BUILDING IMAGE MOSAICS: AN APPLICATION OF CONTENT-BASED IMAGE RETRIEVAL

Yue Zhang *Mario A. Nascimento* *Osmar R. Zaïane*

Dept. of Computing Science, Univ. of Alberta, Canada
{zach | mn | zaiane }@cs.ualberta.ca

ABSTRACT

An image mosaic is an image made up of many other images. In this paper we investigate the automatic generation of such image mosaics, using content-based image retrieval as the underlying framework. Our first contribution is to describe and evaluate a few parameters that control the quality of the mosaic image. Our second contribution is the proposal of an (automatic) measure to assess the quality of the resulting images. Several examples of mosaics built in the context of this research can be found at `http://db.cs.ualberta.ca/Mosaicture`.

1. INTRODUCTION

Using many images (or objects) to build a single image has a long history. Back in the 16th century, the painter Giuseppe Arcimboldo used vegetables, fruit, and flowers images to build image mosaics[1]. In the late 1970s, Salvador Dali created a well known image montage of Abraham Lincoln by putting many other images together including his own wife's picture[2]. More recently, Robert Silvers, devised and pattended a process to generate image mosaics (which he named "Photomosaic") [4] – unfortunately, due its commercial interest, not much is known about his process.

In general, an image mosaic is generated based on an original image, with the requirement that the image mosaic remains visually similar to the original image. A method to generate an image mosaic can be described as follows: (1) divide the original image into many tiles, (2) for each such tile, find a visually similar image from an existing (and sufficiently large) image database, and (3) build the image mosaic by replacing all tiles by their similar image counterparts. Let us illustrate the process by using a simple example. Consider the image in Figure 1(a) as input, i.e., the original image. In step (1) of the process, the original image is tiled (Figure 1(b)). Next, one should search an image database for image(s) which is(are) similar to each tile. In our example, Figure 1(c) depicts some candidates for a few of the target image's tiles. Then, in step (3) all tiles are replaced, resulting in a mosaic (Figure 1(d).

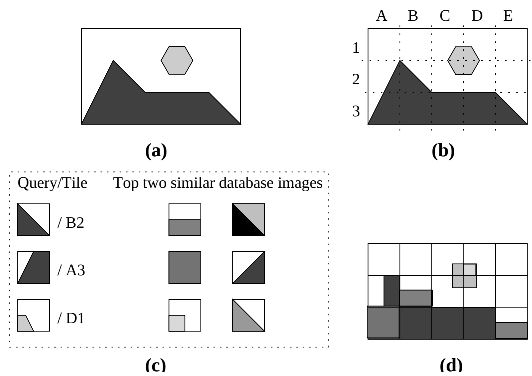

Figure 1: Mosaic Construction Process

In this paper we tackle the problem in step (2), i.e., we investigate the use of content-based image retrieval (CBIR) techniques to rebuild the desired image. In particular we discuss some of the factors elected to be more important when generating image mosaics and how they interact and affect the overall quality of the resulting image. To address the latter issue we also propose an automatic methodology to evaluate a mosaic's quality.

Since there are few image mosaic generating methodologies and most implementations are not thoroughly studied from an academic point of view, we have chosen not to focus on the comparison of our image mosaic generating methodology with others. Instead, we delve into our methodology and analyze the different roles and effects of a variety of factors in the system. Thus, we aim at shedding some light onto the mosaic generation process itself.

This paper is divided as follows. In Section 2 we detail our proposed methodology, and all parameters and implementation details involved. Next, in Section 3, we propose and discuss an automatic mosaic quality evaluation methodology. Section 4 summarizes and discusses some sample experiments we performed, and, finally, we conclude the paper in Section 5 with a discussion of our findings.

Research partially supported by NSERC Canada.

[1] http://www.louvre.fr/francais/audito/20022003/oed/arcimbol.htm

[2] http://www.raymondjames.com/art/images/dali.jpg

2. AN APPROACH FOR BUILDING IMAGE MOSAICS

Due to space constraints we cannot offer an overview of the current state-of-the-art of the content-based image retrieval (CBIR) area. Instead we refer the reader to a recent survey on colour-based image retrieval [5] and a recent book by Castelli and Bergman [1] for a more general reference which covers most of the area. Nonetheless a brief idea of how CBIR works in the context of this paper is warranted.

