

Object-Based Image Description and Query Refinement in a 3D Iconic Environment

Jean-Philippe Turcat, Claude C. Chibelushi, and Adrian A. Low
School of Computing, Staffordshire University
Stafford ST18 0DG, England

J.P.Turcat@staffs.ac.uk, C.C.Chibelushi@staffs.ac.uk, A.A.Low@staffs.ac.uk

Abstract

In this paper, we present a novel approach for image database querying based on a 3D iconic environment, which supports multi-attributes image description and iterative query refinement. A graphical interface enables the user to create a query scene by instantiating and transforming domain-specific 3D objects. The query scene and the database images are each associated with a descriptor of scene attributes, and relevance feedback is used for refining queries. The paper investigates the impact of inexact image description, and relevance feedback, on the effectiveness of the proposed image-retrieval approach. Experimental results show that imperfect image description degrades the precision of image retrieval without relevance feedback. It is also shown that the query refinement approach, proposed in the paper, enhances retrieval precision.

1. Introduction

Content based image retrieval (CBIR) has been studied with much interest in the last twenty years. In particular, some attention has been given to developing techniques that allow the use of a virtual environment to make visual queries [2] [7]. Research into such environments has been identified as a key area for CBIR [13]. Several investigations such as [3] [8], have studied 3D based environments for CBIR. Typically, they render 3D models from several viewpoints, and compare each 2D view against 2D database images.

During the processing of a query, analyzing the content of each image of a database might need long processing time. Query processing time can be reduced by using descriptors associated with each image [15] [7] together with an indexing scheme. Such descriptors can be generated when the image is added to the database. This way, the query will only process the descriptor instead of the image

itself, hence resulting in greater processing speed compared to analyzing the image directly. MPEG7 [1] is a standardisation effort that tries to unify approaches for building descriptors capable of integrating various types of information contained in an image or video.

In the investigations reported herein, we have chosen to focus on CBIR for a narrow application domain, with a constrained vocabulary for describing an image. This idea is reinforced by Smeulders *et al.* [13] who say that the quality of a search engine typically improves when the retrieval is performed on a narrow domain. Although the techniques presented in this paper can be used in other domains, the query environment has been constrained to office furniture and computer equipment, in the application domain considered herein. Such a tool would be useful for office interior designers, for example, who may require access to a database of images from office equipment catalogues. The user can choose objects such as tables, chairs, desks or computers, and place them wherever they want in a 3D virtual room. The latter can be used as a query, which would return real photographs matching 2D views of the virtual room.

An icon can be defined as a symbolic metaphor of an object in a scene. Icons can denote a 2D or 3D environment. The descriptor used herein is a compact representation of a 3D iconic environment; the descriptor contains information on the iconic content of the query. There are four main attributes contained in a descriptor: the identifiers of the icons, their spatial relationship, their texture, and their motion.

Relevance feedback is an important problem [6] [13], which has not been fully solved yet [4] [10]. One CBIR system that integrates relevance feedback is known as MARS [11]. By using a weighted version of the k nearest neighbor rule, the system is able to use feedback information in order to weight significant features. The feature weight is calculated by using the feature variance from the set of retrieved images considered to be relevant by the user. Another system that integrates relevance feedback is known as PicHunter [5]. By using a probabilistic method based on

Bayesian statistics, the system tries to direct the search and predict what image the user wants, by looking at user actions.

This paper addresses two problems associated with the use of object-based CBIR. The first problem is the matching a 2D image descriptor with a 3D-based query. An important issue is the imperfect object detection typically inherent in automatic extraction of image descriptors. The paper investigates the impact of inexact descriptor data on the effectiveness of the retrieval. The second problem under study is the ability to refine a query by using relevance feedback.

The main contributions of the paper are twofold. First, the paper presents a framework adapted to CBIR for 3D queries, using object-based multi-attribute image search. Second, we have developed an approach for relevance feedback based on the weighted feature approach, but the proposed approach differs from others by the way the contribution of each attribute is calculated.

