VISUALIZATION OF LONG SCENES FROM DENSE IMAGE SEQUENCES USING PERSPECTIVE COMPOSITION

Siyuan Fang and Neill Campbell
University of Bristol, Bristol, U.K.

Keywords: Image Generation, Image based Rendering, Visualization.

Abstract: This paper presents a system for generating multi-perspective panoramas for long scenes from dense image sequences. Panoramas are created by combining different perspectives, including both original and novel perspectives. The latter are rendered using our perspective synthesis algorithm, which employs geometrical information to eliminate the sampling error distortion caused by depth parallax of non-planar scenes. Our approach for creating multi-perspective panoramas is different from existing methods in that a perspective composition framework is presented to combine various perspectives to form a panorama without undesired visual artifacts, through suppressing both colour inconsistencies and structural misalignments among input perspectives. We show that this perspective composition can facilitate the generation of panoramas from user specified multi-perspective configurations.

1 INTRODUCTION

A photograph can only capture a portion of long scenes, such as a street, since the field of view of a common camera is usually quite limited. With a single panorama combined from several different images, users are able to view scenes of interest simultaneously. More importantly, a panorama is an effective way of summarizing content of input images with much less redundant data.

Traditional panoramas are generated from images captured at a fixed viewpoint with pure rotational movement (Szeliski and Shum, 1997; Shum and Szeliski, 2000; Brown and Lowe, 2003). In this case, input images can be registered to a reference coordinate based on particular alignment models, of which the most general one is the homography. However, it is usually impossible to place the viewpoint far enough away to encompass the entire street, imagining that we wish to capture a long but narrow street. Obviously, to acquire more scenes, we have to change the viewpoint. Generating panoramas from images captured at different viewpoints is much more challenging, as in this case, the image registration cannot be parameterized by an uniform homography if scenes are not planar.

For non-planar scenes, registering and stitching images with different viewpoints may cause serious visual effects, such as ghost artifacts. To alleviate this problem, these images need to be properly combined, e.g., divide the overlapping area of multiple images into different segments, each of which is only rendered with a single image. The seam is optimized to go through areas that are at a low risk of producing unnatural visual artifacts. However, with only original input images (or perspectives), such an optimized seam would not exist. In addition, being able to view a scene from any arbitrary possible perspective offers a great flexibility in allowing users to depict what they expect to convey in the resultant panorama. This gives rise to the requirement for synthesizing novel perspectives from input images.

Our novel perspective synthesis algorithm is based on the well-known strip mosaic (Peleg et al., 2000; Zomet et al., 2003), which offers an excellent solution to synthesize novel views from dense images. However, since each strip extracted from the input image is rendered from a regular pinhole camera, the synthesized result usually exhibits a sampling error distortion, which is visually unacceptable. In our system, estimated 3D geometrical information is used to eliminate this kind of distortion.

The essence of generating multi-perspective panoramas is to properly combine different perspectives to make the result exhibit a natural appearance. In this paper, a perspective composition framework is presented to overcome visual effects brought by both colour (pixel value) discrepancies and structural

Fang S. and Campbell N..
VISUALIZATION OF LONG SCENES FROM DENSE IMAGE SEQUENCES USING PERSPECTIVE COMPOSITION.
DOI: 10.5220/0003848702270237
In *Proceedings of the International Conference on Computer Graphics Theory and Applications* (GRAPP-2012), pages 227-237
ISBN: 978-989-8565-02-0

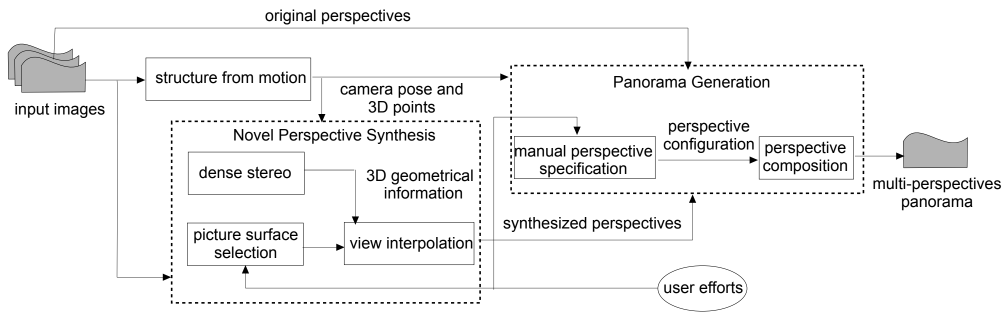


Figure 1: The system framework.

misalignments. The framework consists of two steps: firstly, parts of various perspectives are selected such that visual discontinuities among those parts can be minimized, and then, remaining artifacts are further suppressed through a fusion process.

An overview of our system is presented in Fig 1. In our system, street scenes are captured by a video camera (with a fixed intrinsic camera parameter **K**) moving along the scene to capture it looking sideways. The camera pose of each input image (i.e., the translation vector **T**, the rotation matrix **R** and **K**) is recovered using our Structure from Motion (SfM) system, together with a sparse set of reconstructed 3D scenes points. From recovered camera poses, novel perspectives are synthesized based on 3D geometrical information estimated using our dense stereo algorithm. An interface for manually specifying the multi-perspective configuration is provided based on our perspective composition framework, which combines different perspectives (original or novel) to form the resultant panorama.

The rest of this paper is organized as follows. Section 2 presents background. Section 3 presents our algorithm for synthesizing novel perspectives. Section 4 describes our perspective composition framework. Results and discussions are presented in Section 5 and Section 6 concludes this paper.

2 BACKGROUND

The earliest attempt at combining images captured at different viewpoints is perhaps view interpolation, which warps pixels from input images to a reference coordinate using a pre-computed 3D scene geometry (Szeliski and Kang, 1995; Kumar et al., 1995; Zheng and Kang, 2007). There are two main problems with these approaches: to establish an accurate correspondence for stereo is still a hard vision problem, and there will likely be holes in the resultant image due to sampling issues of the forward mapping and the occlusion problem. Another thread is based on optimal seam (Shum and Szeliski, 2000; Agarwala et al., 2006), which stitches input images with their own perspective and formulates the composition into a labeling problem, i.e., pixel values are chosen to be one of the input images. Results are inherently multi-perspective. However, these approaches only work well for roughly planar scene, as for scenes with large depth variations, it is often impossible to find an optimal partition that can create seamless mosaics.

