Gigapixel Image Mosaicking via Parallel Sparse Tile Decomposition

Hamilton Scott Clouse Xiao Bian
Thanos Gentimis Hamid Krim

April 10, 2014

Abstract

Recently, the rate and quantity at which photographs are being produced and published have greatly increased. Frequently, many of these photos have the same subject, are of the same scene, etc. The ability to combine such images to constitute a single detailed representation would afford the viewer with a succinct and comprehensive understanding of the captured data. In this work, we compute such a mosaic for data captured by the prototype camera AWARE-2. This device produces many high-resolution images of a single scene. Our contribution is a theoretically sound and rigorous approach to combining these high-resolution images into a single consistent gigapixel image.

1 Introduction

Image mosaicking is, like its arts/crafts namesake, the process of combining multiple small tiles/images that, maybe, aren't meaningful on their own, to produce a single larger and complete image. In computer vision, this refers more specifically to the combination of images of a similar/the same scene/subject to produce a more complete representation of that specific scene or subject. Particular instances of this application may be as local as the production of panoramas on an individual's mobile phone or, as global as the combination of images collected from users all over the world and even as broad-reaching as combining astronomical images to create a picture of the cosmos.

Most approaches to image mosaic formation can be described by the following common framework. First, a standardised feature points selection process is applied to each image. This step usually consists of computing some generic feature descriptor; many of which have been developed for such task and successfully employed repeatedly, e.g. SURF [1]. A search must then be performed across all possible pairings of these descriptors. A scoring method, e.g. RANSAC [2], produces a quantitative correspondence between each of the candidate pairings. These values are then compared to compute the best match between some subsets of those points. This set of matching points is used to compute the homography [3] which is subsequently applied to the appropriate images. Of course, variations on the feature descriptor utilized and the scope within which the search for correspondences is performed varies from application to application.

Some preliminary outputs of a naive version of this process are shown in Figure 1. There are discontinuities/misalignments visible in many portions of the constructed mosaic. This construction was performed without the aid of any additional information, i.e. only the images were used as input to the process. In this paper we describe a method for computing more accurate mosaics by utilizing structure in the imaging array as well as in the images themselves. Unlike the feature descriptor based techniques that must be tuned for each application, our global, data-driven technique requires no tuning and is capable of producing the homography between images in parallel.

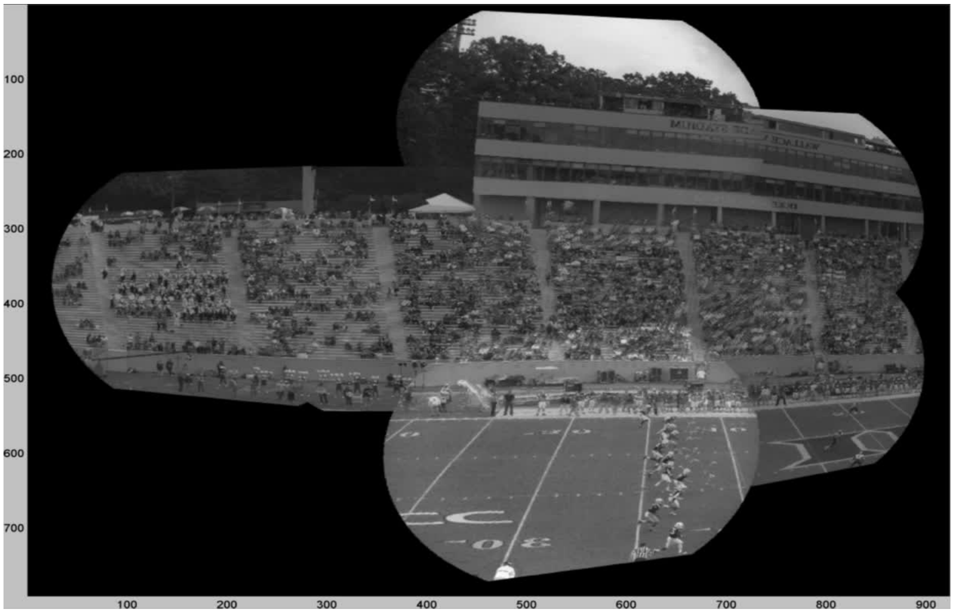


Figure 1: Mosaic output from seven images using current state of the art tools.

