As-Projective-As-Possible Image Stitching with Moving DLT

Julio Zaragoza* Tat-Jun Chin* Quoc-Huy Tran* Michael S. Brown[†] David Suter*
*School of Computer Science, The University of Adelaide
[†]School of Computing, National University of Singapore

Abstract—The success of commercial image stitching tools often leads to the impression that image stitching is a "solved problem". The reality, however, is that many tools give unconvincing results when the input photos violate fairly restrictive imaging assumptions; the main two being that the photos correspond to views that differ purely by rotation, or that the imaged scene is effectively planar. Such assumptions underpin the usage of 2D projective transforms or homographies to align photos. In the hands of the casual user, such conditions are often violated, yielding misalignment artifacts or "ghosting" in the results. Accordingly, many existing image stitching tools depend critically on post-processing routines to conceal ghosting. In this paper, we propose a novel estimation technique called *Moving Direct Linear Transformation (Moving DLT)* that is able to tweak or fine-tune the projective warp to accommodate the deviations of the input data from the idealised conditions. This produces *as-projective-as-possible* image alignment that significantly reduces ghosting without compromising the geometric realism of perspective image stitching. Our technique thus lessens the dependency on potentially expensive postprocessing algorithms. In addition, we describe how multiple as-projective-as-possible warps can be simultaneously refined via bundle adjustment to accurately align multiple images for large panorama creation.

Index Terms—Image Stitching, Image Alignment, Projective Warps, Direct Linear Transformation, Moving Least Squares.

1 Introduction

Image stitching algorithms have reached a stage of maturity where there are now an abundance of commercial tools based on or incorporating image stitching. Most well-known are image editing suites like Adobe Photoshop, web-based photo organization tools like Microsoft Photosynth, smartphone apps like Autostitch, as well as the built-in image stitching functionality on off-the-shelf digital cameras. Such tools are immensely useful in helping users organize and appreciate photo collections. The successful integration of image stitching may lead to the impression that image stitching is solved, but, in fact, many tools fail to give convincing results when given non-ideal data.

Most image stitching algorithms share a similar pipeline: first, estimate the transformations or warping functions that bring the overlapping images into alignment, then, composite the aligned images onto a common canvas. Of course, on real life data perfect alignment is rarely achieved, thus most of the research efforts are put into devising better alignment or compositing techniques to reduce or conceal misalignment artifacts. An excellent survey of state-of-the-art algorithms is available in [2]. Our work concentrates on improving the image alignment stage of the pipeline.

It is pertinent to first briefly mention state-of-the-art compositing techniques for image stitching. Chief among these are the seam cutting methods [3], [4] that optimize pixel selection among the overlapping images to minimize visible seams, and advanced pixel blending techniques such as Laplacian pyramid blending [5], [1] and Poisson image blending [6] that minimize blurring due to misalignments or exposure differences. Though vital to produce visually acceptable results, such post-processing routines are nevertheless imperfect and may not work all the time (see [7] for examples). It is thus strategic to attempt to reduce errors as much as possible during the alignment step.

Research into image alignment for stitching has somewhat culminated into the usage of bundle adjustment [8] to simultaneously optimize the relative rotations of the input images [9], [1], which are then used to align all images to a common frame of reference. This is the technique used in Autostitch as described in [1]. Earlier works incrementally warp multiple images, where for each image a sequence of alignment functions are threaded to warp the image onto the common reference frame [10], [11]. The focus is thus on finding the optimal order of threading such that the errors are not propagated and amplified excessively.

Interestingly, a majority of current techniques (including Autostitch and Photosynth) model the alignment functions as 2D projective transforms or homographies. Homographies are justified only if the images correspond to views that differ purely by rotation, or if the imaged scene is effectively planar (e.g., when the scene is sufficiently far away [2]). Many commercial image stitching tools actually specify this input condition, at least implicitly; see for example the FAQ pages on Autostitch[1] and Photosynth[2]. Violations to this condition predictably yield parallax errors or ghosting in the alignment which must be dealt with in the compositing stage. Row 1 in Fig. 1 is a "raw" result (the

[1] http://www.cs.bath.ac.uk/brown/autostitch/autostitch.html#FAQ
[2] http://photosynth.net/faq.aspx

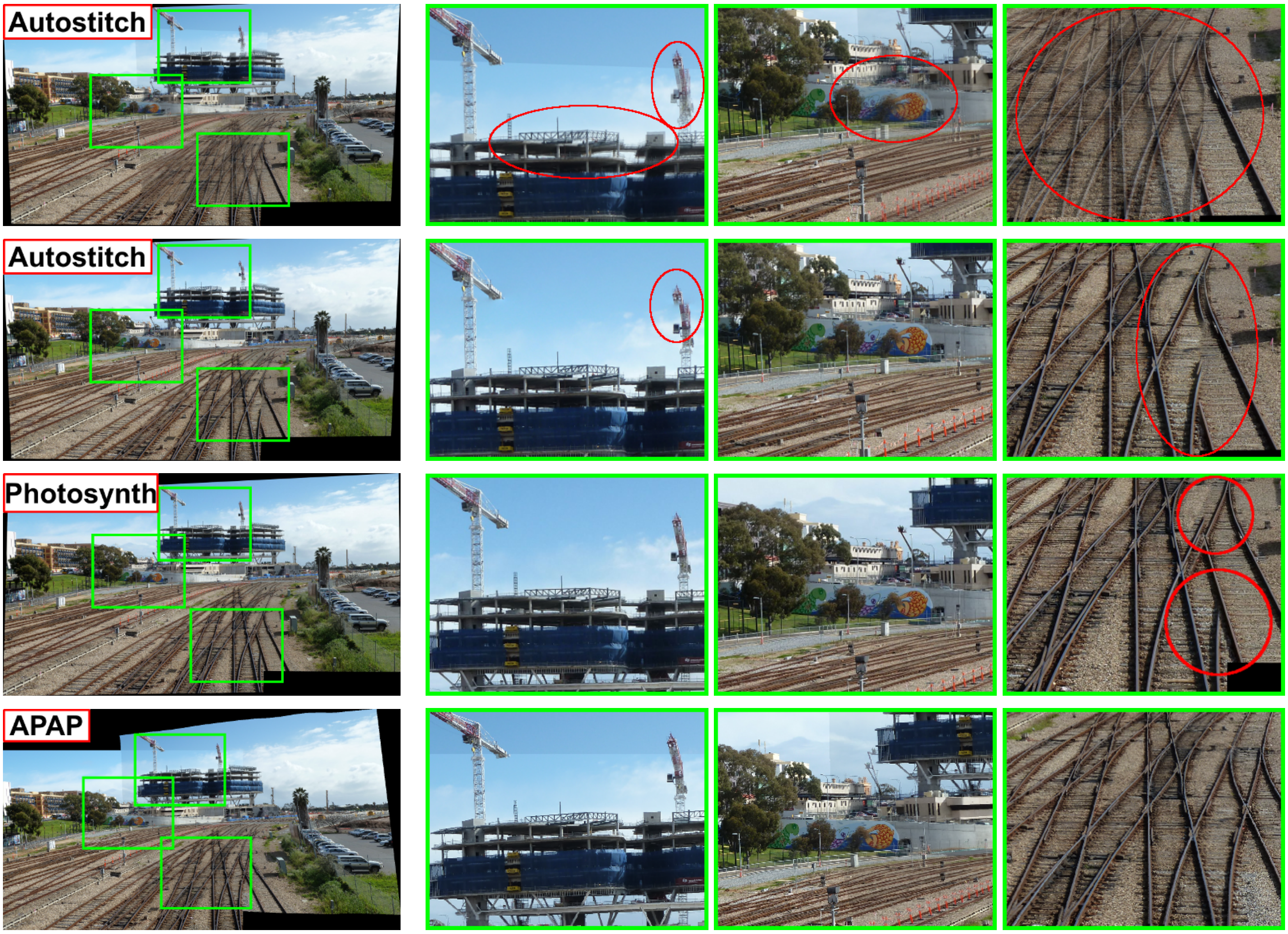


Fig. 1: Row 1: *Raw alignment result* from Autostitch [1] with significant ghosting. Rows 2 and 3: Final results (with advanced pixel compositing) from Autostitch and Photosynth, where glaring artifacts exist. Row 4: *Raw alignment result* using the proposed as-projective-as-possible method, which exhibits little noticeable ghosting.

mosaic is composited only with simple intensity averaging) from Autostitch that exhibits significant parallax errors — note that this problem is due primarily to the *inadequacy of the projective model in characterising the required warp*, and not inaccuracies in the warp estimation. Fig. 2 depicts this condition using a 1D analogy of image stitching.

Realistically, the prescribed imaging conditions are difficult to satisfy by casual users who are generally unfamiliar with image stitching fundamentals. Secondly, the desire to stitch a collection of images may be an afterthought, i.e., when it is not possible to revisit the scene to reshoot under the required imaging conditions. Unfortunately, many state-of-the-art techniques cannot produce convincing results when given uncooperative data, *even with advanced pixel compositing or postprocessing*. Rows 2 and 3 in Fig. 1 are the final (postprocessed) results from Autostitch and Photosynth, where unwanted artifacts evidently still exist.

The above problem raises a strong motivation for improving alignment methods for image stitching. Specifically, we argue that homography-based alignment must account for images that do not satisfy the assumed imaging conditions. To this end, we propose a novel homography estimation technique called *Moving DLT* that is able to tweak or fine-tune the homography to account for data that deviates from the expected trend, thereby achieving *as-projective-as-possible* warps; Fig. 2 shows what we mean by such a warp, while Row 4 in Fig. 1 shows a raw alignment result. Our method significantly reduces alignment errors without compromising the geometric plausibility of the scene.

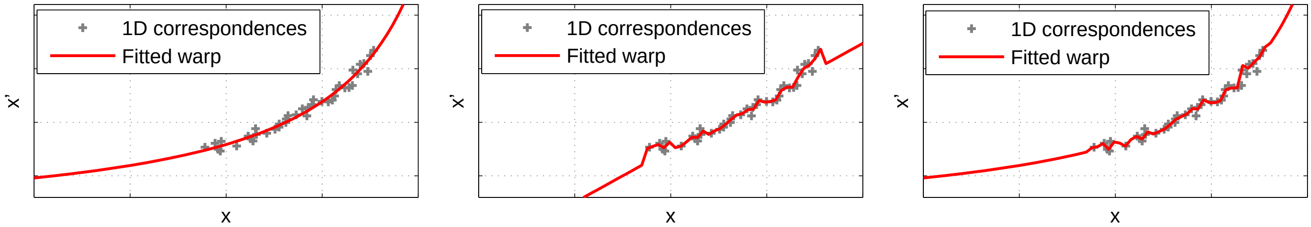
Note that it is not our aim to eliminate the usage of deghosting algorithms, which are still very useful especially if there are serious misalignments or moving objects. However, we argue that it is prudent to achieve accurate image alignment since this imposes a much lower dependency on the success of subsequent postprocessing routines.

