Retrieval of Satellite Cloud Imagery Based on Subjective Similarity

Asanobu Kitamoto and Mikio Takagi
Institute of Industrial Science, University of Tokyo
7-22-1, Roppongi, Minato-ku, Tokyo, 106 Japan
kitamoto@tkl.iis.u-tokyo.ac.jp

Abstract

The goal of this paper is to present an image database method which allows users to retrieve satellite cloud imagery based on subjective similarity of images. First, we propose a technique for shape representation, a region-based deformable model that belongs to the class of shape decomposition methods. The next step is to convert the decomposition result into a mathematical model using a *hierarchical attributed relational graph*, where a similarity measure is defined as the graph matching cost of two graphs with several weighting factors each indicating its importance in similarity. The most important contribution of this paper is that we formulate a *subjective* similarity measure with a set of optimized weighting factors so that it maximally reflects personal subjective impression of satellite cloud imagery. Finally we evaluated the accuracy of the result by comparing the retrieved results to subjective similarity.

1 Introduction

"Intelligent" interface is the keyword for forthcoming image databases in order to effectively utilize a massive number of images. Here we understand the term "intelligent" as "capable of discovering similar images based on image contents." Although text-based information has played an important role in conventional image databases, it is, in general, still a difficult task to provide content-based access to image databases only through keywords. Thus it is natural to make an image representation model that can describe the domain of image contents where keyword-approach does not work effectively.

We have been in charge of receiving and collecting of data from the NOAA meteorological satellites for several years. The daily amount of data received is very large, a content-based retrieval scheme is required to make effective use of the data. Since a practical solution, such as manual assignment of proper keywords to every image, may be a formidable task due to the large amount of data, the goal is to develop an appropriate image representation model that can powerfully describe this satellite imagery.

Here in this paper, we confine the domain of the term "image contents" as "subjective similarity of satellite cloud imagery[1] with regard to both shape and spatial

[1]Satellite cloud imagery is produced from NOAA meteorological satellite data (Channel 4) through a cloud extraction process.

distribution of cloud patterns." Suppose that a user provides a query image as a retrieval key. Then the task of the system is to retrieve images that convey perceptually similar impressions as the query image does.

From the viewpoint of remote sensing, this image database may be interesting for the shape-based and spatial distribution-based access to satellite data. Although pixel-wise information is important for both land and cloud classification classes, the most remarkable difference between these two classes lies in the importance of the region-wise information for cloud class. The air current, by which an infinite number of patterns are produced, accounts for this property.

From the standpoint of computer vision, on the other hand, cloud patterns look different from such patterns as machine parts or human faces. These kinds of objects are definite, articulated objects. As a consequence, the representation of these objects tend to be domain-specific. On the contrary, it is expected that "purer" patterns like clouds will permit us to explore "more primitive" shapes. Cloud patterns surely posses meaningful meteorological parts; however its shape, in general, gives us less association with its functionality.

This paper is organized as follows. In Section 2, we will propose a region-based deformable model to represent shape information of cloud imagery. Next we will introduce a hierarchical attributed relational graph in Section 3 for representing spatial information of cloud imagery. Section 4 explains the derivation of a subjective similarity measure through optimization of weighting factors. Then Section 5 presents the experimental results on subjective similarity-based retrieval. Finally, Section 6 summarizes our conclusion.

2 Shape Information

2.1 Region-based Deformable Model

Since the shape of objects and parts is of paramount importance in visual recognition, there have been numerous papers on shape representation models. Here we categorize various models into two types of approach – 1) region-based, and 2) contour-based. Contour-based models include such methods as chain code, Fourier descriptor and B-spline; while region-based models include skeleton, shape decomposition and quadtree. Because of the jagged shape of clouds, we take the latter approach, in particular shape decomposition.

Our method is inspired from the method of "snakes" [6]. The class of models originating with "snakes" is called "deformable models", because these models simulate elastic contour or surface. Internal and external forces consists of energy functional of the model, which is iteratively deformed until it reaches the minimal energy level. They offer a reasonable approach to solving problems due to their stability, controllability, and their property of regularizing data gathered over regions of the image. Compared to previous contour-based deformable models, our model is a region-based deformable model for solving shape decomposition.

