

# Using an Artificial Imagination for Texture Retrieval

Bart Thomee      Mark J. Huiskes      Erwin Bakker      Michael S. Lew  
*LIACS Media Lab, Leiden University*  
*{bthomee, markh, erwin, mlew}@liacs.nl*

## Abstract

*Our goal is to determine if artificially imagined or synthesized images can be beneficial to interactive visual search. We present a novel approach for using artificially imagined images in relevance feedback. Since the search engine constructs the synthetic images itself, any feedback given by the user on these images allows it to obtain a better understanding of what the user is looking for than it would from feedback on database images alone. We evaluated and compared our image synthesis approach with a normal Rocchio-based system on a well-known texture database with real users.*

## 1. Introduction

The definition of "imagination" according to the Merriam-Webster dictionary is:

**imagination:** the act or power of forming a mental image of something not present to the senses or never before wholly perceived in reality.

This definition of imagination forms the foundation of a paradigm we call *artificial imagination*. Analogous to the concept of artificial intelligence, where the computer is given the ability to be 'intelligent', we intend to give the computer the ability to 'imagine', where the meaning directly ties into our ability to synthesize images of objects or of the world which do not have to conform with reality.

Another important analogy with artificial intelligence is the perspective of intelligence lying on a continuum from a simple game playing program to the highest level of intelligence which arguably would be human intelligence. Similarly, artificial imagination also lies on a continuum where the highest level may be the human imagination, and the lowest level might be a simple random image generator.

Imagination in regards to synthesis has been given significant attention in the field of computer graphics where synthesized imagery is the norm. It is important to

note that imagined images need not have any functional usage – they could simply be artistic for example. In this paper we are not only looking at possibilities for synthesis but also exploring whether the imagined images are *useful* for which we utilize the images in the context of content-based image retrieval.

Content-based image retrieval [1-3,9-16] systems seek to use the pictorial information of the images in the search process. Interactive search, or relevance feedback [1-3], has been posed as a promising solution for obtaining improved retrieval results, as the search engine can discover through interaction with the user what kind of image he is looking for.

In the typical relevance feedback paradigm, the images which are shown to the user are limited to the ones within the database. The novel question we pose and attempt to quantitatively answer is "*Can artificially imagined or synthesized images be beneficial in the search process?*". The idea is that since the search engine has constructed the synthetic images itself, any feedback given by the user on these images allows it to obtain a better understanding of what the user is looking for than it would from feedback on database images alone. As an example, envisage a situation where the search engine is not sure whether the user is searching for an image containing cows, or for one containing sheep; the ability to synthesize two images where one shows cows and the other shows sheep will give the search engine more clarity of the user's interests once it has received feedback on them.

In this paper we construct a novel algorithm for using synthesized images in the search process and present the results of human user experiments on a collection of 3000 texture images. Our goal was to measure the effectiveness of the synthesized images in the relevance feedback process. Therefore, we implemented 2 algorithms, a standard algorithm and an enhancement of the standard algorithm where synthesized images would be introduced. For the standard algorithm, we chose the well-known Rocchio [4] method. For the test database, we used the Ponce Texture Database [7] because it is standardized, well-known, easily available, and considered to be challenging by the texture retrieval community. For the

features, we used the MPEG-7 homogeneous texture descriptor [5], which quantitatively characterizes texture regions by calculating the mean and standard deviation of the image intensity and the mean energy and energy deviation from a set of frequency channels. These frequency channels are partitioned along angular and radial directions. Similarity between two images  $x$  and  $y$  is measured by calculating the rotation-invariant distance between their  $k$ -dimensional feature vectors:

$$d(x, y) = \min_m \left\{ \sum_k \left| \frac{x_{k|m\phi} - y_k}{\alpha_k} \right| \mid m = 0, \dots, 5 \right\}$$

where  $\alpha_k$  are normalization values for the particular image database and  $x_{k|m\phi}$  are the angular-shifted versions of the reference image vector  $x$ , with  $\phi = 30$  degrees.

## 2. Relevance Feedback

Relevance feedback was initially developed to improve document retrieval [1,4]. Under this paradigm, the retrieval system presents a ranked set of images relevant to the user's initial query and asks the user for feedback: the user indicates which images he finds relevant (positive) and which irrelevant (negative). The system then uses the feedback to compose an improved set of results and presents them to the user. The advantage of this approach over a single (keyword-based) query is clear: through interaction the search engine can better determine what the user is looking for.

Over the years different algorithms have been proposed for composing an improved set of results, ranging from traditional algorithms such as query point movement (e.g. [4]) and feature reweighing (e.g. [12]), to machine learning algorithms such as artificial neural networks (e.g. [13]), support vector machines (e.g. [14]) and decision trees (e.g. [15]). In our new approach we currently focus on Rocchio's well-known query point movement algorithm [4]. Rocchio's method takes the current query point in feature space and uses the feedback given by the user to move it towards the positive images and away from the negative images:

$$q_t = \alpha q_{t-1} + \beta \left( \frac{1}{n^+} \sum_{i \in S_t^+} x_i \right) - \gamma \left( \frac{1}{n^-} \sum_{i \in S_t^-} x_i \right)$$

where  $S_t^+, S_t^- \subset \{1, \dots, n\}$  are the index sets of the positive and negative points, respectively, at iteration  $t$ ;  $n^+$  and  $n^-$  the size of the positive and negative index sets, respectively;  $q_t$  and  $q_{t-1}$  are the new and current query points, and  $\alpha, \beta$  and  $\gamma$  are suitable constants.

## 3. Synthetic Imagery

When we are learning new visual concepts, we often construct new mental images which are synthesized from our imagination and which serve the purpose of clarifying

or helping us understand the primary features which are associated with the visual concept. Our approach can be seen as the digital analogy of our own visual imagination. The search engine constructs mental (synthetic) images in order to clarify the primary features that are associated with the image that the user is looking for; ideally the synthetic images have a high relevance to the search query. By asking for feedback on these constructed examples, our assumption is that it benefits the retrieval results in three ways: (i) the user can select more relevant images, giving the search engine more image information to search with, (ii) as the synthetic images are created by the search engine, any positive and/or negative feedback on those images will give the system a better understanding of the user's interests (more than it would from feedback on images from the database), (iii) the amount of iterations necessary to satisfy the user is reduced, since the target image(s) will be found sooner.

Although the MPEG-7 feature space used by the relevance feedback is highly appropriate for finding similar texture images, it is not suitable for the synthesis of images as image reconstruction is not well-defined: there is not a one-on-one mapping from a feature vector to an image. To solve this matter, we introduce a second feature space that is also associated with every image: we use the feature space  $F$  formed by taking the Karhunen-Loeve Transform (KLT, e.g. [6]) of each image in the database and using  $l$  coefficients from the KLT representation of the dataset, thus  $F = \{x_i \in \mathbb{R}^l \mid i=1, \dots, n\}$ . Given a feature vector  $x$  containing the coefficients, a new image  $s$  may be synthesized through  $s = Wx + \mu$ , where the columns of  $W$  represent the eigenvectors of the KLT transform, and  $\mu$  is the average image used for de-normalization.

We base the synthesis on images on which feedback has been given and their locations in KLT feature space: our approach is to examine their relevance in relation to the other database images and attempt to find locations in feature space that will clarify uncertain or emphasize important features when feedback on them is given by the user. An approach inspired by evolutionary algorithms is very well suited for this task, as such algorithms take a population and evolve it towards better solutions; in our situation this leads to evolving a population of images towards a solution ideally containing all images of interest to the user. Our algorithm called *Synthetic*, like an evolutionary algorithm, has four steps: starting population, crossover, mutation and survival, and after the last step the algorithm loops back to the second step. Our algorithm for determining locations in feature space that are likely to result in suitable images when synthesized is defined as follows:

1. *Starting population.* Let  $R_0 \subset \{1, \dots, n\}$  be a random subset of images from the database of size  $N_R \leq N$  and show them to the user. The positive feedback gives the initial population  $S_1^+$ .

2. *Crossover.* In this step we sub-sample  $S_t^+$ , the positive examples from iteration  $t$ . Consider sets  $A_j \in 2^{S_t^+} : |A_j| = 2$  consisting of at least two positive images from the feedback, with  $j = 1, \dots, N_j$  and  $N_j = 2^n - 1 - n$ . Each of these sets gives a new query point  $c_j = \frac{1}{|A_j|} \sum_{i \in A_j} x_i$ . For step 3 we use the set  $C = \{c_1, \dots, c_{N_j}\}$ .
3. *Mutation.* We perturb the points generated in the crossover step by two mechanisms. We first use the negative feedback of iteration  $t$ , similar to how Rocchio uses negative examples to push away points towards a more favorable area in feature space, and then we introduce random elements. Let  $\bar{x}^- = \frac{1}{n} \sum_{i \in S_t^-} x_i$  be the mean of the negative feedback. We use this for each query point in  $C$  to obtain mutation points  $c_j' = (1 - \delta)c_j + \delta\bar{x}^- + r_j$ , with  $0 \leq \delta \leq 1$  and  $r_j$  small random values. Finally, we synthesize images from the mutation points.
4. *Survival.* The synthesized images, together with the images from Rocchio ranking  $R_t$ , are presented to the user and we let the user determine which images are most relevant and survive into the next population  $S_{t+1}^+$ .

In Figures 1 and 2 we give an illustration of the usefulness of synthetic images: if a user is looking for crosshatched images, but only an image containing horizontal lines and an image containing vertical lines are shown on screen, the algorithm is able to synthesize a crosshatched image (Figure 1), or if the user is looking for a color adjusted version of an image the algorithm can synthesize an appropriate image (Figure 2). We expect the search to be steered more quickly into the correct direction if the synthetic images are used in queries.

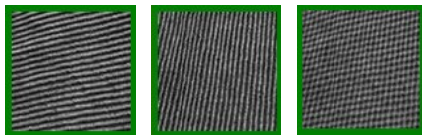


Figure 1. Synthesis of a crosshatched image (right) from images containing horizontal lines (left) and vertical lines (middle).



Figure 2. Synthesis of a color adjusted image (right) from textures containing the pattern (left) and the adjustment color (right).

Unlike general images, textures do not necessarily have any semantic meaning and modifications to a texture generally result in a new valid texture, making our approach appropriate for this category of image search. As our method has no knowledge of semantic concepts (e.g. it cannot synthesize an image containing a dog on a beach given an image of a beach and another with a dog), it is not necessarily suitable for general image search. However, new semantic image synthesis techniques (e.g. [17]) appear very promising and we will look into them in the near future.

## 4. Experiments

The test database was composed of 3000 images taken from the Ponce Texture Database as shown in Figure 3.

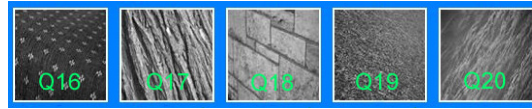


Figure 3. Example texture images.

The 3000 image test set included the 1000 original textures, and 2000 images which were either randomly rotated or scaled from the original versions by up to 15%, resulting in a set of textures that vary in 3D perspective, shape and orientation. The images are represented by both texture MPEG-7 features and are additionally associated with KLT coefficients. Similarity search is performed by applying the Rocchio method on the MPEG-7 features, whereas for the synthesis the KLT features are used.

To minimize bias, we implemented blind user testing where the students were not aware which version they were using: Normal (Rocchio) or Synthetic (Rocchio, Synthesized Images). 30 Students were assigned 6 queries each: 3 text queries and 3 image queries. In total, there were 20 different text queries and 20 image queries, which were randomly assigned to the users.

With an image query, the user was given an example image and was asked to find similar images. With a text query, the user was given a texture category and was asked to find images that would fit that particular category, e.g. ‘marble’, ‘tree bark’. The categories were selected from Getty Images stock photography keywords [8], the Ponce texture classes, and several were suggested by the users, see Table 1.

Table 1. Texture categories

Ponce	Getty Images	User-based
Brick - fine	Horizontal composition	Old stone
Brick - coarse	Vertical composition	Curvy/organic look
Tree - bark	Abstract	Wavy
Tree - wood	Sparse	Diamond pattern on fabric
Marble	Square/rectangular	Tartan pattern
Fabric	Rustic/rural/country	Linear
	Man made/synthetic	
	Nature/natural	

The users were asked to record the number of relevant images at iterations 1, 4, 8 and 12. Per iteration 15 images

were shown on screen; in the Normal algorithm, all images shown were from the database; in the Synthetic algorithm, some of the images were synthetic (given  $n$  positive images:  $2^n - 1 - n$ ) and the remaining ones from the database. At iterations 1, 4, 8 and 12 only the top ranking database images were shown in order to enable a fair comparison between the two methods. The most relevant results to stock photography are shown in Figure 4 and the overall average text and image precision results in Figure 5. Here, we consider precision to be the percentage of relevant results shown on screen per iteration.

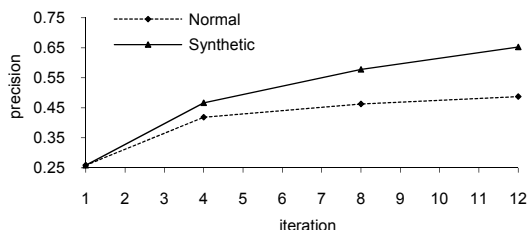


Figure 4. Getty Images text query results

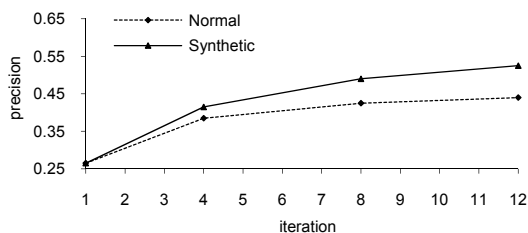


Figure 5. Average text and image query results

As can be seen, the Synthetic method compared with the Normal method shows a constant improvement in the amount of relevant images shown on screen.

## 5. Conclusion

We have developed a new interactive search approach inspired by evolutionary algorithms. Based on user feedback, our method synthesizes images that are constructed to clear uncertain or emphasize important image features; from additional feedback on these images, the search engine is better able to determine what the user is looking for. We performed user experiments on a well-known texture database and the results indicate that the inclusion of synthetic imagery leads to an improvement of the amount of relevant images shown to the user: for the image queries, at iteration 12 the Normal method had a precision 0.40, whereas the Synthetic method had a precision of 0.46, an improvement of 15%; for the text queries, Normal had a precision of 0.48 and Synthetic 0.59, an improvement of 23%; in both cases the Synthetic method outperformed the Normal method. In the future we will focus on more advanced search algorithms, such as support vector machines, as well as other approaches for the creation of synthetic imagery, especially for the case of general images as opposed to textures only.

## Acknowledgements

Leiden University and NWO BSIK/BRICKS supported this research under grant #642.066.603

## References

- [1] X.S. Zhou, T.S. Huang. Relevance feedback in image retrieval: a comprehensive review. *ACM Multimedia Systems Journal*: 536-544, 2003.
- [2] M.S. Lew, N. Sebe, C. Djeraba and R. Jain. Content-based multimedia information retrieval: state of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications* 2(1): 1-19, 2006.
- [3] G. Giacinto. A nearest-neighbor approach to relevance feedback in content-based retrieval. *Proceedings of the ACM International Conference on Image and Video Retrieval*: 456-463, 2007.
- [4] J.J. Rocchio. Relevance feedback in information retrieval. *The Smart Retrieval System: Experiments in Automatic Document Processing*. G. Salton, Ed. Prentice-Hall, 1971.
- [5] Y. M. Ro, M. Kim, H. K. Kang, B. S. Manjunath, and J. Kim. MPEG-7 homogeneous texture descriptor. *ETRI Journal* 23: 41-51, 2001.
- [6] C. Therrien. *Decision, estimation, and classification*. John Wiley & Sons, 1989.
- [7] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(8): 1265-1278, 2005.
- [8] Getty Images, <http://www.gettyimages.com>.
- [9] J. Fan, Y. Gao, H. Luo, and G. Xu. Automatic image annotation by using concept-sensitive salient objects for image content representation. *Proceedings of ACM SIGIR conference on Research and development in information retrieval*: 361-368, 2004.
- [10] Y. Gao and J. Fan. Semantic image classification with hierarchical feature subset selection. *Proceedings of ACM SIGMM international workshop on Multimedia information retrieval*: 135-142, 2005.
- [11] G. Carneiro and N. Vasconcelos. A database centric view of semantic image annotation and retrieval. *Proceedings of ACM SIGIR conference on Research and development in information retrieval*: 559-566, 2005.
- [12] J. Huang, S.R. Kumar, and M. Metra. Combining supervised learning with color correlograms for content-based image retrieval. *Proceedings of ACM international conference on Multimedia*: 325-334, 1997.
- [13] H.K. Lee and S.I. Yoo. A neural network-based image retrieval using nonlinear combination of heterogeneous features. *International Journal of Computational Intelligence and Applications* 1(2): 137-149, 2001.
- [14] S. Tong and E. Chang. Support Vector Machine active learning for image retrieval. *Proceedings of ACM international conference on Multimedia*: 107-118, 2001.
- [15] S. D. MacArthur, C. E. Brodley, and C.-R. Shyu. Relevance feedback decision trees in content-based image retrieval. *Proceedings of IEEE Workshop on Content-Based Access of Image and Video Libraries*: 68-72, 2000.
- [16] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*: 644-655, 1998.
- [17] J. Hayes and A.A. Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics* 26(3), 2007.