A Landscape Photograph Localisation Method with a Genetic Algorithm using Image Features

Hideo Nagashima and Tetsuya Suzuki[a]
Graduate School of Systems Engineering and Science, Shibaura Institute of Technology, Saitama, Japan

Keywords: Geolocation, Genetic Algorithm.

Abstract: It improves the utility value of landscape photographs to identify their shooting locations and shooting directions because geolocated photographs can be used for location-oriented search systems, verification of historically valuable photographs and so on. However, a large amount of labor is required to perform manual shooting location search. Therefore, we are developing a location search system for landscape photographs. To find where and how a given landscape photograph was taken, the system puts virtual cameras in three-dimensional terrain model and adjusts their parameters using a genetic algorithm. The system does not realize efficient search because it has problems such as a long processing time, a multimodal problem and optimization by genetic algorithms. In this research, we propose several efficient search methods using image features and show experimental results for evaluation of them.

1 INTRODUCTION

In general, identifying the shooting location and shooting direction of landscape photographs leads to effective use of them because geolocated photographs can be used for location-oriented search systems, verification of historically valuable photographs and so on. However, a great deal of labor is required to manually identify the shooting location.

Therefore, we worked on developing a system which supports to identify geolocations of landscape photographs (Suzuki and Tokuda, 2008; Suzuki and Tokuda, 2006). The system targets landscape photographs whose shooting locations can be identified by mountain ridges and terrain in them. It puts virtual cameras in three-dimensional terrain model, each of which has a latitude, a longitude, a altitude, a direction and a focal length as parameters. It then adjusts their parameters using a genetic algorithm (GA). The GA uses a fitness function which compares characteristics of a given landscape photograph and those of photographs taken by virtual camera parameters in the three-dimensional terrain model.

It is difficult for the system to find good solutions in general because the landscape of the fitness function constructed from the characteristics of a given landscape photograph tends to be multimodal and include a large plateau. We could not improve the search process enough though we introduced some techniques to the system, which are relaxation of optimization problems, a local search to avoid lethal genes and so on. In addition, much experimental codes added to the system made it difficult to modify the system.

To introduce a more intelligent camera parameter adjustment method based on image features, we decided to abandon extension of the existing system and implement a new system.

The organization of this paper is as follows. In section 2, we summarize related work including our former system. We then explain an overview of our new system in section 3 and explain search methods in the new system in section 4. We explain experiment in section 5 and evaluate the results in section 6. We state concluding remarks in section 7.

2 RELATED WORK

2.1 Our Former System

Fig.1 shows the entire search process in our former system. To find the location of a landscape photograph by our former system, we take the following steps.

[a] https://orcid.org/0000-0002-9957-8229

Nagashima, H. and Suzuki, T.
A Landscape Photograph Localisation Method with a Genetic Algorithm using Image Features.
DOI: 10.5220/0010394712901297
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 2*, pages 1290-1297
ISBN: 978-989-758-484-8

