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

Julio Zaragoza* Tat-Jun Chin* Michael S. Brown† David Suter*
*Australian Centre for Visual Technologies, The University of Adelaide
†School of Computing, National University of Singapore

Abstract

We investigate projective estimation under model inadequacies, i.e., when the underpinning assumptions of the projective model are not fully satisfied by the data. We focus on the task of image stitching which is customarily solved by estimating a projective warp — a model that is justified when the scene is planar or when the views differ purely by rotation. Such conditions are easily violated in practice, and this yields stitching results with ghosting artefacts that necessitate the usage of deghosting algorithms. To this end we propose as-projective-as-possible warps, i.e., warps that aim to be globally projective, yet allow local non-projective deviations to account for violations to the assumed imaging conditions. Based on a novel estimation technique called Moving Direct Linear Transformation (Moving DLT), our method seamlessly bridges image regions that are inconsistent with the projective model. The result is highly accurate image stitching, with significantly reduced ghosting effects, thus lowering the dependency on post hoc deghosting.

1. Introduction

> Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful.
>
> George E. P. Box

This famous advice by an eminent statistician rings true in many scientific disciplines, including computer vision. In this paper, we are primarily concerned with model inadequacies in projective estimation. More specifically, we consider situations where the enabling assumptions for the projective model are not fully met by the data, thus fundamentally limiting the achievable goodness of fit.

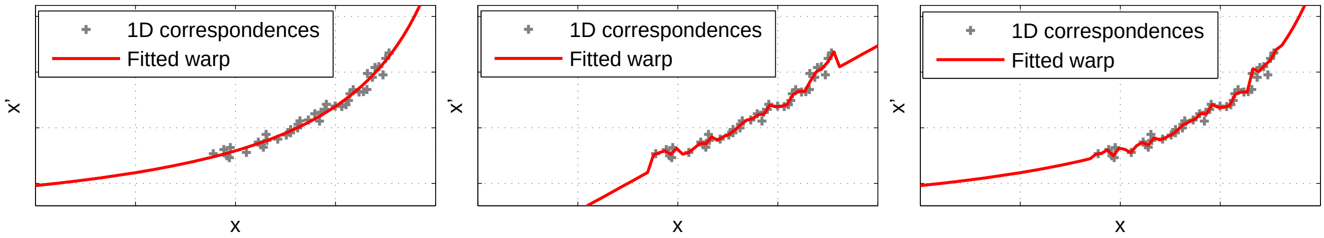
We focus on image stitching, though we envision our methods to be more widely applicable, e.g., in video stabilisation. Image stitching is typically solved by estimating 2D projective warps to bring images into alignment. Parametrised by 3×3 homographies, 2D projective warps are justified if the scene is planar or if the views differ purely by rotation [17]. In reality, in the hands of the casual user the conditions will unlikely be fully satisfied. Thus the projective model cannot adequately characterise the required warp, causing misalignments or ghosting effects. Note that such errors are due to inherent deficiencies in the model and not just noise perturbations; Fig. 1(a) illustrates.

Many commercial stitching software like Autostitch and Photosynth (specifically the panorama tool) use projective warps[1], arguably for their simplicity. When the requisite imaging conditions are not met, their success relies on deghosting algorithms to remove unwanted artefacts [17]. Here, we offer a different strategy: instead of relying on a projective model (which is often inadequate) and then fix the resulting errors, we adjust the model based on the data to improve the fit. We achieve this by our novel *as-projective-as-possible* warps, i.e., warps that aim to be globally projective, yet allow local deviations to account for model inadequacy; Fig. 1(c) illustrates. Our method significantly reduces alignment errors, yet is able to maintain overall geometric plausibility. Fig. 3 shows a sample result.

Note that our aim is not to perform image stitching for arbitrary camera motions (e.g., [12]). Rather, our aim is to tweak the projective model to fit the data as accurately as possible. It is also not our goal to dispense with deghosting algorithms, which are still useful if there are serious misalignments or moving objects. However, we argue that a good initial stitch is very desirable since it imposes a much lower requirement on subsequent deghosting and postprocessing; the result in Fig. 3, for example, was composited using simple pixel averaging with little noticeable ghosting.

