Image Stitching for Drone Mapping

Aditya Jain[1] and Bhavishey Thapar[2]

Abstract— **Aerial mapping using drones provides several advantages in sectors such as construction, agriculture, and urban planning. Low-cost, on-demand, and high-resolution digital maps can be built using data collected by drones. Unlike satellites, drones cover only a small ground area in one image capture, thus, the images needs to be stitched to form a natural-looking artifacts free mosaic. To our knowledge, most off-the-shelf softwares that do high-quality image stitching are paid and their techniques proprietary, which limits their widespread adoption. In this work, we propose a simple, yet an effective framework for stitching large number of images using classical computer vision techniques for application in aerial surveying, precision agriculture, and digital land record management. We collect aerial data using drones for three different terrains and build mosaics using our algorithm. We do various quantitative experiments to evaluate our image stitching algorithm.**

I. INTRODUCTION

Drone mapping, building digital maps from data collected by drones, is enabling precision agriculture [17], digital land record management [9], and aerial surveying of construction sites [6]. Unlike satellites, drones provide us with a low-cost and on-demand method for creating high-resolution maps. Since the drones cover only a small portion of ground in one image capture, the images need to be stitched to form a natural-looking artifacts free mosaic.

Image stitching or mosaicing is a common and popular method in computer vision for effectively increasing the field of view of a camera, by allowing several views of a scene to be combined into a single view. It first arose as an area of research in the field of photography to create panoramic images. But it also has applications in other areas such as document mosaicing, image stabilization, video stitching, high-resolution photo mosaicing of satellite images, and medical imaging. Image stitching can also be used to create 3D models by combining multiple images of a scene taken from different viewpoints [1]. Existing methods for image stitching are either based on image-to-image warping [15], [16], [8] or corner feature localization and matching [4].

In this work, we propose a simple image stitching algorithm with an application in drone mapping. Our method can stitch multiple images ranging from fifty, all the way up to a hundred, to create high-resolution detailed maps of a particular area. We collect aerial data using a quadrotor for three terrains: city, agricultural farm, and a construction site. Our framework consists of four fundamental steps: (i) feature detection, (ii) feature matching, (iii) homography estimation, and (iv) image warping. We test our algorithm on the three datasets and build mosaics for each. We also perform various quantitative experiments to evaluate our image stitching algorithm.

II. RELATED WORK

In this section, we discuss a few prior works in detail. In [15], the authors present an algorithm for multi-image alignment that do not rely on the measurements of a reference image being distortion free. Key advantages of their algorithm are non-reliance on distortion free image, computation of coordinate transformations even when the multiple images are of an extended scene with no overlap between the first and last frame of the sequence, and the ability to handle linear and nonlinear transformations within the same framework.

The work [8] on mosaicing of video sequences describe two different types of mosaics: static and dynamic. It presents a series of extensions to the basic mosaics to provide representations at multiple spatial and temporal resolutions and to handle 3D scene information. It also describes techniques for the basic elements of the mosaic construction process, namely alignment, integration, and residual analysis. The major application of the technique is in video compression and enhancement, enhanced visualization, and other applications in video indexing, search, and manipulation.

The authors of [16] present techniques for constructing panoramic image mosaics from sequences of images. Their mosaic representation associates a transformation matrix with each input image, rather than explicitly projecting all of the images onto a common surface (e.g., a cylinder). To construct a full-view panorama, they introduce a rotational mosaic representation that associates a rotation matrix with each input image. Additionally, to reduce accumulated registration errors, they apply global alignment through block adjustment to the whole sequence of images resulting in an optimally registered image mosaic.

The paper [4] describe mosaicing for a sequence of images acquired by a camera rotating about its center. The work focusses on two key areas. First, in the automation and estimation of image registration: images (60+) are registered under a full (8 degrees of freedom) homography; the registration is automatic and robust, and a maximum likelihood estimator is used. The second is in enhanced resolution: a region of the mosaic can be viewed at a resolution higher than any of the original frames. It is shown that the degree of resolution enhancement is determined by a measure based on a matrix norm.