The strip mosaic offers a better alternative. The basic idea is to cut a thin strip from a dense collection of images and put them together to form a panorama. In its early form, the push-broom model (Zheng, 2003; Peleg et al., 2000), the resultant image is parallel in one direction and perspective in the other, while the crossed-slits (Zomet et al., 2003) model is perspective in one direction and is perspective from a different viewpoint in the other direction. Therefore, the aspect ratio distortion is inherent due to the different projections along the two directions.

In addition, because scenes within each strip are rendered from a regular pinhole perspective, given a certain strip width, there is a depth at which scenes show no distortion. For a further depth, scenes might be duplicately rendered, i.e., over-sampled, while for a closer depth, scenes cannot be fully covered, i.e., under-sampled. In the literature, this kind of artifact is named a sampling error distortion (Zheng, 2003), see Fig 2.

Unlike the view interpolation and optimal seam, even for scenes with complex geometrical structures, strip mosaic can still produce visually acceptable results in spite of the fore-mentioned aspect ratio and sampling error distortions. Therefore, the strip mosaic provides a foundation upon which multi-perspective panoramas in a large scale can be constructed. An interactive approach is presented in (Roman et al., 2004), where several perspectives in the form of vertical slits are specified by users and gaps in-between them are filled with inverse perspectives. Some other approaches attempt to automatically de-

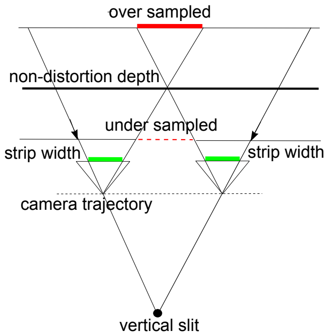


Figure 2: The sampling error distortion is caused by the depth parallax.

tect the multi-perspective configuration through minimizing metrics for measuring undesired effects, e.g., the colour discrepancy between consecutive strips (Wexler and Simakov, 2005) or the aspect ratio distortion (Roman and Lensch, 2006; Acha et al., 2008)

3 NOVEL PERSPECTIVE SYNTHESIS

3.1 Single Direction View Interpolation

The novel perspective is rendered onto a 3D picture surface, which is assumed to be perpendicular to the ground plane of scenes. A working coordinate system (WCS) is fitted from camera poses of input sequence to ensure that the ground plane is spanned by the X and Z axes, so that the picture surface can be simplified as a line in the top-down view of scenes, and extruded along the up (Y) axis. Then input images are rectified according to WCS.

The picture surface is defined by a 3D plane π_f and the X-Z plane of WCS is denoted as π_c. If scenes are exactly located on the picture surface, a point (or pixel) of the resultant image $\mathbf{p}' = [x', y']^\top$ can be mapped to a point $\mathbf{p} = [x, y]^\top$ of the i^{th} input image by a projective transformation, i.e., the homography:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{H}_i \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{R}_i \mid \mathbf{t}_i]\mathbf{G} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \quad (1)$$

where $\mathbf{G}$ is a 4×3 matrix that establishes the mapping between a 2D point of the resultant image and a 3D point $\mathbf{X}_p = [X_p, Y_p, Z_p]^\top$ on the picture surface, such that:

$$\begin{bmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{bmatrix} = \mathbf{G} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x\mathbf{V}_x & s_y\mathbf{V}_y & \mathbf{O} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \quad (2)$$

where $\mathbf{V}_x$ and $\mathbf{V}_y$ are vectors that parameterize X and Y axes of the plane coordinate of the picture surface and $\mathbf{O}$ is the origin of the plane coordinate. We choose $\mathbf{V}_x$ and $\mathbf{V}_y$ as projections of the X and Y axes of WCS onto the picture surface. s_x and s_y define the pixel size along the X and Y axes of the image coordinate. The choice of pixel size may affect the the rendering effect, and the strategy for defining a proper pixel size is presented in Section 3.2.

If scenes do not lie exactly on the picture surface, instead of using a uniform projective transformation, a point from the input image should be individually mapped onto the resultant image based on its actual 3D point $\mathbf{X}_d$. We assume that the (horizontal) projection center $\mathbf{C}_v$ of a novel perspective always lies on plane π_c and the vertical slit $\mathbf{L}$ is the line that passes through $\mathbf{C}_v$ and perpendicular to π_c, as shown in Fig 3. The mapping from a point $\mathbf{p} = [x, y]^\top$ in the i^{th} image onto the picture surface is the intersection of 3 planes: the picture surface π_f, the plane π_v that contains $\mathbf{X}_d$ and the vertical slit $\mathbf{L}$ and the plane π_h that contains $\mathbf{X}_d$ and the X axis of the i^{th} camera that is centered at $\mathbf{C}_i$, see Fig 3.

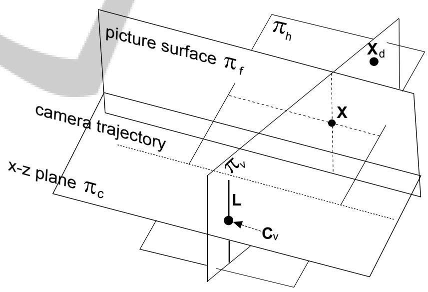


Figure 3: Points warping based on the 3D geometry.