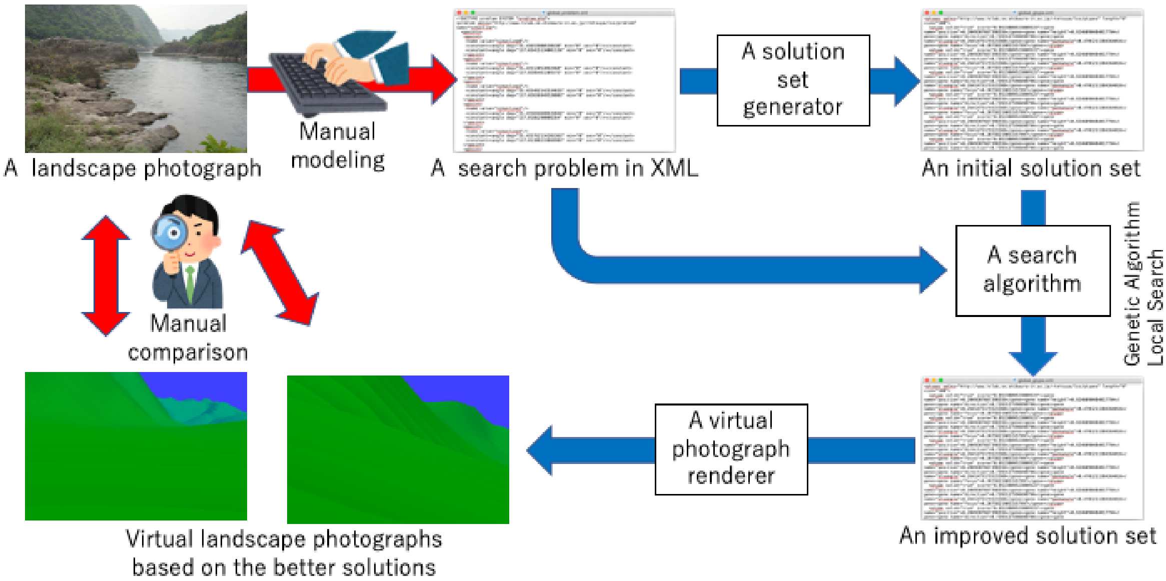

Figure 1: The entire search process using our former system.

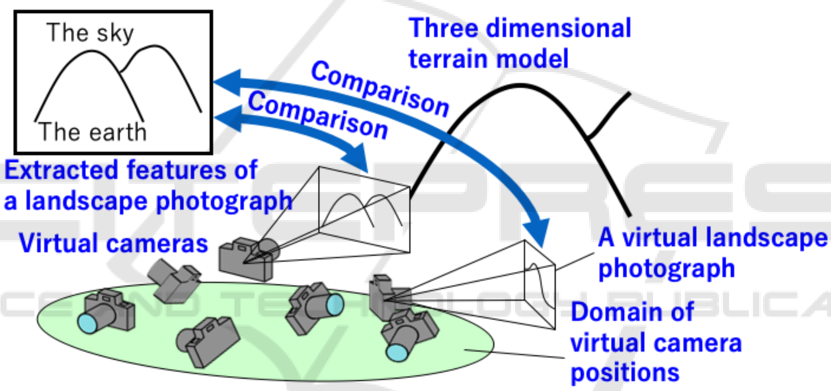

Figure 2: Search in our former system.

1. definition of a search problem
2. generation of initial solution sets
3. search
4. check of the resulting solutions

We explain these steps in the following.

To define a search problem for our system, we describe features of terrain in the photograph, cameras settings, and a search space declaratively. For example, shapes of mountains which are boundaries between the earth and the sky, variables for cameras settings and their domains.

Before we start a search, we generate an initial solution set. The generated solution set is used as start points of a search.

Our former system improves solutions in a given solution set by a specified search algorithm, and outputs the resulting solutions. We have implemented a real-coded genetic algorithm and a local search algorithm as search algorithms for our former system. The real-coded genetic algorithm uses unimodal normal distribution crossover (UNDX)-m as a crossover operation and distance dependent alternation (DDA) model as a generation alternation model. The real-coded genetic algorithm with the combination keeps the variety of solutions during searches (Takahashi et al., 2000). The search algorithm refers to a data elevation model (DEM), which is 50m digital elevation model of Japan published by the Geographical Survey Institute. Fig.2 shows search by a search algorithm.

We finally take virtual photographs based on the resulting solutions for check because better solutions for a search problem may not be solutions we expect if the problem definition is not adequate.

We have problems about the former system as follows.

1. Much experimental codes added to the system makes it difficult to modify the system.

2. The system is very slow in search because it involves many translation between two coordinate systems: the latitude-longitude-altitude coordinate system and the geocentric Cartesian coordinate system. The system uses the former for terrain data and the latter for line of sight from cameras, and the translation between them is needed in collision detection between line of sight from cameras and the earth in the terrain model.
3. We could not improve the search process enough though we introduced some techniques to the system, which are relaxation of optimization problems, a local search to avoid lethal genes and so on.