Aditya Jain[1] and Bhavishey Thapar[2] are with the Institute for Aerospace Studies, University of Toronto, 4925 Dufferin St, North York, ON M3H 5T6, Canada. Contact: aadi.jain@mail.utoronto.ca, bhavishey.thapar@mail.utoronto.ca

III. Image Stitching Algorithm

For a reference image I and a target image I', the objective of image stitching is to map and align I in the frame of I'. There are essentially four core steps involved in this process and we discuss them individually below.

A. Feature Detection

The first step in the process is to extract feature keypoints $\mathbf{x}_I$ and $\mathbf{x}'_I$ from I and I' respectively. Good features are those that are repeatable and invariant to scaling and rotation of images. For each keypoint $\mathbf{x}_{Ii}$ and $\mathbf{x}'_{Ii}$, we also calculate its descriptor $\mathbf{d\_x}_{Ii}$ and $\mathbf{d\_x}'_{Ii}$ respectively. We experiment with different feature detectors and the details are discussed in section IV.

B. Feature Matching

In the next step, we match features $\mathbf{x}_{Ii} \leftrightarrow \mathbf{x}'_{Ii}$ in the two images using the fast library for approximate nearest neighbors (FLANN) based matcher [13]. FLANN is a collection of algorithms optimized for fast nearest neighbor search in large datasets and high dimensional features. It uses the feature descriptors $\mathbf{d\_x}_{Ii}$, $\mathbf{d\_x}'_{Ii}$ for matching and works faster than brute-force matcher [12]. We further filter out good matches using the ratio test suggested in [10].

C. Homography Estimation

Let $\mathbf{x}_i, \mathbf{x}'_i$ be the matched points obtained after FLANN matching and filtering using the ratio test. We now define a projective transformation $\mathbf{H}$, called homography, that aims to map $\mathbf{x}_i$ to $\mathbf{x}'_i$ using the following relation

$$\mathbf{x}'_i = \mathbf{H}\mathbf{x_i}, \tag{1}$$

where $\mathbf{x}_i$, $\mathbf{x}'{}_i$ are homogenous vectors and $\mathbf{H} \in \mathbb{R}^{3\times3}$. Equation (1) can be expressed in terms of the vector cross-product $\mathbf{x}'_i \times \mathbf{H}\mathbf{x}_i = \mathbf{0}$. This form will enable a simple linear solution for $\mathbf{H}$ to be derived. If the j-th row of the matrix $\mathbf{H}$ is denoted by $\mathbf{h}^{jT}$, then we may write

$$\mathbf{H}\mathbf{x_i} = \begin{pmatrix} \mathbf{h}^{1T}\mathbf{x}_i \\ \mathbf{h}^{2T}\mathbf{x}_i \\ \mathbf{h}^{3T}\mathbf{x}_i \end{pmatrix}. \tag{2}$$

Writing $\mathbf{x}'_i = (x'_i, y'_i, w'_i)$, the cross-product can now be given explicitly as

$$\mathbf{x}'_i \times \mathbf{H}\mathbf{x_i} = \begin{pmatrix} y'_i\mathbf{h}^{3T}\mathbf{x}_i - w'_i\mathbf{h}^{2T}\mathbf{x}_i \\ w'_i\mathbf{h}^{1T}\mathbf{x}_i - x'_i\mathbf{h}^{3T}\mathbf{x}_i \\ x'_i\mathbf{h}^{2T}\mathbf{x}_i - y'_i\mathbf{h}^{1T}\mathbf{x}_i \end{pmatrix}. \tag{3}$$