An earlier version of our work [13] introduced Moving DLT for stitching pairs of images. Here, we propose a novel bundle adjustment technique to simultaneously refine multiple as-projective-as-possible warps for large panoramas.

The rest of the paper is organized as follows: Sec. 2 surveys important related work. Sec. 3 introduces the proposed method and its underlying principles, while Sec. 4 extends the method for panorama creation. Results are presented in Sec. 5, and we conclude in Sec. 6.

2 Previous Work

There exist methods that consider image stitching under arbitrary camera motions. A notable example is *manifold*



(a) Projective warp.

(b) As-affine-as-possible warp.

(c) As-projective-as-possible warp.

Fig. 2: To generate a 1D analogy of image stitching, a set of 1D correspondences $\{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^N$ are generated by projecting a 2D point cloud onto two 1D image "planes". Here, the two views differ by a rotation *and* translation, and the data are *not* corrupted by noise. (a) A 1D projective warp, parametrised by a 2×2 homography, is unable to model the local deviations of the data. Note that these deviations are caused purely by model inadequacy since there is no noise in the data. (b) An as-affine-as-possible warp, estimated based on [12], can interpolate the local deviations better, but fails to impose global projectivity. This causes incorrect extrapolation in regions without correspondences. (c) Our as-projective-as-possible warp interpolates the local deviations flexibly and extrapolates correctly following a global projective trend.

mosaicing [14] that is based on pushbroom cameras. Using a standard perspective camera, a pushbroom camera can be approximated by continuously "sweeping" the scene in video. Therefore, the method may not be applicable to stitch still images in a "discrete" photo collection, such as those handled by Autostitch and Photosynth.

2.1 3D reconstruction and plane-plus-parallax

Theoretically, given a set of overlapping views of a scene, it is possible to first recover the 3D structure and camera parameters (e.g., via SfM and dense stereo), then reprojecting each scene point onto a larger reference image to yield the mosaic. A notable approach is [15], which produces panoramas from images taken along long street scenes. However, a full 3D approach can be "overkill" if our goal is just to stitch images; in fact, many advanced compositing methods [3], [6] simply focus on creating perceptually good mosaics with little regard to 3D structure. Also, 3D reconstruction only works for scene points in the overlapping regions. Further, SfM may be brittle in views with small (but not exactly zero) baselines, which represents many of the image stitching cases in real life.

An intermediate approach is to directly align images using a planar projective mapping with a parallax component [16]. Without engaging in full 3D reconstruction, their method can only *approximate* the parallax at each pixel [16], which still results in significant parallax errors.

2.2 Panorama creation

Given a set of overlapping images, state-of-the-art methods [9], [1] perform bundle adjustment [8] to optimise the focal lengths and camera poses (relative rotations) of all views, which then give rise to inter-image homographies to perform alignment. While Shum and Szeliski [9] define the error terms based on pixel values (at regularly sampled patch positions), Brown and Lowe [1] use SIFT keypoint correspondences [17]. Also, Brown and Lowe [1] introduce a *panorama recognition* step based on SIFT matching that is able to determine the subsets of images that belong to the same panorama, given an unordered collection of photos.

A second refinement stage is also conducted in [9], to account for local misalignments in the mosaic. For each patch position, the average of the backprojected rays from each view is taken, which is subsequently projected again onto each view to yield the desired patch position in 2D. The differences between the original and desired patch positions are then interpolated (e.g., using splines) to form a correction field for parallax error removal. However, such a two step approach is cumbersome compared to our method that directly improves the projective warp. A two step approach also raises questions regarding the optimality of the overall process, e.g., how to regularize the correction field from overly distorting the original projective warps. By directly estimating as-projective-as-possible warps, our method avoids a separate refinement step.

Instead of estimating relative image rotations, other works directly estimate inter-image homographies, then chain or thread the homographies to stitch multiple images onto a common reference frame [10], [11]. The focus is thus on finding the optimal order of threading such that the errors are not propagated and amplified excessively. The homographies can also be refined by imposing geometric consistency between triplets of homographies [11]. However the dependence on homographic alignment means that such threading methods cannot handle non-ideal data.

2.3 Direct estimation of flexible warps

Closer to our work are recent methods that depart from the conventional homography model. A smoothly varying affine warp was proposed by [18] for image stitching. Starting from the motion coherence-based point-set registration method of [19], an affine initialization is introduced by [18] which is then deformed locally to minimize registration errors while maintaining global affinity. Conceptually, this warp is similar to the *as-affine-as-possible warp* used in image deformation [12]. Fundamentally, however, using affine regularization may be suboptimal for *extrapolation*, since an affinity is inadequate to achieve a perspective warp [2], e.g., an affine warp may counterproductively preserve parallelism in the extrapolation region. Therefore, while the method can interpolate flexibly and accurately due to local adaptations, it may produce distorted results while *extrapolating*; observe the image contents in the stitching results of [18] in Row 2 of Figs. 6 and 7.

In the context of video stabilization, Liu et al. [20] proposed content preserving warps. Given matching points between the original and stabilized image frames, the novel view is synthesized by warping the original image using an as-similar-as-possible warp [21] that jointly minimizes the registration error and preserves the rigidity of the scene. The method also pre-warps the original image with a homography, thus effectively yielding a locally adapted homography. Imposing scene rigidity minimizes the dreaded "wobbling" effect in video stabilization. However, in image stitching where there can be large rotational and translational difference between views, their method does not interpolate flexibly enough due to the rigidity constraints. This may not be an issue in [20] since the original and smoothed camera paths are close (see Sec. 4 in [20]), i.e., the motion between the views to align is small.

By assuming that the scene contains a ground plane and a distant plane, Gao et al. [22] proposed dual homography warps for image stitching. Basically this is a special case of a piece-wise homographic warp, which is more flexible than using a single homography. While it performs well if the required setting is true, it may be difficult to extend the method for an arbitrary scene, e.g., how to estimate the appropriate number homographies and their parameters.

It is also worth nothing that, unlike the proposed approach, the above flexible image alignment methods do not provide a simultaneous refinement step for multiple image stitching. Thus when creating large panoramas the quality of the result is highly dependent on the accuracy of pairwise stitching and the chaining order of alignment functions.

3 As-projective-as-possible warps

In this section, we first review the 2D projective warp customarily used in image stitching. We then describe the underlying principles of our proposed method.

3.1 The 2D projective warp

Let $\mathbf{x} = [x\ y]^T$ and $\mathbf{x}' = [x'\ y']^T$ be matching points across overlapping images I and I'. A projective warp transforms $\mathbf{x}$ to $\mathbf{x}'$ following the relation

$$\tilde{\mathbf{x}}' \sim \mathbf{H}\tilde{\mathbf{x}}, \tag{1}$$

where $\tilde{\mathbf{x}} = [\mathbf{x}^T\ 1]^T$ is $\mathbf{x}$ in homogeneous coordinates, and $\sim$ indicates equality up to scale. The 3×3 matrix $\mathbf{H}$ is called the homography. In inhomogeneous coordinates,

$$x' = \frac{\mathbf{r}_1[x\ y\ 1]^T}{\mathbf{r}_3[x\ y\ 1]^T} \quad \text{and} \quad y' = \frac{\mathbf{r}_2[x\ y\ 1]^T}{\mathbf{r}_3[x\ y\ 1]^T}, \tag{2}$$

where $\mathbf{r}_j$ is the j-th row of $\mathbf{H}$. The divisions in (2) cause the 2D function to be non-linear, which is crucial to allow a fully perspective warp. Fig. 2(a) shows a 1D analogy.

Direct Linear Transformation (DLT) [23] is a basic method to estimate $\mathbf{H}$ from a set of noisy point matches $\{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^N$ across I and I' (e.g., established using SIFT matching [17]). First, (1) is rewritten as the implicit condition $\mathbf{0}_{3\times1} = \tilde{\mathbf{x}}' \times \mathbf{H}\tilde{\mathbf{x}}$ and then linearised as

$$\mathbf{0}_{3\times1} = \begin{bmatrix} \mathbf{0}_{1\times3} & -\tilde{\mathbf{x}}^T & y'\tilde{\mathbf{x}}^T \\ \tilde{\mathbf{x}}^T & \mathbf{0}_{1\times3} & -x'\tilde{\mathbf{x}}^T \\ -y'\tilde{\mathbf{x}}^T & x'\tilde{\mathbf{x}}^T & \mathbf{0}_{1\times3} \end{bmatrix} \mathbf{h}, \quad \mathbf{h} = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}, \tag{3}$$

where $\mathbf{h}$ is a obtained by vectorizing $\mathbf{H}$ into a vector. Only two of the rows in (3) are linearly independent. Let $\mathbf{a}_i$ be the first-two rows of the LHS matrix in (3) computed for the i-th point match $\{\mathbf{x}_i, \mathbf{x}'_i\}$. Given an estimate $\mathbf{h}$, the quantity $\|\mathbf{a}_i\mathbf{h}\|$ is the *algebraic error* of the i-th datum. DLT minimizes the sum of squared algebraic errors

$$\hat{\mathbf{h}} = \underset{\mathbf{h}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\mathbf{a}_i\mathbf{h}\|^2 \quad \text{s.t.} \quad \|\mathbf{h}\| = 1, \tag{4}$$

where the norm constraint prevents the trivial solution. DLT is thus also referred to as *algebraic least squares* [23]. Stacking vertically $\mathbf{a}_i$ for all i into matrix $\mathbf{A} \in \mathbb{R}^{2N\times9}$, the problem can be rewritten as

$$\hat{\mathbf{h}} = \underset{\mathbf{h}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{h}\|^2 \quad \text{s.t.} \quad \|\mathbf{h}\| = 1. \tag{5}$$

The solution is the least significant right singular vector of $\mathbf{A}$. Given the estimated $\mathbf{H}$ (reconstructed from $\hat{\mathbf{h}}$), to align the images, an arbitrary pixel $\mathbf{x}_*$ in the source image I is warped to the position $\mathbf{x}'_*$ in the target image I' by

$$\tilde{\mathbf{x}}'_* \sim \mathbf{H}\tilde{\mathbf{x}}_*. \tag{6}$$

To avoid issues with numerical precision, prior to DLT the data can first be normalized in the manner of [24], with the estimated $\mathbf{H}$ then denormalized before executing (6).

