found are shown in e, f, g, and h

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.

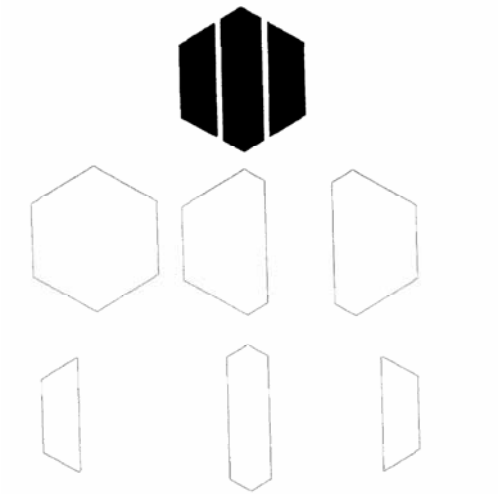


Fig. 7. The six perceptual shapes found by the shape identifier from the trademark image view in the top row.

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.

Proceedings of the Fifth IASTED International Conference
Signal Processing, Pattern Recognition, and Applications
February 13–15, 2008, Innsbruck, Austria
ISBN Hardcopy: 978-0-88986-717-8 / CD: 978-0-88986-718-5

SIMILARITY EVALUATION BASED ON IMAGE PRIMITIVES

Sven Scholz
Institute of Computer Science
Freie Universität Berlin
Berlin, Germany
email: scholz@inf.fu-berlin.de

ABSTRACT
A new framework for the perceptually relevant comparison of figurative images, especially trademark logos is presented in this paper. Images are divided into salient geometric figures such as rectangles, ellipses, and triangles. Parts not fitting into any of those simple classes are represented by their boundaries. The figures are classified, related, and weighted according to their perceptual relevance. For the comparison of two images the figures and the relations are compared independently from each other. For the comparison of single figures a simple measure of similarity based on registration techniques is applied, which is noise tolerant and shows good results for figurative images that have no spatially independent parts. The similarity of the images is then determined by the similarities of the figures and the relations for the best match. The algorithms were tested with a collection of 10 745 trademark images from the UK PTO, with the same set of 24 reference queries that were used to test the ARTISAN System. Each query consists of a query image and a list of relevant images, compiled by experienced trademark examiners. The experiments show that the presented approach allows for a considerable improvement of content based image retrieval in trademark images.

KEY WORDS
shape, content based image retrieval, trademark images

1. Introduction

For the comparison of figurative images that can be represented by a single closed (polygonal) curve, a variety of methods were invented that show respectable performance. Most trademarks on the other hand are way more complex and therefore the comparison has to consider many more aspects. Although one of the laws invented by Gestalt Theory states that configurations cannot be analyzed into parts and relations [1], for such multi-component images the comparison based on the individual image components is more effective than a comparison based on the whole image [2].

With regard to the ground truth provided by professional trademark examiners (see section 3), some observations can be made which are formulated as follows:

- People look for figures in the image that can easily be memorized. These figures may be abstract figures such as squares, circles, and triangles or figures of everyday life such as letters, digits, and stylized eyes or paperclips. If such figures exist within the image, their concrete proportions and positions play a minor role (see the appendix figs. 2 and 3).
  This is supported by the facts that:
  - a small number of common shape elements can form a basis for humans to discriminate between a wide variety of images [3] (cited in [4]).
  - ”there is an unconscious effort to simplify what is perceived into what the viewer can understand”. [5] (cited in [6])
- If the image consists of spatially independent parts, the size of the gaps inbetween plays a minor role (see the appendix fig. 4).
- If an essential part of the image is surrounded by a frame, the shape of the frame and even the existence of the frame play a minor role (see the appendix fig. 4). In [7] experiments on the way humans decompose figurative images were made. 5 of the images had a frame, for 3 of them all subjects completely ignored the frame and for 1 image only the second least significant decomposition (out of 9) contained the frame.
- Looking at a figurative image, the number of essential parts that are perceived is typically very small. For example in a regular pattern of little circles, one does normally not discriminate between the different circles, but group them together to a *’pattern of circles’*. Moreover when comparing such patterns it plays only a minor role if 16 circles form a 4×4 grid or if 25 circles form a 5×5 grid.

Our Framework for improving the comparison of figurative images is based on a very simple idea: try to characterize a figurative image the same way humans would do. If there is a circle in a triangle, characterize it as *‘a circle in a triangle’*, if there is something never seen before, characterize it as *‘something never seen before’* and describe it by what is known about it — in our case its boundaries. Many patent offices use such a characterization based on the so called Vienna classification [8]. The codes for the examples given in fig. 1 would possibly be *‘26.3.10 Triangles*

Figure 1. actual trademark images — some easy to describe by geometric primitives and some not.

containing one or more circles, ellipses or polygons' and *'26.13.25 Other geometrical figures, indefinable designs'* respectively.

Following this idea in our approach, an image is divided into a set of (not necessarily spatially independent) parts — preferably simple and salient geometric figures. These parts are classified, weighted, and related. The relationships are weighted as well. Comparing two images is accomplished by searching for subsets of the parts and their relations that match well.
The comparison of the parts is done independently, leaving aside their relative sizes and positions. It can be done using a similarity measure that works well for shapes whose parts lie close together whereas the resulting measure can handle arbitrary composed shapes.
In [9] a similar approach of dividing the images into geometric primitives and finding a match between these primitives is proposed. Its main drawbacks are 1.) that the comparison of the primitives does not prescind from their concrete positions and 2.) that the similarity between primitives belonging to different categories is defined as being zero, which is contrary to human perception e.g. when comparing a circle and a regular 12-gon.