Since $\mathbf{h}^{jT}\mathbf{x}_i = \mathbf{x}_i^T\mathbf{h}^j$ for j = 1,..., 3, this gives a set of three equations in the entries of $\mathbf{H}$, which can be written in the form

$$\begin{bmatrix} \mathbf{0}^T & -w'_i\mathbf{x}_i^T & y'_i\mathbf{x}_i^T \\ w'_i\mathbf{x}_i^T & \mathbf{0}^T & -x'_i\mathbf{x}_i^T \\ -y'_i\mathbf{x}_i^T & x'_i\mathbf{x}_i^T & \mathbf{0}^T \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}. \tag{4}$$

These equations have the form $\mathbf{A}_i\mathbf{h} = \mathbf{0}$, where $\mathbf{A}_i$ is a 3×9 matrix, and $\mathbf{h}$ is a 9-vector made up of the matrix $\mathbf{H}$,

$$\mathbf{h} = \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix}, \mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \tag{5}$$

with h_i the i-th element of $\mathbf{h}$. The equation $\mathbf{A}_i\mathbf{h} = \mathbf{0}$ is *linear* in the unknown $\mathbf{h}$. Although there are three equations in (4), only two of them are linearly independent because the third row is obtained, up to scale, from the sum of x'_i times the first row and y'_i times the second. Thus, each point correspondence gives two equations in the entries of $\mathbf{H}$. The third equation is usually omitted for solving $\mathbf{H}$ and (4) becomes

$$\begin{bmatrix} \mathbf{0}^T & -w'_i\mathbf{x}_i^T & y'_i\mathbf{x}_i^T \\ w'_i\mathbf{x}_i^T & \mathbf{0}^T & -x'_i\mathbf{x}_i^T \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}. \tag{6}$$

This can be written as $\mathbf{A}_i\mathbf{h} = \mathbf{0}$, where $\mathbf{A}_i$ is now the 2×9 matrix of (6). The equations hold for any homogenous coordinate representation $(x'_i, y'_i, w'_i)^T$ of the point $\mathbf{x}'_i$. One may choose $w'_i = 1$, which means (x'_i, y'_i) are the coordinates measured in the image.

Each point correspondence gives us two independent equations in the entries of $\mathbf{H}$. Given a set of four such point correspondences, we obtain a set of equations $\mathbf{A}\mathbf{h} = \mathbf{0}$, where where $\mathbf{A}$ is the matrix of equation coefficients built from the matrix rows $\mathbf{A}_i$ contributed from each correspondence, and $\mathbf{h}$ is the vector of unknown entries of $\mathbf{H}$. We seek a non-zero solution $\mathbf{h}$, since the obvious solution $\mathbf{h} = \mathbf{0}$ is of no interest to us. Since $\mathbf{H}$, in general, can be determined only up to scale, we have an additional constraint on the norm of $\mathbf{h}$ such as $\|\mathbf{h}\| = 1$, Thus, $\mathbf{H}$ has only eight degrees of freedom and four-point correspondences (that give us eight equations) are sufficient to solve for $\mathbf{H}$.

In practice, we usually have more than four point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ and the set of equations $\mathbf{A}\mathbf{h} = \mathbf{0}$ derived from (6) thus becomes an over-determined problem. In this case, there is no exact solution to $\mathbf{A}\mathbf{h} = \mathbf{0}$, and the objective is to minimize $\|\mathbf{A}\mathbf{h}\|$ subject to the constraint $\|\mathbf{h}\| = 1$. We obtain the SVD of $\mathbf{A}$ and the unit singular vector corresponding to the smallest singular value is the solution $\mathbf{h}$. Specifically, if $\mathbf{A} = UDV^T$ with D diagonal with positive diagonal entries, arranged in descending order down the diagonal, then $\mathbf{h}$ is the last column of V. The matrix $\mathbf{H}$ is then determined from $\mathbf{h}$ using (5). This method of solving for a homography matrix is called as direct linear transformation (DLT). We additionally make use of random sample consensus (RANSAC) [7] to further remove outliers while estimating $\mathbf{H}$.

