Computationally Efficient Image Mosaicing Using Spanning Tree Representations

Nikos Nikolaidis and Ioannis Pitas

Artificial Intelligence and Information Analysis Laboratory,
Department of Informatics, Aristotle University of Thessaloniki, GR-54124
Thessaloniki, Greece
{nikolaid,pitas}@zeus.csd.auth.gr
WWW home page: http://poseidon.csd.auth.gr

Abstract. Optimal image mosaicing has large computational complexity, that becomes prohibitive as the number of sub-images increases. Two methods are proposed, which require less computation time by performing mosaicing in pairs of two sub-images at a time, without significant reconstruction losses, as evidenced by simulation results.

1 Introduction

Very high resolution digital image acquisition is a process that stresses the limits of acquisition devices. Even high-end CCD devices can not offer the level of detail that is required in certain applications. To overcome this obstacle, digitization structures have been proposed that utilize observation of different, albeit overlapping, fields of view (sub-images), sometimes with the aid of sensor/detector arrays, and positioning mechanisms. Mosaicing is a process which is used to reconstruct or re-stitch a single, continuous image from a set of overlapping images. Several mosaicing techniques have been proposed in the literature [1–5]. Image mosaicing is essential for the creation of high-resolution large-scale panoramas for virtual environments. Image mosaicing is also important in other areas that include image-based rendering, creation of high resolution digital images of architectural monuments and works of art (especially of those with considerable dimensions like frescoes and large-size paintings) for archival purposes (Fig. 1) and digital painting restoration, medical imaging [1] aerial and satellite imaging etc [2]. If the field of view is split into M_1 rows of M_2 images, it should be trivial to show that an M_1M_2-fold increase in resolution may be attained, compared to sensor resolution.

The mosaicing process may be broken down into two steps. The first step involves the estimation of optimal displacement of each sub-image with respect to each neighboring one (assuming only translational camera motion and no rotation or zooming). This represents the most computationally intensive part of the entire process. In the general case of an M_1 by M_2 image mosaic, a search should be performed in a m-dimensional space, where $m = 2(2M_1M_2 - M_1 - M_2)$. The term in parenthesis represents the number of all pairs of neighboring sub-images.

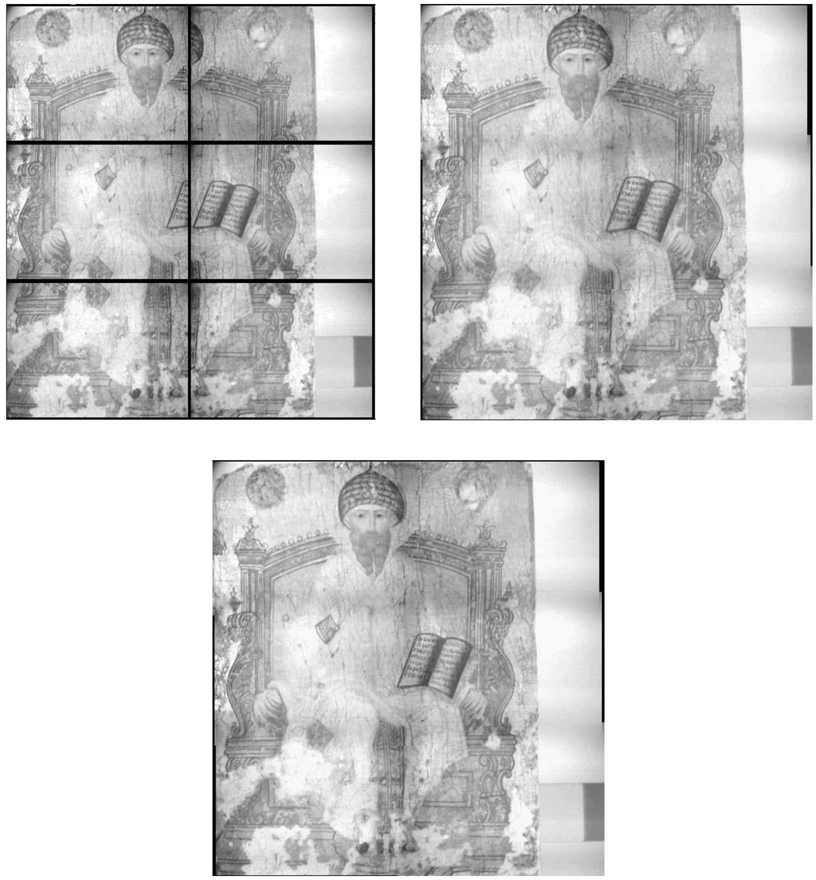

Fig. 1. (a) A $M_1 = 3$ by $M_2 = 2$ sub-image acquisition of a painting. (b) Reconstructed images after STM processing. (c) Reconstructed images after SGSTM processing.

Since these are 2-d searches, this term is multiplied by two. It is evident that computation cost becomes prohibitive, as the number of sub-images increases. Optimal displacement is researched, under the assumption that acquired images are not free of distortions [3]. The second step of the mosaicing process utilizes the previously generated displacement information in order to combine each pair of neighboring sub-images with invisible seams.

This paper shall focus on the first step. The proposed methods attempt to reduce the number of computations required to compute the sub-image displacements. Despite the fact that the methods are illustrated for the particular case where sub-images are only displaced (translated) with respect to each other, the proposed matching methodology is applicable to more complex cases, e.g. cases that involve camera rotation or zooming.

2 Mosaicing Techniques for Two Images

Before we proceed to the general case of mosaicing an arbitrary number of sub-images, the case of two images should be studied first, since it provides significant insight to the problem. In the following it is assumed that the displacement vector $\mathbf{d}$ is constrained to take values in the following set:

$$\mathbf{d} \in \{[d_1 \ d_2]^T : \ d_i \in \{d_{i_{\min}}, \ldots, d_{i_{\max}}\}, \ \ i = 1, 2\} \tag{1}$$

If $I_j(\mathbf{n})$, $j = 1, \ 2$, is the intensity of the j-th image at pixel coordinates $\mathbf{n} = [n_1 \ n_2]^T \in W(\mathbf{d})$, where $W(\mathbf{d})$ denotes the overlap area, then a quantitative expression for the matching error $E(\mathbf{d})$, which is associated with a specific displacement $\mathbf{d}$, can be derived as follows:

$$E(\mathbf{d}) = \frac{\sum\limits_{\mathbf{n} \in W(\mathbf{d})} |I_1(\mathbf{n}) - I_2(\mathbf{n})|^p}{||W(\mathbf{d})||} \tag{2}$$

where $||W(\mathbf{d})||$ denotes the number of pixels in the overlap area $W(\mathbf{d})$. For $p = 1, 2$ (2) expresses the Matching Mean Absolute Error $E_{\mathrm{M}MAE}$ and the Matching Mean Square Error $E_{\mathrm{M}MSE}$, respectively. Subsequently, an optimal value $\mathbf{d}_{opt}$ for the displacement can be estimated as follows:

$$\mathbf{d}_{opt} = \mathrm{a}rg\min_{\mathbf{d}} E(\mathbf{d}) \tag{3}$$

From (1) and (3) it should be evident that this minimization process requires a repeated evaluation of (2) over all possible values of $\mathbf{d}$. Since matching error calculation is the most computationally intensive part of the mosaicing process, alternative forms of (3) should be researched. Block matching techniques can be employed in order to avoid the computation cost which is associated with the exhaustive minimization procedure, which is implied by (3). In this context, procedures such as the 2-d logarithmic search, the three-point search and the conjugate gradient procedure may be utilized for this purpose [6]. These procedures may provide estimates $\hat{\mathbf{d}}_{opt}$ of the optimal displacement value $\mathbf{d}_{opt}$. The 2-d logarithmic search was employed throughout our simulations.

3 Spanning Tree Mosaicing of Multiple Images

Let us suppose that $M_1 \times M_2$ sub-images should be mosaiced. In this case, a displacement matrix $\mathbf{D}$ plays a role similar to that of the displacement vector $\mathbf{d}$ of Sect. 2. The $2M_1M_2 - M_1 - M_2$ columns of $\mathbf{D}$ are 2-dimensional vectors, each one corresponding to a displacement value between two neighboring sub-images. An expression for the quality of matching, similar to the two-image case, can be derived in the multiple image case, by substituting $\mathbf{d}$ with $\mathbf{D}$ in (2) and extending the summation over all neighboring images. The optimal value $\mathbf{D}_{opt}$ of the displacement matrix can be derived from the following expression:

$$\mathbf{D}_{opt} = \mathrm{a}rg\min_{\mathbf{D}} E(\mathbf{D}) \tag{4}$$

Unfortunately, (4) imposes prohibitive computational requirements, since the search is now performed in a much larger space. Indeed, let us suppose that for each one of the column vectors of $\mathbf{D}$, eq (1) holds. It can be shown that $\mathbf{D}$ may assume $\left((d_{1_{\max}} - d_{1_{\min}} + 1)(d_{2_{\max}} - d_{2_{\min}} + 1)\right)^{2M_1M_2-M_1-M_2}$ different values. Thus, computational complexity increases exponentially. Additionally, calculation of $E(\mathbf{D})$ poses other computation problems, since the overlap area W is now a multi-dimensional set.