3.2 Moving DLT

When the views I and I' do not differ purely by rotation or are not of a planar scene, using a basic homographic warp inevitably yields misalignment or parallax errors. To alleviate this problem, our idea is to warp each $\mathbf{x}_*$ using a *location dependent* homography

$$\tilde{\mathbf{x}}'_* \sim \mathbf{H}_*\tilde{\mathbf{x}}_* \tag{7}$$

where $\mathbf{H}_*$ is estimated from the weighted problem

$$\mathbf{h}_* = \underset{\mathbf{h}}{\operatorname{argmin}} \sum_{i=1}^{N} \|w_*^i\mathbf{a}_i\mathbf{h}\|^2 \quad \text{s.t.} \quad \|\mathbf{h}\| = 1. \tag{8}$$

The scalar weights $\{w_*^i\}_{i=1}^N$ give higher importance to data that are closer to $\mathbf{x}_*$, and the weights are calculated as

$$w_*^i = \exp(-\|\mathbf{x}_* - \mathbf{x}_i\|^2/\sigma^2). \tag{9}$$

Here, σ is a scale parameter, and $\mathbf{x}_i$ is the coordinate in the source image I of one-half of the i-th point match $\{\mathbf{x}_i, \mathbf{x}'_i\}$.

Intuitively, since (9) assigns higher weights to data closer to $\mathbf{x}_*$, the projective warp $\mathbf{H}_*$ better respects the local structure around $\mathbf{x}_*$. Contrast this to (6) which uses a single and global $\mathbf{H}$ for all $\mathbf{x}_*$. Moreover, as $\mathbf{x}_*$ is *moved* continuously in its domain I, the warp $\mathbf{H}_*$ also varies

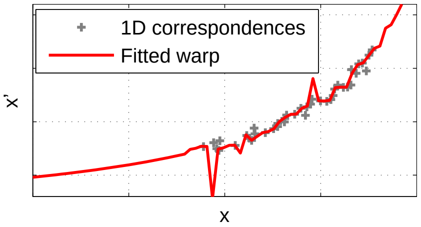


Fig. 3: Results from Moving DLT *without* regularisation for the 1D synthetic image stitching problem.

smoothly. This produces an overall warp that adapts flexibly to the data, yet attempts to preserve the projective trend of the warp, i.e., a flexible projective warp; Fig. 2(c) shows a 1D analogy. We call this method *Moving DLT*.

The problem in (8) can be written in the matrix form

$$\mathbf{h}_* = \underset{\mathbf{h}}{\operatorname{argmin}} \|\mathbf{W}_*\mathbf{A}\mathbf{h}\|^2 \quad \text{s.t.} \quad \|\mathbf{h}\| = 1, \tag{10}$$

where the weight matrix $\mathbf{W}_* \in \mathbb{R}^{2N \times 2N}$ is composed as

$$\mathbf{W}_* = \operatorname{diag}([\, w_*^1 \; w_*^1 \; w_*^2 \; w_*^2 \; \ldots \; w_*^N \; w_*^N]) \tag{11}$$

and diag$(\cdot)$ creates a diagonal matrix given a vector. This is a weighted SVD (WSVD) problem, and the solution is simply the least significant right singular vector of $\mathbf{W}_*\mathbf{A}$.

Problem (10) may be unstable when many of the weights are insignificant, e.g., when $\mathbf{x}_*$ is in a data poor (extrapolation) region. To prevent numerical issues in the estimation, we offset the weights with a small value γ within 0 and 1

$$w_*^i = \max\left(\exp(-\|\mathbf{x}_* - \mathbf{x}_i\|^2/\sigma^2), \gamma\right). \tag{12}$$

This also serves to regularize the warp, whereby a high γ reduces the warp complexity. In fact as γ approaches 1 the resultant warp loses its flexibility and reduces to the original homographic warp. Fig. 3 illustrates a 1D analogy.

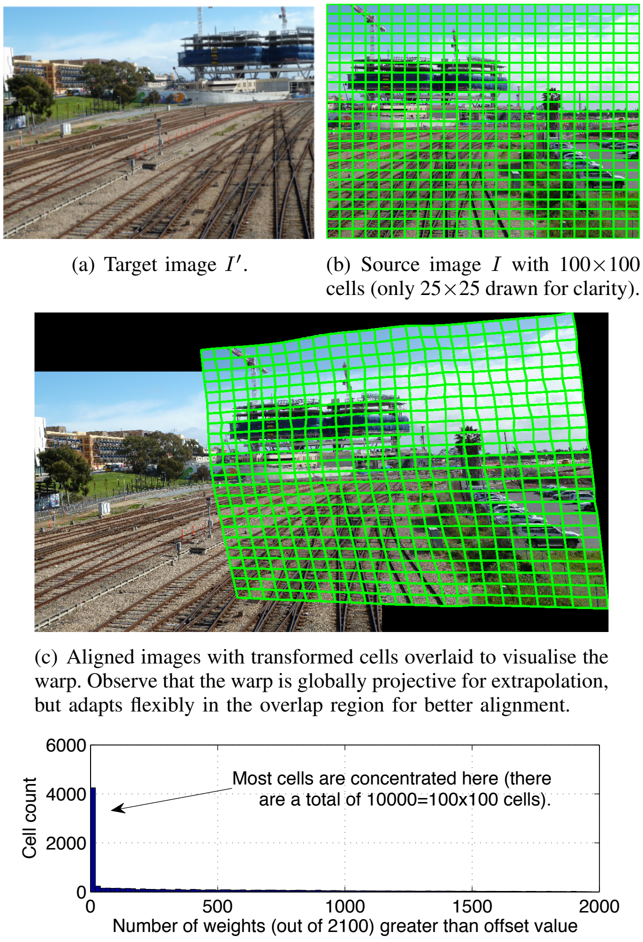
Conceptually, Moving DLT is the homogeneous version of moving least squares (MLS) commonly used in surface approximation [25]. In the context of warping points in 2D for image manipulation [12], MLS estimates for each $\mathbf{x}_*$ an *affine* transformation defined by a matrix $\boldsymbol{F}_* \in \mathbb{R}^{2\times 3}$

$$\mathbf{x}'_* = \boldsymbol{F}_* \begin{bmatrix} \mathbf{x}_* \\ 1 \end{bmatrix}, \quad \text{where} \tag{13}$$

$$\boldsymbol{F}_* = \underset{\boldsymbol{F}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| w_*^i \left(\boldsymbol{F} \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} - \mathbf{x}'_i \right) \right\|^2. \tag{14}$$

Problem (14) is a weighted least squares problem. Including nonstationary weights $\{w_*^i\}_{i=1}^N$ produces flexible warps, but such warps are only as-affine-as-possible; Fig. 2 compares Moving DLT and MLS in a 1D analogy of image stitching.

A homogeneous version of MLS (called *algebraic moving least squares*) was explored before in [26] for surface approximation. In contrast to our formulation here which is based on projective warps, in [26] an elliptical surface is estimated at every weighted instance of DLT. In addition, we also propose a novel bundle adjustment step (see Sec. 4) to simultaneously refine multiple warps.



(a) Target image I'.

(b) Source image I with 100×100 cells (only 25×25 drawn for clarity).

(c) Aligned images with transformed cells overlaid to visualise the warp. Observe that the warp is globally projective for extrapolation, but adapts flexibly in the overlap region for better alignment.

Most cells are concentrated here (there are a total of 10000=100x100 cells).

(d) Histogram of number of weights $\neq \gamma$ for the cells in (b).

Fig. 4: Demonstrating image stitching with our method. The input images correspond to views that differ by rotation *and* translation. The images are both of size 1500×2000 pixels. The number of SIFT matches $\{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^N$ (not shown) after RANSAC is 2100.

3.3 Efficient computation for image stitching

So far we have assumed that no mismatches or outliers exist among the data. Before invoking Moving DLT, we remove outliers using RANSAC [27] with DLT as the minimal solver. Although we consider data where the inliers themselves may deviate from the projective trend, in practice, the outlier errors are orders of magnitude larger than the inlier deviations [28], thus RANSAC can be effectively used.

3.3.1 Partitioning into cells

Solving (10) for each pixel position $\mathbf{x}_*$ in the source image I is unnecessarily wasteful, since neighboring positions will yield very similar weights (9) and hence very similar homographies. We thus uniformly partition the 2D domain I into a grid of $C_1 \times C_2$ cells, and take the center of each cell as $\mathbf{x}_*$. Pixels within the same cell are then warped using the same homography. Fig. 4 illustrates a stitched image from the data in Fig. 1 using 100×100 cells. Observe that the warp is globally projective for extrapolation, but adapts flexibly in the overlap region for better alignment.

Partitioning into cells effectively reduces the number of WSVD instances to $C_1 \times C_2$. Moreover, each of the WSVD instances are mutually independent, thus a simple approach to speed up computation is to solve the WSVDs in parallel. Note that even without parallel processing, solving (10) for all 100×100 cells in the images in Fig. 4 which contain 2100 SIFT matches ($\mathbf{A}$ is of size 4200×9) takes just about 3 seconds on a Pentium i7 2.2GHz Quad Core machine.

A potential concern is that discontinuities in the warp may occur between cells, since cell partitioning effectively downsamples the smoothly varying weights (12). In practice, as long as the cell resolution is sufficiently high, the effects of warp discontinuities are minimal (100×100 is adequate for all the images that we tested in Sec. 5).

3.3.2 Updating weighted SVDs.

Further speedups are possible if we realise that, for most cells, due to the offsetting (12) many of the weights do not differ from the offset γ. Based on the images in Figs. 4(a) and 4(b), Fig. 4(d) histograms across all cells the number of weights that differ from γ (here, $\gamma = 0.0025$). A vast majority of cells ($> 40\%$) have fewer than 20 weights (out of a total of 2100) that differ from γ.

To exploit this observation a WSVD can be updated from a previous solution instead of being computed from scratch. Defining $\mathbf{W}_\gamma = \gamma \mathbf{I}$, let the columns of $\mathbf{V}$ be the right singular vectors of $\mathbf{W}_\gamma \mathbf{A}$. Define the eigendecomposition

$$\mathbf{A}^T \mathbf{W}_\gamma^T \mathbf{W}_\gamma \mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^T \tag{15}$$

as the base solution. Let $\tilde{\mathbf{W}}$ equal $\mathbf{W}_\gamma$ except the i-th diagonal element that has value $\tilde{w}_i$. The eigendecomposition of $\mathbf{A}^T \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} \mathbf{A}$ can be obtained as the rank-one update

$$\mathbf{A}^T \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} \mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^T + \rho \mathbf{r}_i \mathbf{r}_i^T = \mathbf{V}(\mathbf{D} + \rho \bar{\mathbf{r}}_i \bar{\mathbf{r}}_i^T)\mathbf{V}^T,$$

where $\rho = (\tilde{w}_i^2/\gamma^2 - 1)$, $\mathbf{r}_i$ is the i-th row of $\mathbf{A}$, and $\bar{\mathbf{r}}_i = \mathbf{V}^T \mathbf{r}_i$. The diagonalisation of the new diagonal matrix

$$\mathbf{D} + \rho \bar{\mathbf{r}}_i \bar{\mathbf{r}}_i^T = \tilde{\mathbf{C}}\tilde{\mathbf{D}}\tilde{\mathbf{C}}^T \in \mathbb{R}^{m \times m} \tag{16}$$

can be done efficiently using secular equations [29]. Multiplying $\mathbf{V}\tilde{\mathbf{C}}$ yields the right singular vectors of $\tilde{\mathbf{W}}\mathbf{A}$. This can be done efficiently by exploiting the Cauchy structure in $\tilde{\mathbf{C}}$ [29]. The cost of this rank-one update is $\mathcal{O}(m^2 \log^2 m)$.