D. Image Warping

The homography transformation $\mathbf{H}$ is then applied to image I to map it to image I''s coordinate frame. The transformed I and I' are then combined to form a single image or mosaic.

Steps III-A to III-D are repeated for all the images in a dataset to form the final mosaic. The complete stitching framework is sketched in Algorithm 1.

Algorithm 1 Overview of stitching algorithm

```
Initialize the mosaic $\mathbf{M}$ with image $I_1$
for image $i \in \{2, 3, ... n\}$ do
    Find features $\mathbf{x}_{Ii}$, $\mathbf{x}_{Ii-1}$ and corresponding descriptors $\mathbf{d\_x}_{Ii}$, $\mathbf{d\_x}_{Ii-1}$ for $I_i$ and $I_{i-1}$ respectively
    Match $\mathbf{x}_{Ii} \leftrightarrow \mathbf{x}_{Ii-1}$ using FLANN-based matcher
    Filter out good matches using ratio test
    Calculate homography $\mathbf{H}$ using DLT and RANSAC
    Use $\mathbf{H}$ to warp $I_i$ to $I_{i-1}$'s coordinate frame to get $I_{i \rightarrow i-1}$
    $\mathbf{M} \leftarrow$ combine $I_{i-1}$ and $I_{i \rightarrow i-1}$
end for
return $\mathbf{M}$
```

IV. EXPERIMENTS

We use DJI Phantom 4 [5] to collect aerial images and built datasets for three scenarios: city, agriculture, and construction. Fig. 1 shows some sample images of the three datasets. In each of the three datasets, there are a few hundred RGB images with a resolution of 4000×3000 pixels. The data is collected such that there is 70% overlap between the consecutive images. We will leave out any further details of the data collection process and focus on image stitching in this paper.

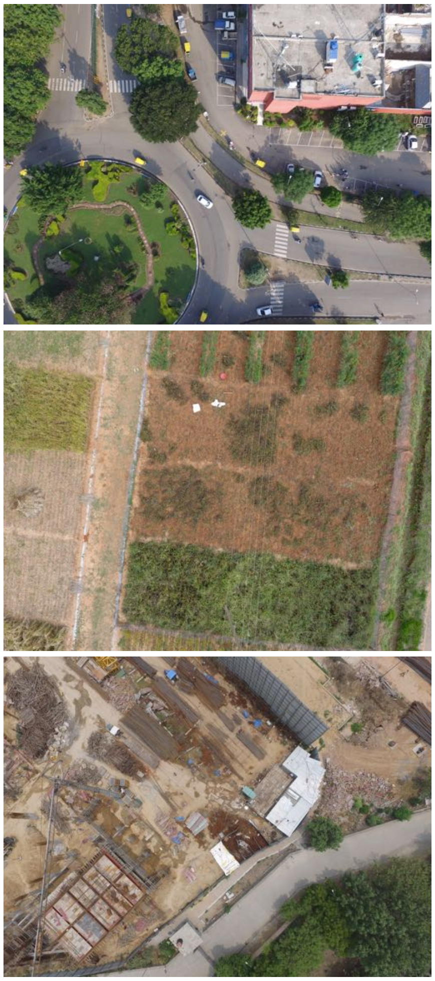
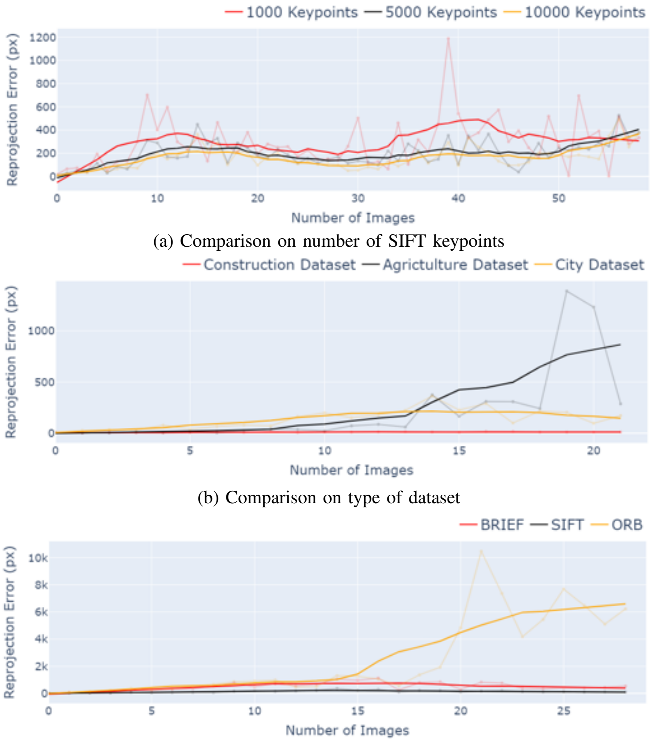


Fig. 1. Sample images for the city (top), agriculture (middle), and construction (bottom) dataset.

The keypoint detection is done using scale-invariant feature transform (SIFT) [10]. The features are invariant to image scale and rotation and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination.

To calculate the keypoint descriptor, a 16×16 neighborhood around the keypoint is taken. It is divided into 16 sub-blocks of 4×4 size and for each sub-block, 8 bin orientation histogram is created. Thus, a total of 128 bin vector is used to represent a descriptor. The keypoints are then matched using FLANN-based matcher and homography transformation $\mathbf{H}$ calculated using DLT and RANSAC. Finally, $\mathbf{H}$ is used to stitch the subsequent image into the existing mosaic. The image stitching pipeline is outlined in Algorithm 1. Fig. 4 shows the mosaics for the city dataset as the stitching grows for different number of images and Fig. 5 contains the final mosaics for the three datasets.

We evaluate the quality of mosaicing using reprojection error $\mathbf{e} = \|\mathbf{x}' - \mathbf{H}\mathbf{x}\|_2$. It can be seen qualitatively in Fig. 4, the reprojection error grows as more images are stitched because the error accumulates with every image warp. We experiment with different components of our stitching algorithm and compare against the reprojection error. The first experiment is using different number of SIFT features for the city dataset and Fig. 2 (a) shows that more the number of keypoints used for stitching, lower is the reprojection error and hence better mosaicing. In the second experiment, we keep SIFT and 10,000 keypoints fixed and test against the three types of dataset. Fig. 2 (b) shows that the agriculture dataset has a significant higher reprojection error than the other two, probably because the images of agricultural fields are quite homogenous, which makes it hard to detect repeatable feature and leads to errors in matching. The last experiment is comparison of different feature detectors: SIFT, binary robust independent elementary features (BRIEF) [3], and oriented FAST and rotated BRIEF (ORB) [14]. SIFT uses a feature descriptor with 128 floating point numbers, which makes it computationally and memory-wise expensive. BRIEF gives an alternative to find binary descriptors that occupy less memory, faster matching, and still higher recoginition rate. ORB is also an efficient alternative to the (earlier) patented SIFT and SURF [2]. For the city dataset and 10,000 keypoints, we compare the three feature detectors against reprojection erorr (Fig. 2 (c)) and stitching run time (Fig. 3). SIFT has the lowest reprojection erorr because of its high-quality keypoints and descriptors but also has the highest run time.



(a) Comparison on number of SIFT keypoints

(b) Comparison on type of dataset

(c) Comparison on type of feature detector

Fig. 2. Plot of reprojection error vs. the number of stitched images for different comparison metrics.

