MVA2000 IAPR Workshop on Machine Vision Applications, Nov. 28–30, 2000, The University of Tokyo, Japan

13—36

Map Generation from Aerial Imagery using Data Compression

Toshinori Watanabe, Ken Sugawara and Hiroshi Sugihara
Graduate School of Information Systems
University of Electro-Communications

Abstract

In this paper, we propose a new method to generate a digital map from the aerial photograph. For map generation, it is necessary to develop a technique to recognize various natural- and artificial-objects in the image by computer processing. In general, a photo image has spatial and spectral informations. It is desirable that these two informations are effectively utilized. A new method proposed here integrates and utilizes them without complicated processing. First, the photo image is represented by a Minimum Spanning Tree (MST) and encoded into one-dimensional text, considering both spatial information and spectroscopic informations. Next, this text is compressed by using a dictionary technique. As the compressibility of text depends on the dictionary, the input image can be described as a n-dimensional compressibility vector using n dictionaries. Using 139 aerial imagery, effectiveness of this new method is illustrated.

1 Introduction

Remote sensing is the technology to survey the earth by satellites and aircrafts using camera, radio wave, specialized sensors etc. It is utilized in various fields such as environment countermeasure, resource survey and so on. Recently, the performance of commercial satellite is drastically improved and high-resolution photographs become available. As a result, it becomes possible to apply them to the urban planning, environment and disaster monitoring, digital map generation and so on. In the near future, it is considered that we can get enough information of the ground in order to generate digital maps from the satellite imagery. Today, making of a map requires huge man power and time. The development of support systems to generate maps automatically is badly needed. How can we realize such a system ? In this paper, we propose a new support system in order to reduce manual burden in map generation.

For map generation, the technology which recognizes various natural- and artificial-objects in the image is necessary. The image usually contains both spectroscopic information such as colors, and spatial information such as positions and shapes of the objects. It is desirable that these two informations are effectively utilized. In addition, heterogeneous image processing methods should be combined in conventional approaches, resulting in complicated systems. Following two problems should be solved.

- Efficient utilization of both spatial and spectral informations.
- Trainability to cope with objects' varieties.
- Simple scheme which is suitable for rapid computation.

Our method proceeds as follows. First, the image is encoded into one-dimensional text considering spatial and spectral informations simultaneously. Second, the data is compressed, outputting a compressibility vector used as the feature of the image in image classification/recognition.

*Address: 1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585 Japan. E-mail: watanabe,sugawara@sd.is.uec.ac.jp

2 Algorithm for Map Generation

To generate a digital map from an aerial photograph, two steps are required.
(1) The image must be divided into regions.
(2) Each region must be recognized as some objects.

Minimum Spanning Tree (MST)[1] is used in (1), Universal Pattern Representation by Data Compression (UPRDC)[2, 3] is used in (2).

2.1 Image Segmentation by MST

Digital image is an arrayed pixels. So, it is possible to express the image as a graph by considering each pixel as a node, and the color difference of adjoining pixels as a weighted edge. By determining MST of this graph, we can get a tree with small edge weights. Cutting the tree into subtrees (regions) at edges of which weight exceeds a certain threshold value, we can get separated regions (Figure 1).

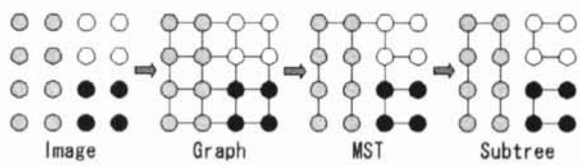


Figure 1: Image segmentation by MST.

This method usually works well, but in some cases it does not. Let's assume the condition where the color values of pixels change gradually (Figure 2).

Figure 2: An anomaly case in segmentation.

In this case, the values of each edge are small and does not exceed the threshold. It means that we cannot divide it into two (or more) regions in spite of the obvious difference between the colors at both ends. There are some proposals to cope with this problem[4, 5], but the execution time takes much time. So, the following faster algorithm is introduced.

