TEXRET: A Texture Retrieval System based on Soft-Computing Technologies

Javier Ruiz-del-Solar, Pablo Navarrete, Patricio Parada
Department of Electrical Engineering, Universidad de Chile, Santiago, CHILE.
Email: {jruizd, pnavarre, pparada}@cec.uchile.cl

To address the problem of texture retrieval from image databases the TEXRET system was developed. TEXRET (TEXture RETrieval) uses soft-computing technologies to allow an interactive communication with the user. TEXRET main features are: (i) direct access from the Internet, (ii) high interactivity, (iii) texture retrieval using human-like or fuzzy description of the textures, (iv) content-based texture retrieval using user-feedback, and (v) generation of the requested textures when these are not found in the database, which allows a growing of the database.

1. Introduction

The aim of the presented work is the construction of the TEXRET system, a texture retrieval system based on soft-computing technologies. The importance of this kind of systems is increasing due to the massive access to digital image databases, which also demand the existence of systems that can understand human high-level requests. The TEXRET system has the following features: (i) direct access from the Internet, (ii) high interactivity, (iii) texture retrieval using human-like or fuzzy description of the textures, (iv) content-based texture retrieval using user-feedback, and (v) synthesis or generation of the requested textures when these are not found in the database, which allows a growing of the database. This paper shows the structure of TEXRET, and explains how a non-expert user can employ each of the system features. The article is structured as follows. TEXRET is outlined in section 2. In section 3 is described the interaction process between the user and TEXRET. In sections 4, 5 and 6 are described the TEXRET main subsystems. Finally, in section 7 some conclusions are given.

2. The TEXRET System

TEXRET, whose block diagram is shown in Figure 1, is made of three main subsystems: *Query-based Texture Retrieval* (A), *Interactive Content-based Texture Retrieval* (B) and *Interactive Texture Generation* (C). These subsystems do not work in parallel but sequentially. First, the user interacts with subsystem A, performing the texture retrieval using a human-like description of the textures. If this process fails, then she interacts with subsystem B, carrying out a content-based texture retrieval that utilizes user-feedback. Finally, if both retrieval subsystems have failed, she has the possibility of performing an interactive generation of the requested texture.

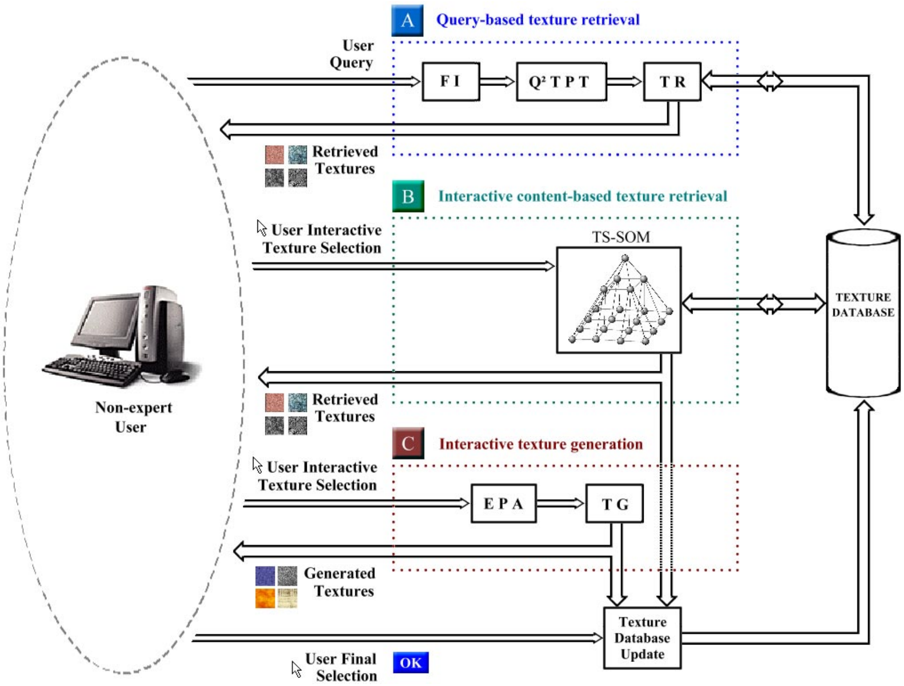

FIGURE 1. Block Diagram of TEXRET.