2.2 Algorithm

The basic figure for our decomposition scheme is *superquadrics* [8], which the following equation describes:

$$\left(\frac{|x|}{a_1}\right)^{\frac{2}{\varepsilon}} + \left(\frac{|y|}{a_2}\right)^{\frac{2}{\varepsilon}} = 1 \tag{1}$$

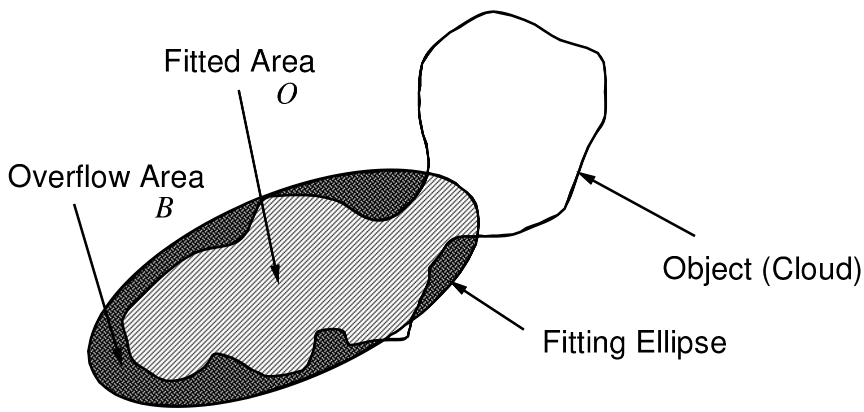

Figure 1: Fitting of an ellipse. The gray area represents the fit region whose area is O, while the black area, the overflow region whose area is B.

It is easily recognized that ellipses are the special case of superquadrics with $\varepsilon = 1$. An ellipse is an attractive figure, since it can approximate elongated objects without any artificial corners. Also respecting its simplicity, we use an ellipse as a component figure for shape decomposition.

Next we formulate the energy function by which the deformable model is guided to an appropriate position. Since our model is region-based, our energy function is as simple:

$$E = O - pB \tag{2}$$

where O is the area of the object inside the fitting ellipse, B is the area of the background where the fitting ellipse grows beyond the object boundary producing overflow area as illustrated in Figure 1. Moreover, p is a penalty constant that controls the degree of overflow of the fitting ellipse. Next we have to find the maximum of E. Note that O works as an expansive force, whereas B works as an contractive force. The stable state is a state where expansive and contractive forces are equalized.

E is a multi-variate function with six degrees of freedom; namely, $(s_x, s_y, \theta, a_1, a_2, \varepsilon)$ for superquadrics. It is a multi-variate optimization — the algorithm we use is a conjugate-gradient method (DFP method). Although the DFP method is weak for global searches, we have to note that we do not need a *global* minimum for the deformable model, but a *local* minimum as pointed out in [5]. However, the initial guess of parameters should be close enough to allow the model to be attracted to an appropriate local minimum. The following is a brief description of the algorithm, and Figure 2 shows a result of decomposition.

1. Initialize parameters using distance transform image (Chamfer distance).
2. Optimize all the parameters using a conjugate-gradient method, and find an optimal decomposition component.
3. Remove all the area included in the optimal component.
4. Start the next search for the remaining part of the object and goto 1.