We do not assume that all parts of all images can be replaced by high level primitives in a meaningful way. Analysis of annotations of trademark images shows that a considerable number of images needs different treatment (see 2.1). In addition, whenever a measure of similarity depends on the way the images are decomposed, there is the risk of underestimating the similarity just because two images get decomposed in different ways (e.g. two triangles forming a square vs. a square plus its diagonal).
For these reasons the comparison based on image primitives is not used as a stand-alone measure of similarity, but it is used in a framework to improve the results of the underlying, simple measure of similarity. Images are first compared using the underlying similarity measure and only if the decomposition leads to a higher value of similarity it is used. In this way the advantages of using high level features is combined with the robustness of the simple, low level comparison.

2. Comparison based on Image Primitives

For the comparison based on image primitives an underlying measure of similarity (e.g., the measure mentioned in section 2.4) is used, that assigns every pair of images or image parts their value of similarity $s \in [0, 1]$.
It is assumed that figurative images are given as a set P of polygonal boundary curves $p_1 \dots p_m$. Based on these polygonal curves a set F of figures $f_1 \dots f_n$ is extracted and their relations $R = r_{1,2} \dots r_{n,n-1}$ are computed.
The process of figure detection is not described in detail here, but the decomposition is assumed to be part of the input. For the experiments in sec. 3 however, a simple proof-of-concept implementation was used.

2.1 Figures

The figures can either be simple geometric objects (*image primitives*) or more complex objects. The primitives considered in our implementation are:

- ellipses (as a generalization of circles)
- rectangles (as a generalization of squares)
- triangles

The choice of these three types of primitives is based on their frequency of occurrence: In a collection of 1 762 395 trademark images for which we had access to the frequencies of the vienna codes, more than 23% of the images contain rectangles (as a special case of quadrilaterals) and 15% contain circles. These two topmost frequencies are followed by 'lines, bands' (which leaves open how to deal with geometrically), and by triangles.
Although these primitives occur very often, more than one half of the images is not annotated with one of them at all. Even with an increased set of primitive types, there will be unclassifiable parts remaining for which even humans have no proper category. The parts of the image that cannot be represented by the three types of primitives are categorized as

- convex polygons
- arbitrary sets of polylines

Analogously to concentric circles, 'concentric' ellipses, rectangles, triangles, and convex polygons resp. are conflated to a single figure with multiple layers.

2.2 Relations

For a pair $(f_i, f_j) \in F \times F, i \neq j$ of figures the relation $r_{i,j}$ consists of numerical values reflecting

- the size of f_j relative to the size of f_i (The size of a figure is defined to be the perimeter of the bounding box that maximizes the aspect ratio.)
- the relative distance of f_j to f_i (The distance of the bounding boxes' centers relative to the size of f_i.)
- the qualitative relation, i.e., the similarity of f_i and f_j under translations, rigid motions and under reflections.

2.3 Comparison of two Images

For the comparison of two images I^1 and I^2 the relevance w_F of the figures and the relevance w_R of the relations is preset such that $w_F + w_R = 1$ — for images consisting only of one type of figures, e.g., only squares, the relations between these figures are of greater importance than for images consisting of totally different figures. The figures and relations get weights $w(f_i)$ and $w(r_{i,j})$ according to their salience, such that for each image all weights sum up to 1, namely: $\sum_{f \in F} w(f) = w_F$ and $\sum_{r \in R} w(r) = w_R$.

For every pair $(f_i^1, f_k^2) \in F^1 \times F^2$ of figures a value of similarity $s(f_i^1, f_k^2) \in [0, 1]$ is computed, using the underlying measure of similarity. For every pair $(r_{i,j}^1, r_{k,l}^2) \in R^1 \times R^2$ of relations a value of similarity $\tilde{s}(r_{i,j}^1, r_{k,l}^2) \in [0, 1]$ is computed, using a simple measure of similarity.

Let $\mathcal{M}$ be the set of all one-to-one matchings between figures of image I^1 and image I^2. The value of similarity S of the two images is then defined as the weighted sum of the similarities of the matched figures, plus the weighted sum of the similarities of the (implicitly) matched relations:

$$S(I^1, I^2) = \max_{M \in \mathcal{M}} \left\{ \sum_{(f_i^1, f_j^2) \in M} s(f_I^1, f_J^2) \cdot \frac{w(f_i^1) + w(f_j^2)}{2} + \sum_{\substack{(f_i^1, f_k^2) \in M \\ (f_j^1, f_l^2) \in M}} \tilde{s}(r_{i,j}^1, r_{k,l}^2) \cdot \frac{w(r_{i,j}^1) + w(r_{k,l}^2)}{2} \right\}$$

The problem of determining whether $S(I^1, I^2) \geq \theta$ for a given threshold $0 < \theta \leq 1$ is an extension of the *quadratic assignment problem* (see e.g. [10]) and therefore is NP-complete. Since the number of essential parts that are perceived is typically very small, the admissible number of figures that represent an image can be bounded by a small constant (see section 3). Thus, the value of similarity $S(I^1, I^2)$ may be computed using a branch and bound algorithm for enumeration of the promising matches.

2.4 Proof of Concept Implementation

Several estimates in the implementation are arbitrarily fixings. Since comprehensive psychological studies on e.g. the relationship between the size and the perceived relevance of figures or on the effect of repeated figures were not available (or at least unknown to the author), the formulas used stem from qualitative considerations but do not necessarily comply with reality in their quantitative behavior.

Weights Every figure f_i gets an absolute weight $w_a(f_i)$ which equals the square root of the figure's size (perimeter of the figure's bounding box that maximizes the aspect ratio). Every relation $r_{i,j}$ gets an absolute weight $w_a(r_{i,j})$ based on the absolute weights of the figures f_i and f_j. The weights w used in the comparison are derived from these absolute weights by normalizing them such that $\sum w(f) = w_F$ and $\sum w(r) = w_R$. If two images I^1 and I^2 with different numbers n^1, n^2 of figures are compared, only the relations for $n_{min} = min(n^1, n^2)$ figures may be selected. In this case the weights of the relations of the image consisting of more figures are adjusted such that the maximum sum of the weights of relations between a n_{min}-subset of the figures equals w_R.