(P1) Initially, all the pixels are considered as independent regions. Individual number is assigned to each node in MST, and the pixel value is adopted as the region value.
(P2) All edges are sorted according to their weights.
(P3) (P4) $\sim$ (P5) is executed in order of increasing weight. However, edges already marked 'done' are skipped.
(P4) The difference of average color values between regions at both ends of the chosen edge is calculated.
(P5) If the difference is less than a given threshold, the edge is marked 'done', the two regions are merged, and the average value of the merged region is calculated. If the difference exceeds a given threshold, they are not merged.
(P6) When all edges are processed, (P7) is executed in case when there is no edge marked 'done', otherwise go to (P3).
(P7) All the edges which has no 'done' mark are removed.

2.2 Object Recognition by UPRDC

We use the text compression method for object recognition. Approaches to text compression can be divided into two classes: statistical method and dictionary method[6]. Dictionary method achieves compression by replacing groups of characters or phrases with indices in some dictionary. The dictionary is a list of characters or phrases that are expected to appear frequently. In general, the length of the indices are shorter than characters or phrases themselves, thereby we say the text is compressed.

There are two remarkable points in this method. One is that the created dictionary depends on the content of the compressed text, and the other is that we can use the dictionary for compressing other texts.

Now let's denote a text as T_i, a dictionary generated from T_i as D_i, and the compression ratio of T_i by D_j as $R(T_i, D_j)$.

$$R(T_i, D_j) = \frac{Compressed\,length\,of\,T_i\,by\,using\,D_j}{Original\,length\,of\,T_i}.$$

Assume we have three texts T_A, T_B and T_M. If the content of T_M resembles to T_A, the following relation holds.

$$R(T_M, D_A) < R(T_M, D_B).$$

If the content of T_M does not resemble to both T_A and T_B,

$$R(T_M, D_A) \sim R(T_M, D_B) >> 0.$$

Using this property, we can express the feature of T_M by a n-dimensional vector $\vec{K}_M$. Here, we used n different dictionaries to compress T_M.

$$\vec{K}_M = (R(T_M, D_1), R(T_M, D_2), ..., R(T_M, D_n)).$$

We refer to the space, the axis of the space, and the vector in this space as "Space of Data Compression(SDC)", "SDC basis", and "Feature vector", respectively. We use this method for classification and recognition of regions in a photo image. A schematic representation of SDC is shown in Figure 3. A feature vector $\vec{K}_X$ representing a region X is close to $\vec{K}_R$ which has been registered as "ROAD" in a pre-instruction.. Therefore, the region X is recognized as "ROAD."

We used LZW data compression algorithm[7, 8] in this study.

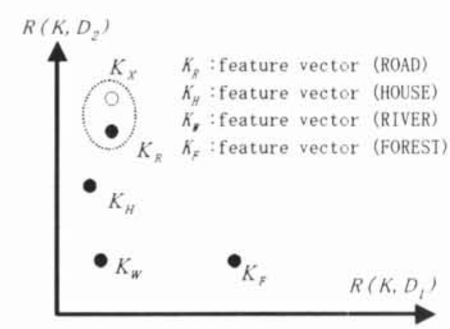
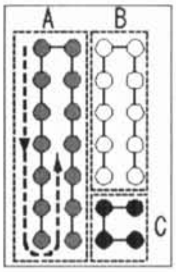


Figure 3: Schematic representation of SDC

2.3 Encoding Region into Text

To use UPRDC, regions have to be encoded into one-dimensional texts which includes spatial and spectral informations. After MST partitioning, each region is expressed as a subtree. So, we encode the subtree by a tree-traversal. As illustrated in Figure 4, an alphabet is assigned to each representative pair of (color, traverse direction). Dummy starting nodes is introduced at north-east end (which is not shown in the figure). The subtree is traversed in minimum-weight-edge-first manner, and at each node, relevant alphabet is chosen, giving a text reflecting both spatial and spectral informations.



Color	Direction of node	Code
C_A	Vertical	a
	Horizontal	b
C_B	Vertical	c
	Horizontal	d
C_C	Vertical	e
	Horizontal	f

Region A = baaaaaabaaaaaa
Region B = dccccdcccc
Region C = feff

Figure 4: An example of Region Textization.

3 Region Recognition

In the textization, infinite color space, *e.g.* RGB vector space, must be represented by a rather small alphabet set. To fill this 'band-gap', we introduced a hierarchical method (Figure 5). Both higher and lower hierarchies are processed using UPRDC.