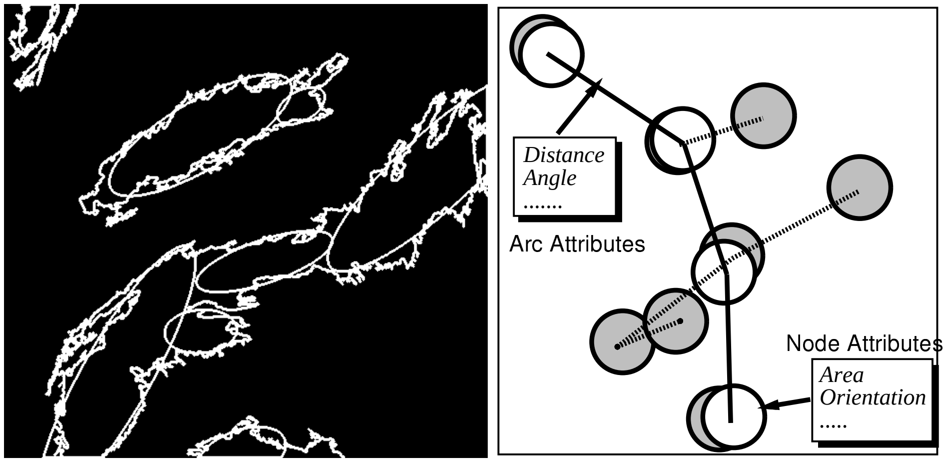

Figure 2: A decomposition result and an hierarchical attributed relational graph. Each cloud region is decomposed into several ellipses, and structured into a HARG. White circles and solid lines correspond to upper level nodes and arcs respectively, whereas gray circles and dotted lines correspond to lower level nodes and arcs.

3 Spatial Information

The decomposition components obtained in Section 2 correspond to what are called “primitives.” Next we describe complex patterns by means of suitable mathematical models – strings, trees, and graphs. Graph models are very powerful with respect to their modelling capacity; their disadvantage is that parsing is complex and inefficient [3]. In this paper, we use one of graph model because of its powerful modelling capacity.

3.1 Hierarchical Attributed Relational Graph

The graph structure we use is a hierarchical version of an attributed relational graph [2, 9], in which nodes and arcs symbolically represents an image, while node and arc attributes describe numerical information of an image. A node represents the position of a cloud region, whereas an arc a relationship between two cloud regions. In this paper, an arc is linked on a “gravity” basis where “gravity” is a heuristic weight force such that mass is replaced with the area of a region. Gravity is calculated between any two regions and a relationship that has maximum gravity among all the links from a node is designated as an arc. Attributes is another important element of the graph; we equip a graph with two types of attributes — scale attribute and shape attributes. The scale attribute describes the size of a node (area) or an arc (length), whereas shape attributes include such attributes as “orientation”, “circularity” and “compactness.”

This graph model is a hierarchical model; the upper level of the graph has the role to showing the global structure of an image, while the lower level of the graph describes the shape of a cloud region using decomposition components. We will make use of this hierarchical structure in Section 5.

3.2 Graph Matching

The similarity measure between two graphs is calculated as a graph matching cost. Matching of a graph ω_1 into a graph ω_2 with the help of inexact matching techniques is a successive transformation of ω_1 with basic transformations — deletion, insertion and substitution of nodes and arcs — until ω_1 becomes *isomorphic* to ω_2.

For each transformation, the calculated matching cost specifies the "difficulty" of the transformation. For example, the matching cost of two attribute vectors is defined as the Mahalanobis distance between two vectors. The summation of matching costs over the all pairs of nodes and arcs produces the final distance between two graphs, in other words "similarity" between two graphs. Note that this similarity is used as similarity between corresponding images.

All the possible pairings of nodes are expanded in the *search space*, and a set of optimal pairings (graph matching) is found for this search space. Here the optimal path is defined as a path that starts from the start node and ends at one of the goal nodes with the least matching cost among all the possible paths. In this paper we search an optimal pairings of nodes and arcs using the *best-first search* [7].

The best first search algorithm manages the queue that keeps all the states in matching cost order. We can utilize this queue to prevent unnecessary matching. Our task is to pick up several similar images, not the calculation of similarity over all the images. Suppose we have already matched N images, and the task, in this case, is to search for the M most similar images. Then we can stop matching immediately if an estimate of the matching cost has exceeded the matching cost of the M-th most similar image. To determine the current estimate of the matching cost, the queue is perfect for this purpose, since the head state of the queue indicates the most optimistic matching cost at that time. Interruption of this kind considerably speeds up the search in large databases.

4 Subjective Similarity

4.1 Criteria on Subjective Similarity

Subjective retrieval of images is one of the key technologies for making image databases more flexible and intelligent. To evaluate the accuracy of such systems, the following criteria have been used in several papers.

1. Give similar images larger similarity values.
2. Retrieve similar images in higher orders.
3. Search similar images with least commitment of users.

All the criteria is intuitively natural; however, only the first and the second criterion is combined into an evaluation function in this paper.

From now on, we study a case that an image is represented parametrically by several parameters. Imagine that someone lets you retrieve similar images out of the image database by just using a set of *weighting factors* each multiplied by each parameter value to indicate the importance of the corresponding parameter. Then you might give larger value to weighting factors that corresponds to image features you regard as salient. Conversely, a subjective similarity measure should be such that the weighting factors you regard as salient have larger values accordingly. Then the task is to inductively learn the optimal value of the weighting factors that maximally reflect personal subjective impression on the images from a small number of examples.

With regard to our graph model, weighting factors for nodes, arcs and attributes are incorporated in the graph matching cost. We must admit that this type of similarity measure is not very powerful; however, we argue that this is a practical approach for the time being.

4.2 Learning

In the previous section, we enumerated a few criteria for subjective similarity-based retrieval systems. Now we give a concrete shape to the criteria by formulating a evaluation function as follows:

$$E = \sum_{i=1}^{N} \left[H_i + ae^{-bS_i} \right] \tag{3}$$

where H_i is a subjective similarity between the query graph and the i-th most similar graph with graph matching cost S_i; a and b are constants. Equation (3) represents such criteria: "if the similarities H_i are equivalent, the smaller matching cost S_i is preferred." The following is a concise description of the algorithm.

1. Set initial values of the weighting factors.
2. Using the current set of weighting factors, retrieve the N most similar images from the training set.
3. Associate the accuracy of the retrieval E with the current set of weighting factors.
4. Optimize weighting factors according to the accuracy value and goto 2.

Here the training set is 162 images, and is a subset of our prototype database which stores 1027 images. Using the whole prototype database would be too redundant a training set.

The search space of E is highly non-linear; which means that the search algorithm should be capable of global search. Respecting this requirement, we use a genetic algorithm, in which weighting factors are represented as a form of genetic code. Each weighting factor is coded in a five bits binary code whose magnitude is between 0 and 16. The population size is 20, and crossover and mutation rates are set empirically to 40% and 10%.

Finally, note that in Equation (3) H_i may be given either off-line or on-line. If they are given values beforehand, weighting factors are optimized without user's interruption. In the other scheme, a user must give his personal evaluation for the retrieval result after every iteration. But at the same time, this scheme allows a user to alter his retrieval criteria even during the retrieval process. This concept, which we call "interactive similarity-based retrieval," is inspired from the works in [1, 4], and very attractive technology for realizing flexible image retrieval.

5 Experimental Results

5.1 Subjective Similarity-based Retrieval

For evaluation, ten testers individually selected five subjectively similar images. They were then categorized into two sets — 1) Image 1, 2, 4, and 2) Image 3, 5 —

Table 1: Comparison of retrieval results. The database contains 1027 images. Image 1 through 5 are subjectively similar images selected by ten testers.

Method	Image 1	Image 2	Image 3	Image 4	Image 5
Template Matching	46	161	367	563	286
Feature Vector 1	25	63	669	15	658
Feature Vector 2	745	850	5	538	2
Graph Matching 1	1	11	637	54	691
Graph Matching 2	8	523	771	414	546