Frames A frame is a — mostly rectangular — part of an image that only surrounds the essential parts, but has only very limited or no influence on the perception of the image. For every figure the likeliness of being a frame is rated based on the following propositions:

- frames are convex and symmetric
- frames contain at least one complex figure or two primitive figures
- frames are not too small compared with surrounding frames
- frames are not surrounded by something that is not a frame

Based on this likeliness the weight of a frame figure is decreased by a factor $\in [1.0, 2.0]$.

Repetitions If a logo contains groups of identical figures, the concrete number of these identical figures plays only a minor role in comparison (see the appendix fig. 3) and some trademark images even contains miscellaneous variants of the actual logo (see the appendix fig. 2). Therefore the weights of such copies are reduced.

Underlying Measures of Similarity For the underlying measures of similarity between figures or relations respectively, values between 0 and 1 are required so that the resulting value will range from 0 to 1. In [11] such a normalized measure of similarity is described which works respectably well for figurative images whose parts lie close together. The basic idea behind this approach is to find a (similarity) transformation $t : \mathbb{R}^2 \to \mathbb{R}^2$ that maps parts of the one figure f^1 into the proximity of corresponding parts of the other figure f^2 and the similarity is rated based on proximity and parallelism of $t(f^1)$ and f^2. For the comparison of image primitives (ellipses, rectangles, triangles) the values of similarity may be predefined, for the comparison of primitives with complex figures the values may be precomputed so that only the values for the comparison of complex figures have to be computed online.
The similarity of 2 relations $r_{i,j}^1$ and $r_{k,l}^2$ is computed by a formula based on the difference in relative distances, the difference in relative sizes, and the qualitative relations i.e. the similarity $s(f_f, f_j)$ under translations, rigid motions, or reflections.

3. Experimental Results

The retrieval performance was tested with the same set of 10 745 trademark images and the same 24 reference queries that were used to test the ARTISAN System [12]. Each query consists of a query image and a list of relevant images from the test set (including the query image). The lists of relevant images had been compiled by experienced trademark examiners (examples of query images with some relevant images can be found in Appendix A). Most of the images depict abstract geometrical figures — black shapes on white background — but some of the figures are hatched or have texture: the number of closed contours (distinguishable black and white areas) exceeds 1 000 for about 800 images (7 %) and the maximum observed is even 92 436.

From every image the set of polygonal boundary curves was extracted and polygons belonging to noise and texture were eliminated[1]. The remaining closed contours for which every vertex corresponds to a pixel, were then simplified using the Douglas-Peucker algorithm [13] (cited in [14]).

The segmented images were automatically decomposed by detecting image primitives and grouping the remaining parts based on their proximity. For images with more than one possible decompositions a value of *simplicity* was computed for every decomposition (based on regularity of the figures, symmetries, and number of figures). More than $90\,\%$ of the images were decomposed into at most 6 figures, the maximum number of perceptually relevant figures in an image that were identified by the segmentation was 14.

For each of the 24 queries, all images were compared to the query image and they were ranked according to the resemblance values. Let N be the number of images, n the number of relevant images for a query, r_i the rank of the i-th relevant image, and r_l the maximum rank of a relevant image for a query. The retrieval performance was rated based on the following values as defined in [12]:

Normalized Recall R_n Value in the range from 0 (worst case) to 1 (perfect retrieval).

$$R_n = 1 - \frac{\sum_{i=1}^{n} r_i - \sum_{i=1}^{n} i}{n(N-n)}$$

The recall gives a higher weight to success in retrieving the first few items.
The average value for the 24 queries achieved by the combined approach was 0.96 (0.90 early artisan, 0.94 late artisan). The average value achieved by the underlying measure of similarity alone was 0.93, so the framework yields an improvement of 0.03.

[1] This noise reduction is important but it is not in the main focus of our work. Therefore, a very simple implementation was used, that was not able to process the entire collection of images. In 116 cases out of 10 745, the texture in the image had to be removed by hand and the segmentation was redone.

Normalized Precision P_n Value in the range from 0 (worst case) to 1 (perfect retrieval).

$$P_n = 1 - \frac{\sum_{i=1}^{n} \log(r_i) - \sum_{i=1}^{n} \log(i)}{\log\left(\frac{N!}{(N-n)!\cdot n!}\right)}$$

The precision gives equal weight to all retrievals.
The average value for the 24 queries achieved by the combined approach was 0.79 (0.63 early artisan, 0.70 late artisan). The average value achieved by the underlying measure of similarity alone was 0.71, so the framework yields an improvement of 0.08.

Normalized Last-Place-Ranking L_n Value in the range from 0 (worst case) to 1 (perfect retrieval).

$$L_n = 1 - \frac{r_l - n}{N - n}$$

The last-place-ranking indicates the number of retrieved items a user has to search in order to have reasonable expectation of finding all relevant items.
The average value for the 24 queries achieved by the combined approach was 0.79 (0.56 early artisan, 0.72 late artisan). The average value achieved by the underlying measure of similarity alone was 0.68, so the framework yields an improvement of 0.11.

Number of Retrieved Images $n_{0.01}$ The number of relevant images ranked within the top 1 percent of the entire collection.
The sum for the 24 queries achieved by the combined approach was 229 (168 early artisan). The sum achieved by the underlying measure of similarity alone was 191, so the framework yields an improvement of 20 %.

For the detailed values of all 24 queries see the appendix table 1.