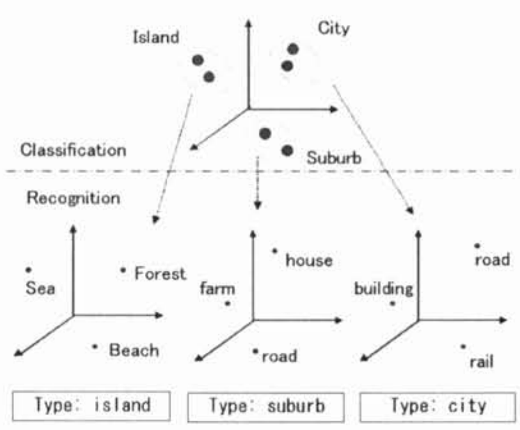


Figure 5: Hierarchical architecture.

(1) Classification
(Cl1) MSTs are generated for input images, and each image is encoded into one-dimensional text. A unique global code table is applied to all input images.
(Cl2) These texts are compressed by SDC basis dictionaries and mapped into SDC.
(Cl3) A dendrogram is generated for them by cluster analysis.
(Cl4) The dendrogram is cut into partial dendrograms, group names are given by the user. Group-specific SDC data, including code table, dictionaries, and feature vectors, are generated. Also, the high level SDC is set up in a similar manner.

(2) Recognition
(Re1) The input image is classified into an appropriate group. This is done by MST generation, mapping into a text using common alphabet, feature vector generation by text compression, and classification into the nearest image group in SDC.
(Re2) The MST in (Re1) is segmented into regions using the method in 2.1.
(Re3) Each sub-MST is encoded into a text. This time, group-specific alphabet is used to cope with the 'band-gap' problem.
(Re4) Each text is mapped to a group-specific SDC vector by compression.
(Re5) Each region is recognized by searching the nearest neighbour in group-specific SDC.
(Re6) New region data is stored into group-specific SDC by the user.
Schematic of these processes are shown in Figure 6.

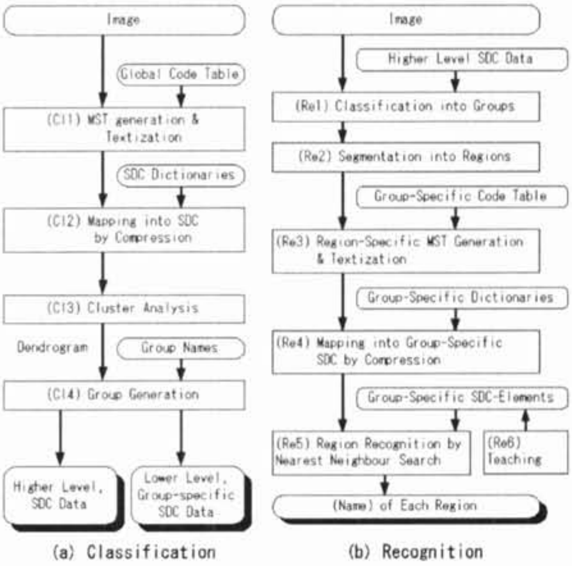


Figure 6: Outline of processing.

4 Experimental Result

In the experiment, we used aerial photographs of 139 districts in Tokyo. The image resolution is close to the current best satellite images. They were photographed from 1000 $\sim$ 1500 m height. The image size is 600 $\times$ 840 pixels.

In classification, they are classified into five groups labeled as "island", "forest", "suburb", "town" and "city".

In recognition, objects are attributed to one of the ten types labeled as "forest", "meadow", "road", "sea", "river", "farm", "field", "building", "railway" and "house".

The recognition accuracy is measured by comparing the computer recognition results and the visual recognition results for 30 regions in each image. Figure 7 summarizes the result.

Figure 8 illustrates an example of the input photo image and the output map. For images with many fragmented regions, the recognition accuracy tends to deteriorate. But the remarkable point is that