2. CBIR framework

The retrieval process is executed in several steps, as illustrated in Figure 1. In the first stage, the user creates an iconic query that represents the scene he or she is looking for. Then, a descriptor is extracted for the iconic representation of each object. The set of descriptors that compose the full query are compared to the object descriptors of images from the database. The comparison incorporates user-specified weights for the descriptor attributes. The user can then select the most relevant images returned by the query, or even modify the query parameters. A refined query will then be executed. User feedback, in form of selection of the most relevant images will affect the consequently refined query, by weighting the next matches to promote images similar to the relevance feedback image set and to the initial query.

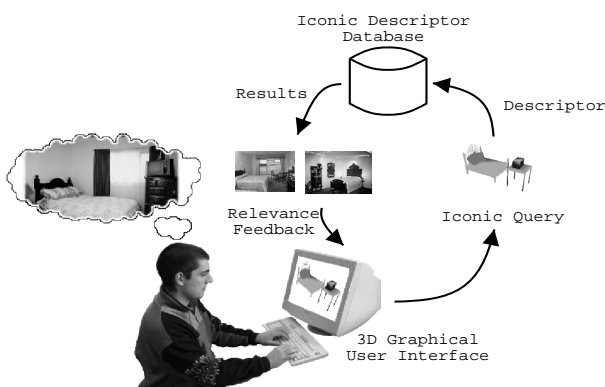


Figure 1. The CBIR cycle.

Smeulders *et al.* [13] stress the importance of a graph-

ical interface, and particularly the interaction between the user and the search engine. We address this challenge by using an interface, which enables the user to build queries by using 3D icons. The interface consists of a virtual environment where the user can create icons, move them, or change icon properties. The motivations for this approach are given hereafter. Unlike visual based queries, text based queries lack precision. It is almost impossible to describe every detail of an image using words. The main reason is that words may carry different meanings in different contexts. A visual query can be constructed using a 2D or 3D environment. When using a 2D query, we face the limitation of restricting the query to a particular viewpoint of a scene. However, a 3D query opens CBIR to searches from any viewpoint. The 3D query will be transformed to a set of 2D views, which will be compared with the database images. However, the relevance feedback is performed using the 2D views represented by the images selected by the user as relevant.

In the rest of this section, we explain what information a descriptor contains, and how the image descriptor attributes are extracted from images. Finally, we explain how the matching is performed.

2.1. Image descriptor set

A descriptor contains four types of information describing the iconic representation of one particular object in an image. First, the identifier, written as a symbolic string representing an object name. Extracting a name is done through shape recognition, based on an approach which uses multiple 3D viewpoints, which is outlined in Section 2.2. The name has an associated quality coefficient which represents the confidence that the shape recogniser had in its object-labelling accuracy. The quality coefficient will be used when processing the query, to favour objects associated with greater levels of confidence.

The second descriptor entry is the spatial location of a particular object relative to others. Hence, the scene configuration is described using an object-centered coordinate system [2]. In this paper, extracting the spatial relationships has been fully automated. By comparing orthogonal projections of the spatial location of objects, the positional relationships between objects is expressed using semantic rules such as "*object A is strictly after object B*". This is based on the work presented in [2].

The third part of a descriptor contains the photometric properties of the object. This part is composed of two attributes, the texture and the colour of the object. In the work presented herein, the colour has been extracted automatically. This has been done by averaging the value of pixels across the surface of the object. However, the texture attribute, in the form of a descriptive word, has been input

manually because automatic texture analysis is outside the scope of the work presented in this paper.

A descriptor could also contain motion information. This part would be useful for video retrieval, but it is not considered in this paper since our database contains only static images.