4. Conclusion

A new framework for content based image retrieval (esp. for trademark images) is presented which does not so much bank on sophisticated computation, but on taking account of some observations concerning perception: Familiar figures in the images are mostly perceived separately and their relevance may differ considerably. According to these observations the computation of image similarity is proceeded as follows: Images are divided up into sets of simple figures and the figures are weighted according to their relevance. The comparison of images is based on comparing the figures as well as their relations separately and on summing up the weighted similarities for the best matching of figures. The results of the experiments encourage further efforts in this direction, e.g., for improving the partitioning of the images, extending the set of image primitives, and refining the underlying measures of similarity for figures and relations.

Acknowledgements

This work was supported by the European Union under contract No. IST-511572-2, Project Perceptually-Relevant Retrieval of Figurative Images (PROFI).

References

[1] H. Helson. The fundamental propositions of gestalt psychology. *Psychological Review*, 40(1):13–32, 1933.

[2] John P. Eakins, K. Jonathan Riley, and Jonathan D. Edwards. Shape feature matching for trademark image retrieval. In *CIVR*, pages 28–38, 2003.

[3] Mary C. Dyson, Hilary Box, and Michael Twyman. The perception of symbols on screen and methods of retrieval from a database. British Library Research and Development Department Report 6163. British Library, London, 1994.

[4] John P. Eakins, Kevin Shields, and Jago Boardman. ARTISAN – a shape retrieval system based on boundary family indexing. In *Storage and Retrieval for Still Image and Video Databases IV. Proceedings SPIE 2670*, pages 17–28, 1996.

[5] Mercedes M. Fisher and Karen Smith-Gratto. Gestalt theory: A foundation for instructional screen design. *Journal of Educational Technology Systems*, 27(4), 1998-1999.

[6] Dempsey Chang, Laurence Dooley, and Juhani E. Tuovinen. Gestalt theory in visual screen design: a new look at an old subject. In *CRPIT '02: Proceedings of the seventh world conference on computers in education: Australian topics*, pages 5–12, Darlinghurst, Australia, 2002. Australian Computer Society, Inc.

[7] Victoria J. Hodge, Garry Hollier, John P. Eakins, and Jim Austin. Eliciting perceptual ground truth for image segmentation. In *CIVR*, pages 320–329, 2006.

[8] International classification of the figurative elements of marks (vienna classification) fifth edition. WORLD INTELLECTUAL PROPERTY ORGANIZATION, 2002. ISBN 92-805-1054-7.

[9] Hui Jiang, Chong-Wah Ngo, and Hung-Khoon Tan. Gestalt-based feature similarity measure in trademark database. *Pattern Recognition*, 39(5):988–1001, 2006.

[10] Eugene L. Lawler. The quadratic assignment problem. *anagement Science*, 9:586–599, 1963.

[11] Helmut Alt, Ludmila Scharf, and Sven Scholz. Probabilistic matching and resemblance evaluation of shapes in trademark images. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 533–540, New York, NY, USA, 2007. ACM Press.

[12] John P. Eakins, Jago M. Boardman, and Margaret E. Graham. Similarity retrieval of trademark images. *IEEE MultiMedia*, 5(2):53–63, 1998.

[13] D. Douglas and T. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. In *The Canadian Cartographer*, volume 10, pages 112–122, 1973.

[14] John Hershberger and Jack Snoeyink. Speeding up the douglas-peucker line-simplification algorithm. In *Proceedings of the 5th International Symposium on Spatial Data Handling*, volume 1, pages 134–143, Charleston, South Carolina, 1992.

Appedix A Examples of Trademark Images

Some examples of query images together with relevant images that can not be handled properly with a simple registration based approach.

Figure 2. Query image (left) and images to retrieve having different proportions.

Figure 3. Query image (left) and images to retrieve having different arrangements.

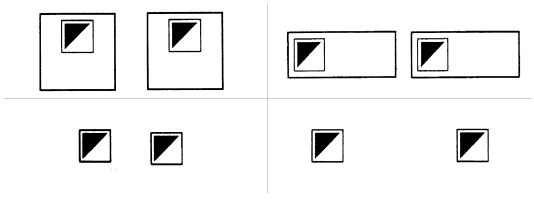


Figure 4. Query image (top left) and images to retrieve having different gaps and different frames.

Appendix B Experimental Results

	query	relevant images	R_n	P_n	L_n	$n_{0.01}$
1.		26	0.99	0.87	0.93	19
2.		16	0.99	0.87	0.89	13
3.		12	0.96	0.89	0.60	10
4.		10	0.92	0.81	0.34	7
5.		10	0.99	0.72	0.97	4
6.		18	0.94	0.80	0.36	12
7.		11	0.97	0.71	0.89	6
8.		20	0.98	0.86	0.73	16
9.		25	1.00	1.00	1.00	25
10.		11	0.92	0.54	0.76	5
11.		10	1.00	0.91	0.98	8
12.		4	1.00	0.99	1.00	4
13.		16	0.97	0.62	0.89	6
14.		6	0.94	0.70	0.74	4
15.		13	0.99	0.85	0.93	10
16.		13	1.00	0.97	0.99	13
17.		17	0.94	0.66	0.72	9
18.		12	0.97	0.60	0.87	6
19.		21	0.67	0.28	0.11	1
20.		8	0.97	0.79	0.85	6
21.		8	1.00	0.91	0.98	7
22.		10	0.99	0.74	0.91	8
23.		23	0.99	0.87	0.93	20
24.		13	0.97	0.86	0.67	10
	mean / sum	333	0.96	0.79	0.79	229
	standard deviation		0.07	0.16	0.23	

Table 1. Results achieved: 24 query images plus values for normalized recall, normalized precision, normalized last place ranking, and number of relevant images ranked within the top 1 percent of the entire collection