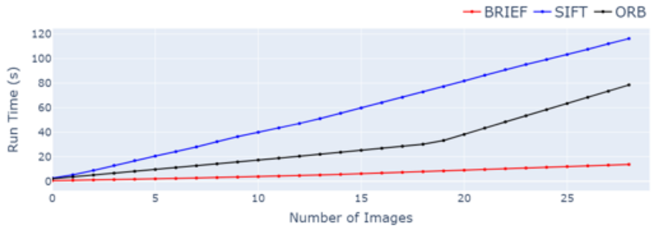


Fig. 3. Run time for different feature detectors vs. the number of stitched images.

V. CONCLUSION

In this work, we present a simple and efficient algorithm for stitching a large number of images using techniques from classical computer vision, with a target application in drone mapping. We collect aerial data using a quadrotor for three different types of terrains and test our image mosaicing algorithm. We conduct quantitative experiments to evaluate the different components of the image stitching algorithm.

Two possible improvements over our work is to use bundle adjustment [11] to reduce the cumulative error issue and blending to remove noticeable seams at the stitching location.

ACKNOWLEDGMENT

We would like to thank Prof. Steven L. Waslander, University of Toronto for helping us scope the project and his continued guidance.

REFERENCES

[1] L Barazzetti, M Previtali, and F Roncoroni. 3d modelling with the samsung gear 360. In *2017 TC II and CIPA-3D Virtual Reconstruction and Visualization of Complex Architectures*, volume 42, pages 85–90. International Society for Photogrammetry and Remote Sensing, 2017.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[3] Michael Calonder, Vincent Lepetit, Christoph Strecha, and FP Brief. Binary robust independent elementary features. In *Proceedings of the European Conference on Computer Vision*, pages 778–792.

[4] David Capel and Andrew Zisserman. Automated mosaicing with super-resolution zoom. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*, pages 885–891. IEEE, 1998.

[5] DJI. Phantom 4. 2022. Available from: https://www.dji.com/phantom-4 [17 December 2022].

[6] Faris Elghaish, Sandra Matarneh, Saeed Talebi, Michail Kagioglou, M Reza Hosseini, and Sepehr Abrishami. Toward digitalization in the construction industry with immersive and drones technologies: a critical literature review. *Smart and Sustainable Built Environment*, 2020.

[7] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[8] Michal Irani, P Anandan, and Steve Hsu. Mosaic based representations of video sequences and their applications. In *Proceedings of IEEE International Conference on Computer Vision*, pages 605–611. IEEE, 1995.

[9] Patricia Kameri-Mbote and Muriuki Muriungi. Potential contribution of drones to reliability of kenya's land information system. *The African Journal of Information and Communication*, 20:159–169, 2017.

[10] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[11] Philip F McLauchlan and Allan Jaenicke. Image mosaicing using sequential bundle adjustment. *Image and Vision computing*, 20(9-10):751–759, 2002.

[12] OpenCV. Brute-force based matcher. 2022. Available from: https://docs.opencv.org/4.x/d3/da1/classcv_1_1BFMatcher.html [17 December 2022].

[13] OpenCV. FLANN based matcher. 2022. Available from: https://docs.opencv.org/3.4/d5/d6f/tutorial_feature_flann_matcher.html [17 December 2022].

[14] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[15] Harpreet S Sawhney and Rakesh Kumar. True multi-image alignment and its application to mosaicing and lens distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):235–243, 1999.

[16] Heung-Yeung Shum and Richard Szeliski. Panoramic image mosaics. Technical report, Citeseer, 1997.

[17] Deepak Vasisht, Zerina Kapetanovic, Jongho Won, Xinxin Jin, Ranveer Chandra, Sudipta Sinha, Ashish Kapoor, Madhusudhan Sudarshan, and Sean Stratman. {FarmBeats}: An {IoT} platform for {Data-Driven} agriculture. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, pages 515–529, 2017.