2.2. Object labelling for image description

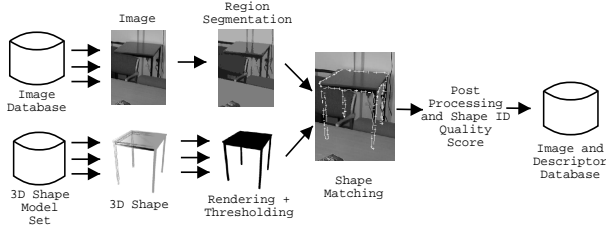


Figure 2. Automatic extraction of object identifier for image description.

There are many different ways to create the descriptors of images contained in a database. In this paper, we use a semi-automated approach. The texture fields of the descriptor are manually created. Object detection and labelling is done automatically, it relies on shape. Unfortunately, such an operation cannot be performed perfectly, and may result in objects not properly detected. Hence, Section 3 presents an assessment of the impact, of extracting inexact descriptor data, on retrieval effectiveness.

The object-detection method can only detect the objects contained in a pre-defined 3D shape-model query set. Consequently, this will limit the choice of objects available to the user during a query. However, such a limitation is unimportant for many narrow application domains.

The object detection process is done in three steps, illustrated in Figure 2. First, the image is segmented into regions. The segmentation algorithm is based on the clustering approach described in [15]. Then, a 3D shape model of the relevant object is rendered in several views. Each rendered view is compared with the segmented image using a method based on the generalised Hough transform [14]. The idea is to use a polygonal approximation of the segmented shape by dividing its contour at several key points joined by line segments. Then, what the authors call a vector-pair transform is applied. A vector pair consists of the two line segments joined at one key point. For each key point, we record the angle between the two segments. Then, we match similar vector-pair angles from the query image and a segmented database image, and increment the corresponding accumulator cells. Accumulator cell values are divided by the size of the shape. The best match, represented by the accumulator maximum, if above a threshold,

indicates where the object, seen from a particular view, is likely to be located. The maximum across all views gives the best view corresponding to the database image. Finally, a similarity score, which indicates how well the detected shape matched the projection of the 3D object model, is attached to the shape identifier stored in the descriptor. It is anticipated that the quality of the shape detection will be influenced by the number of views chosen to render the object. However, this issue is outside the scope of this work, a possible solution for selecting optimal views is described in [9].

2.3. Descriptor matching

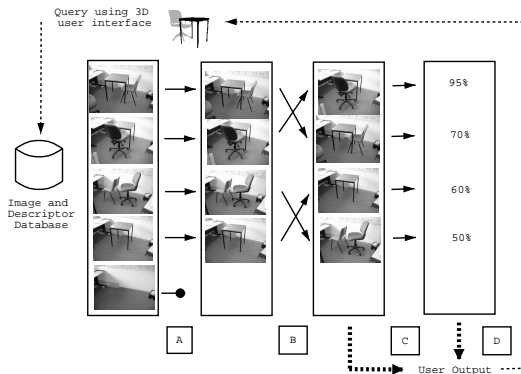


Figure 3. Descriptor comparison process. A : Comparison based on object matching. B : Comparison based on other scene description attributes ranked by the user in order of importance. C : Evaluation of match scores. D : Display of results with support for relevance feedback.

In contrast to our approach, in [2] the user chooses one particular view of the 3D scene. In our system, whenever the user does not specify a view of interest, the camera is moved around the space, generating many possible descriptors, typically for around one thousand viewpoints. This number corresponds to a 10 degree rotation step around the axes of the world reference frame. The distance from the camera to the world is unimportant here since the technique used for comparing spatial configuration is invariant to scaling. We extract from every view of the scene a 2D symbolic description. If this symbolic description is sufficiently different from the previously generated views, then it is compared with the descriptors of the database images.

Comparison of descriptors is done in four steps, illustrated in Figure 3. First, pre-processing is applied in order to extract only the relevant images from the database. This is done by selecting only the images that possess the most

important distinctive features specified in the query. For example, if a user specifies that the object identifier is the most important descriptor entry, we will select only the images that contain many of the icons contained in the query. Up to this stage, no spatial relations or photometric properties are involved yet, only a comparison on the shape-derived icon identifier is performed. Similarly, if any another attribute is selected as the most important distinctive feature, only this attribute will be considered for the first pre-processing step. In the particular case of all the attributes being equally important, the descriptor comparison process can be executed sequentially in any order.