The WSVD for each cell can thus be obtained via a small number of rank-one updates to the base solution, each costing $\mathcal{O}(m^2 \log^2 m)$. Overall this is cheaper than computing from scratch, where for $\mathbf{W}_* \mathbf{A}$ of size $n \times m$, would take $\mathcal{O}(4nm^2 + 8m^3)$ even if we just compute the right singular vectors [30]. Note that in (10), $(n = 2N) \gg (m = 9)$.

4 Simultaneous Refinement

To stitch multiple images to form a large panorama, pairs of images can be incrementally aligned and composited onto a reference frame. However, incremental stitching may propagate and amplify alignment errors, especially at regions with multiple overlapping images [2]. Such errors can be alleviated by simultaneously refining the multiple alignment functions, prior to compositing. Here, we show how bundle adjustment can be used to simultaneously refine multiple as-projective-as-possible warps.

4.1 Selecting the reference frame

Given a set of input images $\{I^k\}_{k=1}^K$, the initial step is to map all the keypoints in the images onto a common reference frame I^R. Though not necessary for bundle adjustment, for simplicity we choose I^R from one of the input images. To this end we apply the keypoint-based panorama recognition method [1, Sec. 3] to identify pairs of overlapping images and construct an image connection graph. The graph is traversed to find the node (image) with the largest number of edges which we choose as I^R.

The byproduct of the panorama recognition step is a set of (rigid) homographies between overlapping images. The homographies are then chained and used to warp the keypoints in all the images onto I^R. To minimise propagation errors during this process, an optimal chaining order can be estimated (e.g., the minimum spanning tree of the connection graph [11]). Within I^R, the coordinates of keypoints that have the same identity (this is inferred from the pairwise image matching conducted in panorama recognition) are averaged. The result of this process is a set coordinates $\{\mathbf{x}_i^R\}_{i=1}^N$ in I^R, where each $\mathbf{x}_i^R$ is (potentially) matched to a keypoint $\mathbf{x}_i^k$ in the k-th image I^k.

4.2 Bundle adjustment

Given an arbitrary location $\mathbf{x}_*$ in I^R, we wish to estimate a set of location dependent homographies $\{\mathbf{H}_*^k\}_{k=1}^K$, where each $\mathbf{H}_*^k$ maps $\mathbf{x}_*$ from I^R to I^k following

$$\tilde{\mathbf{x}}_*^k \sim \mathbf{H}_*^k \tilde{\mathbf{x}}_*. \tag{17}$$

The pixel intensity at $\mathbf{x}_*$ in I^R is composited from the original intensity at $\mathbf{x}_*$ in I^R (if it exists) and the pixel intensities (if they exist) at locations $\{\mathbf{x}_*^k\}_{k=1}^K$ in $\{I^k\}_{k=1}^K$.

To estimate the required homographies $\{\mathbf{H}_*^k\}_{k=1}^K$ for position $\mathbf{x}_*$, we simultaneously minimize the *transfer error* of all correspondences. Specifically, we minimize the cost

$$E_*(\Theta) = \sum_{i=1}^{N} \frac{w_*^i}{\sum_{k=1}^K \delta_{ik}} \sum_{k=1}^{K} \delta_{ik} \|\mathbf{x}_i^k - f(\mathbf{p}_i, \mathbf{H}_*^k)\|^2, \tag{18}$$

where $\Theta = [\mathbf{H}_*^1, \dots, \mathbf{H}_*^K, \mathbf{p}_1, ..., \mathbf{p}_N]$ and $f(\mathbf{p}, \mathbf{H})$ is the projective warp (in inhomogeneous coordinates) defined as

$$f(\mathbf{p}, \mathbf{H}) = \begin{bmatrix} \dfrac{\mathbf{r}_1[\mathbf{p}^T\ 1]^T}{\mathbf{r}_3[\mathbf{p}^T\ 1]^T} & \dfrac{\mathbf{r}_2[\mathbf{p}^T\ 1]^T}{\mathbf{r}_3[\mathbf{p}^T\ 1]^T} \end{bmatrix}^T, \tag{19}$$

where $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ are the three rows of homography $\mathbf{H}$.

The optimized parameters include the point coordinates $\{\mathbf{p}_i\}_{i=1}^N$, which are essential to "couple" the homographies in bundle adjustment [8]. The coordinates $\{\mathbf{p}_i\}_{i=1}^N$ are initialized as the points $\{\mathbf{x}_i^R\}_{i=1}^N$ resulting from the homography chaining in Sec. 4.1. The k-th homography $\mathbf{H}_*^k$ is initialized using Moving DLT on the correspondences between $\{\mathbf{x}_i^R\}_{i=1}^N$ and keypoints in I^k. Note that not all $\mathbf{x}_i^R$ has a correspondence in I^k; if the correspondence $\{\mathbf{x}_i^R, \mathbf{x}_i^k\}$ exists, the indicator $\delta_{ik} = 1$, else $\delta_{ik} = 0$. The division of

Algorithm 1 Simultaneous refinement of multiple as-projective-as-possible warps for panorama creation.

```
Require: Input images $\{I^k\}_{k=1}^K$ with overlaps.
Choose reference frame $I^R$ from $\{I^k\}_{k=1}^K$.
Map all keypoints from $\{I^k\}_{k=1}^K$ onto $I^R$.
Match points $\{\mathbf{x}_i^R\}_{i=1}^N$ in $I^R$ with points in $\{I^k\}_{k=1}^K$.
Define $C_1 \times C_2$ cells in $I^R$.
for each cell in $I^R$ do
    Compute weights (20) for current cell center $\mathbf{x}_*$.
    for $k = 1, \dots, K$ do
        Apply Moving DLT (10) to yield homography $\mathbf{H}_*^k$.
    end for
    Refine all $\{\mathbf{H}_*^k\}_{k=1}^K$ with bundle adjustment (18).
    Using $\{\mathbf{H}_*^k\}_{k=1}^K$, composite pixels in current cell.
end for
```

each error term in (18) by $\sum_{k=1}^K \delta_{ik}$ ensures that points $\mathbf{x}_i^R$ that are matched in many images do not dominate.

Note that we compute the local weights

$$w_*^i = \max\left(\exp(-\|\mathbf{x}_* - \mathbf{x}_i^R\|^2/\sigma^2), \gamma\right) \tag{20}$$

by referring to the coordinates $\{\mathbf{x}_i^R\}_{i=1}^N$. This ensures that the optimized homographies are locally adapted to $\mathbf{x}_*$. One could also refer the weights to the points $\{\mathbf{p}_i\}_{i=1}^N$ which are iteratively updated. However, our simple scheme is sufficient to satisfactorily achieve the desired effect.

To reduce the number of instances of (18) to solve, as in Sec. 3.3 we partition I^R into cells. The center of each cell is taken as $\mathbf{x}_*$, and the estimated homographies for $\mathbf{x}_*$ are applied to the pixels within the same cell. Further, the Moving DLT initialization across the cells can be accomplished as a series of efficient rank-one updates (see Sec. 3.3). Algorithm 1 summarizes our method. Fig. 8 illustrates the multiple as-projective-as-possible warps estimated.

To give an idea of the size of problem (18), the size of the Jacobian is $(9K + 2N) \times (\sum_{i,k} \delta_{ik})$. However, each error term includes only one point $\mathbf{p}_i$, hence the Jacobian is extremely sparse. We use the sparse Levenberg-Marquardt library of [31] to minimize (18). Invoking Algorithm 1 to create the 7-image panorama in Fig. 11 (second row) took $\approx$ 10 mins (time includes pixel compositing), where each image is of size 2000×1329 pixels, I^R is partitioned in 100×100 cells, and the number of points in I^R is 13380. Of course, since the problems (10) and (18) are independent across the cells, they can be solved in parallel for speedups.

5 Results

We compare our approach against state-of-the-art methods for image alignment. In the following, we refer to our method as APAP (as-projective-as-possible). In our experiments, we select or generate input images that correspond to views that differ by rotation *and* translation. While many data have been tested (including those used elsewhere) with convincing results, only a few can be included in this paper; *refer to the supplementary material for more results.*

5.1 Comparisons with flexible warp methods

First we compare APAP against other flexible warp methods for image stitching, namely, content preserving warps (CPW) [20], dual homography warps (DHW) [22], and smoothly varying affine (SVA) [18]. As mentioned in Sec. 2.3, these methods can only stitch two images at a time, since they either cannot be easily extended to simultaneous estimation, or no such extensions exist in the literature. Our aim is to cogently compare the alignment accuracy of the different image warping methods, thus, we avoid sophisticated postprocessing like seam cutting [3] and straightening [22], and simply blend the aligned images by intensity averaging such that any misalignments remain obvious. For completeness, we also compare with the commercial tools of Autostitch[3] [1] and Photosynth by inputting two images at once. For Photosynth, the final postprocessed results are used since "raw" alignment results are not obtainable in the standard version of the software.

Preprocessing and parameter settings. Given a pair of input images, we first detect and match SIFT keypoints using the VLFeat library [32]. We then run RANSAC to remove mismatches, and the remaining inliers were given to CPW, DHW, SVA and APAP. The good performance of these methods depend on having the correct parameters. For CPW, DHW and SVA, we tuned the required parameters for best results[4]; refer to the respective papers for the list of required parameters. For APAP, we varied the scale σ within the range $[8\ 12]$ for images of sizes 1024×768 to 1500×2000 pixels. The offset γ was chosen from $[0.0025\ 0.025]$. The grid sizes C_1 and C_2 were both taken from the range $[50\ 100]$; on each dataset, the same grid resolution was also used in the CPW grid. In addition, following [20], for CPW we pre-warp the source image with the global homography estimated via DLT on the inliers returned by RANSAC. For Photosynth and Autostitch the original input images (with EXIF tags) were given.