Once the intersection is recovered, it is mapped to the resultant image using $\mathbf{G}^+$, the *pseudo-inverse* of $\mathbf{G}$. This approach can be further simplified, since the Y component of $\mathbf{p}'$, i.e., y', can be directly computed using the homography $\mathbf{H}_i$. The value of the X component x' depends on the actual 3D point. Suppose that the picture surface π_f intersects π_v at a 3D line, and $\mathbf{X}_s$ and $\mathbf{X}_t$ are two points on that 3D line, then we have:

$$\begin{bmatrix} ((G^+)^{2\top}X_s)((G^+)^{3\top}X_t) - ((G^+)^{2\top}X_t)((G^+)^{3\top}X_s) \\ ((G^+)^{3\top}X_s)((G^+)^{1\top}X_t) - ((G^+)^{3\top}X_t)((G^+)^{1\top}X_s) \\ ((G^+)^{1\top}X_s)((G^+)^{2\top}X_t) - ((G^+)^{1\top}X_t)((G^+)^{2\top}X_s) \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = 0 \quad (3)$$

where $(\mathbf{G}^+)^{k\top}$ denotes the k^{th} row of the matrix $\mathbf{G}^+$. With this equation, the value of x' can be solved from the known value of y'. Since with one direction the mapping adopts the original projective transformation, and the other is based on the real 3D geometry, this rendering strategy is named a “single direction view interpolation” as opposed to the full perspective interpolation.

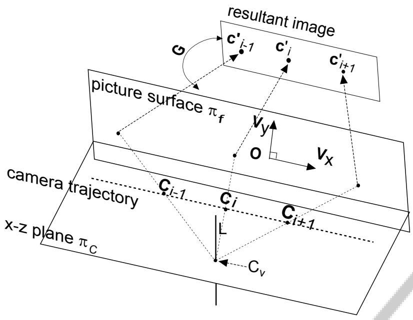


Figure 4: Rendering from a novel perspective. The projection center of the novel perspective is projected onto the picture surface and then mapped to the final resultant image.

The point mapping is followed by the determination of which input image is selected to render a point in the result. Such selection is inspired by the the strip mosaic. We project each camera center $\mathbf{C}_i$ onto a point in the resultant image $\mathbf{c}'_i$ along the line connecting $\mathbf{C}_i$ and the projection center of the novel perspective $\mathbf{C}_v$, see Fig4. We define a vertical center line $\mathbf{CL}_i$ that passes $\mathbf{c}'_i$ on the resultant image. A vertical split line $\mathbf{BL}_{i,i+1}$ is drawn between any consecutive camera center projections. The center line $\mathbf{CL}_i$ is then mapped to $\widehat{\mathbf{CL}}_i$ in the corresponding input image $\mathbf{I}_i$. We only examine pixels within a region around $\widehat{\mathbf{CL}}_i$. For each row of $\mathbf{I}_i$, we take the pixel on $\widehat{\mathbf{CL}}_i$ as the starting point and search on both sides. Once the warped point onto the result is beyond the split line $\mathbf{BL}_{i,i+1}$ or $\mathbf{BL}_{i-1,i}$, we proceed to the next row, see Fig 5.

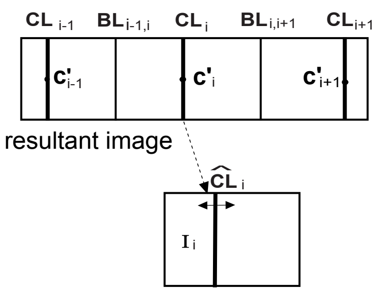
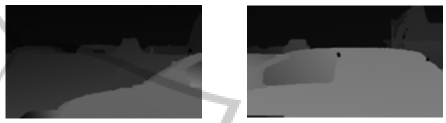


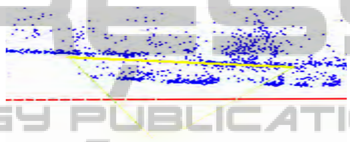
Figure 5: Center lines and split lines on the resultant image. Pixel warping is carried out within a region around the center line mapping.

Fig 6(d) shows the result synthesized using our single direction view interpolation. As compared to that without the interpolation shown in Fig 6(e), the sampling error distortion is removed. However, the aspect ratio distortion still exists. For example, in Fig 6(d), the car in front of the middle low wall is apparently squashed.

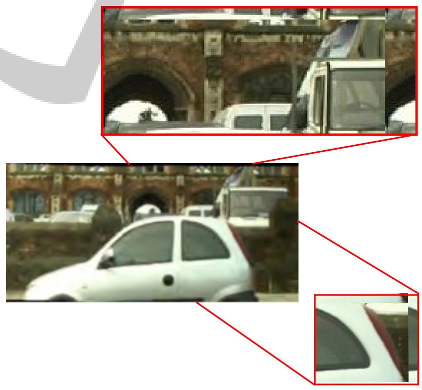
(a) Input images.



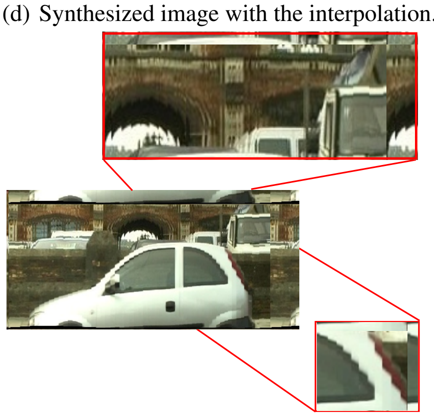
(b) Depth maps.



(c) The novel perspective configuration.



(d) Synthesized image with the interpolation.



(e) Synthesized image without the interpolation.

Figure 6: A result of the novel perspective synthesis.

3.2 Rendering from Point Samples

Each projected point from an input image provides a sample, and to render the resultant image is equivalent to reconstructing a continuous function from these scattered samples. This is done by convolving samples with a Gaussian filter. Then, a question naturally arises as to what size of the pixel of the resultant image is. The pixel size of the input image along the X (horizontal) direction is known, i.e., s'_x. We compute the average distance d_{ave} of the picture surface deviating from the camera trajectory. For simplicity, we assume that the projection of the picture surface is a straight line segment. In this case, the pixel size is defined as:

$$s_x = \frac{d_{ave}}{f} s'_x \tag{4}$$

where f is the focal length. The aspect ratio is chosen as that of the input image, so the pixel size along the Y (vertical) direction is: $s_y = \alpha s_x$.