Proceedings of the Fifth IASTED International Conference
Signal Processing, Pattern Recognition, and Applications
February 13–15, 2008, Innsbruck, Austria
ISBN Hardcopy: 978-0-88986-717-8 / CD: 978-0-88986-718-5

TOPOLOGICAL AND DIRECTIONAL LOGO LAYOUT INDEXING USING HERMITIAN SPECTRA

Reinier H. van Leuken
Information and Computing Sciences
Universiteit Utrecht, The Netherlands
reinier@cs.uu.nl

Olga Symonova
Fondazione Graphitech
Trento, Italy
olga.symonova@graphitech.it

Remco C. Veltkamp
Information and Computing Sciences
Universiteit Utrecht, The Netherlands
remco.veltkamp@cs.uu.nl

ABSTRACT
To evaluate similarity between two images, the layout or configuration of the shapes is an important feature besides geometrical shape similarity. In particular, trademark image retrieval is an application domain where layout similarity is important, and in many cases overlooked. In this paper, we present a graph-based encoding of layout, in which both directional and topological layout information is stored. A Hermitian matrix is associated to each graph, and contains all the information that is present in the graph. The spectra of these Hermitian matrices are used for indexing purposes. By obeying several constraints on the construction of the Hermitian matrices, we can mimic the spectral behaviour of Laplacian matrices, which are proven to be successful representations in retrieval environments. Experiments show the improved representational power of the proposed approach over spectral methods using Laplacian matrices.

KEY WORDS
Indexing, image retrieval, trademarks, Laplacian, Hermitian, spectra

1. Introduction

The key function of any indexing algorithm is to speed up content-based retrieval of objects or models that are stored in a database, by selecting a small set of candidate objects that are either presented to the user, or passed on to a more refined matching unit in the retrieval pipeline. At this matching level, more accurate and more expensive matching algorithms can be deployed because of the reduced size of the set of objects that is under inspection. At the indexing level however, comparison of objects should be efficient and it must be possible to prune the database, i.e. the database must be partitioned in such a way that similar models are positioned close to each other. Only then objects that are far from the query object can be discarded without further inspection.

Naturally, the representation of the objects in the index and the accuracy and efficiency with which non-similar objects can be discarded are closely related. The objects that are under investigation in this work are logo and trademark images, or any kind of image in general where the layout of the individual image components (as opposed to their shape characteristics) is important for similarity evaluation [10]. In content-based trademark image retrieval, layout can play a large role in identifying trademark infringement. See for an example Figure 1, where the configuration of the individual shapes is one of the most important properties. Suppose that in all three cases the five circles are returned as a result of image segmentation (which would be the ideal segmentation), it is impossible to distinguish between the images without any notion of layout in the representation. In this case, indexing algorithms (without layout information) will be less efficient because the set of candidate models will be unnecessary large. More importantly, indexing algorithms can be less accurate by ignoring layout. See Figure 2 for an illustration of a case where a low similarity score will be calculated for similar images, if only shape similarity is taken into account. If one of these images is a query, neither of the other two will be returned based on shape similarity. However, according to trademark experts, if these image were to be registered as real trademarks within similar product or service categories, a conflict of uniqueness may arise [10].

Within the area of content-based image retrieval, a lot of work has been devoted to spatially oriented retrieval. One of the most popular techniques often used for this purpose is based on string matching. To produce the strings that encode layout, the centres of mass of all objects are projected on the x and y axes. By taking objects from left to right and from below to above, and by representing these objects by a class identifier, two one-dimensional strings are formed that together form the 2D-String [3]. A number of modifications and extensions to this idea have been presented, see [9, 6, 2] for a some examples. A major drawback of these symbolic projection methods is that in general they are not rotation invariant.

In this paper, we propose a new spectral encoding for layout of shapes that can be represented and compared efficiently. Recent studies [11] have shown how spectral representations of layout can be used to index trademark collections. With the proposed encoding however, that follows some of the ideas of [12], we are able to discriminate better between different configurations, as we take into account more information without sacrificing any efficiency.

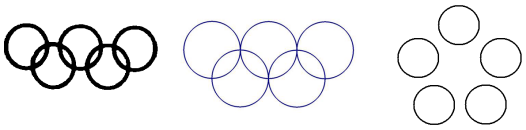


Figure 1. Example of different configurations of the same primitive shapes, with decreasing layout similarity from lefttoright.

Figure 2. Three trademarks with similar layouts, but dissimilar primitive shapes.

1.1 Our contributions

The main contribution of this paper is a new method for efficient retrieval of trademark images, or images in general, that is based on the layout of the different shapes the image is composed of.

To this end, a graph is constructed for each image in which the layout of the trademark is encoded. After associating a matrix with each graph, the spectra (sorted sets of eigenvalues) of these matrices are compared for similarity evaluation. However, unlike most spectral methods, that usually focus on connectivity, several types of additional information are taken into account as well. For this purpose, we will use the spectrum of a Hermitian matrix. In Section 2 details about Hermitian spectra and how to match them are given.

This work proposes a way to encode both precise directional and topological relations between the components. These additional (graph) properties are reflected in the spectrum that is used for similarity evaluation during indexing. Details on the graph construction and calculation of attributes are given in Section 3. By obeying several constraints on the definition of the graph's topology and geometry measurements, and by encoding these values in a Hermitian matrix, we can mimic the spectral behaviour of the Laplacian matrix. The obtained spectrum can therefore be used for efficient retrieval, as the Laplacian spectrum has been proven to be reliable for this purpose in recent studies [11, 4]. Finally, in Section 4, experiments show the increase in representational power of the encoding over existing methods.