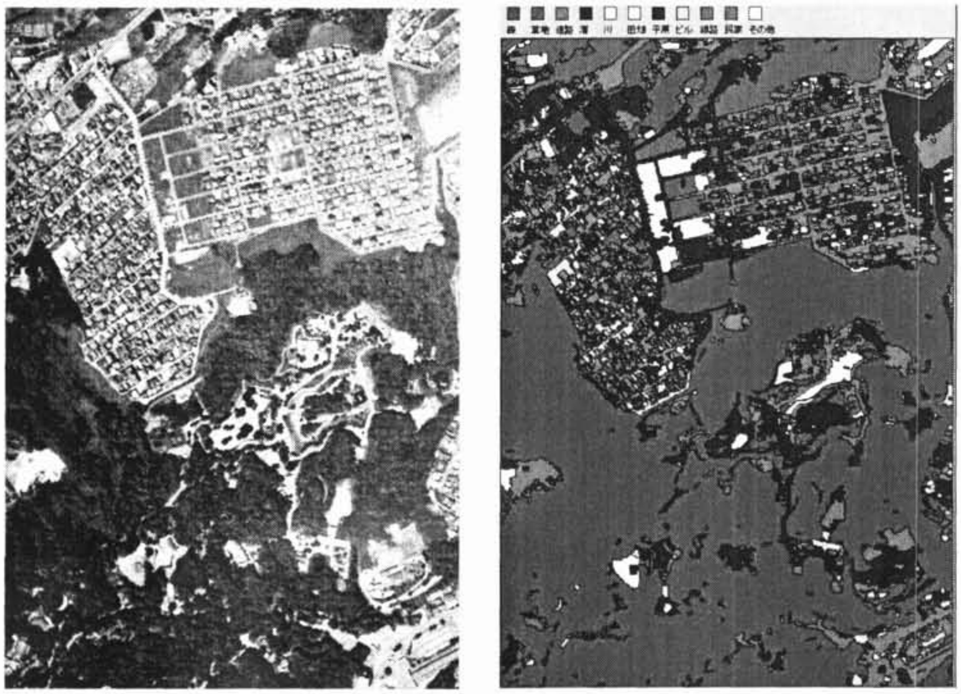


Figure 8: Experimental result of map generation.

$$\text{Accuracy} = \frac{\text{the number of pixels recognized correctly}}{\text{total number of pixels in 30 regions}}$$

Type	Island	Forest	Suburb	Town	City
recog. Acc.	88%	88%	70%	80%	77%

Figure 7: Recognition accuracy.

the system can distinguish similar colored objects ("building" and "road", "sea" and "river", and so on). This is due to the simultaneous coding of spatial and spectral information adopted by our system.

5 Conclusion

We proposed a new method to generate digital maps from aerial imagery automatically. The method is examined using more than 100 aerial imagery and as a result, the effectiveness of the method was confirmed.

In our method, the input image was described by MST and encoded into one-dimensional text. The text is compressed by using n dictionaries and its feature is characterized by a n-dimensional vector. And both classification and recognition is performed in SDC. This scheme is simple and flexible. As mentioned above, the unknown object is recognized by measuring distances from other known objects. If these distances are too large, the object is added as a new object. It means that this system can easily learn unknown objects.

References

[1] D.Cheriton, R.E.Tarjan, "Finding Minimum Spanning Trees",*SIAM J.Comput.*, Vol.5, No.4 (1976) pp.724-742.

[2] T.Watanabe, "UPRDC: Universal Pattern Representation by Data Compression", *proc. JSPRS*, (2000) pp.253-258 (in Japanese).

[3] K.Kondou, N.Kato and T.Watanabe, "OSR: Online Sketch Recognition by Data Compression Technique", *J.IPSJ*, Vol.38, No.12, (1997) pp.2468-2478 (in Japanese).

[4] J.R.Lersch, A.E.Iverson and B.N.Webb, "Segmentation of Multi b and Imagery Using Minimum Spanning Tree", *Algorithms for Multi-spectral and Hyperspectral Imagery* Ⅱ *Proc. SPIE*,2758 (1997) pp.10-18.

[5] M.Suk, T.H.Cho, "Segmentation of Images using Minimum Spanning Trees", *Applications of Digital Image Processing V, Proc. SPIE* 397 (1983) pp.180-185.

[6] T.C.Bell, J.G.Cleary and I.H.Witten, "TEXT COMPRESSION", Prentice-Hall (1990).

[7] J.Ziv, A.Lempel, "Compression of Individual Sequences via Variable-Rate Coding",*IEEE Trans. on Inform. Theory*, vol.IT-24, no.5 (1978) pp.530-536.

[8] T.A.Welch, "A Technique for High Performance Data Compression",*IEEE Comput.*, vol.17 (1984) pp.8-19.