International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

Range Based Search Algorithm For Photomosaic Generation

Jinal Shah[1], Jinali Gala[2], Krunal Parmar[3], Manan Shah[4], Mansi Kambli[5]

Student, Computer Engineering, K. J. Somaiya College of Engineering, Mumbai, India[1]

Student, Computer Engineering, K. J. Somaiya College of Engineering, Mumbai, India[2]

Student, Computer Engineering, K. J. Somaiya College of Engineering, Mumbai, India[3]

Student, Computer Engineering, K. J. Somaiya College of Engineering, Mumbai, India[4]

Assistant Professor, Computer Engineering, K. J. Somaiya College of Engineering, Mumbai, India[5]

Abstract: Photomosaic is a technique which transforms an input image into a rectangular grid of thumbnail images preserving the overall appearance. When a photomosaic is viewed from a distance, it resembles a desired target image. The typical photomosaic algorithm searches from a large database of images, one picture that approximates a block of pixels in the main image. Synthesizing photomosaics typically requires very large image databases in order to produce pleasing results. In this paper, we propose an approach in which we apply a constraint on the number of times a tile can be repeated. We use Range Based Search Algorithm to find the appropriate tiles for each rectangular grid. Visual responses change depending on the proximity to the photomosaic, leading to many creative prospects for publicity.

Keywords: Photomosaic ; range search algorithm (RS), Selection criteria , threshold.

I. INTRODUCTION

Photographic mosaic, also known as Photomosaic, is a picture that has been divided into rectangular sections, each of which is replaced with another photograph that matches the section. When viewed at low magnifications, the individual pixels appear as the primary image, while close examination reveals that the image is in fact made up of many hundreds or thousands of smaller images. The concept of photomosaics originated in a computer graphics system called DominiPix [1], in which pictures are constructed from sets of dominoes.

Fig.1 A 10 × 10 photomosaic assembled from 1600 images. The input image is miniaturized at the bottom-right for reference.

Silvers later conceived of the idea of dividing a target image into smaller regions, eventually founding a company that now produces photomosaics on a commercial basis. In the simpler kind, each part of the target image is averaged down to a single color. Each of the database images is also reduced to a single color. Each part of the target image is then replaced with one from the database where these colors are as similar as possible. Current photomosaic generation algorithms, however, require huge amounts of images in order to produce attractive results. Repetitions are allowed to occur, which may locally bias the mosaic.

Tuning photomosaics manually is obviously exhaustive and impractical. This paper presents alternatives to maximize the usage of available images while still preventing repetitions.

The photomosaicing can be used to an artistic view of an image which can also be used for animations, magazines and advertisements . Various applications based on mobile platforms have been developed and served to users, such as PhotoMosaic in Mac, Mosaicture Lite in Android.

II. DEVELOPING PHOTOMOSAIC

Given a target image I and database of tiles $T = \{t_1, \ldots, t_n\}$ synthesizing a photomosaic P digests to:

i. Subdivide the target image into rectangular grids $P=\{p_1, \ldots, p_m\}$.
ii. For each patch p_i, search for an appropriate tile t_j to replace the patch; this step may require manipulation of the tile images (resizing, cropping, etc...)

The goal is to maximize the usage of different tiles in the resulting photomosaic. However, maximization itself does not guarantee that the output photomosaic P will resemble the input image I when observed from distance. Therefore, colour similarity also plays an important role which cannot be neglected.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

Consider size of target image as A×B, and that of patch as $M \times N$, then the mosaic image will consist of A/M rows and B/N columns. The number of patches can be calculated as:

Number of patches (m) = $\frac{Size\ of\ target\ image}{Size\ of\ patches}$

Let t be the threshold which gives the number of times a tile can be used for replacement. Then, photomosaic generation becomes an optimization problem, where a set of tiles has to be arranged to resemble a target image such that no tile is used more than t times.

The threshold should be selected with care because a large value can make the mosaic biased and if proper threshold is selected then the mosaic will resemble closely to the target image.

III. RANGE BASED SEARCH

The greedy approach[1] begins by assigning tiles to patches based on the "best-match" criteria. It locates the most similar matches and sequentially revamps repetitions with unmatched ones; likely to fall into local minima solutions.

Range based search algorithm is a technique which finds potential solutions based on specified range. It has almost no constraints and is relatively easy to implement. Moreover, it is capable of finding very close approximations to the global optimum in lesser time.
Initially, the procedure is same as greedy algorithm[1] where we calculate RGB values of each patch and tile. Similarities between patch and tiles are done on the basis of the selection criteria.

A. Selection Criteria

Individual range for the red, green and blue components of the patches are selected. the differences between the corresponding RGB components of a patch with every tile from the database is calculated using the given equations. The tiles of the database whose values are less than specified range are first collected. For those collected tiles the mean values of RGB difference is calculated and the tile with lowest mean is selected for that respective patch. Care is taken that the number of times a tile is used does not cross the threshold.

$$RS_R = Pi,r - Tj,r \tag{1}$$
$$RS_G = Pi,g - Tj,g \tag{2}$$
$$RS_B = Pi,b - Tj,b \tag{3}$$
$$RS_{sum} = RS_R + RS_G + RS_B \tag{4}$$

Where $P_{i,r}$, $P_{i,g}$ and $P_{i,b}$ are red, green and blue component of i^{th} patch respectively. Similarly $T_{j,r}$, $T_{j,g}$ and $T_{j,b}$ are components of tiles. Here, instead of calculating the Euclidean distance i.e. cost matrix, we store the difference between all red, green and blue components of only the collected tiles. RS_{sum} is used to store sum of differences of the other three equations.

B. Algorithm description

The algorithm generates photomosaics with limited reuse of tiles. In the algorithm, assign is a two-dimensional array containing the tile numbers assigned to the photomosaic. The counter array keeps track of the number of times a tile has been repeated. The equations of the selection criteria are used for selection of tiles.

```
Target ← input image
Patches ← target
Tiles ← fromdb()
Declare a "selected" array to store the patch
with all its collected tiles with their RGB
differences.
For i=1….no_of_patches
For j=1….no_of_tiles
Calculate RS_R, RS_G, RS_B
If (RS_R, RS_G, RS_B <individual range)
Selected(select) ← RS_R+ RS_G +RS_B
End if
Declare threshold value and counter to store
the no of a times a tile is repeated.
For i=1…no_of_patches
From selected array select the tile with
minimum difference with each patch.
End if
Assign the tiles to patches
Mosaic ← from assign array.
```

IV. EXPERIMENTAL RESULTS

We show the results on a iPod reference photograph (Figure 2, below left). A total of four configurations were studied by varying the size of the patches and the number of available tiles collected based on the threshold. For convenience, the target image was resized in each case so that the patch resolution matches the one of the tiles. As for the colour similarity function evaluation (Equation 1), the target image was partitioned into a rectangular grids of 10×10 and 5×5 pixels each. The initial RS parameters were determined, so this experimental result shows the difference of generating time and visual quality. The four configurations studied are enumerated below, with qualitative information available in Figure 2:

1) 5×5 = 6400 patches with threshold=20 are selected from database of tiles; target image resized to 400×400 pixels.

2) 5×5 = 6400 patches with threshold=40 are selected from database of tiles; target image resized to 400×400 pixels.

3) 10×10 = 1600 patches with threshold=20 are selected from database of tiles; target image resized to 400×400 pixels,

4) 10×10 = 1600 patches with threshold=40 are selected from database of tiles; target image resized to 400×400 pixels.

If the range is selected very low, then sometimes it so happens that no tile will be obtained for a particular patch. At the same time, if the range is very large, a large number of tiles will be obtained for a patch, from which selecting one will be difficult. Also, it may affect the time complexity.

The results show that when the threshold value is too small, the quality of the mosaic obtained is not upto satisfaction. Also, if the threshold value is too large, a patchy appearance will be observed in the mosaic. Hence, it is recommended to select the threshold value with care and based on the application.

The aim of this method is that the mosaic created should reflect the target image for which range and threshold should be selected properly.

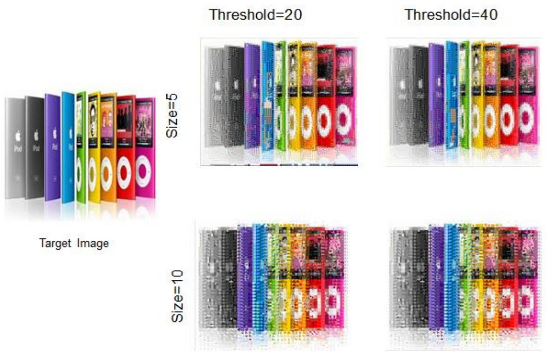

Fig. 2. Summary of experimental results: the leftmost image is the target image; whereas the remaining mosaics are obtained from a database of 1200 tiles by using range based search algorithm, varying the size of patches and the threshold.

V. CONCLUSIONS AND FUTURE WORKS

We have implemented an algorithm for photomosaic generation with limited repetition of tiles. A range search algorithm with user-defined range value has been used to create high quality photomosaics. The performance of the proposed algorithm has been compared with that of greedy based search algorithm. It has been shown that RS is more effective with its approach. Future direction of work would be to come with a better quality mosaic for the image. In this paper, the photomosaic is generated using a pixel-by-pixel difference between patches and tiles In this project we have done mosaicing for coloured images only which can then be later applied for grey scale images. The threshold for repetitions should be selected with care because a large value can make the mosaic biased and if proper threshold is selected then the mosaic will resemble closely to the target image. The database of tiles should contain a variety of images.

ACKNOWLEDGMENT

The authors are grateful to the anonymous reviewers for useful suggestions and comments.

REFERENCES

[1] Mikamo, M.; Slomp, M.; Yanase, S.; Raytchev, B.; Tamaki, T.; Kaneda, K "Maximizing Image Utilization in Photomosaics" Networking and Computing (ICNC), 2010 First International Conference Publication Year: 2010.

[2] Harikrishna Narasimhan, Sanjeev Satheesh "A Randomized Iterative Improvement Algorithm for Photomosaic Generation" 2009 World Congress on Nature & Biologically Inspired Computing Publication Year: 2009.

[3] B. J. Meier, "Painterly rendering for animation," in SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. New York, NY, USA: ACM, 1996, pp. 477–484.

[4] S. Battiato, G. D. Blasi, G. M. Farinella, and G. Gallo, "Digital mosaic frameworks- an overview," Eurographics - Computer Graphic Forum, vol. 26, no. 4, pp. 794–812, 2007.

[5] K. Knowlton, "DominoPix," US Patent No. 4398890, Representation of Designs, 1983, http://www.metron.com/DominoPix, Visited 25-Jul-2009.

[6] R. Silvers and M. Hawley, "Photomosaics," Henry Holt, New York,1997.

[7] S. Battiato, G. Di Blasi, G. M. Farinella, and G.A. Gallo, "Survey of digital mosaic techniques," Proc. Eurographics Italian Chapter Conference, 2006.

[8] N. Tran, "Generating photomosaics: an empirical study," Proc. 1999 ACM Symposium on Applied Computing San Antonio (SAC '99),Texas, United States, Feb. 28 – Mar. 02, 1999. SAC '99, ACM, NewYork, NY, pp. 105-109.