2.1 Query-based Texture Retrieval

The stages of this subsystem are *FI* (Fuzzy Interface), Q^2TPT (Qualitative to Quantitative Textural Properties Transformation), and *TR* (Texture Retrieval). The on-line phase of the texture retrieval process works as follows: A human user makes a query of a texture using a subjective, linguistic or human-like texture description. The FI module enters this description into the system using a fuzzy representation of it. The Q^2TPT module interprets the query and translates it into a quantitative texture description that is implemented using Tamura Descriptors [7]. This quantitative description is used in the TR module to search the texture in the database (see description in [5]). The subjective or human-like texture description that TEXRET accepts was determined by a psychological study described in [6].

2.2. Interactive Content-based Texture Retrieval

There are many forms in order to develop an efficient system for content-based image retrieval. We have developed our system based in the PicSOM implementation [2]. That system uses a tree-structured self-organizing map (TS-SOM [3]) for auto-organizing similar feature vectors (Tamura Descriptors in our case) together in the database, and user-feedback to search the requested texture using these vectors. Under this approach, called *relevance feedback*, previous human-computer interactions are employed to refine subsequent queries, which iteratively approximate the wishes of the user [2]. In order to know in which part of the TS-SOM the requested texture is mapped, TEXRET asks the user to select textures, which she consider are similar to the requested one, from a given set of textures. Then, the system shows the user new textures (from the database), which have neighbor positions in the TS-SOM (their corresponding descriptors), respect to the ones selected by the user. The user and TEXRET iterate until the interaction process converges.

2.3. Interactive Texture Generation

The stages of this subsystem are *EPA* (Evolutionary Parameter Adjustment) and TG (Texture Generation). In the case that the requested texture is not found in the database, i.e. the retrieved textures does not satisfy previous queries or selections, the user can choose the automatic generation of it. The TG module generates the texture using an Autoregressive model [1], whose parameters are calculated from the retrieved textures. Within this framework other generation models, as for example Markov Random Fields, can be used without problems (see explanation in [4]). As a result of the generation process a set of textures is presented to the user. If she considers that one of the generated textures satisfies her query, the process finishes here. If not, the user enters into an iterative process. The iterative generation of the textures is implemented in the EPA module.

3. User-TEXRET Interaction

TEXRET has many possibilities to succeed in retrieving a texture within a large database. In order to allow that a non-expert user can employ all these possibilities, a user-TEXRET interaction procedure was designed (see figure 2). Within this procedure, steps 1, 2 and 3 correspond to the query-based texture retrieval process. Steps 4, 5 and 6 correspond to the interactive content-based texture retrieval process. Finally, steps 7, 8 and 9 correspond to the interactive texture generation process. The steps indicated by (*) correspond to actions performed by TEXRET. All the rest correspond to user actions.

The texture retrieval procedure starts in step 1. The user enters a description of a texture into the system using a qualitative representation of it. After processing this information, in step 2, the system gives the user a set of retrieved textures. In step 3 the user has to take a decision on the retrieved textures. If the requested texture is one of the retrieved textures, the user exits the system (3.a). If no similar textures were found two things could happen: the user is tired and she exits the system (3.b.1) or the user is not tired and she goes into step 1 again to redefine her query (3.b.2). Finally, if similar textures were found (3.c), the process continues with step 4.

In step 4 the user selects similar textures within the retrieved ones. In step 5 this information is processed, the system searches for textures similar to those selected (using the TS-SOM) and gives the user a new set of retrieved textures. In step 6 the user has to take a decision on the retrieved textures. If one of the retrieved textures is the requested texture, then the database is updated (the database quantitative description associated to the requested texture is refined considering the initial textural query), and the user exits the system (6.a). If no similar textures were found (6.b), then the user must decide whether or not to go into the texture generation subsystem. If not, the user exits the system (6.b.1), and in other case the process continues with step 7 (6.b.2). Finally, if similar textures were found (6.c), the user goes into step 4 again.

In step 7 the user must select the most similar textures within the generated or retrieved textures. In step 8, the system generates a new set of textures. In step 9 the user has to take a decision on the generated textures. If the user finds the generation was successful, then the database is updated and she exits the system (9.a). If no similar textures were generated two things could happen: the user is tired and she exits the system (9.b.1), or the user is not tired and she wants another texture generation iteration, going to step 8 (9.b.2). In this last case, TEXRET should automatically generate a texture-set seed for the next generation iteration. Finally, if similar textures were found (9.c), the user goes into step 7 again.