Unlike this uniform sampling strategy, another choice is to use a non-uniform sampling, where the pixel size varies according to the distance of the picture surface deviating from the camera trajectory. Given a point $\mathbf{p}$ on the picture surface, suppose that the corresponding distance deviating from the camera trajectory is represented by a function $d(\mathbf{p})$, then the pixel size at $\mathbf{p}$ is written as:

$$s_x = \frac{d(\mathbf{p})}{f} s'_x \tag{5}$$

The result shown in Fig 6 is rendered using the non-uniform sampling strategy. Fig 7 compares results rendered using uniform and non-uniform sampling strategy.

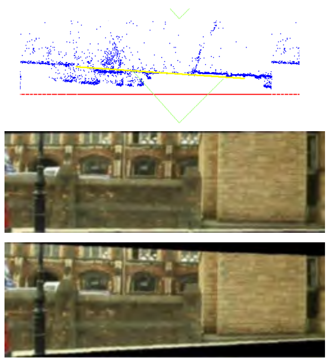


Figure 7: Results of the non-uniform and uniform sampling strategy. The result of the non-uniform sampling (middle), and that of uniform sampling (bottom). The uniform sampling strategy is aware of the shape of the picture surface, while the result of the non-uniform sampling strategy more agrees with human perception.

3.3 Dense Stereo

To estimate the depth (or, 3D geometry) map for each point (pixel) in an image $\mathbf{I}_i$, a stereo process is performed to $\mathbf{I}_i$ and its neighboring image $\mathbf{I}_{i+1}$. The stereo is accomplished in two steps: firstly, a correspondence between $\mathbf{I}_i$ and $\mathbf{I}_{i+1}$ is detected, and then the depth map is computed from the correspondence together with camera poses of $\mathbf{I}_i$ and $\mathbf{I}_{i+1}$.

To construct the correspondence, we adopt the concept of the surface correspondence as suggested in (Birchfield and Tomasi, 1999). A surface can be parameterized by the motion of its projections on $\mathbf{I}_i$ and $\mathbf{I}_{i+1}$, such that: $\mathbf{p} + S(\mathbf{p}) = \mathbf{p}'$. In this sense, the correspondence detection is converted to determining for each point $\mathbf{p}$ in $\mathbf{I}_i$ which surface it should belong to, and to calculating the motion parameter of that surface. Since the stereo pair is assumed to be rectified and the vertical movement is ignorable, the surface is represented by a 1D affine model:

$$S(\mathbf{p}) = \begin{bmatrix} a_1 * x + a_2 * y + b \\ 0 \end{bmatrix} \tag{6}$$

We adopt a similar framework to that proposed in (Birchfield and Tomasi, 1999). The basic idea is to iteratively refine the estimation by alternating between two steps:

1. Given a labeled map of each point, we need to find the affine motion parameter for each connected segment. This is done by minimizing the cost function $\sum_{p \in \Omega} (\mathbf{I}_i(\mathbf{p}) - \mathbf{I}_{i+1}(\mathbf{p} + S(\mathbf{p})))^2$, where Ω denotes the set of all points in a segment. This cost function is minimized using the the iterative method proposed in (Shi and Tomasi, 1994).

2. Given a set of surfaces characterized by their affine motion parameters, each pixel is labeled as belonging to one surface. The problem is solved by a Markov Random Field (MRF) optimization implemented using the Graph Cut algorithm (Kolmogorov and Zabih, 2004). The cost function of the MRF consists of a data term that computes the cost for a pixel $\mathbf{p}$ to be assigned with a surface $S(\mathbf{p})$, and a smooth term penalizing a pixel $\mathbf{p}$ and its neighboring point $\mathbf{q}$ for having different surface labels.

Fig 6(b) presents an example result of our dense stereo algorithm.

4 PERSPECTIVE COMPOSITION

Our perspective composition framework consists of two steps. Firstly, a decision must be made for each pixel of the resultant panorama as to which perspective should be adopted. Significant pixel value differences, and complex structures together with large geometrical misalignments could make this labeling process challenging and hence a sophisticated cost function is proposed to take into account these vital factors. Visible discontinuities might still exist, which mainly occur along lines bordering adjacent segments rendered from different perspectives. Therefore, in the second step, we suppress such discontinuities through blending information along these boundary lines.

4.1 Perspective Selection

Given n perspectives: $\{\mathbf{V}_i\}_0^{n-1}$, for each point $\mathbf{p}$ of the resultant panorama, the perspectives selection is represented by a labeling function: $L(\mathbf{p}) = i$. Labels of all points constitute a labeling configuration: $\mathbf{L}$, the cost of which is formulated as a Markov Random Field (MRF):

$$E(\mathbf{L}) = \sum_{\mathbf{p}} E_D(\mathbf{p}, L(\mathbf{p})) + \sum_{\mathbf{p}} \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} E_S(\mathbf{p}, \mathbf{q}, L(\mathbf{p}), L(\mathbf{q})) \tag{7}$$

E_S denotes the smooth term and E_D denotes the data term.

The smooth term E_S consists of three terms: a depth term, a colour term and a structure term, and the measuring function is a weighted sum of these three terms:

$$E_S(\mathbf{p}, \mathbf{q}, L(\mathbf{p}), L(\mathbf{q})) = E_{d\_S} + \mu_0 E_{c\_S} + \mu_1 E_{g\_S} \tag{8}$$

μ_0 and μ_1 are weights.