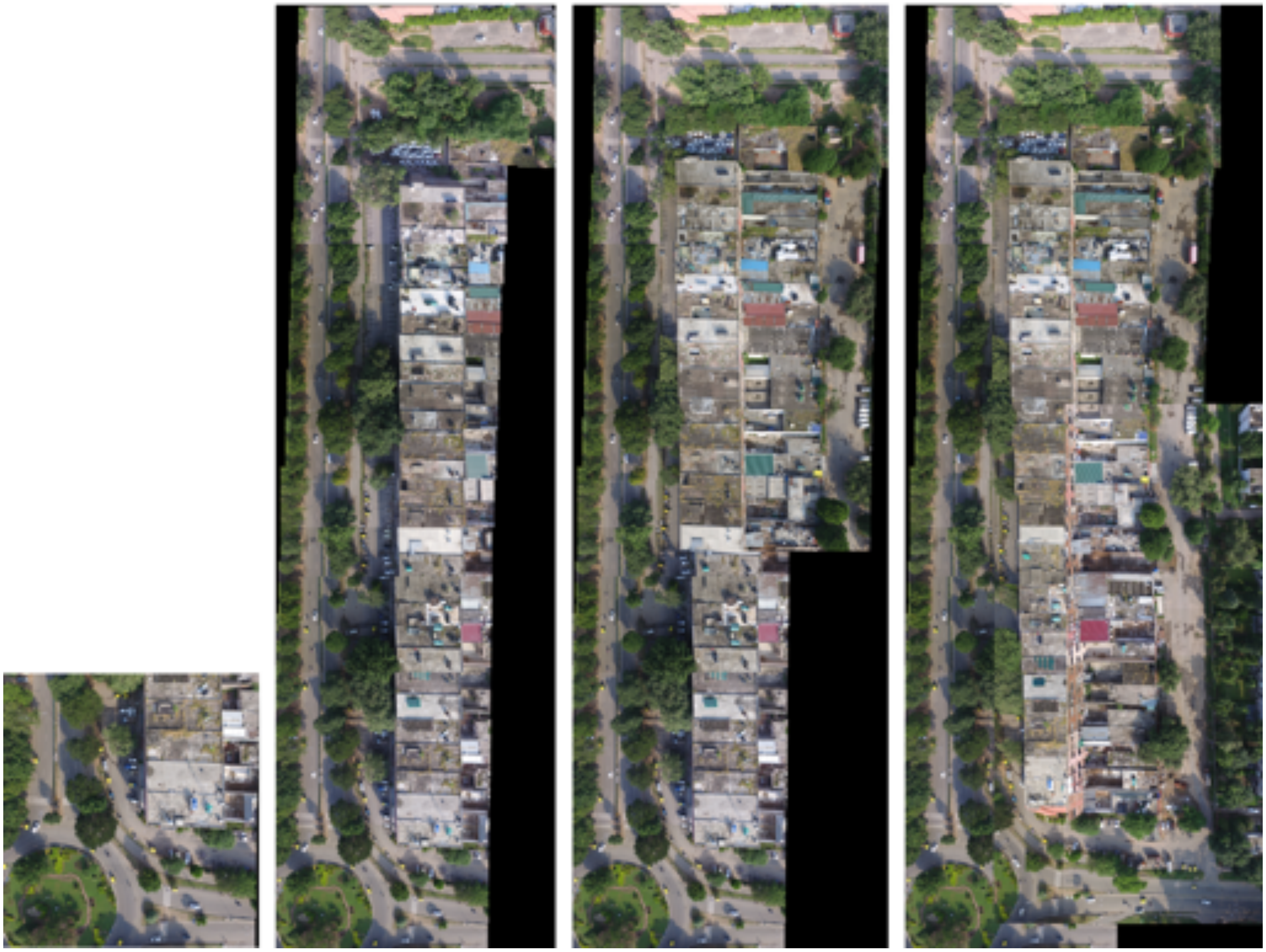


Fig. 4. Mosaics for the city dataset with different number of images n (3, 20, 30, and 50) stitched together.