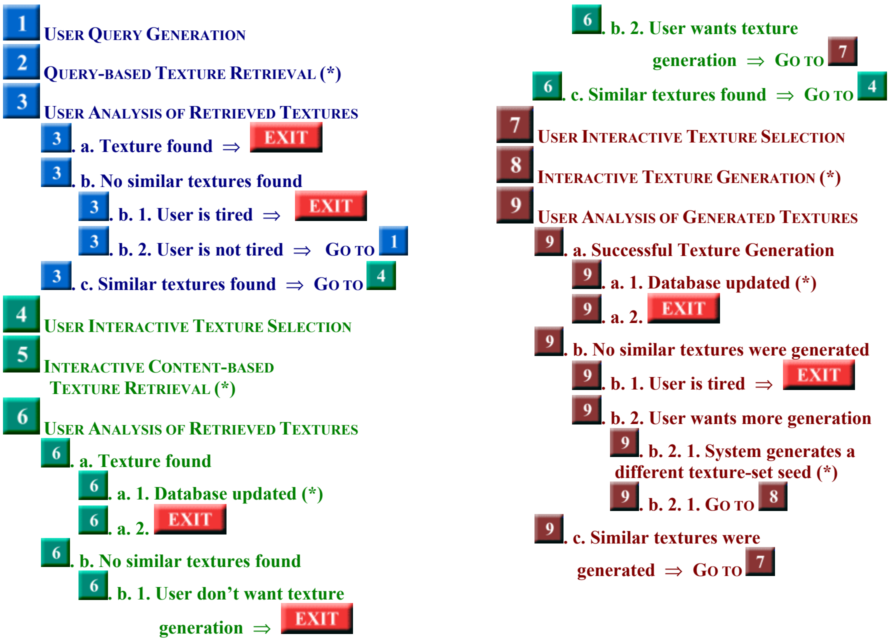

FIGURE 2. User-TEXRET interaction procedure. The actions performed by TEXRET are indicated by (*).

4. Query-based Texture Retrieval

As mentioned, the Q^2TPT module should translate a human-like or qualitative texture description into a quantitative or mathematical description. The quantitative descriptors or features we use are the Tamura ones (coarseness, directionality, contrast, line-likeness, regularity and roughness). Tamura gave a mathematical procedure to calculate them. A description of our own implementation can be found in [5].

The human-like texture description that TEXRET accepts was determined by a psychological study, described in [6]. This study consisted in three surveys in which we asked directly to humans how they describe textures. As a result of this study, we found a set of 12 adjective pairs to be used in our system: "soft / rough", "fine / coarse", "natural / artificial", "clear / dark", "homogeneous / non homogeneous", "defined / diffuse", "regular / irregular", "symmetrical / non symmetrical", "simple / complex", "flat / non flat", "geometrical / non geometrical", and "with lines / without lines". It should be mentioned that this study was performed in Chile, using Spanish words. However, each of the used adjectives has a direct translation into English. For this reason we believe that the results obtained in the Spanish language can be directly applied to English.

To that point a question is still open. How can the Q^2TPT module make the translation from qualitative to quantitative description of textures? That is, how can it relate the both mentioned kinds of texture descriptions? To solve this problem we took 100 textures and we obtained both kinds of descriptors from them. Then, we trained a neural network to relate both kinds of descriptors (see description in [5]). This network performs a kind of association between these descriptors. The quantitative description of the 100 textures was directly obtained using the Tamura equations. To obtain the human-like description of the textures was necessary to ask humans again. This was the subject of a last survey, performed in the Internet (see http://cipres.cec.uchile.cl/~pnavarre/).

5. Interactive Content-based Texture Retrieval

As mentioned TEXRET uses a tree-structured self-organizing map (TS-SOM [3]) for auto-organizing similar feature vectors (Tamura Descriptors in our case) together in the database, and user-feedback to search the requested texture using these vectors. In order to know in which part of the TS-SOM the requested texture is mapped, TEXRET asks the user to select textures, which she consider are similar to the requested one, from a given set of textures. Then, it shows the user new textures (from the database), which have neighbor positions in the TS-SOM (their corresponding descriptors), respect to the ones selected by the user. The user and TEXRET iterate until the interaction process converges. The specific mechanism employed is the following: a so-called selection matrix is updated using the user-feedback (textures selected or not selected), by increasing/decreasing those elements corresponding to the nodes on the TS-SOM network in which the texture images selected/ignored are respectively located. A low-pass filter is then applied on the selection matrix in order to spread the positive/negative responses. The new set of retrieved textures is selected by choosing images whose nodes have the highest values in the selection matrix. The hypothesis assumed under this approach is that the Tamura Descriptors together with the similarity measure used in the off-line TS-SOM training (usually

Euclidean metric) are consistent with the user-criteria utilized for the selection of similar texture images. If this happened, then the highest values in the selection matrix should move the searching process towards the zone in the TS-SOM map in which the requested texture is located.