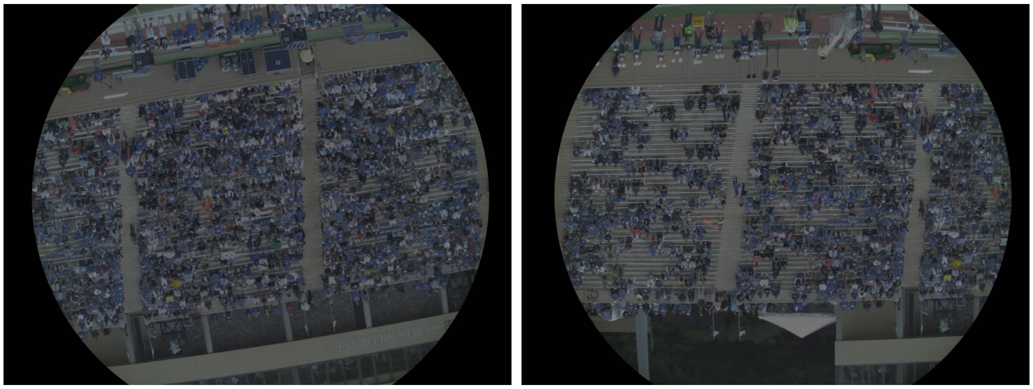
The remainder of the paper is organized as follows: Section 2 covers background material related to the AWARE-2 sensor and the techniques applied to its data, Section 3 describes our proposed approach, Section 4 describes our experimental results and we conclude in Section 5.

2 Background

2.1 Data

The apparatus by which the images utilized in this work was captured is a prototype camera array named AWARE-2. This is a product of the Duke Imaging and Spectroscopy Program led by Dr. David Brady at Duke University in Durham, NC, USA. This device consists of a semi-spherical array (shown in Figure 3a), of 98 micro-cameras, each with a 14-megapixel resolution [4]. Sample images from these micro-cameras are shown in Figure 2. The resulting imagery can be combined to form single images with resolutions nearing 1-gigapixel.

While the camera apparatus is constructed to hold approximately 200 micro-cameras, currently only 98 are populated. The configuration of these selected micro-cameras is illustrated in Figure 3b, as if one were looking into the imaging sensors.



(a) (b)

Figure 2: Two example high-resolution images from the micro-camera array.

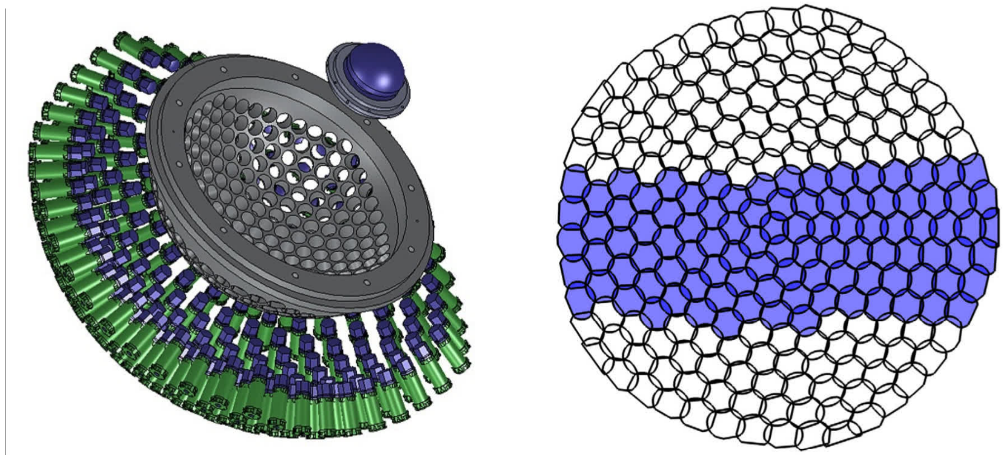
2.2 Problem Formulation

Suppose that we are given an image $\mathcal{I}$, and a set of sub-images (tiles):

$$\mathfrak{I} = \{I_1, \ldots, I_s\}, \tag{1}$$

where the I_i's can be viewed as an intensity function from the region of the image-plane U_i to a set of positive integers $[0, 255]$ for the case of gray-scale mosaicking. Without many changes we can extend this procedure to colored pictures by changing the function $I_i : U_i \rightarrow [0, 255]^3$, RGB-values.

We say that two sub-images I_1, I_2 are *related* if $I_1 \cap I_2 \neq \emptyset$. We can then turn the set of sub-images into a graph where the nodes are the sub-images and an edge exists between nodes I_x and I_y if they are related. We denote this graph also by $\mathfrak{I}$ since it is completely determined by the set $\mathfrak{I}$ for a given picture $\mathcal{I}$.



(a) Schematic of the AWARE-2 camera array [5].

(b) Graph colored to indicate populated micro-cameras in the array [6]. Shaded regions indicate populated micro-cameras. Intersection between regions indicates overlap between the respective micro-camera images.

Figure 3: Depictions of the AWARE-2 camera array.

In the mosaicking problem we require the two following conditions to hold:

1. $\mathcal{I} = \bigcup_{i=1}^{s} I_i$ and (2)
2. the graph of the sub-images is connected.