2.2 A CNN-based Method

Tobians Weyand et al. proposed a method to identify the shooting area using convolution neural network (CNN) (Hays and Efros, 2008). It divides the world map into multiple areas and learns areas in which given landscape photographs were taken. When a new landscape photograph without location information is input, it estimates an area in which the landscape photograph was highly likely taken.

The method can narrow down the shooting area, but cannot specify the shooting location in detail. To specify the shooting location in detail, it is necessary to divide the world map in more detail and use a lot of photographs with shooting location information for training.

3 AN OVERVIEW OF OUR NEW SYSTEM

To resolve the problems of our former system pointed out in section 2.1, we decided to develop a new system which supports to identify geolocations of landscape photographs from scratch.

The main differences between our former system and the our new system are as follows.

1. We changed the implementation language from Java to Python. One of the reason is that Python is in widespread use as Java. Another reason is that we plan to introduce image segmentation based on deep neural network for automatic image feature extraction to define a search problem from a given landscape photograph and Python libraries for deep learning such as TensorFlow (Abadi et al., 2015) and PyTorch (Paszke et al., 2019) are also in widespread use. In addition, it would be easy to implement a domain specific language for search problem in Python than in Java.
2. We changed the mainly used coordination system for terrain data from the latitude-longitude-altitude coordinate system to the geocentric Cartesian coordinate system. To realize the change, we use three dimensional polygons converted from the DEM used in our former system as terrain data.
3. We introduced a camera parameter adjustment method based on image features to make parameter adjustment more efficient.

4 SEARCH IN OUR SYSTEM

In this section, we explain how our new system search the shooting location of a given landscape using a real-coded genetic algorithm.

4.1 An Overview of the Search Process

The input for our new system are a color-coded landscape photograph whose colors corresponds to attributes such as the sky and the ground surface, and a search range. If the input is given, our new system works as follows.

1. It extracts image features such as ridge lines from the given color-coded photograph.
2. It sets the search range as a geographical area.
3. It generates an initial solution set, each solution of which corresponds to initial parameters of a virtual camera. It uses the terrain data of the geographical area to place virtual cameras where the surrounding scenery can be looked over.
4. It repeats the following steps until the best fitness function's value converges or the number of repetition reaches the specified number of times.
   (a) For each virtual camera, it calculates the ridge line in the virtual landscape photograph, the color-coded virtual landscape photograph whose colors correspond to attributes such as the sky and the ground surface, and the depth image of the virtual landscape photograph.
   (b) For each virtual camera, it compares the features of the input landscape photograph with the features of the ridge line taken by the camera and calculates the fitness function value of the camera based on the degree of similarity between them. The higher the degree of similarity is, the lower the fitness function value is.

(c) For each camera with a fitness function value above a certain level, it improves the camera parameters using the depth image.

(d) For each camera with a fitness function value above a certain level, it adjusts the direction of the camera to maximize the fitness function value without changing its latitude and longitude.

(e) It updates the solution set by the crossover operation UNDX-*m* and the generation alternation model DDA.

4.2 Evaluation Function by Image Feature

This is the step 4(b) of our new system described in section 4.1. It uses an image feature matching algorithm AKAZE (Alcantarilla et al., 2013) to compare the color-coded image of an input landscape photograph and that of the virtual landscape photograph taken by a virtual camera. AKAZE is an algorithm that is resistant to scaling and rotation. It apply smoothing filters to the images before comparison by AKAZE. It combines AKAZE with template matching, which is vulnerable to scaling, because some similar images found by AKAZE are quite different when compared by humans though AKAZE can find partial matching of images.

4.3 Initial Solution Set Generation

This is the step 3 of our new system described in section 4.1. In our former system, initial solutions are randomly generated within the specified range. The system initially places virtual cameras on mountains to make it possible to overlook many ridge lines because the system uses ridge lines for comparison photographs during search.

The procedure to place cameras on mountains is shown below. The range can be a quadrangle. It takes the number of initial solutions and a rectangular geographical area as input.

1. It stores the elevation data of the specified geographical area into a two-dimensional array E.
2. It repeats the following steps until the number of the extracted summit reaches the specified number of initial solutions.
   (a) It finds the summit which is the highest place in E.
   (b) It records the array coordinate of the summit.
   (c) It ignores the summit and the arbitrary range around it hereafter.
3. It outputs the recorded summits as initial solutions.

4.4 Improvement by Distance Adjustment

This is the step 4(c) of our new system described in section 4.1. Even if a mountain to be searched is in the image taken by a virtual camera during search, our former system can not adjust its camera parameters in consideration of it. Our new system, however, adjusts its camera parameters using both the depth image by the camera as follows if the fitness value of a virtual camera is higher than the threshold value.

1. It finds the n-best positions of image feature matching in the image and the scale ratio of the image.
2. It creates the depth image by the camera and a ridge line image with the n-best positions.
3. It obtains the average distance of the ridge line by the depth image and the ridge line image.
4. It determines the amount of camera movement based on the distance, the amount of pixel movement in the image, and the scale ratio.

4.5 Improvement by Camera Direction Adjustment

This is the step 4(d) of our new system described in section 4.1. The step tries to improve the fitness function value of a virtual camera by changing the direction of the camera. The procedure is as follows.

1. For each camera with a fitness function value above a certain level, it performs the following steps.
   (a) It takes virtual landscape photographs while horizontally rotating the camera by 10 degrees.
   (b) Let d and s be the direction of a virtual landscape photograph among them which maximizes the degree of similarity to the given landscape photograph and the maximum degree of similarity respectively.
   (c) If s is greater than the degree of similarity between the given landscape photograph and the virtual landscape photograph of the camera, it changes the camera direction to d.

5 EXPERIMENT

We conducted a comparative experiment to verify the effect of our new system and the the search method described in section 4.1 under the following conditions.

- The search range is 20 km square around Mt. Fuji.
- The number of generations of the genetic algorithm was fixed at 50 generations.
- The input image used was the image of the summit of Mt. Fuji. The image is a color-coded image for each attribute.
- The following three methods were compared.

  Baseline The initial solutions are randomly generated, and the following three methods are not used.

  Method 1 The initial solution generation method described in section 4.3

  Method 2 The improvement-by-distance-adjustment described in section 4.4

  Method 3 The improvement-by-camera-direction-adjustment described in section 4.5
- The experiment was performed with five patterns: the baseline, each of the method 1, 2 and 3, and the combination of the method 1, 2 and 3.

The input image and its camera parameter are shown in Fig.3 and Table 1 respectively. The sky and the ground are painted in black and green respectively in Fig.3.

The resulting images of the five patterns are shown in Fig.4, Fig.5, Fig.6, Fig.7, and Fig.8. The sky and the ground are painted in black and green respectively in the figures as in Fig.3. The resulting images' camera parameters and the evaluation value are shown in Table 2, Table 3, Table 4, Table 5, and Table 6. The evaluation values should be low because they are the degree of difference from the input image.

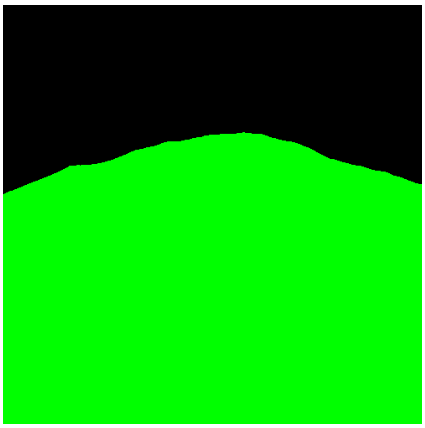

Figure 3: The input image.

Fig.9 and Fig.10 show the fitness of the optimal solution at each generation in each method. The initial fitness values in the method 1, which is the initial solution generation method described in section 4.3, and all the methods are higher than other three methods. The fitness values of the method 1 and all the methods are at least 86 until the 21th generation while those of other methods are at most 24. The fitness value in the method 3, which is the improvement-

Table 1: The input image's camera parameters.