The TS-SOM training procedure begins with the highest map and it continues with the lower ones. As in any Self-Organizing Map, each node (or neuron) has a weight-vector that represents a certain area of the high-dimensional input space. In addition, the TS-SOM structure allows connections between each node (the parent) and four nodes in the next level (the children). The TS-SOM is trained using the Tamura descriptors of the database textures. In order to determine the weight-vectors of all nodes at all TS-SOM levels, using the training samples (Tamura descriptors), the TS-SOM algorithm carries out the following iteration procedure:

0) In the first level (that has one node) the weight-vector is the mean of all the training samples.
1) In the other levels, weight-vectors of the previous level are copied into the children of the current level (initialization).
2) The centroid associated to each node is determined as the mean of the closest training samples. The set of closest samples is found based on limited search, that is comparing the distance of each sample to the node with the distance of each sample to a selected set of nodes. This set is composed by the children nodes of the parent (best matching unit) of the previous level and the children of the parent's neighbors.
3) The new weight-vectors are calculated as the mean of the neighbor centroids, weighted by the number of closest samples (of each node) and a kernel function that gives more importance to the closest neighbors (usually a Gaussian function).
4) If the weight-vectors do not change more than a given value, then the procedure continues with the next TS-SOM level in step 1). If not, then it goes into step 2).

6. Interactive Texture Generation

Interactive Texture Generation or Synthesis is a dynamic process, where the user plays an important role in the synthesis of textures. This approach follows the "synthesis-by-analysis" paradigm, which means that the system analyzes a texture-sample and then it creates a new image. In the proposed subsystem (see block diagram in Figure 3) a set of generated textures is presented to the user at each iteration. She selects a subset of textures that better achieve her requirements. This information is used to adjust the parameters of the texture-generation model. The subsystem and the user iterate until the required texture is generated. What really happens is that the user performs an interactive adjustment of the texture-generation model parameters by using Interactive Genetic Algorithms [4] (IGA). IGA can be defined as genetic algorithms that do not use a standard fitness function but a user's choice as fitness criterion.

As in any genetic algorithm application, the parameters of the model are packed into a chromosome (array), whose size depends on the texture-generation model being used. The texture generation process starts with the result of the retrieval process. The first vector parameters population is constructed using the parameter estimation performed by the *Texture Analysis* module (see block diagram in figure 3). Then, these parameters are coded into a vector by the *Parameter Codification* module. Thereafter the subsystem creates *N* chromosomes using mutation and crossover operations in the *Population Generation* module.

After these operations have been carried out, the *Texture Synthesis* module builds *N* texture images using the parameter vectors recently generated. Then, these textures are shown to the user. In that moment, the user can interact in three different ways with the system(see interaction procedure in figure 2).

The user selection of the textures is the only interactive part in the evolutionary procedure, and replaces the usual fitness selection. By selecting the texture images the corresponding chromosomes are selected as well. From these *M* parameter vectors, crossover and mutation operators create *N* new chromosomes. After that, *N* new texture images are generated (*Texture Synthesis* module) and presented to the user, who again has the same three options before described to interact with the system. The whole procedure is repeated until the generated textures satisfy the user wishes or after the user stop the generation process. A detailed description of the whole process can be found in [4].