The second property implies that there is an overlap between each tile and some other *neighboring* ones, hence, a matching can be produced between them. This matching, or mapping, between pairs of images is the crux of image mosaicking. Since our graph of sub-images is connected we can create a process which will gradually stich neighboring images together to create a global picture. The first property guarantees that the output picture will be achieved by this mosaicking.

2.3 Current Methods

The methods followed by many current efforts such as [7] and others enumerated in [8] and [9], can be summarized as follows. The first step is to compute feature descriptors of the image tiles. The techniques may vary in the choice of feature descriptor, as mentioned in Section 1. Once each image is succinctly described, the descriptors are compared using some measure, e.g. RANSAC [2]. The parsimony with which this comparison is performed, i.e. whether each descriptor set is exhaustively compared to every other descriptor set, is another facet by which the various techniques vary [10].

A set of corresponding points $\{\mathbf{x}_1^j \ldots, \mathbf{x}_p^j\} \in U_j$ and $\{\mathbf{x}_1^k, \ldots, \mathbf{x}_p^k\} \in U_k$ is then considered. To align the respective images I_j and I_k via these point sets, a homography $\mathbf{H}$ must be computed such that

$$\mathbf{x}_i^j = \mathbf{H}\mathbf{x}_i^k \ , \forall i. \tag{3}$$

The result is computed by solving the system of equations represented by Equation (3) as described in [3, 11].

3 Methods

3.1 Image Decomposition

From the geometry of the camera array [12], we can extract the relative positions of the sub-images. Using this information in conjunction with the extents of the sub-images, it was possible to compute the intersection, or overlap, between each sub-image pair and form our graph of sub-images.

The neighborhood of a sub-image I_i is defined to be the set of sub-images related to I_i, or in other words, the set of sub-images that have some non-empty intersection with I_i:

$$\mathcal{N}_i = \{I_j | I_i \cap I_j \neq \emptyset\}. \tag{4}$$

Obviously $I_i \in \mathcal{N}_i$. In the graph theoretic language this is the *vertex star* of the node I_1.

Given a pair $(I_i, I_j), I_j \in \mathcal{N}_i$, their extent is used to compute the size and shape of the overlap between them. By performing this computation for all such pairs including I_i, the minimal overlap between I_i and any of its neighbors is determined. We compute the rectangle with maximal area, entirely contained within this overlap and denote its dimensions by $p \times q$. Then we partition the sub-image I_i and all $I_j \in \mathcal{N}_i$ into $\frac{m}{p} \times \frac{n}{q}$ regions, e.g.

$$I_i = \bigcup_{k=1}^{\frac{mn}{pq}} I_i^k. \tag{5}$$

This partitioning, although seemingly increases the complexity of the algorithm by increasing the number of tiles, it ultimately decreases the complexity of the problem by decreasing the size of the regions being compared. Further simplification is gained by allowing for the comparison of only the overlapping regions.

3.2 Sparse Subspace Recovery

In contrast to the methods commonly utilized for image mosaicking, as described and reference in Section 2.3, the proposed approach highlights similarity between portions of the image data itself. By computing a low-rank representation of the query image over the support of the column-space of its neighborhood, we are able to localize the overlap among the sub-images. This method follows closely to that of Robust Subspace Recovery via Dual-Sparsity Pursuit (RoSuRe-DSP), described in [13] and recounted here.

3.2.1 Robust Subspace Recovery vis Dual Sparsity Pursuit

Consider a data set $L \in R^d$ uniformly sampled from a union of subspaces $S = \cup_{i=1}^{J} S^i$, then assuming sufficient sample density, each sample can be represented by the others from the same subspace with probability 1. Mathematically, we represent the data matrix by $L = [l_1|l_2|\dots|l_n]$, yielding

$$L = LW,$$

where W is $n \times n$ block-diagonal matrix given.

It is worth noting that, to recover the underlying data sampled from UoS, it is equivalent to find a matrix L and W under the above constraints. Let us now define, mathematically, what the method does.

Definition 1 (k-block-diagonal matrix) *We say that a $n \times n$ matrix M is k-block-diagonal if and only if*

1. *There exists a permutation matrix P, such that the matrix $\tilde{M} = PMP^{-1}$ is a block-diagonal matrix*
2. *the maximum dimension of each block of $\tilde{M}$ is less or equal to $k+1$.*

The set of all such matrices is denoted as BM_k.

We next define the set of self representative matrices, based on the space BM_K as follows:

Definition 2 (k-self-representative matrix). *We say that a $d \times n$ matrix X with no zero columns is k-self-representative if and only if*

$$X = XW, W \in BM_k, w_{ii} = 0.$$

The space of all such $d \times n$ matrices is denoted by SR_k

The problem can then be formulated as

$$\begin{aligned} &\min \|W\|_0 \\ &s.t. X \in SR_k. \end{aligned} \tag{6}$$

where $||\cdot||_0$ is the l_0 vector pseudo-norm. We have a fundamental difficulty in solving this problem on account of the combinatorial nature of $\|\cdot\|_0$ and the complicated geometry of SR_k. For the former one, there are established results of using the l_1 norm to approximate the sparsity of W [14] [15]. The real difficulty, however, is that SR_k is a non-convex set and, thus the minimization process is impossible, in general.

To avoid solving Eqn(6) with a non-convex region, we opt to integrate this constraint into the objective function, and see the problem from a different angle. We hence have the following definition:

Definition 3 ($\mathcal{W}_0$-function on a matrix space). *For any $d \times n$ matrix X, if there is $W \in BM_k$, such that $X = XW$, let*

$$\mathcal{W}_0(X) = \min_{W} \|W\|_0, \quad s.t.\ X = XW, w_{ii} = 0, W \in BM_k.$$

Otherwise, $\mathcal{W}_0(X) = \infty$

Then instead of Eqn(6), we consider the following optimization problem:

$$\begin{aligned} &\min_{L,E} \mathcal{W}_0(L) \\ &s.t. X = L \in SR_k. \end{aligned} \tag{7}$$

Next we will leverage the parsimonious property of the l_1 norm to approximate $\|\cdot\|_0$. First, the definition of $\mathcal{W}_0(\cdot)$ is extended to an l_1 norm based function:

Definition 4 ($\mathcal{W}_1$-function on a matrix space). *For any $d \times n$ matrix X, if there exists $W \in BM_k$, such that $X = XW$, let*

$$\mathcal{W}_1(X) = \min_{W} \|W\|_1, \quad s.t.\ X = XW, w_{ii} = 0, W \in BM_k.$$

Otherwise, $\mathcal{W}_1(X) = \infty$

We then have the following problem,

$$\begin{aligned} &\min \mathcal{W}_1(L) \\ &s.t. X = L \in SR_k \end{aligned} \tag{8}$$

3.2.2 Application of RoSuRe-DSP

For a particular sub-image I_i with $|\mathcal{N}_i| = s$, we construct a dictionary matrix X whose columns are the columns of the regions I_j^k for all sub-images I_j in the neighborhood $\mathcal{N}_i$ of I_i:

$$X = [I_1^1, \ldots, I_1^{\frac{mn}{pq}}, I_2^1, \ldots, I_s^{\frac{mn}{pq}}]. \tag{9}$$

Having constructed this dictionary whose atoms are columns representing portions of the neighborhood, we proceed to optimize for a low-rank representation, Equation 8.

$$\min_i ||W_i||_1, \text{ s.t. } X = XW_i \text{ and } W_i \in BM_k. \tag{10}$$

Here, $\| \cdot \|_1$ is the vector l_1 norm. For a matrix X and an index set J, we let X_J be the submatrix containing only the columns of indices in J.

As shown in [13], this procedure will result in a sparse and low-rank coefficient matrix W that is also block diagonal. The blocks in W will correspond to the different subspaces inherent in the data represented by the matrix X_i. In this particular application, these subspaces are the intersections between the regions. Additionally, the sparsity of the coefficients guarantees high values indicating correspondence between the similar columns in those regions. In this fashion, all the computations are performed on small, local neighborhoods without the need for global information about the mosaic. This is the parallel nature of the method.

In the following sections we demonstrate this methods ability to define and localize similarity between images while expounding on the structure of the coefficient matrix and how it can be leveraged to construct the desired mosaic.

3.3 The Structure of the Coefficients

3.3.1 Synthetic Data

To evaluate the proposed method for representation, increasingly realistic data sets were utilized as input. The first two tests focused on synthetically generated mosaic tile sets. The synthetic data was produced from a larger test image, shown in Figure 4, by spatial windowing.

3.3.2 Tile Translation

To determine the proposed representation method's functionality in regard to simple translation, the test image, Figure 4, referred to as I_{orig}, was

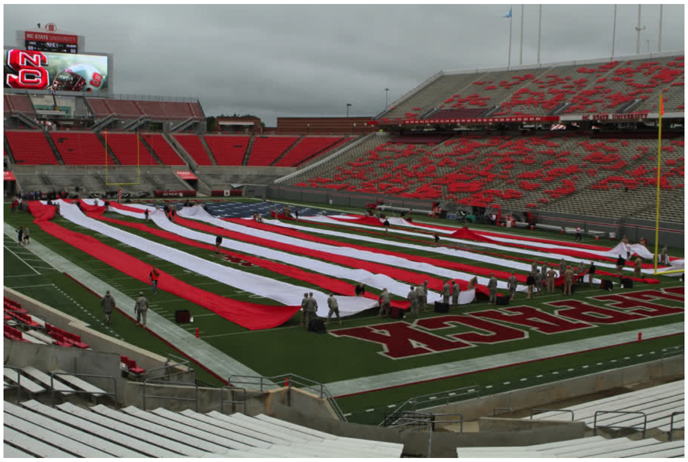


Figure 4: Test image from which synthetic sub-images, or tiles, were generated. [16]

partitioned into regular blocks on a 3 × 3 grid:

$$I_{orig} = \begin{bmatrix} A & B & E \\ C & D & F \\ G & H & K \end{bmatrix}. \tag{11}$$

These blocks were then grouped into four overlapping sub-images as indicated in Equations (12)-(15).

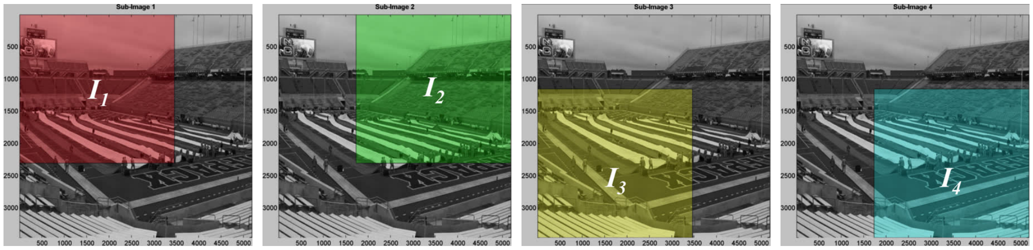
$$I_1 = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{12}$$

$$I_2 = \begin{bmatrix} B & E \\ D & F \end{bmatrix} \tag{13}$$

$$I_3 = \begin{bmatrix} C & D \\ G & H \end{bmatrix} \tag{14}$$

$$I_4 = \begin{bmatrix} D & F \\ H & K \end{bmatrix} \tag{15}$$

Masks to indicate this subdivision of the test image can be seen in Figure 5.



(a) Sub-image I_1 indicated.

(b) Sub-image I_2 indicated.

(c) Sub-image I_3 indicated.

(d) Sub-image I_4 indicated.

Figure 5: Masks indicating the portions of the test image, Figure 4, used to create synthetic tiles.

In this experiment, we explicitly compute the similarity between the sub-images for both the column- and row-spaces. To compute this similarity amongst the columns, we follow the procedure exactly as outlined in Section 3 by constructing the dictionary and constraint as shown in Equation (16). For this test, we've decomposed the image such that there is minimal similarity between the columns of I_1 and I_3 as well as between the columns of I_2 and I_4.

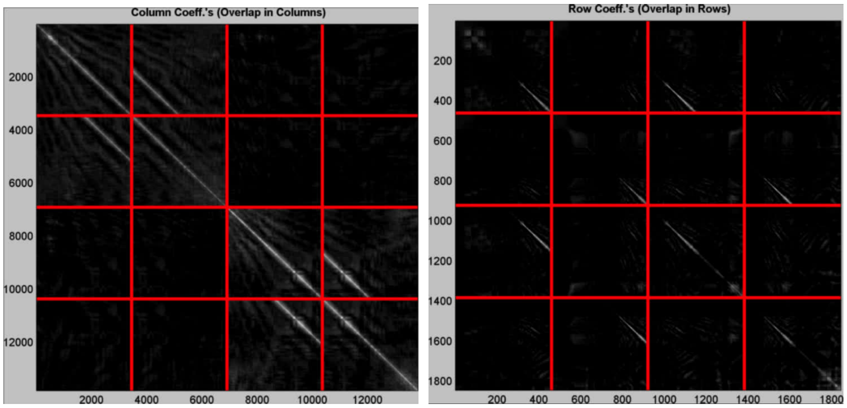
$$X_c = X_c W_c,\ X_c = [I_1, I_2, I_3, I_4] \tag{16}$$

Also for illustrative purposes, we've designed the sub-images such that there is little similarity between the rows of I_1 and I_2 as well as between I_3 and I_4. To compute the row-space similarity, we simply need to transpose the sub-images, as shown in Equation (17).

$$X_r = X_r W_r,\ X_r = [I_1^t, I_2^t, I_3^t, I_4^t] \tag{17}$$

Once the optimization from Equation (10) has been completed for both

Equations (16) & (17), the similarity between the sub-images can be observed from the coefficient matrices W_c and W_r, as shown in Figure 6. Red lines have been added to the coefficient matrices, artificially, to highlight the block structure. The lighter colors in the matrix represent higher-valued coefficients and the darker colors represent lower-valued coefficients.



(a) W_c

(b) W_r

Figure 6: Coefficient matrices W_c and W_r at computed in Equations (16) & (17). Lighter colors indicated high values. Blocks in the matrix, demarcated by artificially inserted horizontal and vertical lines, are the representation of the sub-image corresponding to the column in the space of the sub-image corresponding to the row.

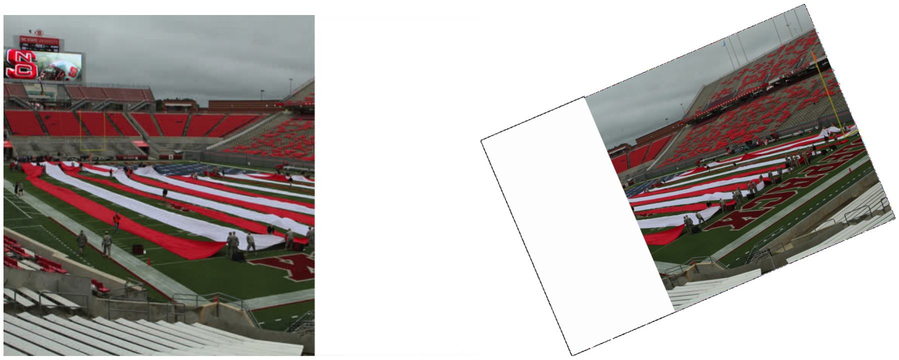
Considering Figure 6a, the bright bands of coefficients in the off-diagonal blocks correspond precisely to the duplicate columns in the sub-images. For example, the high-valued coefficients in the first block of the second row indicate that the last few (right-most) columns of I_1 are quite similar to the first few (left-most) columns of I_2, as expected from our construction of the test imagery. Similarity between the columns of I_3 and I_4 can also be determined by the high-valued coefficients in the last block of the third row and the third block of the last row. Likewise, considering Figure 6b allows for the determination of the similarity between the rows of the sub-images. Knowing the correspondence between rows and columns of the images allows for recovery of the possible translation that is necessary to align the images. It's important to note, however, that not only is a correspondence indicated by these coefficient values but, also the strength of the similarity.

3.3.3 Tile Rotation

The other portion of the problem presented by the real data for this application is that of rotation of the sub-images. To test the proposed method's robustness to rotation, the same test image shown in Figure 4 was divided into two overlapping sub-images, I_1 and I_2. The test proceeded by applying a rotation to I_2 about its centroid to mimic the possible rotation in the AWARE-2 data. An example of I_1 and I_2 rotated is shown in Figure 7.

Once this rotation had been applied, the RoSuRe-DSP optimization was performed. This procedure was followed for several rotations of I_2. Selected blocks of the resulting coefficient matrices are shown in Figure 8 for three such rotations.

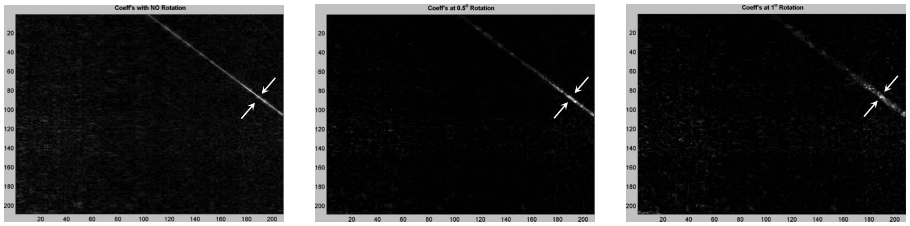
Figure 8a illustrates the type of coefficient band to expect in the blocks of W for perfect alignment. Here, a single coefficient value in each column represents a $1-1$ correspondence between the columns in I_1 and the



(a) Portion of test image not rotated: I_1.

(b) Portion of test image rotated: I_2. Shown with arbitrary rotation.

Figure 7: Image decomposition used for rotation tests.



(a) Coefficient matrix W corresponding to no rotation in I_2.

(b) Coefficient matrix W corresponding to a CCW 0.5^o rotation in I_2.

(c) Coefficient matrix W corresponding to a CCW 1^o rotation in I_2.

Figure 8: Subsets of the coefficient matrix W corresponding to various rotations of I_2.

columns in I_2. When I_2 was rotated, the local self-similarity in I_{orig} allowed for matching to still occur between columns of the sub-images. However, a spreading in the coefficient band can be observed in Figures 8b & 8c. As the rotation of I_2 is increased, the width of the band is increased. Indeed, the case illustrated in Figure 8a is a global minimum for the width of the coefficient band. Consequently, the width of this band can be exploited to compute the appropriate rotation of I_2 to best align with I_1. Besides the actual white bands that represent common features, one can see in Figure 6 other non-zero entries of the matrix W. This is due to two reasons:

1. The self-similarity within the big pictures which comes from the fact that it is a “natural image”.

2. Artifacts coming from the RoSuRe-DSP implementation.

For 40 test images the prescribed decomposition was performed at 360 different rotations of I_2, covering a full circular rotation in 1^o increments. To illustrate the degradation of the detection of these bands, the signal-to-noise ratios (SNR) for the examples in Figure 8 were computed and are shown in Table 1.

Rotation	SNR
0^o	1.56dB
0.5^o	1.40dB
1^o	0.95dB

Table 1: Table of SNR values for band of strong coefficients in W corresponding to the listed rotations in I_2.

The maximum rotation that allowed for detection of the strong coefficient band was 5^o. This rotation tolerance will be utilized in future imaging as a calibration parameter for the insertion of the micro-cameras into the array.

The simple fact that RoSuRe-DSP is producing useful decompositions when the input images are rotated at all deserves some discussion. The nature in which the input matrices are rotated in these experiments, as well as in the actual data, are not mathematical functions. The physical rotation of a picture leads to a highly non-linear transformation of the corresponding pixel representation matrix. Theoretically, when concerned with matrices, it is unreasonable to expect similarity between a matrix and a rotated version of itself. However, in this case we are targeting "natural" images where there is some smoothness of the features from column to column. Even when we rotate the image, the fact that the intensity is locally a continuous function allows small patches to be still written as a linear combination of similar columns. Without such local homogeneity this method would not be viable at all.

4 Results on Captured Data from AWARE-2

The proposed method was applied to all neighborhoods of each of the 98 images captured by the AWARE-2 camera array. The resulting mosaic is shown in Figure 9. The final image had a resolution of 0.38-gigapixels and was 483.3MB in size.

As noted in [17], there is no standard way to compare the results of image

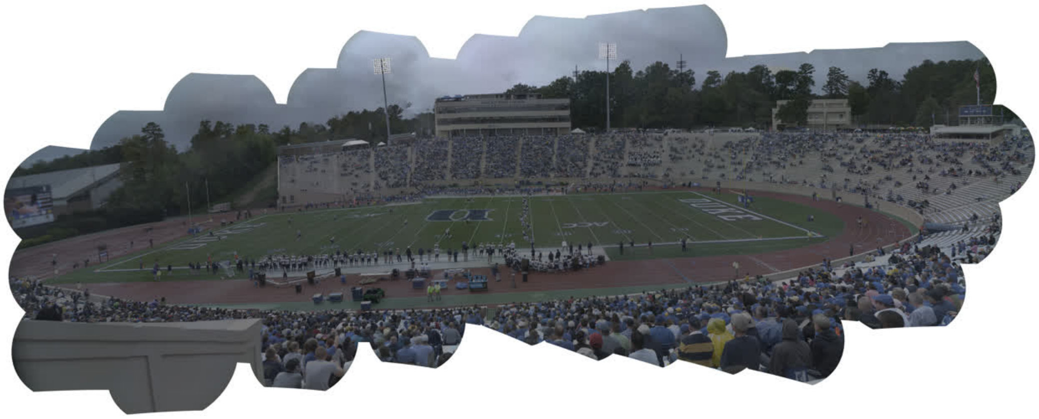
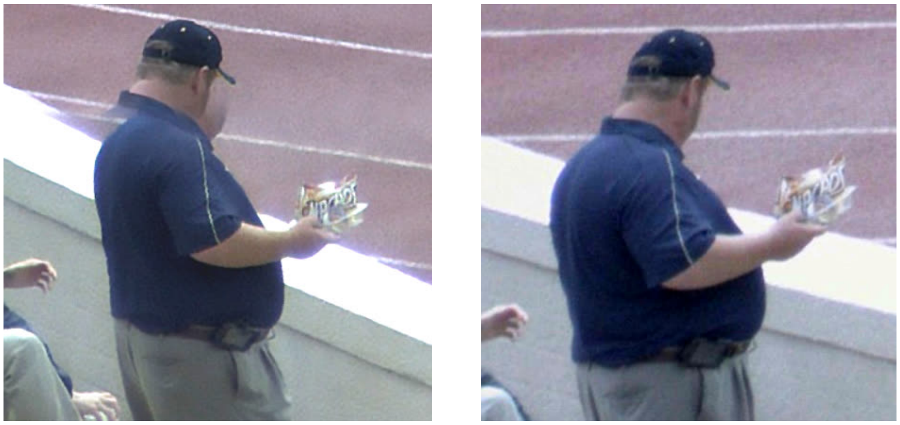


Figure 9: Computed mosaic.

mosaicking algorithms except a visual comparison. To that end, a small sample of the image is shown, in Figure 10, in comparison to the same portion produced by the methods described in Section 2.3. In this specific instance, our method is shown to not suffer from a particular boundary effect and corrects the floating head. However, since all methods suffer from some error-inducing effects, an objective measure of quality would prove invaluable.