Qualitative comparisons. Figs. 6 and 7 depict results on the *railtracks* and *temple* image pairs (note: the results of Autostitch, Photosynth and APAP on *railtracks* are already shown in Fig. 1). The *railtracks* data is our own, while *temple* was contributed by the authors of [22]. The baseline warp (single homography via DLT on inliers) is clearly unable to satisfactorily align the images since the views do not differ purely by rotation. SVA, DHW and Autostitch are marginally better, but significant ghosting remains. Further, note the highly distorted warp produced by SVA, especially in the extrapolation regions. The errors made by Photosynth seem less "ghostly", suggesting the usage of advanced blending or pixel selection [2] to conceal the misalignments. Nonetheless it is clear that the postprocessing was not completely successful; observe the misaligned rail tracks and tiles on the ground. Contrast the above methods with APAP, which cleanly aligned the two images with few artefacts. This reduces the burden on postprocessing. We

[3] We used the commercial version of Autostitch: "Autopano".

[4] Through personal communication, we have verified the correctness of our implementation of CPW, DHW and SVA and their parameter settings.

have confirmed that feathering blending [2] is sufficient to account for exposure differences in the APAP results.

While CPW with pre-warping is able to produce good results, the rigidity constraints (a grid like in Fig. 4(b) is defined and discouraged from deforming) may counterproductively limit the flexibility of the warp (observe the only slightly nonlinear outlines of the warped images[5]). Thus although the rail tracks and tiles are aligned correctly (more keypoint matches exist in these relatively texture-rich areas to influence the warp), ghosting occurs in regions near the skyline. Note that although APAP introduces a grid, it is for computational efficiency and not to impose rigidity.

Sec. A in the supplementary material shows qualitative comparisons on more image pairs.

Run time information. For DHW, CPW, SVA and APAP (without parallel computation and rank-1 updates), we record the total duration for warp estimation (plus any data structure preparation), pixel warping and blending. All methods were run in MATLAB with C Mex acceleration for warping and blending. DHW and APAP take in the order of seconds, while CPW typically requires tens of seconds. In contrast, SVA scales badly with image size (since larger images yield more keypoint matches), most likely due to the underlying point set registration method [19]. While 8 mins was reported in [18] for 500×500-pixel images, in our experiments SVA takes 15 mins for *temple* (1024×768) and 1 hour for *railtracks* (1500×2000). Autostitch and Photosynth typically take < 7 seconds in our experiments.

Quantitative benchmarking. To quantify the alignment accuracy of an estimated warp $f : \mathbb{R}^2 \mapsto \mathbb{R}^2$, we compute the root mean squared error (RMSE) of f on a set of keypoint matches $\{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^N$, i.e., $\text{RMSE}(f) = \sqrt{\frac{1}{N}\sum_{i=1}^N \|f(\mathbf{x}_i) - \mathbf{x}'_i\|^2}$. Further, for an image pair we randomly partitioned the available SIFT keypoint matches into a “training” and “testing” set. The training set is used to learn a warp, and the RMSE is evaluated over both sets.

We also employed the error metric of [33]: a pixel $\mathbf{x}$ in the source image is labeled as an outlier if there are no similar pixels in the neighbourhood of $f(\mathbf{x})$ in the target image. Following [33], neighbourhood is defined by a 4-pixel radius, and two pixels are judged similar if their intensities differ by less then 10 gray levels. The percentage of outliers resulting from f is regarded as the warping error. Note that pixels that do not exist in the overlapping region are excluded from this measure. Also, in our experiments f is estimated using only the data in the training set.

Table 1 depicts the average errors (over 20 repetitions) on 10 challenging real image pairs, 6 of which were provided by the authors of [22], [18] (see Sec. C in the supp. material for the data). It is clear that APAP provides the lowest errors (RMSE and % outliers) in most of the image pairs.

Dataset	Base	DHW	SVA	CPW	APAP
railtracks -TR	13.91	14.09	7.48	6.69	**4.51**
-TE	13.95	14.12	7.30	6.77	**4.66**
-% outliers	21.14	20.48	16.73	**16.42**	16.86
conssite -TR	12.43	11.24	11.36	7.06	**5.16**
-TE	12.88	11.87	11.56	7.43	**5.88**
-% outliers	11.28	11.24	10.79	11.18	**10.38**
train -TR	14.76	13.38	9.16	6.33	**5.24**
-TE	15.16	13.52	9.84	6.83	**6.06**
-% outliers	18.17	21.01	**10.61**	11.83	11.11
garden -TR	9.06	8.76	8.98	6.36	**5.19**
-TE	9.12	9.01	9.47	7.06	**5.31**
-% outliers	14.03	16.01	13.20	13.54	**13.15**
temple -TR	2.66	6.64	12.30	2.48	**1.36**
(from [22]) -TE	2.90	6.84	12.21	2.54	**2.04**
-% outliers	11.65	12.27	12.29	12.65	**11.52**
carpark -TR	4.77	4.36	4.19	3.60	**1.38**
(from [22]) -TE	4.85	5.67	4.05	3.86	**1.67**
-% outliers	9.50	9.32	9.01	9.28	**8.04**
apartments -TR	10.23	9.06	9.84	6.86	**6.23**
(from [22]) -TE	10.48	9.76	10.12	7.02	**6.40**
-% outliers	4.16	3.10	3.69	3.27	**2.83**
chess/girl -TR	7.92	10.72	21.28	9.45	**2.96**
(from [18]) -TE	8.01	12.38	20.78	9.77	**4.21**
-% outliers	23.35	22.87	22.98	23.44	**21.80**
rooftops -TR	2.90	4.80	3.96	3.16	**1.92**
(from [18]) -TE	3.48	4.95	4.11	3.45	**2.82**
-% outliers	8.66	10.48	10.17	8.24	**8.44**
couch -TR	11.46	10.57	12.04	5.75	**5.66**
(from [18]) -TE	11.84	10.86	12.93	5.92	**5.68**
-% outliers	39.10	38.80	37.20	39.56	**36.68**

TABLE 1: Average RMSE (in pixels, TR = training set error, TE = testing set error) and % outliers over 20 repetitions for 5 methods on 10 image pairs. See Sec. A of the supp. material to view the qualitative stitching results.

To further investigate, we produce synthetic 2D images by projecting 3D point clouds onto two virtual cameras. The point clouds were laser scanned from parts of buildings in a university campus; see Column 1 in Fig. 5 for the point clouds used. The camera intrinsics and poses were controlled such that the projections fit within 200×200-pixel images. The projections yield a set of two-view point matches that permit the direct application of the various warp estimation methods. For each point cloud, we fixed the relative rotation between the cameras at 30°, and varied the *distance* between the camera centers along a fixed direction.

To generate different data instances, we randomly sample 1500 points per point cloud. Fig. 5 shows the average (over 50 repetitions) training and testing RMSE plotted against camera distance (% outlier measure [33] cannot be used here since there are no image pixels). Expectedly, all methods deteriorate with the increase in camera distance. Note that the errors of SVA and CPW do not diminish as the translation tends to zero. For SVA, this is due to its affine instead of projective regularization (in other words, the affine model is the incorrect model, even with no camera translations). For CPW, this indicates that its rigidity preserving distortions may sometimes overly perturb the pre-warping by the homography. In contrast, APAP reduces

[5] As explained in Sec. 2.3, imposing warp rigidity is essential to prevent wobbling in video stabilisation [20]. Note that the original purpose of CPW was for video stabilisation and not image stitching.

gracefully to a global homography as the camera centers coincide, and provides overall the lowest error.

5.2 Comparisons with bundle adjustment

Here, we compare our novel APAP bundle adjustment scheme against Autostitch [1] in simultaneously refining multiple alignment functions. Autostitch uses bundle adjustment to optimize the relative rotations and homographies between a set of overlapping images. Again, to directly compare alignment accuracy, we avoid any advanced compositing, and simply blend the aligned images with intensity averaging (the Autopano commercial version of Autostitch allows postprocessing to be switched off). Since Autostitch prewarps the images onto a cylindrical surface, we also conduct the same prewarping for APAP.

Figs. 9, 10 and 11 show the alignment results respectively on the *construction site*, *garden* and *train* image sets. The images correspond to views that differ by more than pure rotation, as one would expect from a typical tourist's photo collection. The Autostitch results exhibit obvious misalignments; these are highlighted with red circles in the figures. Fundamentally, this is due to being restricted to using homographies for alignment. In contrast, our APAP method (Moving DLT initialization and bundle adjustment refinement) produced much more accurate alignment that maintains a geometrically plausible overall result.

However, both methods (without photometric postprocessing) cannot handle moving objects, which give rise to motion parallax in the mosaic. This is evident in the *train* scene (Fig. 11), where there are many walking pedestrians. Nonetheless, our APAP method handles the static components of the scene much better than Autostitch.

5.3 Stitching full panoramas with postprocessing

Our premise is that more accurate image alignment imposes lower dependency on deghosting and postprocessing. Therefore, methods that can align more accurately tend to produce more satisfactory final results. To demonstrate this point, we stitch full panoramas by incrementally stitching multiple images onto a canvas using Moving DLT. After each image is warped onto the canvas, we apply seam cutting and feathering blending to composite the pixels. The resulting mosaic allows us to compare on an equal footing with Autostich and Photosynth, which by default conduct postprocessing. Sec. B in the supplementary shows results on the *construction site*, *garden* and *train* image sets.

It is evident that our results show much fewer artifacts than Autostitch and Photosynth. In particular, in *train* the motion parallax errors have been dealt with by seam cutting after Moving DLT, without introducing noticeable alignment errors in the other parts of the scene. Photosynth's results show signs of seam cutting and sophisticated pixel blending methods. While noticeable artifacts exist, given the potentially very bad errors from using basic homography alignment, the results of Photosynth show the remarkable ability of postprocessing methods to reduce or conceal much of the misalignment artifacts. Nonetheless our contributed method allows the remaining errors to be eliminated thoroughly via improved image alignment.

6 Conclusion

We have proposed as-projective-as-possible warps for image stitching. The warps are estimated using our novel Moving DLT method. Our method was able to accurately align images that differ by more than a pure rotation. We also propose a novel locally weighted bundle adjustment scheme to simultaneously align multiple images. The results showed that the proposed warp reduces gracefully to a global homography as the camera translation tends to zero, but adapts flexibly to account for model inadequacy as the translation increases. Combined with commonly used postprocessing methods, our technique can produce much better results than state-of-the-art image stitching softwares.

Acknowledgements

The authors express their gratitude to Rhys Hill and Anders Eriksson for helpful discussions on using Ceres [31]. This research was carried out in part at the SeSaMe Centre, supported by the Singapore NRF under its IRC@SG Funding Initiative and administered by the IDMPO.