In order to avoid this exhaustive matching process, certain constraints can be imposed on the way images are matched. Indeed, a faster method may be devised by performing simple matches only, i.e. matches between an image and one of its neighbors. The proposed method may be easily understood with the aid of a mosaicing example. In Fig. 2 (a) a mosaic of $M_1 = 2$ by $M_2 = 2$ sub-images is depicted. By associating each image with a graph node and each local matching of two sub-images with an edge, the mosaicing of the four images can be described by the graph of Fig. 2 (b). Computation of $\mathbf{D}_{opt}$ requires an exhaustive search in 8-dimensional space. To avoid this complexity, the entire mosaicing process is decomposed into simpler steps of mosaicing two images at a time. *Spanning trees* offer an elegant representation of the possible mosaicing procedures, under this constraint. Figs. 2 (c)-(f) illustrate the four spanning trees that correspond to the graph of Fig. 2 (b). For example, in the case depicted in Fig. 2 (c) three two-image matches should be performed: image A to B, A to C, and C to D, while in Figs. 2 (d)-(f) the other three possible mosaicing procedures are illustrated. The final resulting image is the one produced by the procedure which is associated with the smallest matching error. It should be obvious that this sub-optimal procedure offers a significant decrease in computational complexity. The number of trees of a graph can be calculated by the matrix-tree Theorem [7]:

Theorem 1 (Matrix-Tree Theorem). *Let G be a non-trivial graph with adjacency array $\mathbf{A}$ and degree array $\mathbf{C}$. The number of the discrete spanning trees of G is equal with each cofactor of array $\mathbf{C} - \mathbf{A}$.*

Both $\mathbf{A}$ and $\mathbf{C}$ are matrices of size $(M_1M_2) \times (M_1M_2)$. If node v_i is adjacent to node v_j, $\mathbf{A}(i,j) = 1$, otherwise $\mathbf{A}(i,j) = 0$. Additionally, the degree matrix is of the form:

$$\mathbf{C} = \mathrm{d}iag(d(v_1), \ldots, d(v_{M_1M_2})) \tag{5}$$

where $d(v_i)$ denotes the number of nodes adjacent to v_i.

The spanning tree mosaicing (STM) procedure is outlined below:

1. For each pair of neighboring images, calculate the optimal displacement and the associated matching error. Notice, that each pair of neighboring images corresponds to an edge of the graph.
2. For each spanning tree that is associated with the specific graph, calculate the corresponding matching error, by summing the local matching errors which are associated with the two-image matches depicted by the given tree.
3. Select the tree that is associated with the smallest matching error.
4. Perform mosaicing of two images at a time, following a route of the selected spanning tree.

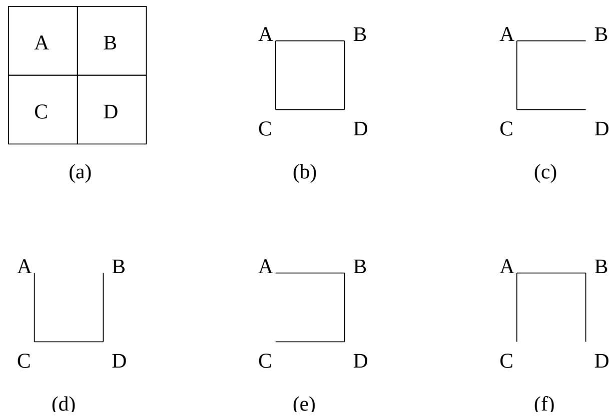

Fig. 2. (a) A labelled mosaic of $M_1 = 2$ by $M_2 = 2$ sub-images, (b) the corresponding graph, (c)-(f) the four possible spanning trees.

It should be clarified that sub-optimal results are obtained when the STM approach is followed. Specifically, an approximation $\hat{\mathbf{D}}_{opt}$ of the optimal matrix is computed. However, this is contemplated by the speed gains provided by the algorithm.

As will be shown in Sect. 5, similar results are obtained by using either the MMAE or the MMSE criterion. Thus, MMAE may be preferred since it is faster to compute.

4 Sub-Graph STM

The number of trees that correspond to graphs of sizes up to 5 by 5 images are tabulated in Table 1. Unfortunately, for large values of M_1 and M_2 this method can not be utilized, since the number of trees grows very fast with respect to grid size.

Table 1. Number of spanning trees in a graph-grid of size $M_1 \times M_2$

		M_2				
		1	**2**	**3**	**4**	**5**
	1	0	1	1	1	1
	2	1	4	15	56	209
M_1	**3**	1	15	192	2415	30305
	4	1	56	2415	100352	4140081
	5	1	209	30305	4140081	5.6×10^8

Sub-graph STM (SGSTM) may, partially, address this issue. In SGSTM, a graph may be partitioned into sub-graphs, by a process that splits the original graph vertically and/or horizontally. In Fig. 3, a sample partitioning of this form is depicted. By splitting vertically first and then horizontally, four sub-graphs are created. STM can be applied separately to each one of the four sub-graphs of Fig. 3 (d). Using data of Table 1, it can be easily shown that a total of $192 + 1 + 0 + 1 = 194$ spanning trees should be examined. Since four images will be produced by the STM process (one for each sub-graph), a further STM step will be required, in order to produce the final image. Thus, 4 more trees should be added to the 194 trees examined in the previous step, to produce a total of 198 trees. In contrast, an STM of the original image set would require matching error calculations in 100352 cases.

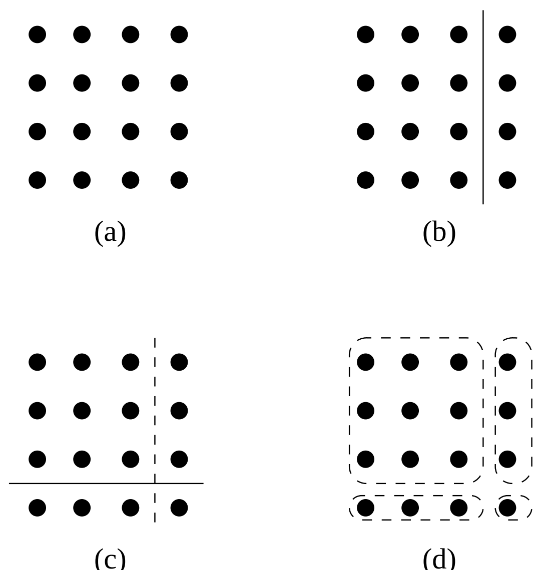

Fig. 3. (a) A graph of $M_1 = 4$ by $M_2 = 4$ image mosaic. (b) Vertical split of (a). (c) Horizontal split of (b). (d) Resulting partition of graph.

If the original graph is of size $M \times M$ ($M = 2^\nu$), the image can be gradually mosaiced by decomposing the original graph into an appropriate number of 2×2 sub-graphs, performing STM on each one, decompose once more the resulting $\frac{M}{2} \times \frac{M}{2}$ graphs and so on, until one image emerges. After mosaicing a partition's sub-graphs, new displacement matrices should be calculated that correspond to the resulting sub-images. It is obvious that the number of 2×2 graphs is equal to $\frac{M}{2}\frac{M}{2} + \frac{M}{4}\frac{M}{4} + \ldots + 1 = \frac{M^2-1}{3}$. Since four spanning trees exist for a 2×2 graph, the matching error of only $4\frac{M^2-1}{3}$ trees should be evaluated. Speedup values are depicted in Fig. 4. It is obvious that SGSTM represents a vast improvement over the STM approach, in terms of computation needs. The results of both methods on an image consisting of 6 sub-images can be seen in Fig. 1. Obviously, SGSTM mosaicing quality may be inferior to the one provided by STM, since a smaller number of possible sub-image displacements is examined.

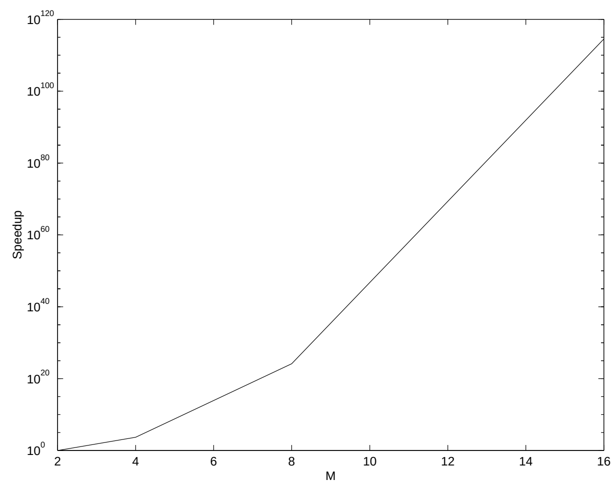

Fig. 4. Theoretical speedup of SGSTM compared to STM, for $M \times M$ ($M = 2^\nu$) graphs.