Depth Smooth Term: The depth smooth term encourages the seam to go through regions where 3D geometry coincides with the picture surface. We calculate for each pixel of a perspective the residual error with respect to the picture surface. Each point mapped from the input image onto the synthesized image of a perspective constitutes a point sample. Suppose that the point sample is extracted from the input image $\mathbf{I}_i$, and let us denote the depth under the camera coordinate of $\mathbf{I}_i$ as $d_{\mathbf{I}_i}(\mathbf{x})$. Then, we calculate the depth of the corresponding point projected onto the picture surface, denoted as $d_{\mathbf{I}_i}(\mathbf{x}')$. The residual error of the point sample is:

$$r_{\mathbf{I}_i}(\mathbf{x}') = \begin{cases} 1.0 - \frac{d_{\mathbf{I}_i}(\mathbf{x})}{d_{\mathbf{I}_i}(\mathbf{x}')} & d_{\mathbf{I}_i}(\mathbf{x}') \geq d_{\mathbf{I}_i}(\mathbf{x}) \\ 1.0 - \frac{d_{\mathbf{I}_i}(\mathbf{x}')}{d_{\mathbf{I}_i}(\mathbf{x})} & d_{\mathbf{I}_i}(\mathbf{x}') < d_{\mathbf{I}_i}(\mathbf{x}) \end{cases} \tag{9}$$

The residual error of a grid point $r_{\mathbf{V}_i}(\mathbf{p})$ of the i^{th} perspective is computed by convolving these samples with the Gaussian filter. Given a pair of neighboring pixels $\mathbf{p}$ and $\mathbf{q}$ of the resultant panorama, with assigned labels as $L(\mathbf{p})$ and $L(\mathbf{q})$, the depth smooth term is:

$$\begin{aligned} E_{d\_S} = \; & r_{\mathbf{V}_{L(\mathbf{p})}}(\mathbf{p}) + r_{\mathbf{V}_{L(\mathbf{q})}}(\mathbf{p} \\ & + r_{\mathbf{V}_{L(\mathbf{p})}}(\mathbf{q}) + r_{\mathbf{V}_{L(\mathbf{q})}}(\mathbf{q}) \end{aligned} \tag{10}$$

Colour Smooth Term: To place the seam in regions where pixel values from different perspectives are similar, the colour smooth term is defined as:

$$\begin{aligned} E_{c\_S} = \; & \frac{1}{N} \sum_{\mathbf{x} \in \mathbf{W}} \mid V_{L(\mathbf{p})}(\mathbf{p}+\mathbf{x}) - V_{L(\mathbf{q})}(\mathbf{p}+\mathbf{x}) \mid \\ & + \frac{1}{N} \sum_{\mathbf{x} \in \mathbf{W}} \mid V_{L(\mathbf{p})}(\mathbf{q}+\mathbf{x}) - V_{L(\mathbf{q})}(\mathbf{q}+\mathbf{x}) \mid \end{aligned} \tag{11}$$

where $\mathbf{W}$ is a window for the aggregation of difference, and N is the size of $\mathbf{W}$.

Structural Smooth Term: To suppress structural discontinuities, we define the structural smooth term as (assuming the gradient $\nabla.$ captures the most structural information of an image):

$$\begin{aligned} E_{g\_S} = \; & \mid \nabla \mathbf{V}_{L(\mathbf{p})}(\mathbf{p}) - \nabla \mathbf{V}_{L(\mathbf{q})}(\mathbf{p}) \mid \\ & + \mid \nabla \mathbf{V}_{L(\mathbf{p})}(\mathbf{q}) - \nabla \mathbf{V}_{L(\mathbf{q})}(\mathbf{q}) \mid \end{aligned} \tag{12}$$

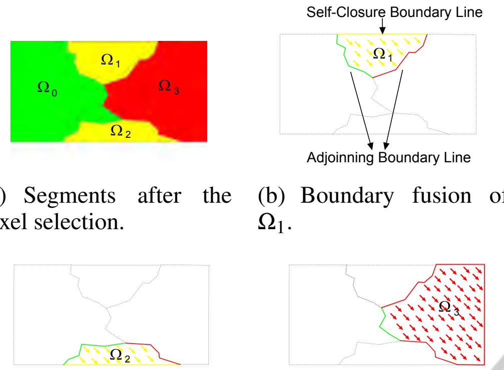
The Data Term: A general form of the data term is written as:

$$E_D(\mathbf{p}, L(\mathbf{p})) = \begin{cases} U_{L(\mathbf{p})}(\mathbf{p}) & \mathbf{p} \in \mathbf{V}_{L(\mathbf{p})} \\ \infty & \mathbf{p} \notin \mathbf{V}_{L(\mathbf{p})} \end{cases} \tag{13}$$

$U_{L(\mathbf{p})}(\mathbf{p})$ measures the fitness of a pixel to be assigned with the label. We adopt a simple solution, i.e., use a uniform function such that: $U_{L(\mathbf{p})}(\mathbf{p}) \equiv 0$.

4.2 Perspective Fusion

Discontinuities along the seam (or boundary line) between two segments rendered from different perspectives need to be eliminated to produce a natural transition from one perspective to another, whilst, in the meantime, we need to preserve original appearance as much as possible. Here, we adopt a local method, i.e., information of one segment along the boundary line is blended into the interior of the other one. Such information could simply be pixel values, and warping vectors compensating for structural misalignments.



(a) Segments after the pixel selection.

(b) Boundary fusion of Ω_1.

(c) Boundary fusion of Ω_2.

(d) Boundary fusion of Ω_3.

Figure 8: The boundary fusion in a monotonic order. The order of involved perspectives is: $L(0) < L(3) < L(1) = L(2)$.

4.2.1 The Fusion Paradigm

Segments of each input perspective are extracted from the result of the perspective selection. Let us assume that there are m such segments, and for each segment $k \in \{0,...,m-1\}$, the mapping function $L(k)$ denotes the index of the corresponding perspective i, $i \in \{0,...,n-1\}$. A segment Ω_k is enclosed by its boundary lines $\partial\Omega_k$. There are two kinds of boundary lines: 1) adjoining boundary lines that are adjacent to other segments, and 2) self-closure boundary lines that do not border with any other segments, see Fig 8. As for the former, information from other segments need to be blended into the interior part of Ω_k, whilst, for the latter, we only consider information from Ω_k itself.

We propose a monotonic fusion paradigm that only performs the boundary fusion in a single direction. Let us suppose that two neighboring segments Ω_j and Ω_k share a adjoining boundary line. Without losing generality, it is assumed that $L(j) < L(k)$. Information from Ω_j along the boundary line are blended into the interior part of Ω_k, while, Ω_j remains unchanged. Fig 8 illustrates this monotonic fusion.

A proper order of input perspectives is needed. We compute for each perspective the number of its neighboring perspectives $N(i)$, and then sort all input perspectives with an ascending order of $N(i)$. If two perspectives have the same value of $N(i)$, then the one with more pixel number is placed before the other.