More fundamentally, we learn the proposed warp based on a novel estimation technique called Moving DLT. It is inspired by the Moving Least Squares (MLS) method [2] for image manipulation [14], but our method applies projective regularisation instead of rigid or affine regularisation. This is essential to ensure that the warp extrapolates correctly beyond the image overlap (interpolation) region to maintain perceptual realism. Figs. 1(b) and 1(c) contrast warps from

[1] Both tools require the camera to rotate about a point, or that the photos be taken from the same spot and with the same focal length. See:
- http://www.cs.bath.ac.uk/brown/autostitch/autostitch.html#FAQ
- http://photosynth.net/faq.aspx



(a) Projective warp.

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

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

Figure 1. A 1D analogy of image stitching, with a set of 1D correspondences $\{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^N$ generated by projecting a 2D point cloud onto two 1D image "planes". 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 [14], 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.

Moving DLT and MLS. Being able to interpolate flexibly to minimise ghosting and extrapolate corectly to maintain geometric consistency are vital qualities for image stitching.

Closer to our method is the surface approximation work of [6], where spheres are fitted using *algebraic* MLS onto point clouds. Our work is different in that we fit projective functions instead of geometric surfaces. Further, function extrapolation is a crucial aspect that was not stressed in [6].

The rest of the paper is organised as follows: Sec. 1.1 surveys related work. Secs. 2 and 3 introduce the proposed warp and its efficient learning for image stitching. Results are presented in Sec. 4, and we conclude in Sec. 5.

1.1. Related work

While the fundamentals of image stitching are well studied (see [17] for an excellent survey), how to produce good results when the data is noisy or uncooperative is an open problem. In our context, we categorise previous works into two groups: (1) methods that reduce ghosting by constructing better alignment functions, and (2) methods that reduce ghosting *after* alignment using advanced methods in compositing, pixel selection or blending. In the second group, seam cutting [1, 3] and Poisson blending [13] are influential. Since our approach belongs to the first group, we review such methods in the following. Ideally, methods from both groups should be jointly used for best results.

Shum and Szeliski [15] first perform bundle adjustment to optimise the rotations and focal lengths of all views. For each feature, the average of the backprojected rays from each view is taken, which is subsequently projected again onto each view to yield the revised feature positions in 2D. The function of the remaining registration errors are then modelled with bilinear kernels, and used in the final alignment. While a very principled method, the backprojection requires camera intrinsics which may not be available.

In the context of video stabilisation, Liu et al. [10] proposed content preserving warps. Given matching features between the original and target image frames, the novel view is synthesised by warping the original image using an as-similar-as-possible warp [8] that jointly minimises 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 smoothly interpolating projective warp. Imposing scene rigidity minimises the dreaded "wobbling" effect in the smoothed video. However, as we show in Sec. 4, 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 [10] since the original and smoothed camera paths are close (see Sec. 4 in [10]), i.e., the views to align are close to begin with.

A recent work proposed smoothly varying affine warps for image stitching [9]. The basis of their idea is the point-set registration method based on motion coherence [11]. An interesting innovation of [9] is an affine initialisation of the registration function, which is then deformed locally to minimise registration errors while maintaining global affinity. Fundamentally, using affine regularisation may be sub-optimal, since an affinity does not contain sufficient degrees of freedom to achieve a fully perspective warp [17], e.g., an affine warp may counterproductively preserve parallelism. Indeed, as Figs. 4 and 5 (second row) show, while the method can interpolate flexibly, it produces highly distorted results in the extrapolation region, where there are no data to guide the local deformation and the warp reverts to global affinity; Fig. 1(b) provides a 1D analogy.

By assuming that the scene contains a ground plane and a distant plane, Gao et al. [4] proposed dual homography warps for image stitching. Essentially theirs is a special case of a piece-wise projective 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 number of required homographies and their parameters.

2. As-Projective-As-Possible Warps

We first review the estimation of projective transformations customarily used in image stitching, and then describe the proposed as-projective-as-possible warp.

2.1. The 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 or homography aims to map $\mathbf{x}$ to $\mathbf{x}'$ following the relation

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

where $\tilde{\mathbf{x}}$ is $\mathbf{x}$ in homogeneous coordinates, and $\mathbf{H} \in \mathbb{R}^{3\times 3}$ defines the homography. In inhomogeneous coordinates,

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

where $\mathbf{h}_j^T$ is the j-th row of $\mathbf{H}$. The divisions in (2) cause the warp to be non-linear, as Fig. 1(a) shows for the 1D case.