2. Hermitian spectral representation

One of the most natural and informative algebraic structures to associate with a graph is its Laplacian matrix. This matrix is defined as $L(G) = D(G) - A(G)$, where $D(G)$ is the diagonal matrix containing node degrees, and $A(G)$ is representing G's connectivity; the entry $A_{i,j}$ is 1 if nodes i and j are connected, 0 otherwise. As a result, for all rows in $L(G)$ the sum of the entries is 0. The spectrum of the Laplacian matrix can be used as a signature representation for the graph, and thus for the model that is represented by the graph. This signature representation can be used for efficient retrieval purposes (indexing), [4]. One of the main reasons for this is that many graph properties and invariants are implicitly or explicitly reflected by the Laplacian spectrum [8]. Moreover, cospectrality for non-isomorphic graphs tends to be rare [13] and similar Laplacian matrices have similar spectra due to the interlacing theorem for two graphs where one is a slightly modified version of the other [7].

In the case of a weighted graph, $L_w(G) = D_w G - A_w G$ can be obtained. In this case, $D_w(G)$ is a diagonal matrix containing for each node the sum of edge weights of its incident edges. Correspondingly, in the adjacency matrix the entry $A_{i,j}$ represents the weight associated with nodes i and j, which is 0 if there is no connecting edge between them. Therefore, all information with respect to the graph's connectivity is still present in $L_w(G)$, since every non-zero entry indicates the existence of an edge between the corresponding nodes.

In order to preserve the useful properties of a normal Laplacian spectrum, every edge weight $w_{a,b}$ should satisfy the following conditions:

$$W_{a,b} = W_{b,a}, \text{ where } a, b \in V \quad (1)$$

$$W_{a,b} \geq 0, \text{ where } a, b \in V \quad (2)$$

$$W_{a,b} \neq 0, \text{ iff } a \text{ and } b \text{ are adjacent in } G \quad (3)$$

Equation 1 ensures a symmetric matrix, whereas equation 3 ensures that the connectivity of the graph remains unchanged after weighting the edges.

Unfortunately, it is not possible to store more information in a Laplacian matrix than the graph's connectivity together with the edge weights. As a consequence, a spectral representation using this matrix will suffer in most cases from significant information loss, since other graph characteristics such as node labels, node locations (planar graphs, 3D graphs) or additional edge measurements are not captured by the encoding.

Therefore, following the ideas of [12], we use a Hermitian matrix to store graph characteristics. However, to really mimic the spectral behaviour of a Laplacian matrix, we added two additional constraints to the construction of the Hermitian matrices. First, we give a brief theoretical background on Hermitian matrices, and then we impose the constraints for mimicking Laplacian spectral properties.

A Hermitian matrix H (or self-adjoint matrix) is a square matrix with complex entries that is equal to its own conjugate transpose. In other words, $H_{i,j}$ is equal to the complex conjugate of $H_{j,i}$. Fortunately, every Hermitian matrix has a real valued spectrum. The corresponding

eigenvectors however contain complex entries. By adding several additional constraints to the construction of H, we can mimic the spectral behaviour of a Laplacian matrix, i.e. we can construct a property matrix $H(G)$ for $G = (V, E)$ in such a way that we can use its spectrum for retrieval purposes equally well as the Laplacian spectrum.

To this end, the off-diagonal elements of H are chosen to be complex numbers written in polar form using Euler's formula:

$$H_{a,b} = -W_{a,b}e^{iy_{a,b}} \tag{4}$$

where each edge has the pair of observations $(W_{a,b}, y_{a,b})$. The second observation, represented as the phase of the complex matrix entry, must satisfy the following conditions:

$$y_{a,b} = -y_{b,a} \tag{5}$$

$$-\pi < y_{a,b} < \pi \tag{6}$$

The first condition (5) ensures that H is equal to its own conjugated transposed matrix. By obeying the second constraint (6), phase wrapping can be avoided.

The on-diagonal entries (that are required to be real) are chosen to be

$$H_{aa} = \sum_{b \neq a} W_{a,b} \tag{7}$$

In this way, the entries in each row of the matrix now sum up to zero. This on-diagonal entry is necessary, because all edge weights $W_{a,b}$ (magnitudes) are inserted as $-W_{a,b}$, see (4). By summing up the edge weights and inserting this sum as on-diagonal entry, the sum of the entries in each row is zero. We would like to stress that this is a necessary property to correctly mimic the spectral behaviour of Laplacian matrices, contrary to the Hermitian matrix that is used in [12] (where additional node measurements on the diagonal are allowed). Furthermore, edge weights (magnitudes of the complex entries) should be calculated in such a way that an edge between two nodes can never be weighted 0, for it would destroy the connectivity of the graph.

2.1 Retrieval based on spectra

It is the key function of any indexing algorithm to speed up the retrieval process by selecting a small set of candidate models that are either presented to the user, or passed on to a more refined matching unit in the retrieval pipeline. The representation used here during indexing is a spectral one, which is basically a d-dimensional vector of features where d is the number of nodes in the graph, or the size of the Hermitian matrix. Therefore, to evaluate similarity between two objects, we calculate the Euclidean distance between their feature sets, i.e. between their Hermitian spectra. When trademarks are of different size, the spectra are of different dimension. There are several ways to deal with this problem. It is possible to enlarge the spectrum of the smaller trademark by inserting zeros. This is semantically correct, since it means isolated nodes are added to the graph. Another possibility is to decompose the graph into several subgraphs, and match only subgraphs of the same size. For more details on how to handle graphs of different sizes, we refer to [11]. In the rest of this paper we will assume graphs are of the same size.

In order to index a large data set efficiently, the vectors can be accessed through a Balanced-Box-Decomposition Tree (BBD-Tree), as introduced in [1]. This data structure is proven to be optimal for $(1 + \epsilon)$-approximate nearest neighbour searching [1], where k approximate nearest neighbours in a d-dimensional space can be reported in $O(kd \log n)$ time.

3. Graph attributes

With the goal to describe a trademark, we construct the graph whose nodes represent the shapes of the trademark revealed after the segmentation phase. We connect each node with its six nearest neighbours based on the distance between the barycenters of the corresponding shapes. There are many possible attributes that can be used to enrich a graph structure with additional shape information. To name a few, the attributes can be the area, perimeter, curvature of the corresponding segment, whereas the edges can be weighted with the distance or the angle between the shapes. The scope of our work is to represent the layout of the trademark. To this end, we will use the information about the location and intersection of shapes with respect to each other. Moreover, the use of the Hermitian matrix for the graph encoding imposes the constraints (1)-(3), (5) and (6) which the graph attributes should satisfy.

3.1 Directional attributes

For the description of the position of one shape with respect to the other we chose the angular measure. Precisely, we compute the angle between the two lines formed by the end points of an edge and the barycenter of the trademark. See Figure 3 for an example. This attribute satisfies the conditions (5) and (6) and thus can be used as the phase of the complex off-diagonal entries of the Hermitian matrix.

3.2 Topological attributes

Egenhofer and Franzosa [5] pointed out that there are 8 basic topological relations: disjoint, contains, inside, meet, equal, covers, covered-by and overlap. These relations, or intersection types, can be partially captured with one intersection measure on two components, which we define as

$$W_{ab} = \frac{Area_{ab}}{Area_a + Area_b}$$

[1] An object is a $(1 + \epsilon)$-approximate k-nearest neighbour of the query if its distance to the query is within a factor of $(1 + \epsilon)$ to the distance between the query and its true k-nearest neighbour.

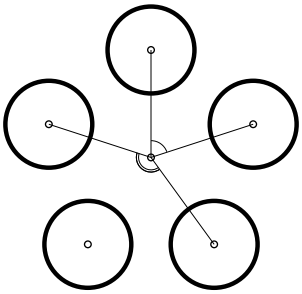


Figure 3. Computation of directional attributes: angle between lines formed by connecting the two end points of each edge to the trademark's barycenter.

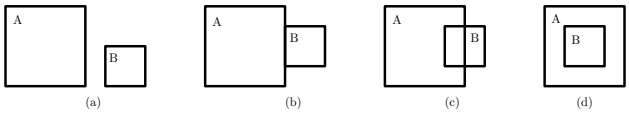


Figure 4. Different intersection types for the shapes $Area_A = 4$, $Area_B = 1$. (a) separate shapes $W_{AB} = 1$, (b) touching shapes $W_{AB} \approx 1$, (c) intersecting shapes $W_{AB} = 0.91$, (d) shape a includes shape b $W_{AB} = 0.8$.

where $Area_{ab}$ is the area of the union of the components a and b. The area of a component is measured by the number of pixels occupied including boundary pixels. For two separated segments the intersection measure is equal to one and decreases as the intersection area increases. Figure 4 illustrates different types of the intersections. The intersection measure satisfies the conditions (1)-(3) and thus can be used as the magnitude of the complex elements of the Hermitian matrix.

4. Experiments

To evaluate the effectiveness of the proposed approach, we focus on comparing several typical examples of shape configuration. Therefore, in this section we will assume that segmentation reveals the individual shapes, and calculates the angular and topological values. Furthermore, we will assume that the graphs that are compared are of the same size, i.e. they have the same number of vertices. For details on an appropriate segmentation technique, and on how to work with graphs of different sizes we refer to a recent study [11].

In this Section, we will evaluate how distances between pairs of trademarks change when topological and directional changes in the configuration occur. At this point, we would like to point out that every distance will be 0, should each image be represented by the spectrum of its normal Laplacian matrix. When a weighted Laplacian matrix is chosen as associated structure, angular or directional changes in the configuration are not revealed during similarity evaluation.

Table 1. Distance matrix for different topological configurations of 12 circles.

dist					
	0	0.023	0.107	1.067	5.441
	0.023	0	0.084	1.044	5.420
	0.107	0.084	0	0.960	5.343
	1.067	1.044	0.960	0	4.474
	5.441	5.420	5.343	4.474	0

In the first experiment, all pairwise distances between 5 configurations of 12 circles are calculated. The angles between the circles are the same in all images, the overlap varies from disjoint to touching, overlapping, more overlapping and inclusion. See Table 1 for the results of this experiment together with the images that are used for calculation. The results clearly show how distances increase when overlap increases. The experiment is repeated with configurations of four squares in Table 2. Again, the angles between the squares remain constant, while the overlap increases from no overlap to inclusion. For these examples, distances grow proportionally with increasing overlap as well. Furthermore, this experiment shows that calculation of the topological attributes is dependent on the shape of the components. For instance, a larger distance is found for configurations of squares than of circles between a disjoint configuration and a touching configuration (first row, second column of both Tables 1 and 2).

The third experiment, of which the results are given in Table 3, shows the benefit of the directional information in the encoding. All these distances would have been 0 using normal or even weighted Laplacian matrices as a representation. The distances listed in Table 3 coincide with the perceived similarity between the images. For example, the first and the third images both appear to have a smaller distance to each other than to all other images, which is a desired result in this case.

The images used for the final experiment have variations in both topological and directional configuration. As the results show in Table 4, even with these combined alterations, distances reflect the similarity in layout. Take for instance the pair of the first and fourth images, that have a closer distance to each other than to all other images. Furthermore, the influence of the enclosing frame in the fifth image is clearly present, since it has a large distance to all other models. Finally, the two images containing only

Table 2. Distance matrix for different topological configurations of 4 squares.