4.2.2 Blending of Pixel Values

To smoothly propagate pixel values of boundary lines, the blending is constrained by the gradient field of Ω_k as suggested in (Pérez et al., 2003). Let the function $f(\cdot)$ be the original pixel value at a certain pixel and $f'(\cdot)$ denote the pixel value to be speculated in Ω_k. In addition, the function $f^{\star}(\cdot)$ denotes the pixel value at the boundary line. The pixel value blending is casted into a minimization problem, such that:

$$\min_{f'} \iint_{\mathbf{p}\in\Omega_k} |\nabla f'(\mathbf{p}) - \nabla f(\mathbf{p})|^2 \quad \textbf{with } f'(\mathbf{p}) = f^{\star}(\mathbf{p}) \textbf{ for } \mathbf{p} \in \partial\Omega_k \tag{14}$$

According to the Euler-Lagrange theorem for quadratic functions, such minimization can be converted to a group of equations: $\Delta f'(\mathbf{p}) = div f(\mathbf{p})$ for all $\mathbf{p} \in \Omega_k$ with $f'(\mathbf{p}) = f^{\star}(\mathbf{p})$ for $\mathbf{p} \in \partial\Omega_k$, where $\Delta.$ denotes the second-order derivative and $div f(\cdot)$ is the divergence. The second-order derivative can be discretized using the Laplacian operator, yielding a linear equation system, which is solved by our implementation of the multi-grid V-cycle algorithm specialized for irregular shaped segments.

4.2.3 Blending of Image Warping

Misalignments can be roughly grouped into two categories: a small structural misalignment, which is usually brought by breaking edges along the adjoining boundary line, and a large structural misalignment, which is mainly caused by significant geometrical misalignments. Fig 9(a) and 10(a) presents real examples of these two types of misalignments. We introduce two corresponding algorithms based on image warping. Our system enables these two optional algorithms to be selected by users for a given adjoining boundary line.

Structure Re-alignment. It is a common practice to compensate for small structural misalignments through locally re-aligning deviated edges, e.g., (Fang and Hart, 2004; Jia and Tang, 2008). In the following, this strategy is termed as "structure re-alignment".

For each input perspective, we detect salient edges with sufficiently large magnitudes. Given two neighboring segments Ω_j and Ω_k, it is assumed that $L(j) < L(k)$, and hence the boundary information from Ω_j is blended into Ω_k. We match edges striding over the adjoining boundary line bordering these two segments. We enforce a one-to-one mapping that minimizes the sum of edge difference measuring three factors: the edge direction similarity, geometrical distance and pixel value similarity. For a pair of matched edges, we calculate the backward warping vector. Then we interpolate warping vectors for those non-edge points on the adjoining boundary line between Ω_j and Ω_k. Warping vectors are blended into the interior of Ω_k

(a) A small structural misalignment. (b) After the structure re-alignment.

Figure 9: A result of the structure re-alignment. In (a), edges are broken by the seam. After the structure re-alignment (b), edges are correctly aligned, and thus the structural discontinuity is eliminated.

(a) A large structural misalignment. (b) After the segment shift.

Figure 10: A result of the segment shift. Due to the geometrical misalignment, a duplicate gutter is shown in (a), after the segment shift, the ghost gutter is eliminated.

using the Poisson image editing (Pérez et al., 2003): $\min_{\mathbf{z}'} \iint_{\mathbf{p}\in\Omega_k} |\nabla z'(\mathbf{p})|^2$ and for $\mathbf{p} \in \partial\Omega_k : z'(\mathbf{p}) = z^\star(\mathbf{p})$, where $z'(\mathbf{p})$ denotes the warping vector to be speculated in Ω_k and $\mathbf{z}^\star(\mathbf{p})$ is warping vector from the boundary line.

Once the warping vector is calculated for each pixel in Ω_k, its pixel value of the warped image is determined using bilinear interpolation. If there still exists a large pixel value difference, we warp the gradient field of Ω_k, and then, pixel value blending presented in the previous section is applied to the warped gradient field.

Segment Shift. To fix the large structural misalignment caused by the geometrical misalignment, the image warping is based on a robust match between corresponding perspectives of the two adjacent segments. The matching process is constrained by the geometrical information. Firstly, a match region $\mathbf{R}$ is placed to enclose the adjoining boundary line in $\mathbf{V}_{L(k)}$, and for each pixel $\mathbf{p}$ in $\mathbf{R}$, its corresponding depth information is calculated from point samples for constructing the perspective $\mathbf{V}_{L(k)}$ through convolving such samples with a Gaussian filter, and then it is re-projected onto the perspective $\mathbf{V}_{L(j)}$ as $\mathbf{p}'$. The similarity between $\mathbf{p}$ and $\mathbf{p}'$ is measured. The measurement is applied to a patch centered at $\mathbf{p}$ (and $\mathbf{p}'$) to allow for an offset. The one with the highest similarity degree is chosen as the measuring result, and if it is above certain threshold, then these two pixels are regarded as a correct match. A robust measuring function could be the Normalized Cross-Correlation (NCC) performed over a small window around a pixel. We also integrate into our system the SIFT feature (Lowe, 2004), which is reliable but sometimes too sparse.

Two matched pixels provide a backward warping vector. For those pixels in $\mathbf{R}$ without robust matches, their warping vectors are estimated using known data. In our system, we integrate two methods. If pixels with known warping vectors are dense, a convolution with a Gaussian filter is applied. On the other hand, if such pixels are sparse (e.g., extracted from SIFT feature matching), a Radial Basis Function (RBF)-based interpolation with thin-plate spline kernal (Wahba, 1990) is adopted.

Now, a warping vector is associated with each pixel on the adjoining boundary between Ω_j and Ω_k. These warping vectors are then blended into the interior of Ω_k using the poisson image editing as described above. This strategy often induces a large image deformation, which would cause parts of one segments shifted towards the other, and therefore we name it as "segment shift".