DLT 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'. Eq. (1) is rewritten as the implicit condition $\mathbf{0}_{3\times 1} = \tilde{\mathbf{x}}' \times \mathbf{H}\tilde{\mathbf{x}}$ and linearised

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

Only two of the rows are linearly independent. Let $\mathbf{a}_i \in \mathbb{R}^{2\times 9}$ be the first-two rows of (3) computed for the i-th datum $\{\mathbf{x}_i, \mathbf{x}'_i\}$. DLT estimates the nine elements of $\mathbf{H}$ as

$$\hat{\mathbf{h}} = \underset{\mathbf{h}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\mathbf{a}_i\mathbf{h}\|^2 = \underset{\mathbf{h}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{h}\|^2 \tag{4}$$

with the constraint $\|\mathbf{h}\| = 1$, where matrix $\mathbf{A} \in \mathbb{R}^{2N\times 9}$ is obtained by stacking vertically $\mathbf{a}_i$ for all i. The solution is simply the least significant right singular vector of $\mathbf{A}$.

Given the estimated $\mathbf{H}$ (reshaped from $\hat{\mathbf{h}}$), to align the images, an arbitrary pixel at position $\mathbf{x}_*$ in the source image I is warped to the position $\mathbf{x}'_*$ in the target image I' by

$$\tilde{\mathbf{x}}'_* = \mathbf{H}\tilde{\mathbf{x}}_*. \tag{5}$$

To avoid issues with numerical precision, prior to DLT the data can first be normalised in the manner of [7], with the estimated $\mathbf{H}$ then denormalised before executing (5).

2.2. Moving DLT

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

$$\tilde{\mathbf{x}}'_* = \mathbf{H}_*\tilde{\mathbf{x}}_*, \tag{6}$$

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

$$\mathbf{h}_* = \underset{\mathbf{h}}{\operatorname{argmin}} \sum_{i=1}^{N} \|w_*^i \mathbf{a}_i\mathbf{h}\|^2 \tag{7}$$

subject to $\|\mathbf{h}\| = 1$. The scalar weights $\{w_*^i\}_{i=1}^N$ change according to $\mathbf{x}_*$ and are calculated as

$$w_*^i = \exp(-\|\mathbf{x}_* - \mathbf{x}_i\|^2/\sigma^2). \tag{8}$$

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 (8) 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 (5) which uses a single and global projective warp $\mathbf{H}$ for all $\mathbf{x}_*$. Moreover, as $\mathbf{x}_*$ is *moved* continuously in its domain I, the warp $\mathbf{H}_*$ also varies smoothly. This produces an overall warp that adapts flexibly to the data, yet attempts to be as-projective-as-possible. Figs. 1(c) and 3(c) illustrate such a warp in 1D and 2D. We call this estimation procedure *Moving DLT*.

The problem in (7) can be written in the matrix form

$$\mathbf{h}_* = \underset{\mathbf{h}}{\operatorname{argmin}} \|\mathbf{W}_*\mathbf{A}\mathbf{h}\|^2, \tag{9}$$

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

$$\mathbf{W}_* = \operatorname{diag}([\, w_*^1 \; w_*^1 \; \dots \; w_*^N \; w_*^N \,]) \tag{10}$$

and diag() 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 (9) may be unstable when many of the weights are insignificant, e.g., when $\mathbf{x}_*$ is in a data poor or extrapolation region. To prevent numerical issues in the estimation, we offset the weights with a small value $\gamma \in [0 \; 1]$

$$w_*^i = \max\left(\exp(-\|\mathbf{x}_* - \mathbf{x}_i\|^2/\sigma^2), \gamma\right). \tag{11}$$

This also serves to regularise the warp, whereby a high γ reduces the warp complexity; in fact as $\gamma \rightarrow 1$ the warp reduces to the global projective warp. Fig. 2 depicts results from MDLT without regularisation, while Fig. 1(c) shows results on the same data with regularisation.

Conceptually, Moving DLT can be seen as the projective version of MLS [2]. In the context of warping points in 2D for image manipulation [14], MLS estimates for each $\mathbf{x}_*$ an *affine* transformation defined by a matrix $\boldsymbol{F}_* \in \mathbb{R}^{2\times 3}$

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

Including nonstationary weights $\{w_*^i\}_{i=1}^N$ produces flexible warps, but such warps are ultimately only as-affine-as-possible; see Fig. 1(b) for a 1D analogy. Moreover, the concern in [14] is on further limiting the overall flexibility of the warp, in order to avoid undesired shearing of shapes.