References

[1] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *IJCV*, vol. 74, no. 1, pp. 59–73, 2007.

[2] R. Szeliski, "Image alignment and stitching," in *Handbook of Mathematical Models in Computer Vision*. Springer, 2005, pp. 273–292.

[3] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, "Interactive digital photomontage," in *ACM SIGGRAPH*, 2004.

[4] A. Eden, M. Uyttendaele, and R. Szeliski, "Seamless image stitching of scenes with large motions and expoure differences," in *CVPR*, 2006.

[5] P. J. Burt and E. H. Adelson, "A multiresolution spline with applications to image mosaics," *ACM Transactions on Graphics*, vol. 2, no. 4, pp. 217–236, 1983.

[6] P. Perez, M. Gangnet, and A. Blake, "Poisson image editing," in *ACM SIGGRAPH*, 2003.

[7] J. Jia and C.-K. Tang, "Image stitching using structure deformation," *IEEE TPAMI*, vol. 30, no. 4, pp. 617–631, 2008.

[8] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – a modern synthesis," in *Vision Algorithms: Theory and Practice, LNCS*. Springer Verlag, 2000, pp. 298–375.

[9] H.-Y. Shum and R. Szeliski, "Construction of panoramic mosaics with global & local alignment," *IJCV*, vol. 36, no. 2, 2000.

[10] E.-Y. Kang, I. Cohen, and G. Medioni, "A graph-based global registration for 2D mosaics," in *ICPR*, 2000.

[11] R. Marzotto, A. Fusiello, and V. Murino, "High resolution video mosaicing with global alignment," in *CVPR*, 2004.

[12] S. Schaefer, T. McPhail, and J. Warren, "Image deformation using moving least squares," in *SIGGRAPH*, 2006.

[13] J. Zaragoza, T.-J. Chin, M. S. Brown, and D. Suter, "As-projective-as-possible image stitching with Moving DLT," in *CVPR*, 2013.

[14] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet, "Mosaicing on adaptive manifolds," *IEEE TPAMI*, vol. 22, no. 10, 2000.

[15] A. Agarwala, M. Agarwala, M. Cohen, D. Salesin, and R. Szeliski, "Photographing long scenes with multi-viewpoint panoramas," in *ACM SIGGRAPH*, 2006.

[16] F. Dornaika and R. Chung, "Mosaicking images with parallax," *Signal Processing: Image Communication*, vol. 19, no. 8, pp. 771 – 786, 2004.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal on Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

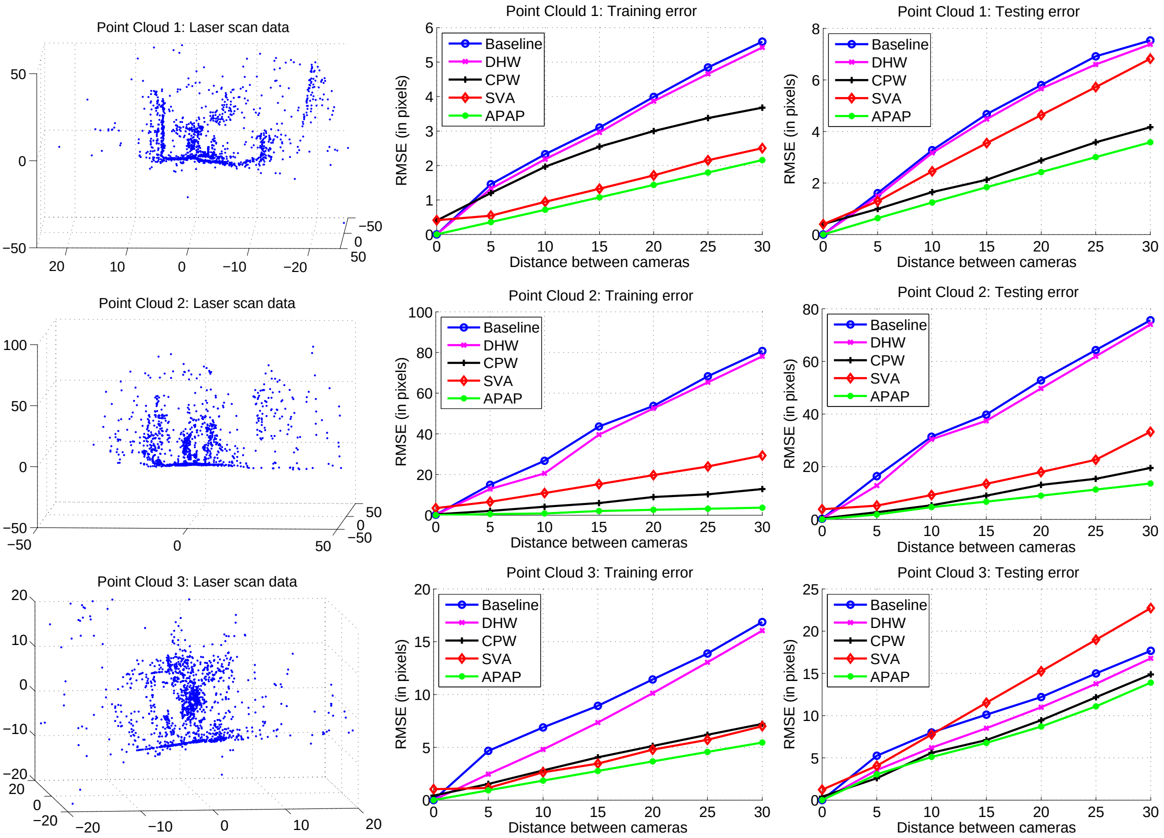


Fig. 5: Point cloud (left) and average RMSE on the training set (middle) and the testing set (right) as a function of inter-camera translational distance.

[18] W.-Y. Lin, S. Liu, Y. Matsushita, T.-T. Ng, and L.-F. Cheong, "Smoothly varying affine stitching," in *CVPR*, 2011.

[19] A. Mynorenko, X. Song, and M. Carreira-Perpinan, "Non-rigid point set registration: coherent point drift," in *NIPS*, 2007.

[20] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, "Content-preserving warps for 3d video stabilization," in *ACM SIGGRAPH*, 2009.

[21] T. Igarashi, T. Moscovich, and J. F. Hughes, "As-rigid-as-possible shape manipulation," *ACM TOG*, vol. 24, no. 3, 2005.

[22] J. Gao, S. J. Kim, and M. S. Brown, "Constructing image panoramas using dual-homography warping," in *CVPR*, 2011.

[23] Z. Zhang, "Parameter estimation techniques: a tutorial with application to conic fitting," *IVC*, vol. 15, no. 1, pp. 59–76, 1997.

[24] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE TPAMI*, vol. 19, no. 6, pp. 580–593, 1997.

[25] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Computing and rendering point set surfaces," *IEEE TVCG*, vol. 9, no. 1, pp. 3–15, 2003.

[26] G. Guennebaud and M. Gross, "Algebraic point set surfaces," *ACM TOG*, vol. 26, no. 3, 2007.

[27] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[28] Q.-H. Tran, T.-J. Chin, G. Carneiro, M. S. Brown, and D. Suter, "In defense of RANSAC for outlier removal in deformation registration," in *ECCV*, 2012.

[29] P. Stange, "On the efficient update of the singular value decomposition," in *App. Mathematics and Mechanics*, 2008.

[30] G. H. Golub and C. F. van Loan, *Matrix computations*, 3rd ed. The Johns Hopkins University Press, 1996.

[31] S. Agarwal and K. Mierle, "Ceres solver," https://code.google.com/p/ceres-solver/.

[32] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[33] W.-Y. Lin, L. Liu, Y. Matsushita, and K.-L. Low, "Aligning images in the wild," in *CVPR*, 2012.

Julio Zaragoza received his B.Eng. in Computer Systems from ITM (Morelia, Mexico) in 2006 and his masters degree from the National Institute of Astrophysics, Optics and Electronics (INAOE, Puebla, Mexico) in 2010. He is a PhD student within the Australian Centre for Visual Technologies (ACVT) in the University of Adelaide, Australia. His main research interests include model fitting, image stitching and probabilistic graphical models.

Tat-Jun Chin received a B.Eng. in Mechatronics Engineering from Universiti Teknologi Malaysia in 2003 and subsequently in 2007 a PhD in Computer Systems Engineering from Monash University, Victoria, Australia. He was a Research Fellow at the Institute for Infocomm Research in Singapore 2007-2008. Since 2008 he is a Lecturer at The University of Adelaide, Australia. His research interests include robust estimation and statistical learning methods in Computer Vision.

Quoc-Huy Tran graduated with first class honours in computer science and engineering from Ho Chi Minh City University of Technology, Vietnam in 2010. He is currently working towards the PhD degree in Mathematical and Computer Sciences at the Australian Centre for Visual Technologies in the University of Adelaide, Australia. His research interests include model fitting, deformable registration and object tracking. He is a student member of the IEEE.

Michael S. Brown obtained his BS and PhD in Computer Science from the University of Kentucky in 1995 and 2001 respectively. He was a visiting PhD student at the University of North Carolina at Chapel Hill from 1998-2000. Dr. Brown has held positions at the Hong Kong University of Science and Technology (2001-2004), California State University - Monterey Bay (2004-2005), and Nanyang Technological University (2005-2007). He joined the School of Computing at the National University of Singapore in 2007 where he is currently an associate professor and assistant dean (external relations). His research interests include computer vision, image processing and computer graphics. He has served as an area chair for CVPR, ICCV, ECCV, and ACCV and is currently an associate editor for the IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI). Dr. Brown received the HKUST Faculty Teaching Award (2002), the NUS Young Investigator Award (2008), the NUS Faculty Teaching Award (for AY08/09, AY09/10, AY10/11), and the NUS Annual Teaching Excellence Award (ATEA) for AY09/10.

David Suter received a B.Sc.degree in applied mathematics and physics (The Flinders University of South Australia 1977), a Grad. Dip. Comp. (Royal Melbourne Institute of Technology 1984)), and a Ph.D. in computer science (La Trobe University, 1991). He was a Lecturer at La Trobe from 1988 to 1991; and a Senior Lecturer (1992), Associate Professor (2001), and Professor (2006-2008) at Monash University, Melbourne, Australia. Since 2008 he has been a professor in the school of Computer Science, The University of Adelaide. He is head of the School of Computer Science. He served on the Australian Research Council (ARC) College of Experts from 2008-2010. He is on the editorial boards of International Journal of Computer Vision. He has previously served on the editorial boards of Machine Vision and Applications and the International Journal of Image and Graphics. He was General co-Chair or the Asian Conference on Computer Vision (Melbourne 2002) and is currently co-Chair of the IEEE International Conference on Image Processing (ICIP2013).

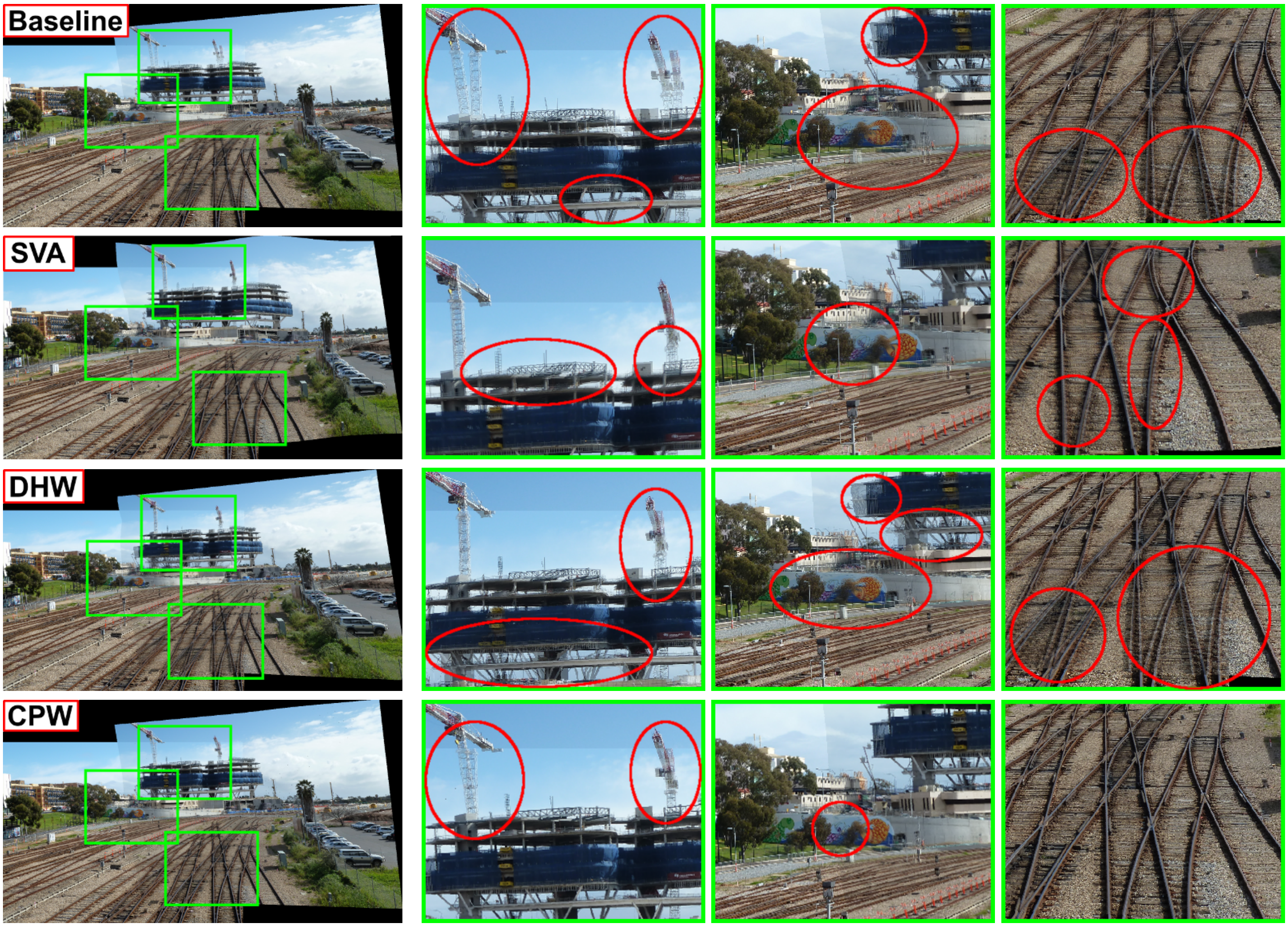


Fig. 6: Qualitative comparisons (best viewed on screen) on the *railtracks* image pair. Red circles highlight errors. List of abbreviations: SVA-Smoothly Varying Affine, DHW-Dual Homography Warps, CPW-Content Preserving Warps. Note that results for Autostich, Photosynth and APAP are shown in Fig. 1.

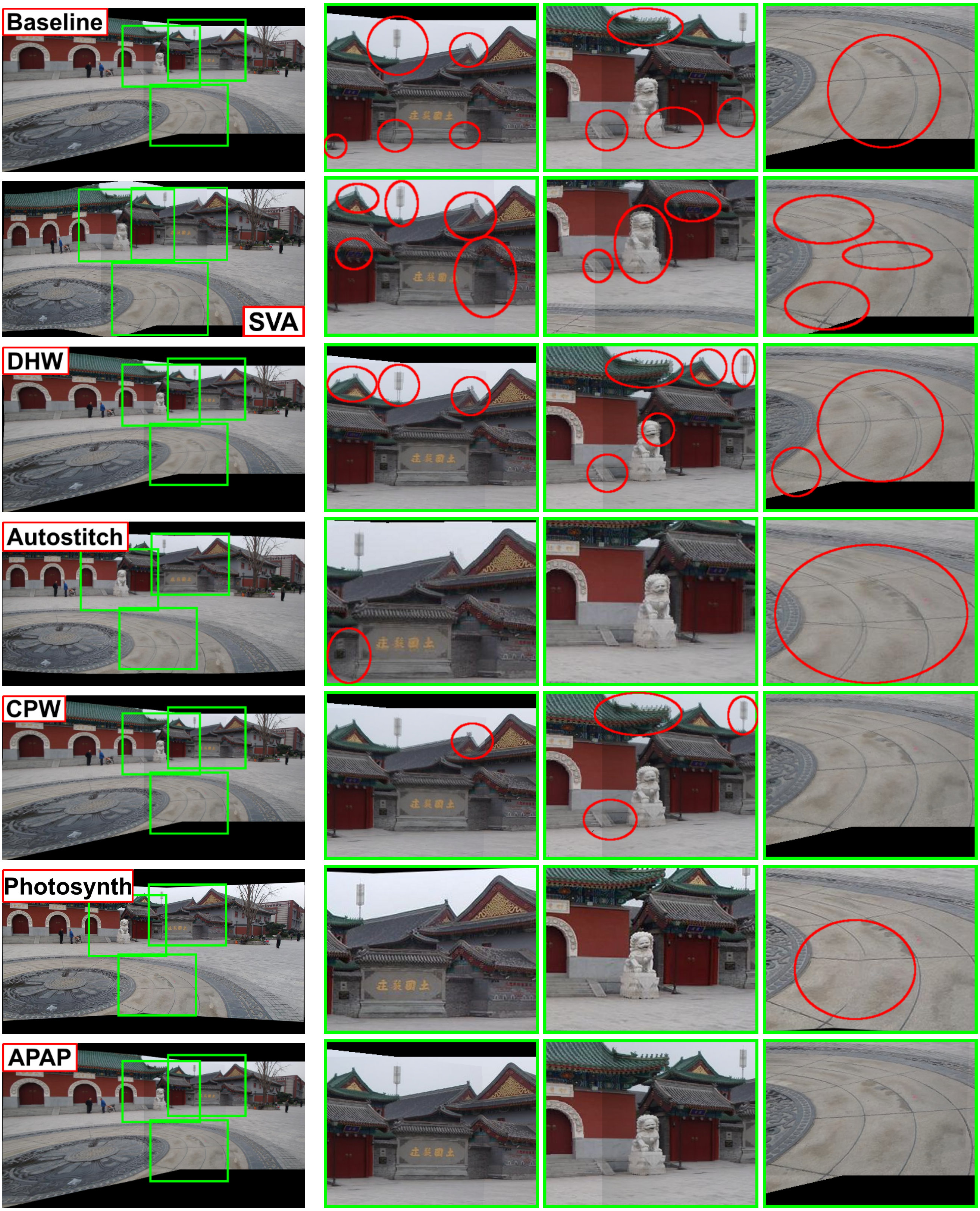


Fig. 7: Qualitative comparisons (best viewed on screen) on the *temple* image pair. Red circles highlight errors. List of abbreviations: SVA-Smoothly Varying Affine, DHW-Dual Homography Warps, CPW-Content Preserving Warps, APAP-As Projective As Possible Warps.

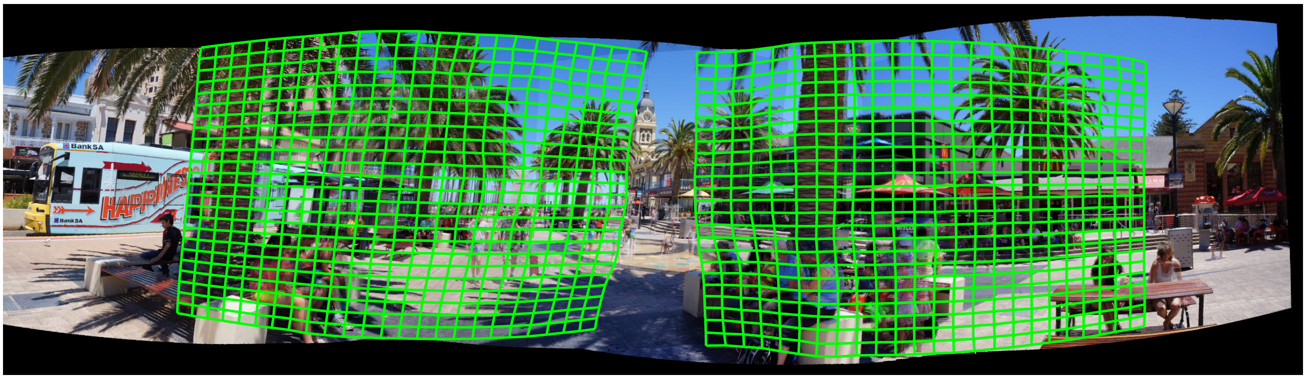


Fig. 8: Aligned images with transformed cells overlaid to visualize the multiple as-projective-as-possible warps refined using our novel bundle adjustment scheme (Sec. 4).