5 Simulation Results

Simulations were carried out in order to assess the performance of the proposed methods, on several image sets. In the following, comments and results are presented for one of these sets, consisting of 12 sub-images, which were arranged in a grid of $M_1 = 4$ rows of $M_2 = 3$ sub-images each. Each sub-image had a resolution of 951×951 pixels. For this graph 2415 spanning trees exist.

For each one of the $2M_1M_2 - M_1 - M_2 = 17$ pairs of neighboring sub-images, matching errors were calculated under both the MMAE and the MMSE criteria. The 2-d logarithmic search was utilized in order to obtain the optimal displacement. Subsequently, for each one of the 2415 spanning trees, the total matching error (MMAE and MMSE) was calculated.

The total time that was required to find all spanning trees that correspond to the given graph, calculate optimal neighboring image displacements under a specific criterion, and output the overall matching error of each tree is tabulated in Table 2, in seconds. Results are included for both STM and SGSTM approaches.

In the SGSTM case, the graph was decomposed into four sub-graphs: two 2×2 and two 2×1, which required the calculation of overall error for 14 trees, compared to the 2415 of STM. It is evident that SGSTM is more than an order of magnitude faster than STM. More specifically, the speedup provided by the SGSTM method over the STM method was 13.9 when the MMSE criterion was used and 11.4 for the MMAE criterion, albeit with a significant increase in the matching error. Furthermore, the use of MMAE proved to be faster than that of the MMSE. Sample minimum, maximum, mean and variance of matching error, over the entire spanning tree set for the STM approach, are recorded in Table 3.

Table 2. STM and SGSTM performance results under the MMAE and MMSE criteria

	MMAE		*MMSE*	
Method	*Error*	*Time*	*Error*	*Time*
STM	6.4	782.7	107	1265
SGSTM	9.7	68.9	228	91

Table 3. STM matching error statistical measures

Measure	*MMAE*	*MMSE*
Maximum	10.7	378
Minimum	6.4	107
Mean	7.9	186
Variance	0.45	1771

By studying the spanning trees that exhibited the lowest MMAE scores the following observations were made:

- MMAE optimality is closely related to MMSE optimality. Indeed, the trees that exhibited the lowest MMAE figures, exhibited also the lowest MMSE figures.
- The most characteristic feature of the trees with the lowest error figures was that optimal matching began from the center and proceeded outwards. In other words, the central nodes of the graph were connected in the best performing trees. Currently, it is assumed that matching quality of the central nodes is more crucial to the overall mosaicing quality, than matching quality of the other nodes. This issue is currently under investigation.

6 Conclusions

In this paper two novel methods that can be used for the mosaicing of large images were proposed. Spanning trees are utilized for describing the order of the mosaicing process. Since matches are performed between pairs of neighboring images, an exhaustive search for the optimal placement of sub-images with respect to each other is avoided. The SGSTM method offers significant speedup, when compared to the STM method. Despite the fact that this performance improvement comes at the cost of increased matching error, SGSTM can be utilized for fast visualization of mosaicing results (e.g. mosaic previews).

7 Acknowledgement

The presented work was developed within VISNET, a European Network of Excellence (http://www.visnet-noe.org), funded under the European Commission IST FP6 program.

References

1. V. Swarnakar, M. Jeong, R. Wasserman, E. Andres and D Wobschall, "Integrated distortion correction and reconstruction technique for digital mosaic mammography", in *Proc. SPIE Medical Imaging 1997: Image Display*, Vol. 3031, p. 673-681, 1997.
2. H.-Y. Shum and R. Szelisky", Construction of Panoramic Image Mosaics with Global and Local Alignment", *International Journal of Computer Vision*, vol 36, no 2, pp. 101-130, February 2000.
3. H. S. Sawhney and R. Kumar, "True multi-image alignment and its application to mosaicing and lens distortion, in *Proc. CVPR 1997*, pp. 450-456, 1997.
4. W. Puech, A. G. Bors, I. Pitas and J-M. Chassery, "Projection distortion analysis for flattened image mosaicing from straight uniform generalized cylinders", *Pattern Recognition*, vol. 34, no. 8, pp. 1657-1670, August 2001.
5. S. Peleg and J. Herman, " Panoramic Mosaics with VideoBrush", in *Proc IUW-97*, pp. 261-264, May 1997.
6. A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation and Compression*, Plenum Press, 1988.
7. N. L. Biggs, E. K. Lloyd and R. J. Wilson, *Graph Theory 1736-1936*, Clarendon Press, 1986.