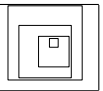
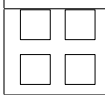
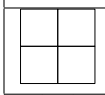
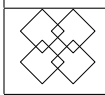
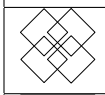
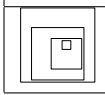
dist					
	0	0.036	0.167	0.489	1.046
	0.036	0	0.161	0.453	1.014
	0.167	0.161	0	0.333	0.899
	0.489	0.453	0.333	0	0.643
	1.046	1.014	0.899	0.643	0

Table 3. Distance matrix for different angular configurations of 5 circles.

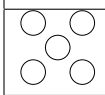
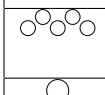
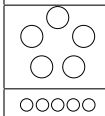
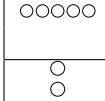
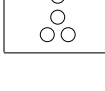
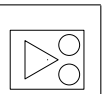
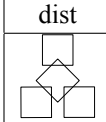
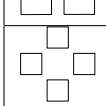
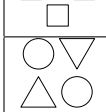
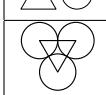
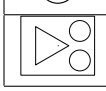
dist					
	0	1.119	0.531	1.933	1.24
	1.119	0	0.917	0.875	0.237
	0.531	0.917	0	1.782	1.108
	1.933	0.875	1.782	0	0.716
	1.24	0.237	1.108	0.716	0

Table 4. Distance matrix for different configurations 4 of shapes with mixed properties.

dist					
	0	0.446	0.922	0.173	1.446
	0.446	0	0.582	0.513	1.208
	0.922	0.582	0	0.902	0.695
	0.173	0.513	0.902	0	1.356
	1.446	1.208	0.695	1.356	0

disjoint components (second and third image) are close to each other, but still have a nonzero distance because of differences in directional attributes.

5. Conclusion

In this paper we have presented a new approach for encoding layout between image components that, together with the shapes of the components, is important for evaluating similarity between images. Both directional and topological relations between image components that are near each other, are encoded in a rich graph structure. By associating a Hermitian matrix to the graph, and by obeying several constraints on the computation of edge weights, we are able to capture more edge information (together with the connectivity) in a spectral representation that mimics the behaviour of Laplacian spectra. Therefore, similarity evaluation is efficient and accurate, and the proposed approach can be successfully applied as an indexing mechanism.

The next step will be to evaluate the new approach within the context of a real retrieval environment. Although it was shown before that spectral representations are well suited for this kind of retrieval purposes, and we have shown in this paper that the new Hermitian spectral representation is more discriminating and provides distance values that reflect layout similarity better, it is important to investigate retrieval performance on a real data set of trademark images. To do so, we will make use of a large collection of real trademark images that has been classified by trademark experts who evaluate trademark similarity on a daily basis. We will compare our results to other methods using popular and representative performance measures such as Average Dynamic Precision, Mean Cumulative Gain Vectors and Mean Discounted Cumulative Gain Vectors.

Furthermore, it is one of our interests in the near future to explore part-based similarity between graphs using a spectral approach. Since the eigen decomposition of a Hermitian matrix reveals the eigenvectors as well as the spectrum, we automatically obtain the eigenvector associated with the second smallest eigenvalue (the so-called Fiedler vector and Fiedler value respectively) [8]. The Fiedler vector can be used for partitioning the graph in sensible parts, avoiding the computationally expensive inspection of all possible subgraphs of all different sizes. These subgraphs can be represented again by their Hermitian spectra. A voting schema will be necessary to combine search results for complete and partial graphs.

Acknowledgements

This research was supported by FP6 IST projects 511572-2 PROFI and 506766 AIM@SHAPE.

References

[1] S. Ayra, D.M. Mount, N.S. Netanyahu, R.Silverman, and A.Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. Journal of the ACM, 45(6):891–923, 1998.

[2] S.K. Chang, E. Jungert, and Y. Li. Representation and retrieval of symbolic pictures using generalized 2D strings. In Conference on Visual Communications and Image Processing , volume 3, pages 1360–1372, November 1989.

[3] S.K. Chang, Q.Y. Shi, and C.W. Yan. Iconic indexing by 2-d strings. Pattern Analysis and Machine Intelligence, 9(3):413–428, 1987.

[4] M.F. Demirci, R.H. van Leuken, and R.C. Veltkamp. Indexing through laplacian spectra. Computer Vision and Image Understanding , In press, 2007.

[5] M.J. Egenhofer and R.D. Franzosa. Point Set Topological Relations. Geographical Information Systems, 5(2):161–174, 1991.

[6] S.Y. Lee and F.J. Hsu. 2D C-string: a new spatial knowledge representation for image database systems. Pattern Recognition , 23(10):1077–1087, 1990.

[7] R. Merris. Laplacian matrices of graphs: a survey. Linear Algebra and its Applications , 197(1):143–176, 1994.

[8] B. Mohar. The laplacian spectrum of graphs. Graph Theory, Combinatorics and Applications, 2:871–898, 1991.

[9] E.G.M. Petrakis and S.C. Orphanoudakis. A Methology for the Representation, Indexing, and Retrieval of Images by Content. Image and Vision Computing, 8(11):504–512, October 1993.

[10] J. Schietse, J.P. Eakins, and R.C. Veltkamp. Practice and challenges in trademark image retrieval. In Conference on Image and video retrieval , pages 518–524, 2007.

[11] R.H. van Leuken, M.F. Demirci, V.J. Hodge, J. Austin, and R.C. Veltkamp. Layout indexing of trademark images. In Conference on Image and video retrieval , pages 525–532, 2007.

[12] R.C. Wilson, E.R Hancock, and B. Luo. Pattern vectors from algebraic graph theory. Pattern Analysis and Machine Intelligence, 27:1112–1124, 2005.

[13] P. Zhu and R.C. Wilson. A study of graph spectra for comparing graphs. In British Machine Vision Conference, 2005.