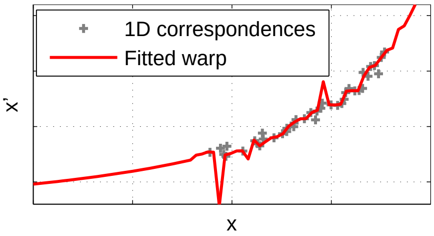


Figure 2. Results from Moving DLT *without* regularisation for a 1D projective estimation problem on synthetic data.

3. Efficient Learning for Image Stitching

Here we describe an efficient algorithm for image stitching based on the proposed warp. We first remove the mismatches among $\{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^N$ using RANSAC [17] with DLT as the minimal solver for a global homography. One might argue against RANSAC since we consider cases where the inliers may deviate from the projective model. In practice, the error of the outliers is orders of magnitude larger than the inlier deviations, thus RANSAC can be effectively used.

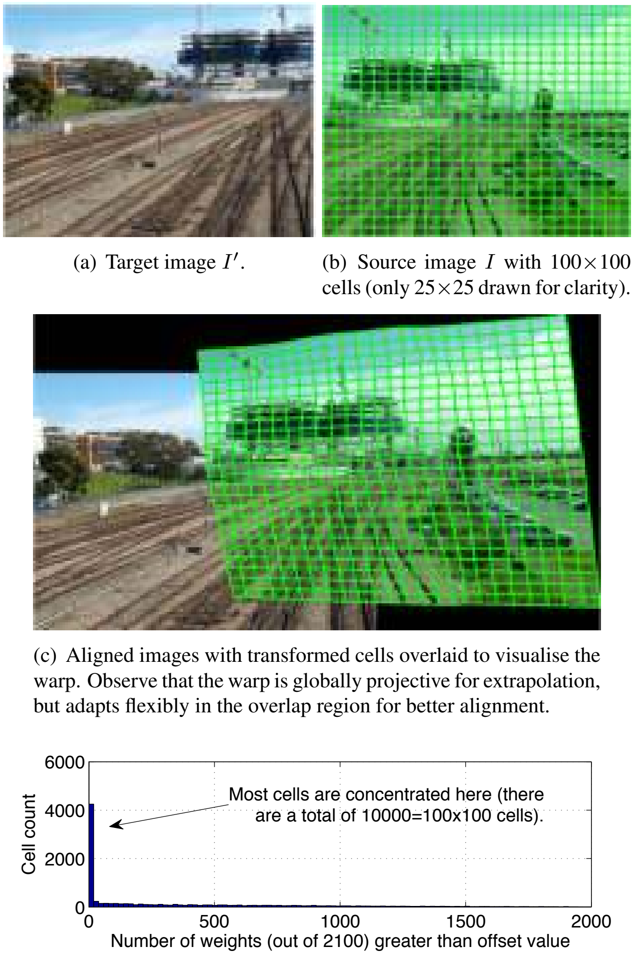
Partitioning into cells. Solving (9) for all pixel positions $\mathbf{x}_*$ in the source image I is wasteful, since neighbouring positions yield practically the same estimates of $\mathbf{H}_*$. Following [14], we partition the source image I into a grid of $C_1 \times C_2$ cells. For each cell, the centre coordinate is chosen as $\mathbf{x}_*$, and all pixels within the same cell are warped using the same $\mathbf{H}_*$. We thus reduce the number of instances of WSVD to $C_1 \times C_2$. Fig. 3(c) illustrates a warp learnt with 100×100 cells for a 1500×2000-pixel image pair.

In addition, observe that the WSVD for all cells can be solved independently. Thus a straightforward approach for speedup is to solve the multiple instances of WSVD in parallel. Note that, even without parallel computations, learning the warp in Fig. 3 with 100×100 cells and $N = 2100$ keypoint matches ($\mathbf{A}$ is of size 4200×9) takes less than a minute on a Pentium i7 2.2GHz Quad Core machine.

Updating weighted SVDs. Further speedups are possible if we realise that, for most cells, due to the offsetting (11) many of the weights do not differ from the offset γ. Based on the images in Figs. 3(a) and 3(b), Fig. 3(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 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

$$\mathbf{A}^T \mathbf{W}_\gamma^T \mathbf{W}_\gamma \mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^T \tag{12}$$


6000
4000
2000
0
Cell count
0
500
1000
1500
2000
Number of weights (out of 2100) greater than offset value
Most cells are concentrated here (there are a total of 10000=100x100 cells).