5 RESULTS AND DISCUSSION

Experiments have been conducted on real urban scenes. We demonstrate in Fig 11 how the perspective composition can be used to create a multi-perspective panorama from a manually specified perspective configuration. The perspective configuration is shown in Fig 11(d), which is mixed with both novel and original perspectives. The original perspective is rendered through mapping the corresponding input image onto the picture surface using a projective transform as defined in 1. The novel perspective is synthesized using our single direction view interpolation. Fig 11(a) shows the result of the MRF optimization for the perspective selection, where adjoining boundary lines (seams) are highlighted. Fig 11(b) shows the map of residual error with respect to the picture surface, from which one can see that seams produced by the MRF optimization are roughly placed in areas with low residual errors. Fig 11(c) presents the final panorama after the boundary fusion. More results are

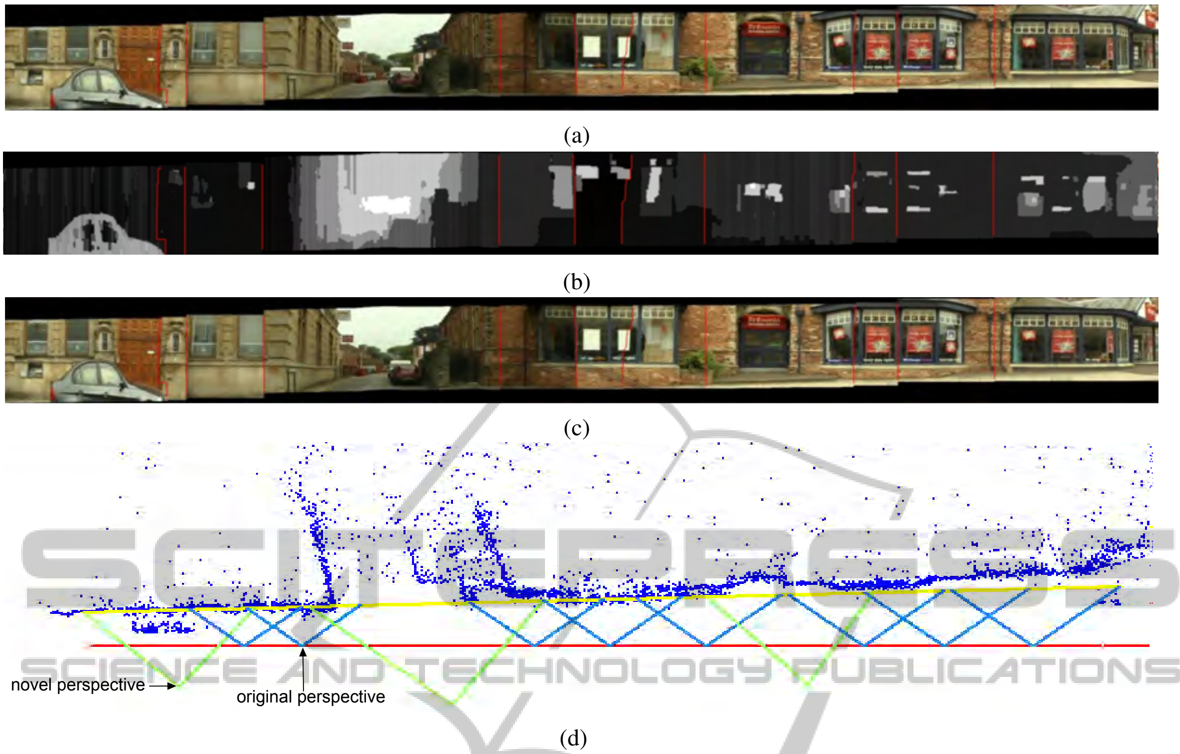


Figure 11: A Panorama created from the perspective composition. In the multi-perspective configuration (d), original perspectives are denoted as blue and novel ones are denoted as green. The result of MRF optimization is shown in (a) and the composed residual error map is shown in (b). The final panorama after boundary fusion is shown in (c).

shown in Fig 12. (Parts of fusion results are already shown in Fig 9(b) and 10(b)).

We are not the first to generate multi-perspective panoramas through perspective composition. The approach presented in (Agarwala et al., 2006) takes original perspectives as input, and use the MRF optimization to select a perspective for rendering each pixel in the resultant panorama. This approach works quite well for mainly planar scenes. However, due to the lack of facility to synthesize novel perspectives that are wide enough to cover scenes not on the main plane (picture surface), a seam placed at the area corresponding to the off-plane scenes would induce serious visual artifacts. Fig 13 presents an example result of this approach, which is visually unacceptable [1].

There are several existing approaches that address the problem using synthesized novel perspectives. However, they assume that input perspectives are precisely registered with each other, and therefore no further composition processing is required in their system. For example, the interactive approach described in (Roman et al., 2004) only allows a set of disjoint perspectives to be specified, and these disjoint perspectives are simply connected by a set of inverse perspectives in between them. Obviously, their approach restricts content that can be conveyed in the resultant panorama, e.g., the perspective configuration as presented in Fig 11(d) can never be achieved with their approach.

[1] Actually, they use a fish-eye camera to expand the filed of view (FOV) of input images. However, the FOV of an image is still limited.

6 CONCLUSIONS

This paper presents a system for producing multi-perspective panoramas from dense video sequences. Our system uses estimated 3D geometrical information to eliminate the sampling error distortion in the synthesized novel perspective. Then a perspective composition framework is presented to combine different perspectives by suppressing their pixel value and structural discrepancy. Compared to the existing methods, the perspective composition not only removes noticeable artifacts, but also relieves some constraints imposed on the perspective configuration that a resultant panorama can be properly generated from.