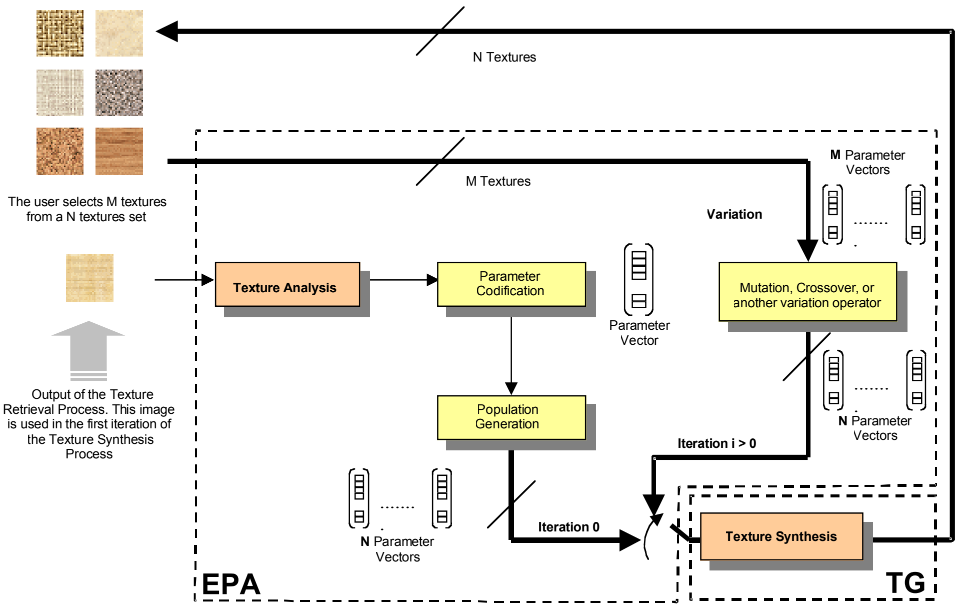

FIGURE 3. Block Diagram of the Interactive Texture Synthesis.

7. Conclusions

Providing the facility of a user-friendly texture retrieval system comes out to be a very complex task. The TEXRET project corresponds to an effort in this direction. To address the problem of texture retrieval from image databases TEXRET uses a neural network to translate user queries in the form of linguistic texture descriptions into quantitative texture descriptions. The corresponding quantitative textural features (Tamura features) are computed and used to retrieve the texture images from a database. If this process fails, the textures can be interactively retrieved using self-organizing maps. In particular, the PicSOM approach is utilized. The interactive retrieval process allows also an update of the texture database. When the requested texture is found, its quantitative description, the one stored in the database, is refined considering the initial textural query. Finally, if the requested texture can not be found in the database, i.e. both retrieval subsystems fail, the user can choose to interactively generate it. The generation of the requested textures when these are not found in the database is one of the main TEXRET characteristics. In order to satisfy the requirements of the user an interactive genetic algorithm synthesizes missing textures. When the user has interactively found her texture, the database can be expanded and a new pair of data is supplied as well (the user-query and the texture selected by her). From this, even a small-at-the-beginning texture database can grow from the very beginning on.

ACKNOWLEDGEMENTS

This research was supported by the DID (U. de Chile) under Project ENL-2001/11 and by the join "Program of Scientific Cooperation" of CONICYT (Chile) and BMBF (Germany).

REFERENCES

[1] R. Chellappa, R. Kashyap, "Texture Synthesis Using 2D-Noncausal Autoregressive Models", *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. ASSP - 33, No. 1, February 1985, pp. 194 - 203.

[2] J. Laaksonen, M. Koskela, S. Laakso and E. Oja, "PicSOM – content-based image retrieval with self-organizing maps", *Pattern Recognition Letters*, Vol. 21, pp. 1199-1207, 2000.

[3] P. Koikkalainen, E. Oja, "Self-organizing hierarchical feature maps", *Proc. of 1990 Int. Joint Conf. on Neural Networks*, Vol. II. IEEE, INNS, San Diego, CA, 1990.

[4] P. Parada, J. Ruiz-del-Solar, M. Köppen, and W. Plagges, "Interactive Texture Synthesis", *Proc. of the 11th Int. Conf. on Image Analysis and Processing ICIAP 2001*, September 2001, pp. 434 - 439.

[5] J.Ruiz-del-Solar, "Neuro-fuzzy System for the Administration of Texture Databases", *Proc. of the XIII Chilean Congress on Electrical Eng.*, Nov. 1999, pp. 55 - 60 (in Spanish).

[6] J. Ruiz-del-Solar, and M. Jochmann, "On Determining Human Description of Textures". *Proc. of the 12th Scandinavian Conf. on Image Analysis – SCIA 2001*, Bergen, Norway, June 2001, pp. 288 – 294.

[7] H. Tamura, S. Mori, and T. Yamawaki, "Textural Features Corresponding to Visual Perception", *IEEE Trans. on System, Man and Cybernetics*, Vol. SMC – 8, No. 6, July 1978, pp. 460 – 472.