(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.

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

Figure 3. 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.

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 updated diagonal matrix

$$\mathbf{D} + \rho \bar{\mathbf{r}}_i \bar{\mathbf{r}}_i^T = \tilde{\mathbf{C}} \tilde{\mathbf{D}} \tilde{\mathbf{C}}^T \in \mathbb{R}^{m \times m} \tag{13}$$

can be done efficiently using secular equations [16]. 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 $\tilde{\mathbf{C}}$ [16]. 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 [5]. Note that in (9), $(n = 2N) \gg (m = 9)$.

4. Results

We compare our as-projective-as-possible (APAP) warp against other warp improvement methods for image stitching, namely, content preserving warps (CPW) [10], dual homography warps (DHW) [4], and smoothly varying affine (SVA) [9]. To cogently differentiate the methods, we avoid sophisticated postprocessing like seam cutting and straightening such as in [4], and simply blend the aligned images by intensity averaging such that any misalignments remain obvious. We also compare against Autostitch and Photosynth's panorama tool. For Photosynth the final postprocessed results are used since the raw alignment is not given.

We select testing images which correspond to views that differ by more than a pure rotation. While a number of images 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*.

Preprocessing and parameter settings. Given a pair of input images, we first detect and match SIFT keypoints using the VLFeat library [18]. We then run RANSAC as described in Sec. 3 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[2]; 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 image pair, the same grid resolution was also used in the CPW grid. In addition, following [10], 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.

4.1. Qualitative comparisons

Figs. 4 and 5 depict results on the *railtracks* and *temple* image pairs. The former is our own data, while the latter was contributed by the authors of [4]. The baseline warp (global 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 [17] 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 have confirmed that pyramid blending [17] is sufficient to account for exposure differences and to smoothen the blend.

While CPW with pre-warping is able to produce good results, the rigidity constraints (a grid like in Fig. 3(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[3]). 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.

Run time information. For DHW, CPW, SVA and APAP (without WSVD updating), we record the total duration for warp estimation (plus any data structure preparation time), pixel warping and blending. All methods were run in MATLAB with C Mex acceleration for warping and blending. DHW runs in the order of seconds, While CPW and APAP typically take tens of seconds. In contrast, SVA scales badly with image size (since larger images yield more keypoint matches). While 8 mins was reported in [9] for 500×500 images, in our experiments SVA takes 15 mins for *temple* (1024×768) and 1 hour for *railtracks* (1500×2000).

Constructing full panoramas. Given more than two images, we first choose a central image to initialise the panorama. We then incrementally warp the other images via APAP onto the panorama. Refer to the supplementary material for the outcomes, where we simply blend with pixel averaging to highlight the accuracy of the proposed warp. While a simultaneous alignment of all images [17] is possible with our APAP method, we leave this as future work.

4.2. 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., $\mathrm{RMSE}(f) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|f(\mathbf{x}_i) - \mathbf{x}'_i\|^2}$. 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. Table 1 depicts the average RMSE (over 20 repetitions) of the different methods on 5 challenging real image pairs, 4 of which were used in [4, 9]. It is clear that APAP consistently outperformed the others. Refer to the supplementary material for qualitative comparisons.

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

[3] As explained in Sec. 1.1, imposing warp rigidity is essential to prevent wobbling in video stabilisation, which is the original aim of [10].

Image pair	Baseline	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**
temple -TR	2.66	6.64	12.30	2.48	**1.36**
(from [4]) -TE	2.90	6.84	12.21	2.54	**2.04**
carpark -TR	4.77	4.36	4.19	3.60	**1.38**
(from [4]) -TE	4.85	5.67	4.05	3.86	**1.67**
chess/girl -TR	7.92	10.72	21.28	9.45	**2.96**
(from [9]) -TE	8.01	12.38	20.78	9.77	**4.21**
rooftops -TR	2.90	4.80	3.96	3.16	**1.92**
(from [9]) -TE	3.48	4.95	4.11	3.45	**2.82**

Table 1. Average RMSE (in pixels) over 20 repetitions for 5 methods on 5 image pairs. TR = training error, TE = testing error.