The main problem of our approach is the aspect ratio distortion associated with the synthesized novel

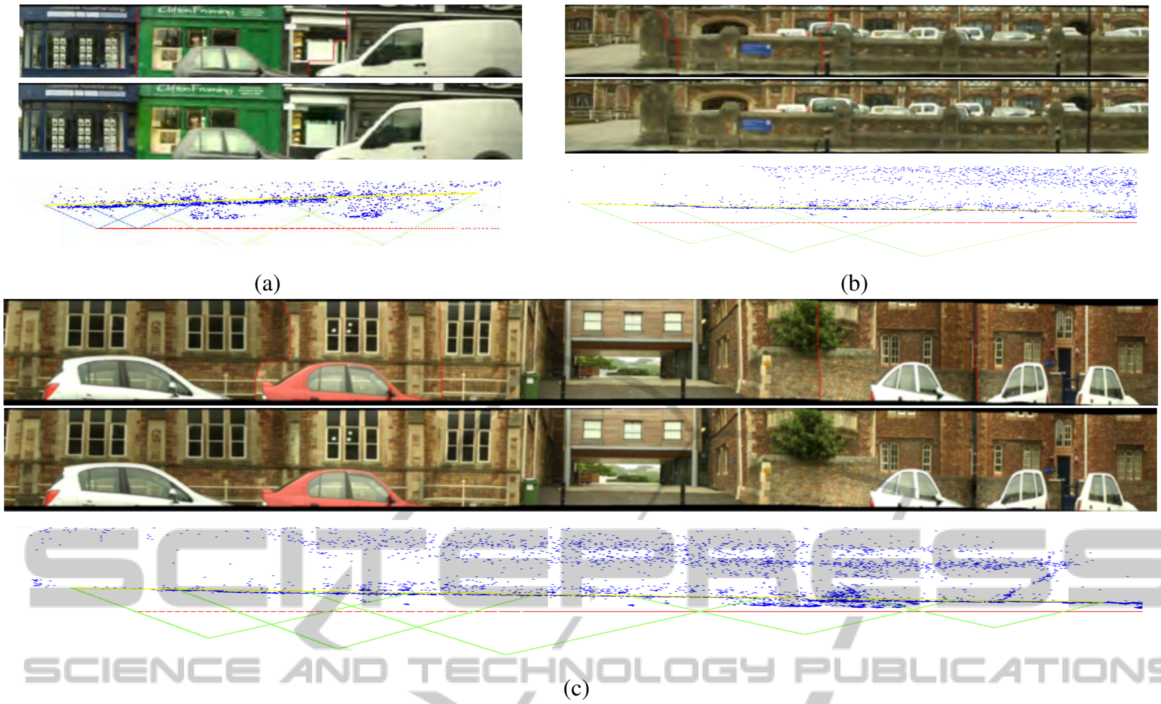


Figure 12: Multi-perspective panoramas of urban scenes. For each result, the outcome of the MRF optimization (top) is blended using our boundary fusion algorithm (middle).

Figure 13: A Panorama created from pure original perspectives (Agarwala et al., 2006). Input image sequence is resampled to get a set of sparse original perspectives, which are combined using the MRF optimization. For scenes off picture surface, visual artifacts are noticeable.

perspective. The cause of this problem lies in the fact that we are lack of information along the direction perpendicular to the direction of the camera movement. In the future, we shall look into the use of an array of cameras mounted on a pole to collect enough information along the direction perpendicular to the camera movement. Another interesting extension is to introduce into our system some kinds of interactive viewing facility, so that users can choose to view scenes of interest at a high resolution or from a particular perspective such as the Street Slide system (Kopf et al., 2010).

REFERENCES

Acha, A., Eagel, G., and Peleg, S. (2008). Minimal Aspect Distortion (MAD) mosaicing of long scenes. *International Journal of Computer Vision*, 78(2-3):187–206.

Agarwala, A., Agrawala, M., Cohen, M., Salesin, D., and Szeliski, R. (2006). Photographing long scenes with multi-viewpoint panoramas. *ACM Transactions on Graphics*, 25(3):853 – 861.

Birchfield, S. and Tomasi, C. (1999). Multiway cut for stereo and motion with slanted surfaces. In *Proceedings of the International Conference on Computer Vision*, pages 489–495.

Brown, M. and Lowe, D. G. (2003). Recognising panoramas. In *Proceedings of IEEE International Conference on Computer Vision*, volume 2, pages 1218–1225.

Fang, H. and Hart, J. C. (2004). Textureshop: texture synthesis as a photograph editing tool. In *Proceedings of SIGGRAPH*, volume 23, pages 354–359.

Jia, J. and Tang, C.-K. (2008). Image stitching using structure deformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):617–631.

Kolmogorov, V. and Zabih, R. (2004). What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159.

Kopf, J., Chen, B., Szeliski, R., and Cohen, M. (2010). Street slide: Browsing street level imagery. *ACM Transactions on Graphics*, 29(4):96:1–96:8.

Kumar, R., Anandan, P., Irani, M., Bergen, J., and Hanna, K. (1995). Representation of scenes from collections of images. In *Proceedings of IEEE Workshop on Representation of Visual Scenes*, pages 10–17.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Peleg, S., Rousso, B., Rav-Acha, A., and Zomet, A. (2000). Mosaicing on adaptive manifold. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1144–1154.

Pérez, P., Gangnet, M., and Blake, A. (2003). Poisson image editing. In *Proceedings of SIGGRAPH*, volume 22, pages 313–318.

Roman, A., Garg, G., and Levoy, M. (2004). Interactive design of multi-perspective images for visualizing urban landscapes. In *Proceedings of IEEE Visualization*, pages 537–544.

Roman, A. and Lensch, H. (2006). Automatic multiperspective images. In *Proceedings of Eurographics Symposium on Rendering*, pages 161–171.

Shi, J. and Tomasi, C. (1994). Good features to track. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600.

Shum, H. and Szeliski, R. (2000). Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101–130.

Szeliski, R. and Kang, S. (1995). Direct methods for visual scene reconstruction. In *Proceedings of IEEE Workshop on Representation of Visual Scenes*, pages 26–33.

Szeliski, R. and Shum, H. (1997). Creating full view panoramic image mosaics and environment maps. *Journal of Computer Graphics*, 31:251–258.

Wahba, G. (1990). *Spline Models for Observational Data*. SIAM.

Wexler, Y. and Simakov, D. (2005). Space-time scene manifolds. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 858 – 863.

Zheng, J. (2003). Digital route panoramas. *IEEE Multimedia*, 10(3):57– 67.

Zheng, K. C. and Kang, S. B. (2007). Layered depth panoramas. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Zomet, A., Feldman, D., Peleg, S., and Weinshall, D. (2003). