Fast Photomosaic

Gianpiero Di Blasi Maria Petralia

D.M.I. - University of Catania
Via A. Doria, 6
95125, Catania, Italy

gdiblasi@dmi.unict.it ankh76@virgilio.it

ABSTRACT

Photomosaic is a technique which transforms an input image into a rectangular grid of thumbnail images preserving the overall appearance. The typical photomosaic algorithm searches from a large database of images one picture that approximates a block of pixels in the main image. Since the quality of the output depends on the size of the database, it turns out that the bottleneck in each photomosaic algorithm is the searching process. In this paper we present a technique to speed-up this critical phase using the Antipole Tree Data Structure. This improvement allows the use of larger databases without requiring much longer processing time.

Keywords
Photomosaic, Antipole tree, non-photorealistic rendering, image processing and enhancement

1. INTRODUCTION

Photomosaic [Sil97] is a technique which transforms an input image into a rectangular grid of thumbnail images.
The bottleneck in each photomosaic algorithm is the search in a large database of images to find a best match. This search is usually sequential and the time required to perform this task is, hence, high. Some photomosaic algorithm relies on the off-line construction of a suitable data structure to speed-up the matching process. However the overall time required to create this structure could be excessive.
In this paper we propose a new method to speed-up the search process. This technique is based on the Antipole Tree Data Structure [Can04] and allows us to obtain impressive effects in an efficient manner. The Antipole Tree is suitable for searches over large record sets embedded into a metric space (X, d).
The rest of this paper is organized as follows: Section 2 introduces the Antipole Clustering Strategy, Section 3 explains our algorithm. In Section 4 we show the experimental results. Finally in Section 5 we suggest directions for future work and research.

POSTERS proceedings ISBN 80-903100-8-7
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.
Copyright UNION Agency – Science Press

2. THE ANTIPOLE CLUSTERING STRATEGY

The Antipole Clustering Strategy of bounded radius is a top-down procedure that starts with a given finite set of points X in a metric space. The first check is if there exists a pair of points in X such that their distance is longer than the radius. If this is the case, the set is partitioned by assigning each point of the splitting subset to the closest endpoint of the pair (A, B). Otherwise the splitting is not performed and the given subset is itself a cluster.
Once the data structure is built a suitable nearest neighbor algorithm can be designed. The search, starting from the root, proceeds by following the path in the tree, which guarantees to find the nearest cluster centroid pruning the impossible branches. A backtracking search explores the remaining branches of the tree to assure a correct answer.

3. THE PROPOSED ALGORITHM

The proposed algorithm can be divided into two steps: database acquisition and photomosaic creation.

3.1 Database Acquisition

The acquisition of the database of images is very simple: we partition each image of the database into 9 equal rectangles arranged in a 3x3 grid and compute the RGB mean values for each rectangle. This leads us to a vector x composed by 27 components (three RGB components for each rectangle). x is the feature vector of the image in the data structure. When all the images in the database have their own feature vector the Antipole clustering can be performed.

3.2 Photomosaic Creation

The photomosaic creation is very simple and easy to explain in few steps. First we subdivide the input image into a regular grid, then each cell of the grid into another 3x3 sub-grid. Second we compute the RGB mean values for each sub-cell of the sub-grid. This leads us to a vector x composed by 27 components (three RGB components for each sub-cell). x is the feature vector of the cell. After performing the best matching we resize the selected tile to fit and paint it over the cell.

4. EXPERIMENTAL RESULTS

In this Section we report some examples and quantitative results. All experiments have been carried out on a PC Athlon XP-M 1800+. To allow the reader to test directly the quality of our algorithm an applet is available at the URL [Dib04].

The examples in Figure 1 show the effectiveness of the proposed technique. Timing results (Table 1) show that our algorithm is fast enough to be used as a plug-in in a typical user-end software.

5. CONCLUSIONS & FUTURE WORK

In this paper we presented a new method to speed-up the creation of photomosaic images. Experimental results show the soundness of our algorithm.

There are several ways to improve the aesthetic of our results and several ideas started from this work:

1. extending photomosaic technique to other kind of mosaics as proposed in [Dib04b] and [Elb03];
2. the use of Antipole Tree or other data structures in other fields of non-photorealistic rendering to speed-up the rendering process, for example the JIM algorithm proposed in [Kim02];
3. extension of our method for photomosaic rendering of 3D surface is probably the most exciting direction of research.

6. REFERENCES

[Can04] Cantone D., Ferro A., Pulvirenti A., Reforgiato Recupero D., Shasha D. Antipole Tree indexing to support range search and K-nearest neighbor search in metric spaces. Accepted to IEEE Transactions on Knowledge and Data Engineering, 2004

[Dib04] Di Blasi G., Petralia M. The Photomosaic Creator applet.
www.dmi.unict.it/~gdiblasi/photomosaic/photomosaic.html
JGimp plug-in and Java application
www.dmi.unict.it/~gdiblasi/photomosaic/photomosaic.jar

[Dib04b] Di Blasi G., Gallo G. Artificial Mosaic. Submitted to The Visual Computer, 2004

[Elb03] Elber E, Wolberg G. Rendering Traditional Mosaics. The Visual Computer 19, pp 67-78, 2003

[Kim02] Kim J. and Pellacini F. Jigsaw Image Mosaics. In proceedings of SIGGRAPH2002, pp 657-664, 2002

[Sil97] Silvers R. and Hawley M. Photomosaics. Henry Holt, New York, 1997

Size	Total Mean Time (sec.)	Size	Total Mean Time (sec.)
320x240	5.980	800x600	19.058
640x480	16.044	1024x768	32.487

Table 1. Timing results

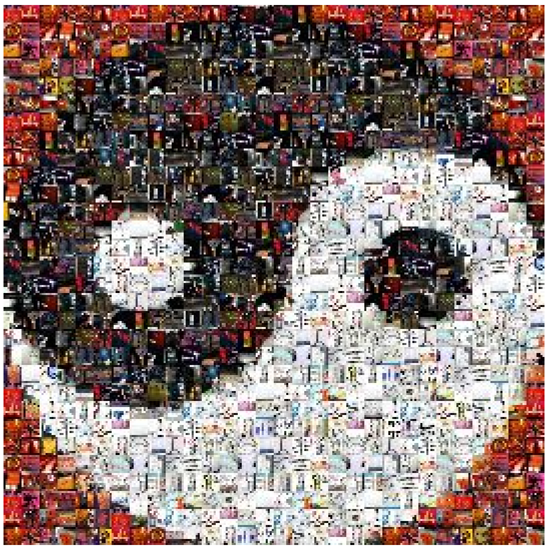
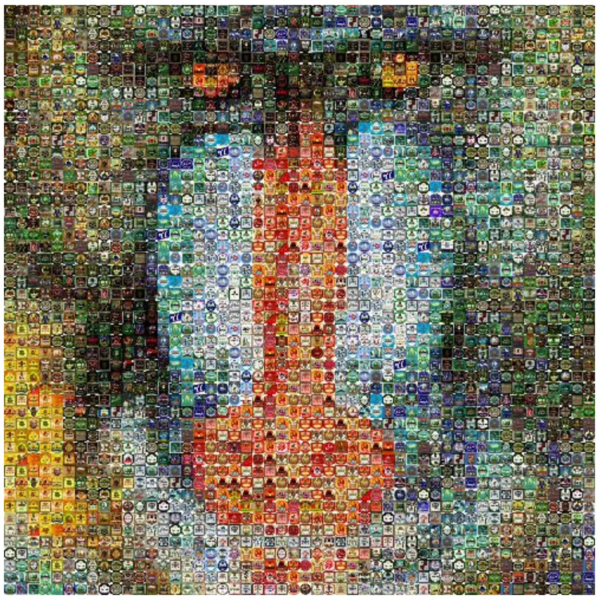

Figure 1. Some examples of our photomosaic algorithm