We assume the Query-by-Example paradigm, i.e., given a large image database the query goal is to find images that are similar to the query image and rank those accordingly. It is commonly accepted in the CBIR literature that colour does play an important role in assessing the similarity between images. It is undeniable that the colour feature by itself can hardly capture image semantics. Nonetheless, for the purpose of our work, image semantics can be safely ignored since the ultimate goal is to obtain an image mosaic that visually resembles the original image.

We have identified five main variables that could play a role in the quality of an image mosaic: (1) size of the image database, (2) colour quantization, (3) image tessellation scheme, (4) number of candidates/tile considered, and (5) how to chose the best of such candidates. Even though we have investigated thoroughly all of these five aspects elsewhere [8], due to space limitations we will detail in this paper only the third and fifth ones. A brief summary of the influence of the other parameter are as follows. The larger the database the better the result and the slower the image selection process – though we must note that for the experiments we performed, a database of about 13,000 images yielded results nearly as good as those using another database three times as large. This can be explained by the fact that the smaller database was already diverse enough in term of colour contents. As for colour quantization, using the RGB colour space with 64 colours, yielded much better results (though slightly slower) than when using 8 or 27 colours, as one would expect. When replacing a tile, we consider τ lists of image candidates depending upon the tessellation scheme (discussed next). We fixed the number of candidates per list at 30, which seemed to be a good compromise between processing time (larger with the increase in the number of candidates) and mosaic quality.

Next we detail how the tile replacement takes place. There are basically two phases in this process: how to model similarity between tiles and how to rank the candidate images. Before dealing with those issues, an important observation to be made when looking for a tile replacement is that a perfect tile match may not be as desirable if it does not visually "flow" with its neighbours. Again, let us resort to a visual example. Consider Figure 2(a) and its three tiles: A, B and C. It should be rather clear that Figure 2(b) is more similar to Figure 2(a) than Figure 2(c). However, looking at their colour composition they are all identical. It is the spatial composition and flow of the colours in Figure 2(b) that causes it to be an overall better match to Figure 2(a).

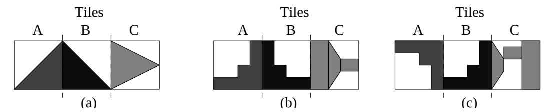

Figure 2: Visual flow matters.

In order to take such visual flow into account, we decompose the tiles into τ partitions. We explore two alternatives with $\tau = 4$ and $\tau = 5$. In the first one we decompose the tile in four overlapping partitions, i.e., the upper, the bottom, the left and the right ones. We refer to this approach as 4-tessellation. In the second alternative, called 5-tessellation, we consider five regions, four similarly as in the first case, with the addition of a centre tile which is not overlapped by the other ones. Figures 3(b) and (c) show examples of the two schemes applied to Figure 3(a) Therefore, given a tile from the target image, it is tessellated as described above and compared to the corresponding tessellations of all images stored in the databases[3].

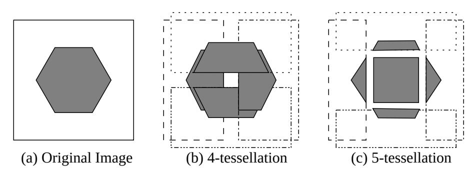

Figure 3: Illustration of used tessellation schemes.

This brings the issue of tile similarity. For the sake of argument let us consider each tile partition obtained in the tessellation stage an image by itself. Since we are mainly concerned with the visual (colour) aspect of the tiles, we compare only the colour contents of the images. For that we used the approach proposed in [2] which has been shown to be more effective and efficient than the use of traditional global colour histograms (GCHs) [6] and colour coherence vectors (CCVs) [3]. The idea is to use a non-linear discretization of the colour bins (obtained as in GCH) which reflects the non-linear human response to many stimuli. The obtained feature vector can be efficiently encoded in a compact bit-string signature, e.g., one could encode the (colour) signatures of 100,000 images in only 4 Mbytes. More importantly, the use of such signatures lends itself to efficient indexing via Signature Trees. Note that the use of GCHs (or CCVs) would likely imply the use of spatial indices, which are notably not efficient in high-dimensions, e.g., those we

[3]We assume that all database images are pre-processed (tessellated) beforehand, i.e., off-line.

consider in this work. Furthermore efficiency is an aspect of utmost importance to be considered since the task of finding similar images will be performed once for each of the four partitions of all (64 $\times$ 64) tiles in the target image. E.g., assuming the 4-tessellation tile partition scheme, for each original image we need to query the image database 16,384 times! It is noteworthy pointing out that more complex image similarity measures, presented elsewhere, could be used. However, we claim that they would be hardly worth it since, in the scope of this work, the overall result dominates that of the individual ones, and total mosaic construction time grows fast with the complexity of the individual image matching process.

Recall that for every tile, four or five partitions are obtained by the tessellation process, and for each partition a list of best candidate images (with respect to that particular partition) is obtained. The question now becomes how to chose the best candidate to replace the whole tile. We have designed and experimented two techniques, MOSS and DOSS. DOSS selects as best candidate the image that appeared most often in all lists regardless of its rank within each list. MOSS instead uses the individual list's ranks. The idea is that the image with the minimum sum (over all lists) of the distances between the candidates and the tile is the best candidate.

At this point we are able to, given a target image, tile it, and retrieve for each such tile the best candidate. By doing so, we can build an image mosaic. The next question is: how to measure the quality of the obtained mosaic?

3. QUALITY ASSESSMENT MEASURE

To our knowledge, the only published proposal for assessing the quality of an image mosaic with respect to the original target image is due to Tran [7]. His idea was to measure the similarity as a direct ratio of how physically close one would have to get to note differences between the original and the mosaic images, i.e., the closer one would need to get the better the mosaic quality. Although fairly intuitive, his proposal suffers from a major drawback. It requires human subjects to perform the evaluation, who, by their very nature, may be inclined to bias their judgment based on their subjective interpretation of quality. We propose an *automatic* human-independent measure, and discuss its correspondence with Tran's proposal. As a by-product we can also perform a much larger number of experiments since no human intervention is necessary.

Our reasoning is as follows. Consider an original image I and two mosaics, $M1$ and $M2$, of the same image but obtained with a different set of parameters. If when one looks increasingly closer at all three images, and one mosaic, say $M1$, becomes more different faster, then one can say that the other mosaic $M2$ is better (or vice-versa). In order to mimic this observation, we first calculate the GCH distance between the mosaic and the original image according to the L_2 norm [6]. We then tile both images and obtain their average GCH distances, over all corresponding pair of tiles. (In our work, we have tiled both images into 5x5 blocks.) Finally, we use the difference between these two distances as the distance between the original image and the mosaic. The smaller this distance is, the better an image mosaic is.

Other possibilities for assessing a mosaic's quality could be trying compare them to the original image on a pixel-by-pixel basis or using image segmentation approaches. The former clearly does not work since we are mostly interested in the overall "look-and-feel" rather than microscopic details. The latter did not succeed in our experiments since the particular segmentation techniques we used were not very sensitive to noise, i.e., all mosaics were nearly as good (or nearly as bad).

4. EXPERIMENTAL RESULTS

The C programming language, csh scripts, and ImageMagick APIs[4] were used to implement our image mosaic generating system under Linux OS. All the image mosaics in our experiments were generated using a PC with a 550MHz Celeron CPU and 128 Mbytes of main memory. We used a test image set consisting of 33 different images to perform the experiments, and their average result values are reported in the analysis presented below. These 33 colour images were selected from the "Corel GALLERY 1,000,000" CDROM. They had different semantic content and colour distributions in order to avoid any bias. In addition, they are all of the same dimension, which ensures that the processing time and image mosaic's quality is dependent only on the different parameters being used.

We now discuss the implications of the variables we discussed above (Section 2) in terms of mosaic building time and quality (Section 3). Due to lack of space we give an overview of the results we obtained. Unless otherwise noted, all reported figures were obtained using a database of 13,846 images, all colours were quantized into 64 RGB colours and for each tile partition a list of 30 candidates was considered. A thorough investigation of all parameters (including some not discussed here) as well as a large sample of image mosaics we have built can be found in [8].

- Tessellation Scheme

  Using the 4-tessellation scheme consumed about 197 minutes, with the 5-tessellation scheme taking only 8% more time. While one could think that it should take 20% more time since it has 20% more tiles to process, one must remember that in the former scheme the tiles contain more pixels to be processed than in

[4] http://www.imagemagick.org

the former. Hence the smaller gain. In terms of quality the average distance to the original images yielded by the 5-tessellation scheme was slightly smaller than that yielded by the 4-tessellation scheme. This suggests that although not very representative, some savings in processing time could be enjoyed by using the 4-tessellation scheme without much loss in the final mosaic's quality.

- Tile Replacement Selection

  In this case, neither approach showed a representative advantage in terms of query processing time, both required nearly the same 196 minutes in average to build a mosaic. When comparing the quality though, the MOSS scheme produced mosaics with an average distance approximately 25% better (i.e., closer to the original image) than the mosaics obtained by using the DOSS approach. Hence, using the MOSS approach is the clear choice to be made.

Before we conclude this section there is a particular implementation issue we should discuss: caching. The question is whether the use of caching would help speeding up the retrieval process. The intuition is positive since it is reasonable to expect that many neighbouring tiles are virtually identical, e.g., parts of the same texture/background or a large similarly coloured region. Hence, once one tile is retrieved one could avoid searching the database by simply reusing a recently fetched image. In our experiments an average of nearly half of the tiles in the target image were unique with respect to their abstraction, i.e., the (colour-based) bit-strings. That means that a perfect caching technique would save nearly half of the query processing time. Unfortunately this is not the case. The use of our best performing cache technique (with a hit ratio of 96%) was able to save no more than 20% of processing time. Although apparently surprising there is a reasonable explanation for this result. The image querying step, when caching is an issue, is only one of the many steps involved in the candidate tile retrieval, and it is not the most expensive one. Since the original image is not known beforehand, all other tasks (e.g., image tiling, tessellation, colour abstraction, etc.) have to be done on-line and the savings by using the cache are overshadowed. Nevertheless we did use caching in our implementation, and all results above take that into account.

5. CONCLUSIONS

We have presented a clear methodology for mosaic generation, i.e., given an image one can build a mosaic image visually similar to the original one but built only with other images. We discussed and evaluated some parameters that would affect a mosaic's processing time as well as quality. In summary, we came to the conclusion that[5]: (1) the larger image database the better, but what really matters is whether the database is sufficiently diverse colour-wise; (2) colours should be quantized so that they can capture the diversity of the original image's colours; (3) among the two approaches we used for exploring the image's colours location, the 4-tessellation scheme offers a good compromise in terms of processing speed and final mosaic quality; (4) the cardinality of the candidate image set for a given tile plays little importance in terms of mosaic quality but has non-negligible effect in terms of processing time, and a list as small as 30 yielded good results; and finally (5) the way of choosing the best candidate for a tile is important regarding the overall quality, with MOSS being the best of the methods we tried, but bearing little important regarding processing time. It is also worth noting that we have devised a totally automatic measure for a mosaic's quality while still attempting to mimic human judgment.

Regarding future directions we believe further experimental work could be done using colour spaces different than RGB's, devising other tessellation schemes (e.g., using overlapping partitions), using indices faster than the Signature Tree and/or other abstractions for the colour content of images, and incorporating the possibility of avoiding an image to be used repeatedly (or close to another instance of itself) too often.

6. REFERENCES

[1] V. Castelli and L. D. Bergman, editors. *Image Databases: Search and Retrieval of Digital Imagery*. Wiley, 2001.

[2] M. A. Nascimento et al. Image indexing and retrieval using signature trees. *Data and Knowledge Engineering Journal*, 43(1):57–77, 2002.

[3] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *Proc. of ACM Multimedia Intl. Conf.*, pages 65–73, 1996.

[4] R. Silvers. Digital composition of a mosaic image, Oct. 2000. US Patent and Trademark Office. http://www.uspto.gov.

[5] R. Stehling, M. Nascimento, and A. Falcão. Techniques for color-based image retrieval. In C. Djeraba, editor, *Multimedia Mining - A Highway to Intelligent Multimedia Documents*. Kluwer Academics, 2002.

[6] M. J. Swain and D. H. Ballard. Colour indexing. *International Journal of computer vision*, 7:11–32, 1991.

[7] N. Tran. Generating photomosaics: An empirical study. In *Proceedings of the 1999 ACM Symposium on Applied computing*, pages 105–109, 1999.

[8] Y. Zhang. On the use of CBIR in image mosaic generation. Master thesis, Computing Science Department, University of Alberta, 2002. (Available on-line via http://db.cs.ualberta.ca/Mosaicture/).

[5] Some conclusions are based on experiments not fully discussed in this paper but elsewhere [8].