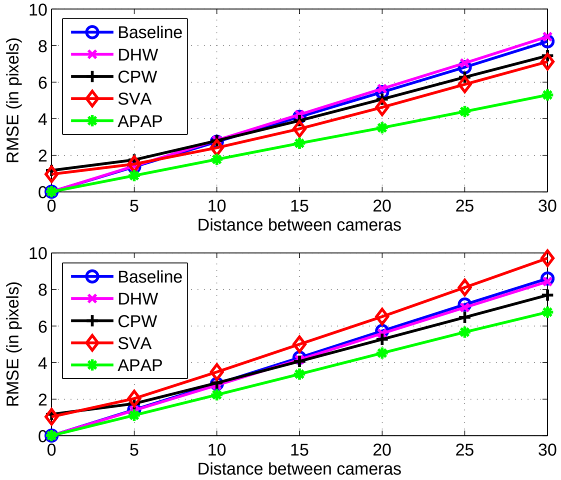


Figure 6. Average RMSE on (top) the training set and (bottom) the testing set as a function of inter-camera translational distance.

To further investigate, we produce synthetic 2D images by projecting randomly generated 3D point clouds onto two cameras. In each instance, 200 points are created, where the 3D coordinates and camera intrinsics are controlled such that the projections fit within 200 × 200-pixel images. This permits the direct application of the various warp estimation methods. For each point cloud, we fix the relative rotation between the cameras at 60°, but vary the *distance* between the camera centres along a fixed direction. As before, we partition the point matches into a training and testing set.

Fig. 6 shows the average RMSE (over 50 repetitions) plotted against distance. Expectedly, all methods deteriorate with the increase in distance. However, observe that the errors of SVA and CPW do not reduce to zero as the translation tends to zero. For SVA this is most likely due to its affine instead of projective regularisation; cf. Fig. 1(b). Additionally, for CPW, it appears that enforcing rigidity has perturbed the effects of the pre-warping by a global homography. In contrast, APAP reduces gracefully to a global homography as the camera centres coincide, and provides the most accurate alignment as the translation increases.

5. Conclusion

We have proposed an as-projective-as-possible estimation method for 2D warping functions. The results on image stitching showed encouraging results, where our method was able to accurately align images that differ by more than a pure rotation. The experiments also demonstrated 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. This yields a highly accurate image stitching technique.

Acknowledgements

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] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *ACM SIGGRAPH*, 2004. 2

[2] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. *IEEE TVCG*, 9(1):3–15, 2003. 1, 3

[3] A. Eden, M. Uyttendaele, and R. Szeliski. Seamless image stitching of scenes with large motions and expoure differences. In *CVPR*, 2006. 2

[4] J. Gao, S. J. Kim, and M. S. Brown. Constructing image panoramas using dual-homography warping. In *CVPR*, 2011. 2, 5, 6

[5] G. H. Golub and C. F. van Loan. *Matrix computations*. The Johns Hopkins University Press, 3rd edition, 1996. 4

[6] G. Guennebaud and M. Gross. Algebraic point set surfaces. *ACM TOG*, 26(3), 2007. 2

[7] R. I. Hartley. In defense of the eight-point algorithm. *IEEE TPAMI*, 19(6):580–593, 1997. 3

[8] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. *ACM TOG*, 24(3), 2005. 2

[9] W.-Y. Lin, S. Liu, Y. Matsushita, T.-T. Ng, and L.-F. Cheong. Smoothly varying affine stitching. In *CVPR*, 2011. 2, 5, 6

[10] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Content-preserving warps for 3d video stabilization. In *ACM SIGGRAPH*, 2009. 2, 5

[11] A. Mynorenko, X. Song, and M. Carreira-Perpinan. Non-rigid point set registration: coherent point drift. In *NIPS*, 2007. 2

[12] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet. Mosaicing on adaptive manifolds. *IEEE TPAMI*, 22(10), 2000. 1

[13] P. Perez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM SIGGRAPH*, 2003. 2

[14] S. Schaefer, T. McPhail, and J. Warren. Image deformation using moving least squares. In *SIGGRAPH*, 2006. 1, 2, 3, 4

[15] H.-Y. Shum and R. Szeliski. Construction of panoramic mosaics with global & local alignment. *IJCV*, 36(2), 2000. 2

[16] P. Stange. On the efficient update of the singular value decomposition. In *App. Mathematics and Mechanics*, 2008. 4