Latitude (deg.)	35.39333333
Longitude (deg.)	138.81333333
Azimuth Angle (deg.)	122
Elevation Angle (deg.)	0

Figure 4: The resulting best image (the baseline).

Table 2: The resulting best image's camera parameter (the baseline).

Latitude (deg.)	35.37055556
Longitude (deg.)	138.84944444
Azimuth Angle (deg.)	2
Elevation Angle (deg.)	0
Distance (m)	4141.52
Evaluation Value	23.90

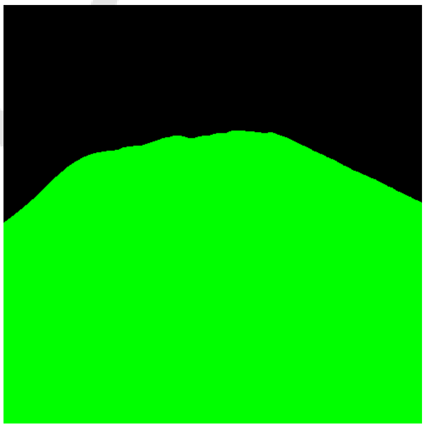

Figure 5: The resulting best image (the method 1).

Table 3: The resulting best image's camera parameter (the method 1).

Latitude (deg.)	35.33305556
Longitude (deg.)	138.77111111
Azimuth Angle (deg.)	-1
Elevation Angle (deg.)	-16
Distance (m)	7710.35
Evaluation Value	9.70

Figure 6: The resulting best image (the method 2).

Table 4: The resulting best image's camera parameter (the method 2).

Latitude (deg.)	35.35638889
Longitude (deg.)	138.82666667
Azimuth Angle (deg.)	9
Elevation Angle (deg.)	0
Distance (m)	4274.21
Evaluation Value	14.09

Figure 7: The resulting best image (the method 3).

by-camera-direction-adjustment described in section 4.5, is lower than other methods from the beginning, which is at most 19.

From the viewpoint of the fitness values, the best

Table 5: The resulting best image's camera parameter (the method 3).

Latitude (deg.)	35.36055555
Longitude (deg.)	138.85527777
Azimuth Angle (deg.)	-37
Elevation Angle (deg.)	0
Distance (m)	5267.98
Evaluation Value	6.84

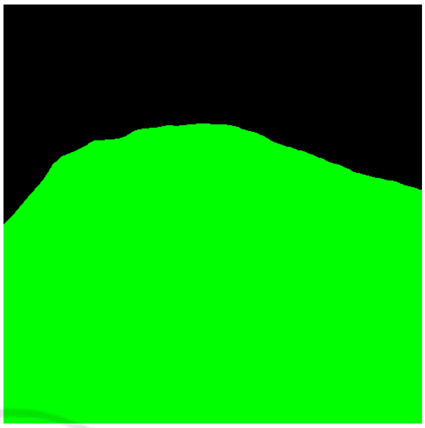

Figure 8: The resulting best image (all the methods).

Table 6: The resulting best image's camera parameter (all the methods).

Latitude (deg.)	35.35944444
Longitude (deg.)	138.87444444
Azimuth Angle (deg.)	88
Elevation Angle (deg.)	-10
Distance (m)	6706.20
Evaluation Value	16.11

result was obtained when only the method 3, which is the improvement-by-camera-direction-adjustment described in section 4.5, was used. From the viewpoint of the distance from the position where the input image was taken, the best result was obtained in the baseline.

6 EVALUATION

We evaluate the experimental results by Fig.9 and Fig.10 as follows.