Then, a comparison of the remaining attributes of the iconic descriptor is performed. The weight associated with each attribute comparison is chosen by the user when building the query. Object identifier entries in the descriptor are compared by matching corresponding identifier values. The number of matching objects, between the database image and the query scene, is divided by the number of objects in the query scene. This gives a percentage that represents the quality of the match.

Comparing the spatial configuration of the scene is based on the method described in [2], which uses a representation language, describing the spatial relation between scene objects. Del-Bimbo *et al.* [2] use a symbolic description, which is captured by a set of formulas, expressing the mutual relationship between pairs of objects. Each spatial relation produced from the query scene, is compared with the corresponding relation in the image. A similarity score s , expressed as a percentage, is returned as part of the query result. The score is calculated using the equation :

$$s = 100 * \frac{g}{t} \quad (1)$$

where g represents the number of relations that satisfy the correct configuration of the query scene, and t the total number of spatial relations specified in the query scene.

The colour and texture are currently used as placeholders. However, in order to produce experimental results, we have used a simple method illustrating their contribution to the comparison. Colour is compared using the differences in hue and saturation colour components. A percentage is returned which represents the ratio between the number of colours found in the image compared to the number of different colours specified in the queried. We do not consider common problems such as noise or colour variation, because they are challenges in their own right, and they are outside scope of this paper. Texture is compared using an algorithm for matching texture labels, based on an approach similar to the one for matching object identifier descriptor entries.

The final step is to compute a quantitative score for the match. The score, reflecting the level of similarity of scene attributes, is calculated by a weighted averaging of

the match score for individual descriptor attributes. The formula used is

$$f = \sum_{n=1}^4 w_n c_n \quad (2)$$

where c_n is the absolute value of the difference between corresponding descriptor entries, each c_n is represented as a percentage. w_n represents the weight specified by the user for the n^{th} descriptor attribute (the weights satisfy the constraint $\sum_n w_n = 1$). We use percentages for descriptor attribute similarity scores for two reasons. First, all scores are normalised to the same scale. Another reason is that percentages are quite meaningful as feedback to the user.

2.4. Relevance Feedback

The relevant images chosen by the user orient the next iteration of the search. This is done by replacing w_n by $w_n * p_n$ in Equation (2), which becomes :

$$f = \sum_{n=1}^4 w_n p_n c_n \quad (3)$$

where p_n is the frequency of occurrence of descriptor entries which are similar between the relevance-feedback image set and the original query image. p_n is calculated as a count, of similar entries, divided by the total number of entry-similarity comparisons. As in Section 2.3, c_n measures the difference between a database image and one view from the 3D query.

3. Experimental results and discussion

The first objective of our experiments is to assess the impact of imperfect image description on the effectiveness of image retrieval, based on the approach described in the preceding section of the paper. The second objective is to assess the improvement accruing from relevance feedback.

The test database comprises 70 images showing a room which contains office furniture and computer equipment. Shots are taken from different locations, and with different configuration of the scene. Images which do not belong to the application domain (office furniture and computer equipment) are not included in the test database. However, there is great variability in object location and type across the database, which increases the level of difficulty for the retrieval task. For query formulation, the set of shape models is composed of eight different 3D mesh models which represent three different tables, a computer screen, a printer, a computer tower case and two types of chairs. In the experiments reported herein, we have used fixed weights in Equation 3, with corresponding values of $w_1 = 0.5$, $w_2 = 0.3$,

$w_3 = 0.15$, $w_4 = 0.05$. These values were obtained experimentally, they were observed to offer a good balance across the weights of individual descriptor attributes, for effective discrimination between images. Retrieval effectiveness is assessed using a precision measure [13]. The precision measure represents the proportion of relevant images found within the ten best-ranked images in the query result.