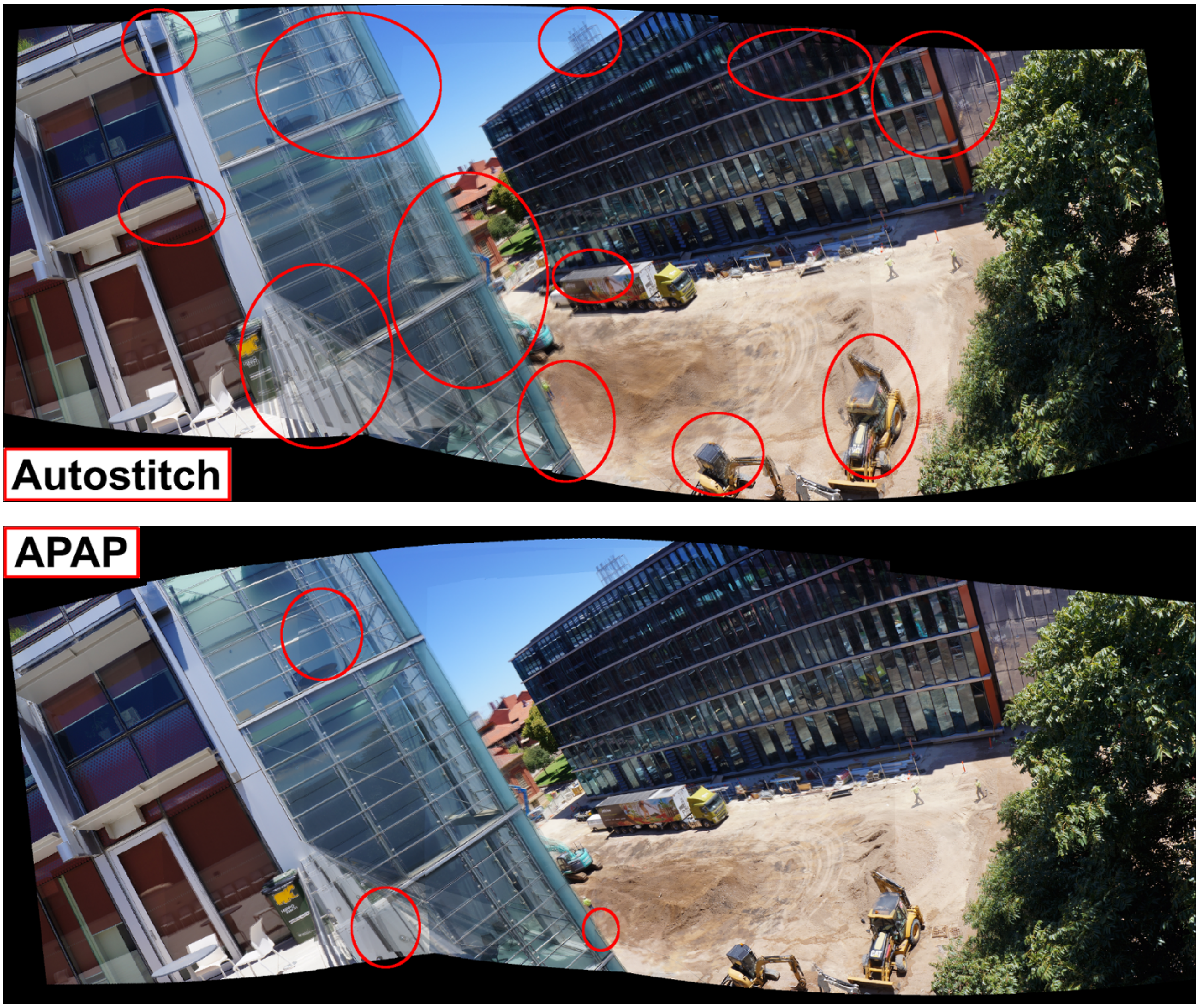


Fig. 9: Alignment results on the *construction site* image set (best viewed on screen). Red circles highlight errors. Note that photometric postprocessing was *not* applied on Autostitch and APAP.

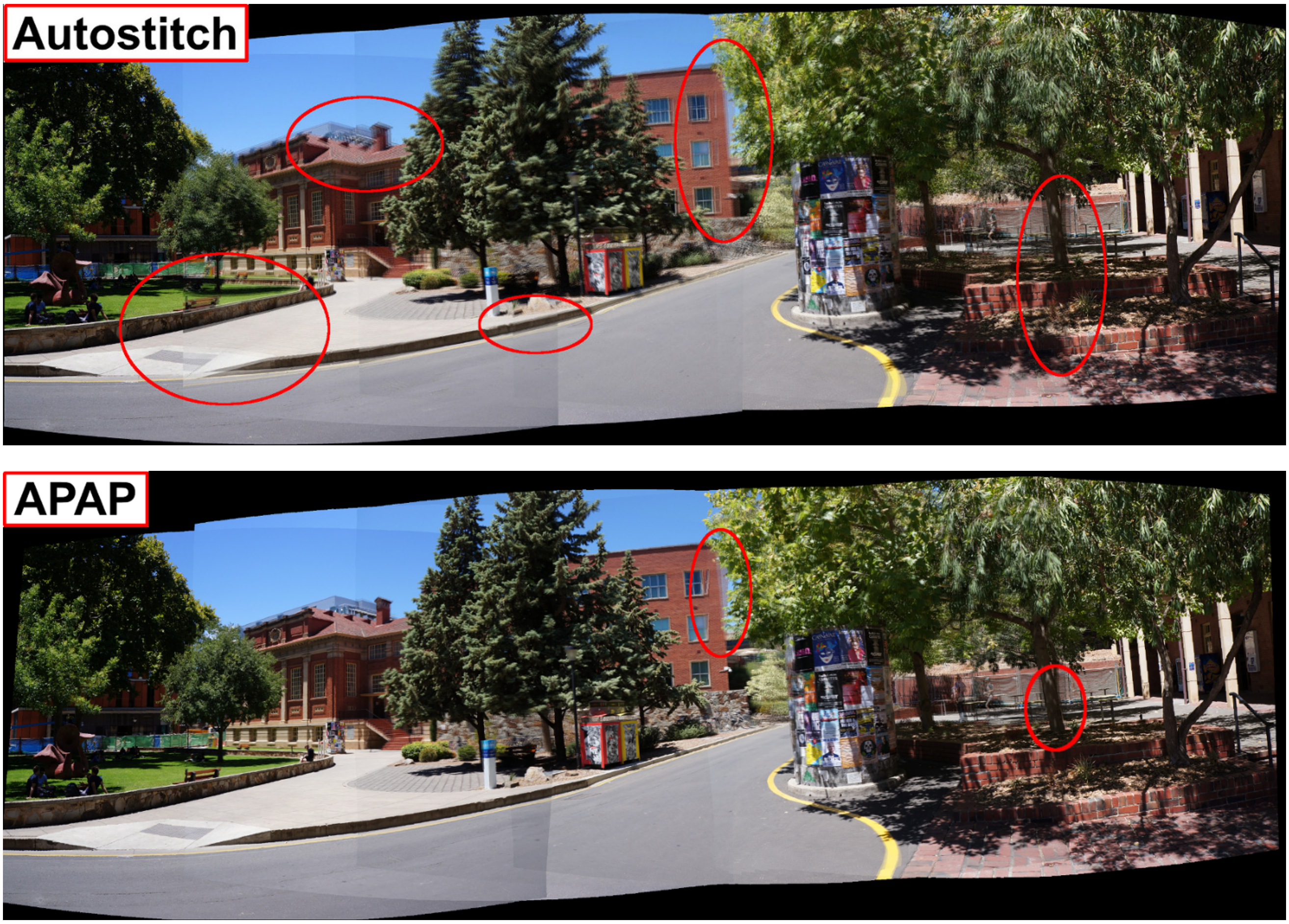


Fig. 10: Alignment results on the *garden* image set (best viewed on screen). Red circles highlight alignment errors. Note that photometric postprocessing was *not* applied on Autostitch and APAP.

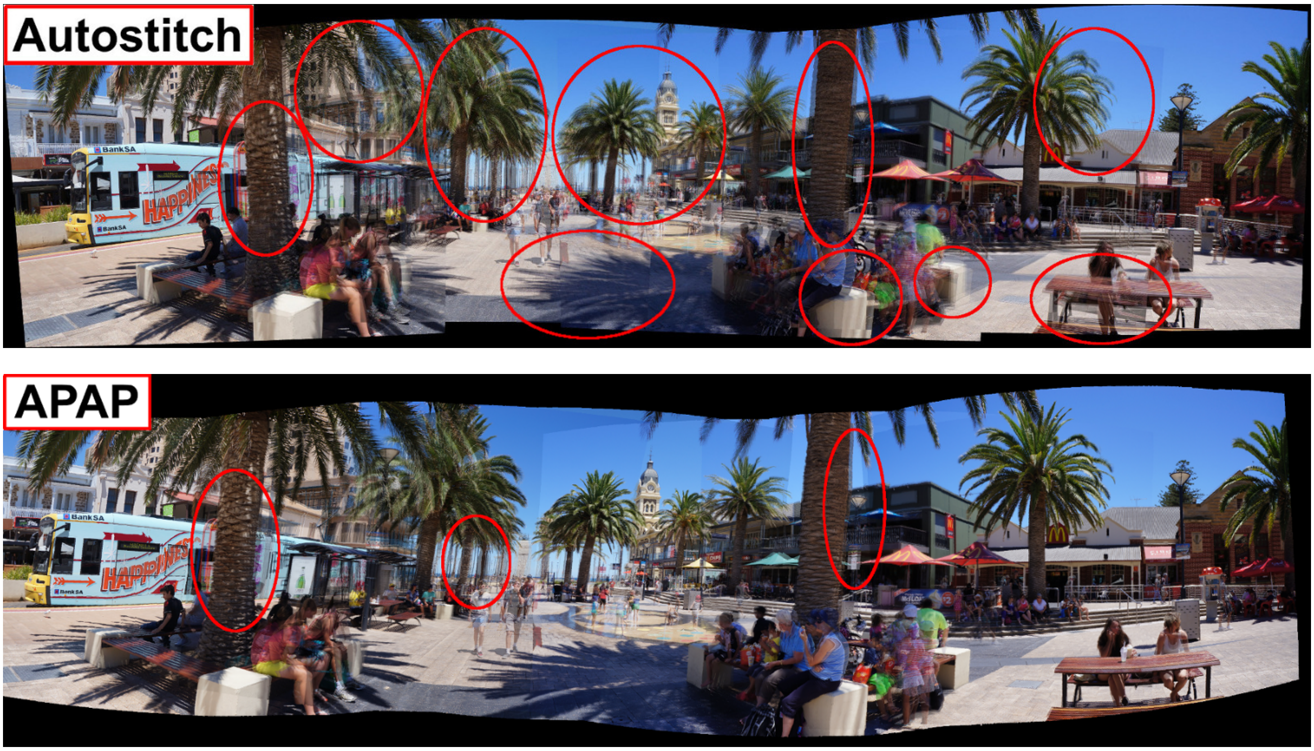


Fig. 11: Alignment results on the *train* image set (best viewed on screen). Red circles highlight errors. Note that photometric postprocessing was *not* applied on Autostitch and APAP.