We would like to, additionally, propose a specific method, leveraging the subjective analysis currently employed, to quantify the quality of the mosaics constructed by various methods for comparison. Starting with a set of 50 natural images, each should be broken into 100 tiles utilizing some readily available software, e.g. the crop command in ImageMagick[1]. These tiles are then used as input to each of three mosaicking algorithms: the proposed method, the method described in Section 2.3 and the graph-cuts

[1] http://www.imagemagick.org/Usage/crop/#crop_tile



(a) Portion of a computed mosaic from [5]

(b) Similar portion of our computed mosaic

Figure 10: Comparison of other work to our computed mosaic via a sampled portion.

variation [18]. Once each method has reconstructed each of the 50 images, ten sets of ten will be randomly selected. An individual will be assigned to each set of ten and all three reconstructions for those ten images will be presented to him/her. The task given them will be to rank the reconstructions for each image by quality. Quality will be probed by considering the smoothness, warping or lack thereof, and overall aesthetic appeal of each reconstruction. The result will be a statistical ranking of the subjective quality of each reconstruction.

5 Conclusion and Future Work

In this paper we have proposed a novel utilization of sparse-decomposition techniques within a rigorous set-theoretical framework for image mosaicking. While the results cannot be quantitatively compared to other work due to the lack of a standard measure, the results can be compared visually. Accordingly, the results are comparable to existing techniques and, yet, the method does not rely on heuristically defined tools. The authors wish to continue this line of thought, focusing on a quantitative measure of mosaic quality. To that end, the authors will implement the testing procedure outlined in Section 4.

6 Acknowledgment

This work was funded via the ATR Center of the Air Force Research Laboratory's Sensors Directorate. Special assistance was provided by Dr. Wesam Sakla.

References

[1] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *In ECCV*, 2006, pp. 404–417.

[2] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[3] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, “A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry,” *Artificial Intelligence*, vol. 78, no. 1–2, pp. 87 – 119, 1995, special Volume on Computer Vision. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0004370295000224

[4] D. J. Brady, M. E. Gehm, R. A. Stack, D. L. Marks, D. S. Kittle, D. R. Golish, E. M. Vera, and S. D. Feller, “Multiscale gigapixel photography,” *Nature*, vol. 486, no. 7403, pp. 386–389, 06 2012. [Online]. Available: http://dx.doi.org/10.1038/nature11150

[5] S. Feller. Aware2 multiscale gigapixel camera. [Online]. Available: http://disp.duke.edu/projects/AWARE/

[6] D. R. Golish, E. M. Vera, K. J. Kelly, Q. Gong, P. A. Jansen, J. M. Hughes, D. S. Kittle, D. J. Brady, and M. E. Gehm, “Development of a scalable image formation pipeline for multiscale gigapixel photography,” *Opt. Express*, vol. 20, no. 20, pp. 22 048–22 062, Sep 2012. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-20-20-22048

[7] A.-S. Iliopoulos, J. Hu, N. Pitsianis, X. Sun, M. Gehm, and D. Brady, "Big snapshot stitching with scarce overlap," in *High Performance Extreme Computing Conference (HPEC), 2013 IEEE*, Sept 2013, pp. 1–6.

[8] P. M. Jain and V. K. Shandliya, "A review paper on various approaches for image mosaicing," *International Journal of Computational Engineering Research*, vol. 3, no. 4, 2013.

[9] U. B. Patel and H. N. Mewada, "Review of image mosaic based on feature technique," *International Journal of Engineering and Innovative Technology*, vol. 2, no. 11, 2013.

[10] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[11] R. Abraham and P. Simon, "Review on mosaicing techniques in image processing," in *Advanced Computing and Communication Technologies (ACCT), 2013 Third International Conference on*, April 2013, pp. 63–68.

[12] H. S. Son, D. L. Marks, J. Hahn, J. Kim, and D. J. Brady, "Design of a spherical focal surface using close-packed relay optics," *Opt. Express*, vol. 19, no. 17, pp. 16 132–16 138, Aug 2011. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-19-17-16132

[13] X. Bian and H. Krim, "Robust subspace recovery via dual sparsity pursuit," *CoRR*, vol. abs/1403.8067, 2014.

[14] E. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489–509, Feb 2006.

[15] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, “Sparse representation for computer vision and pattern recognition,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, June 2010.

[16] N. W. Communications. Carter-finley stadium. [Online]. Available: http://www.ncsu.edu/facilities/buildings/carter-finley.html

[17] T. Buyukyazi, S. Bayraktar, and I. Lazoglu, “A novel mosaic quality measurement method for uav surveillance and remote sensing,” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-7/W2, pp. 55–60, 2013. [Online]. Available: http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-7-W2/55/2013/

[18] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, “Graphcut textures: Image and video synthesis using graph cuts,” *ACM Trans. Graph.*, vol. 22, no. 3, pp. 277–286, Jul. 2003. [Online]. Available: http://doi.acm.org/10.1145/882262.882264