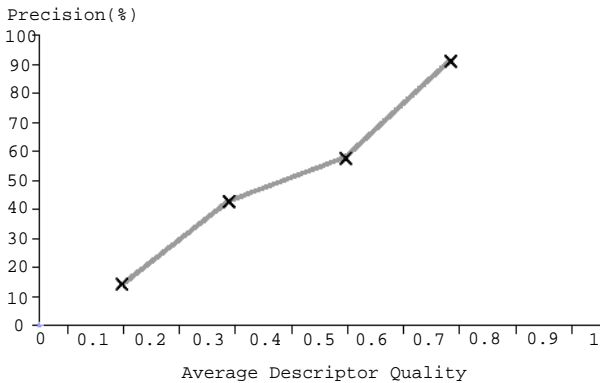


Figure 4. Effect of descriptor quality on retrieval effectiveness, without relevance feedback.

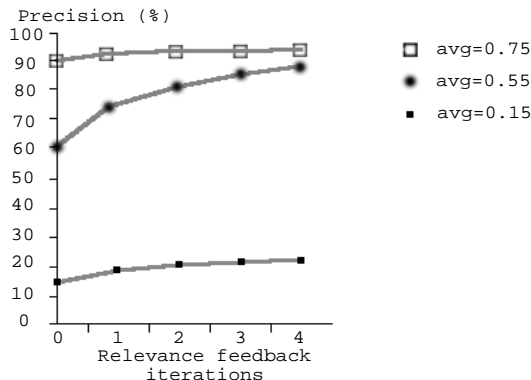


Figure 5. Effect of relevance feedback on retrieval precision. *avg* is the average descriptor quality coefficient.

Only the quality of the shape detection, during the extraction of the object identifier descriptor attribute, is considered in the investigation of the effect of descriptor quality. Hence, the similarity score attached to the object identifier attribute of the descriptor, is used as a measure of descriptor quality. Descriptor quality has been measured as the average of the similarity scores for every object in the descriptor. The images used in the test data, cover a wide range of descriptor quality. Figure 4 shows the effect of

descriptor quality on retrieval effectiveness. It reveals that retrieval precision increases with descriptor quality.

The last set of tests concerns relevance feedback. Figures 5 and 6 show that, for a given descriptor quality, retrieval precision improves with relevance feedback.

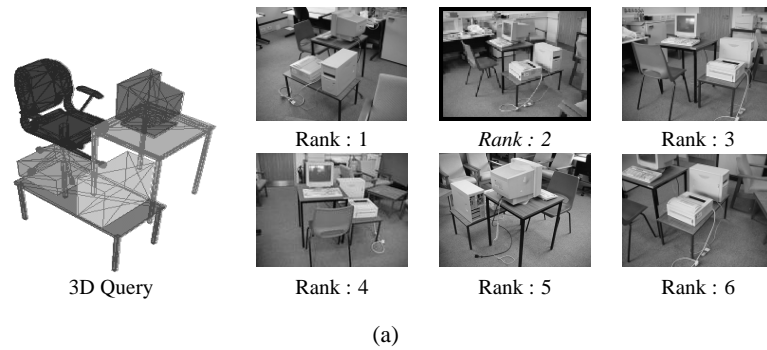
4. Conclusion

The paper has presented approaches for object-based multi-attribute image search using 3D queries and query refinement. Image description is built on an iconic representation of the scene. A 3D query can be refined by specifying images representing relevant 2D views. It has been shown that retrieval effectiveness increases with the quality of data contained in the image descriptor. It has also been shown that the proposed relevance feedback approach enhances retrieval precision.

A possible future direction of research would be to optimise the search for the relevant 3D viewpoint to match a 2D database image. This would reduce the time required for query processing and descriptor extraction. Another direction for further research is toward a better object detection technique. A technique capable of interpreting occlusion would enable the addition of depth relationship to the image descriptor. Finally, the way relative spatial location is represented requires optimizing. The current object-centered coordinates system will result in many redundant relations when processing cluttered images. Solutions from the field of spatial databases are under study [12]. Also, the concurrent processing of descriptor attributes is under investigation. The aim is to minimise the effect of poor shape identification quality on query matching.