[17] R. Szeliski. Image alignment and stitching. In *Handbook of Mathematical Models in Computer Vision*, pages 273–292. Springer, 2005. 1, 2, 4, 5

[18] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. In *ICM*, MM ’10, pages 1469–1472, New York, NY, USA, 2010. ACM. 5

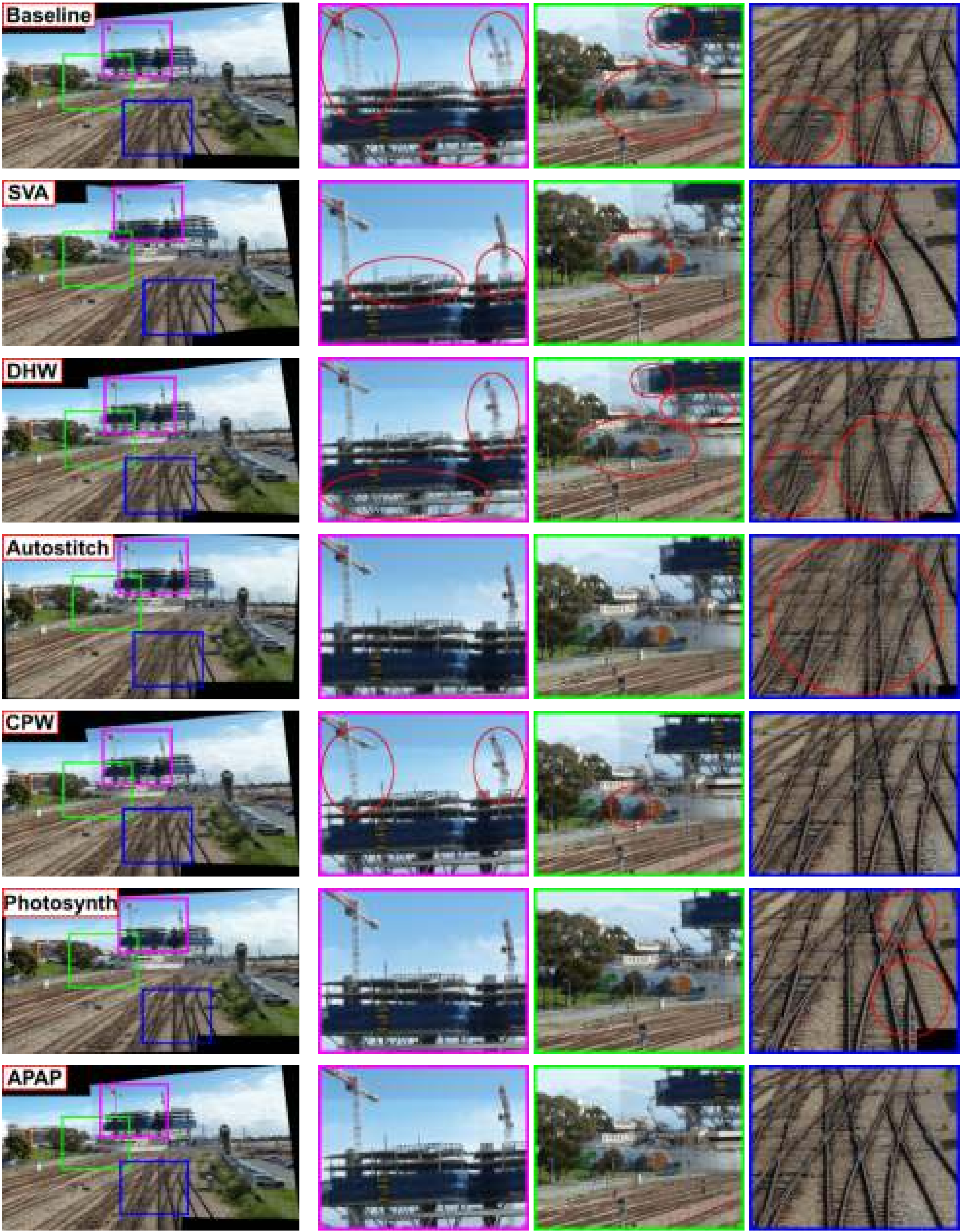


Figure 4. Qualitative comparisons (best viewed on screen) on the *railtracks* image pair. Red circles highlight errors. List of acronyms and initialisms: SVA-Smoothly Varying Affine, DHW-Dual Homography Warps, CPW-Content Preserving Warps, APAP-As Projective As Possible Warps.