- Because the best fitness value of the initial solutions of the method 3 was the best among those of the methods in the experiment, we confirmed that initial solutions can be improved well by optimizing the camera orientations even if they are randomly generated.
- The fitness values of the method 1 is the worst

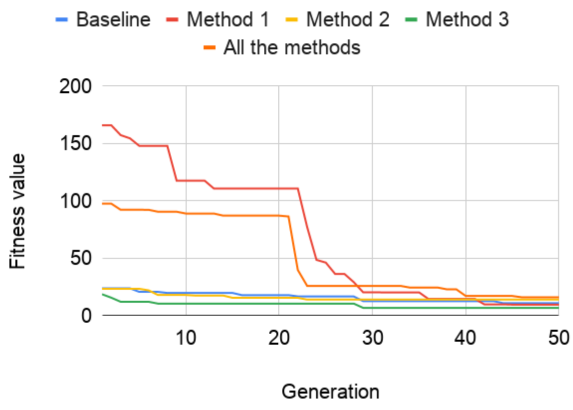

Figure 9: Fitness values of the best individuals at each generation.

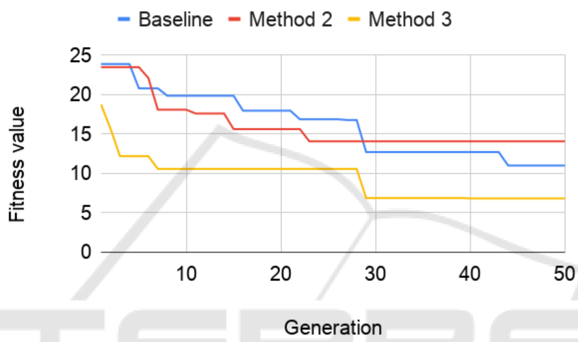

Figure 10: Fitness values of the best individuals at each generation (except the method and all the method).

among those of the methods in the experiment, and that of all the methods which includes the method 1 is the second worst. We guess the reasons as follows.

– The method 1 placed initial cameras on the summit of the target mountain in the input image because the target mountain is a single peak mountain and no other mountains exist in the search area. As the result, the camera can not take pictures of the mountain.
– The diversity of initial solutions was decreased by the method 1.

7 CONCLUSION

We proposed a new shooting location search system and a new search method. The system targets landscape photographs whose shooting locations can be identified by mountain ridges and terrain in them. The system places virtual cameras in three dimensional terrain model and searches camera parameters which can take a landscape photograph similar to a given landscape photograph using a real-coded genetic algorithm. We conducted experiments to verify effectiveness of the proposed three camera parameter methods, and confirmed that the initial solution set generation method which places cameras on mountains and a camera parameter adjustment method which find better direction by rotating cameras are effective. We also confirmed that the fitness function in the genetic algorithm which compares two images is still insufficient.

Our future works are improvement of initial solution set generation, the fitness function in the genetic algorithm, image segmentation for automatic image feature extraction to define a search problem from a given landscape photograph, and coping with foggy input photographs which hide ridge lines.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Alcantarilla, P. F., Nuevo, J., and Bartoli, A. (2013). Fast explicit diffusion for accelerated features in nonlinear scale spaces. In Burghardt, T., Damen, D., Mayol-Cuevas, W. W., and Mirmehdi, M., editors, *British Machine Vision Conference, BMVC 2013, Bristol, UK, September 9-13, 2013*. BMVA Press.

Hays, J. and Efros, A. A. (2008). Im2gps: estimating geographic information from a single image. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Suzuki, T. and Tokuda, T. (2006). A system for landscape photograph localization. In *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications - Volume 01*, ISDA '06, page 10801085, USA. IEEE Computer Society.

Suzuki, T. and Tokuda, T. (2008). Initial solution set improvement for a genetic algorithm in a metadata generation support system for landscape photographs. In Tokunaga, T. and Ortega, A., editors, *Large-Scale Knowledge Resources. Construction and Application*, pages 67–74, Berlin, Heidelberg. Springer Berlin Heidelberg.

Takahashi, O., Kita, H., and Kobayashi, S. (2000). A real-coded genetic algorithm using distance dependent alternation model for complex function optimization. In *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*, GECCO'00, page 219226, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.