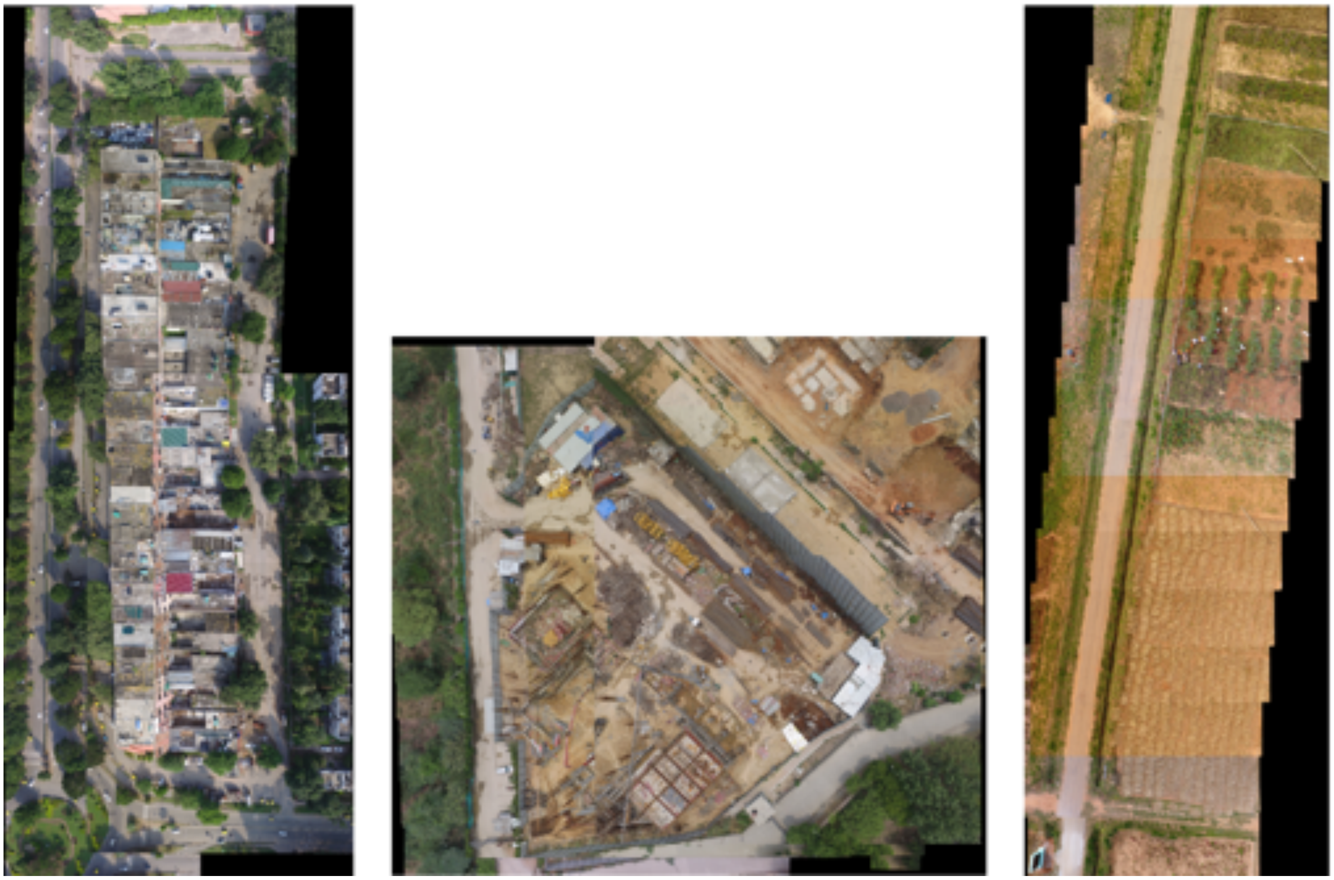


Fig. 5. Final mosaics for the city (left), construction (middle), and agriculture (right) dataset. 50, 23, and 15 images are stitched for city, construction, and agriculture dataset respectively.