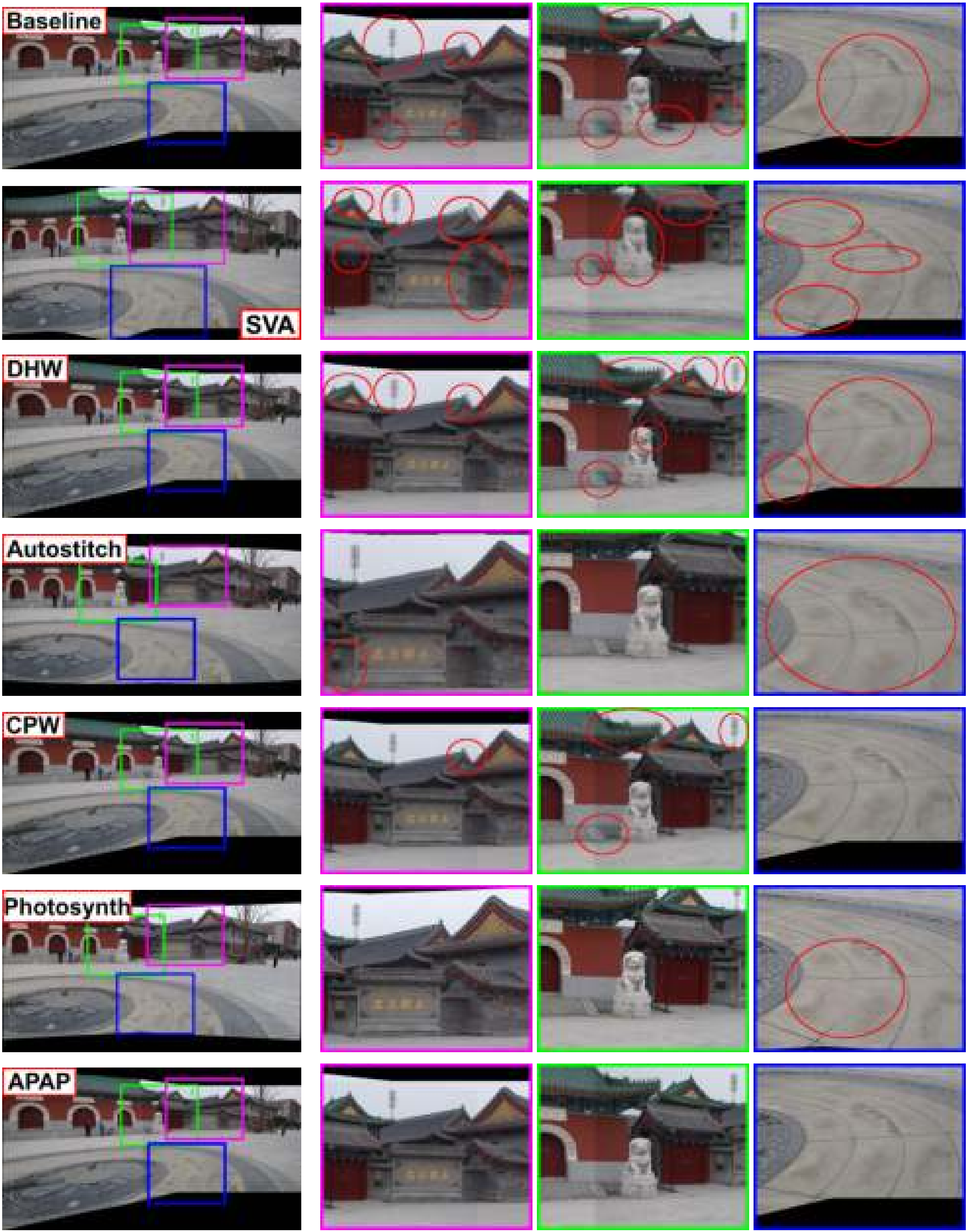


Figure 5. Qualitative comparisons (best viewed on screen) on the *temple* image pair. Red circles highlight errors. List of acronyms and initialisms: SVA-Smoothly Varying Affine, DHW-Dual Homography Warps, CPW-Content Preserving Warps, APAP-As Projective As Possible Warps.