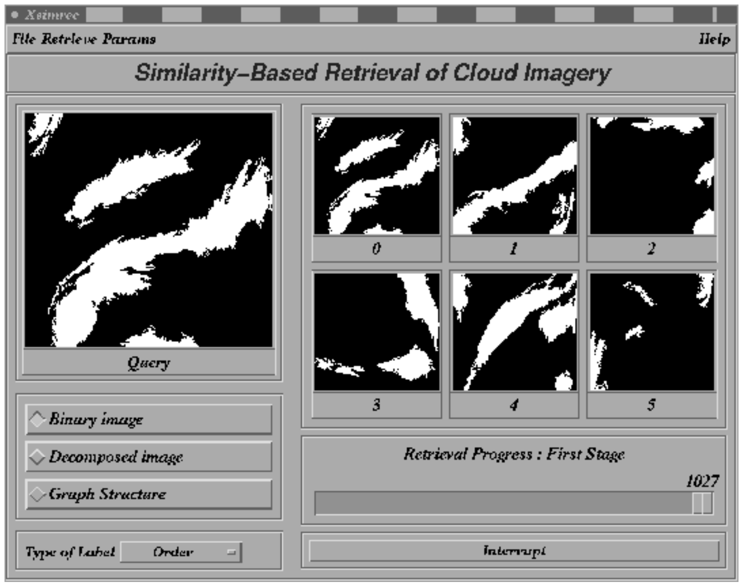

Figure 3: An example of subjective similarity-based retrieval of satellite cloud imagery. The left-side figure is the query image, and the numbers below the right-side images indicate the similarity retrieval order.

based on the qualitative difference of similarity.

Table 1 shows a comparison among the three methods — (naive) template matching, matching using feature vector and graph matching. Concerning Set 1, the graph matching methods produced a better result, although for Set 2 the feature vector method outperformed the graph matching method. This result suggests that our model works effective in some cases, yet it is not sufficient to cover all personal viewpoints on similarity.

5.2 Speeding Up the Retrieval

Since graph matching is highly time-consuming, as we stated earlier, we should have measures to speed up the retrieval. Our model provides three matching methods — 1) matching using feature vectors (FV), 2) graph matching using only the upper level (GMU), and 3) graph matching using entire graphs (GM). Here FV is a method that simply calculates the Mahalanobis distance between two node attributes where an array of nodes is sorted by node area.

Table 2: Retrieval time for a prototype database of 1027 images.

Methods	**Time (sec)**
GM	292
FV-GM	86
GMU-GM	185
FV-GMU-GM	141

Among three methods, FV is the fastest, whereas GM is the slowest. Then our scenario for reducing retrieval time goes as follows. First a simple and fast algorithm is applied in order to discard unpromising matches from the matching list, and at the same time roughly sort the matching list by matching cost. This greatly saves the retrieval time of the following stages, because unpromising matchings tend to appear later. Then a more powerful method is applied over the reduced matching list. Applying an arbitrary number of methods, several combinations are possible.

Table 2 shows the average computation time for three cases. In this paper, FV discards the worst 20% matchings, and GMU, 50%. This means that if we apply FV first, about 200 graphs are eliminated from the matching list after the FV stage. It is obvious that FV-GM method was the fastest. This is because the first stage FV works as both sorting and filtering algorithm, thereby GM can spare unnecessary matchings. Other combinations also outperformed the simple GM method.

6 Conclusion

We have presented a method of subjective similarity-based retrieval of satellite cloud imagery. We experimented over a prototype database of 1027 images, and evaluated our method. We have also addressed several measures for speeding up the search; however we feel that we need a faster algorithm to scale up the image database. Moreover, we should further develop our model to make image databases intelligent enough to "understand" personal viewpoints on similarity.

References

[1] E. Baker. Evolving line drawings. In *Graphics Interface*, 1994.

[2] H. Bunke and G. Allermann. Inexact graph matching for structural pattern recognition. *Patt. Recog. Lett.*, 1:245–253, 1983.

[3] H. Bunke and A. Sanfeliu. Introduction. *Patt. Recog.*, 19(4):249–254, 1986.

[4] C. Caldwell and V.S. Johnston. Tracking a criminal suspect through face-space with a genetic algorithm. In *Proc. of Int. Conf. on Genetic Algorithm*, pages 416–421, 1991.

[5] L.D. Cohen and I. Cohen. Finite-element methods for active contour models and ballons for 2-D and 3-D images. *IEEE Trans. PAMI*, 15(11):1131–1147, 1993.

[6] M. Kass, A. Witkin, and D. Terzopoulos. Snakes : Active coutour models. In *Proc. of 1st ICCV*, pages 259–268, 1987.

[7] N.J. Nilsson. *Principles of Artificial Intelligence*. Springer-Verlag, 1982.

[8] A.P. Pentland. Perceptual organization and the representation of natural form. *Artificial Intell.*, 28:293–331, 1986.

[9] L.G. Shapiro and R.M. Haralick. Structural descriptions and inexact matching. *IEEE Trans. PAMI*, PAMI-3(5):504–519, 1981.