References

- [1] M. Abdel-Mottaleb, N. Dimitrova, L. Agnihori, S. Dagtass, S. Jeannin, S. Krishnamachari, T. McGee, and G. Vaithilingam. MPEG-7: a Content Description Standard Beyond Compression. *42nd Midwest Symposium on Circuits and Systems*, 2:770–777, 2000.
- [2] A. D. Bimbo, M. Campanai, and P. Nesi. A Three-Dimensional Iconic Environment for Image Database Querying. *IEEE Transactions on Software Engineering*, 19(10):997–1011, October 1993.
- [3] M. Brendan. *Learning to Recognise 3D Objects from 2D Intensity Images*. PhD thesis, James Cook University of North Queensland, Australia, 1995.
- [4] R. Brunelli and O. Mich. Image Retrieval by Examples. *IEEE Transactions on Multimedia*, 2(3):164–171, September 2000.
- [5] I. Cox, M. Miller, T. Minka, T. Papatomas, and P. Yianilos. The Bayesian Image Retrieval System, PicHunter: Theory, Implementation and Psychological Experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, January 2000.







	Rank : 1 Object : 100% Location : 97% Texture : 100% Colour : 90% Average : 98%		Rank : 2 Object : 100% Location : 95% Texture : 100% Colour : 67% Average : 96%
	Rank : 3 Object : 100% Location : 82% Texture : 100% Colour : 65% Average : 92%		Rank : 4 Object : 100% Location : 63% Texture : 100% Colour : 83% Average : 88%

Figure 6. Illustrative example of query refinements. (a) Initial query and its results. (b) Results after the image ranked second, in the result of the initial query, is selected as relevant feedback. The percentages show the similarity of each descriptor attribute between the query and the returned image. The average similarity score is calculated using Equation 3.

- [6] V. N. Gudivada and V. Raghavan. Picture Retrieval Systems: A Unified Perspective and Research Issues. Technical report, Department of Computer Science, Ohio University, 1995.
- [7] B. Lamiroy. *Reconnaissance et Modélisation d'objets 3D à l'aide d'invariants projectifs et affines*. PhD thesis, Institut National Polytechnique de Grenoble, 1998.
- [8] D. Lowe. Three-Dimensional Object Recognition from Single Two-Dimensional Images. *Artificial Intelligence*, 31:355–395, 1987.
- [9] F. Mokhtarian and S. Abbasi. Automatic Selection of Optimal Views in Multi-view Object Recognition. *Proceedings British Machine Vision Conference*, 1:272–281, 2000.
- [10] H. Muller, W. Muller, S. Marchan-Maillet, and T. Pun. Strategies for Positive and Negative Relevance Feedback in Image Retrieval. *Proceedings of 15th IEEE International Conference on Pattern Recognition*, 1:1043–1046, 2000.
- [11] Y. Rui, T. Huang, and S. Mehrotra. Content-based Image Retrieval With Relevance Feedback in MARS. *Proceedings of IEEE International Conference on Image Processing*, 2:815–818, 1997.
- [12] S. Shekhar, S. Chawla, S. Ravada, A. Fetterer, X. Liu, and C. Lu. Spatial databases - accomplishments and research needs. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):45–55, 1999.
- [13] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, December 2000.
- [14] J. P. Turcat, C. C. Chibelushi, and A. A. Low. Comparative Assessment of Hough Transform Techniques for Image Retrieval Using Approximate Shape Queries. *Proceedings of the IASTED International Conference on Visualization, Imaging and Image Processing*, pages 527–531, 2001.
- [15] J. Z. Wang, J. Li, and G. Wiederhold. “SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries,” *Lecture Notes in Computer Science. Special Issue on Advances in Visual Information Systems*, Robert Laurini (ed.), 1929:360–371, November 2000.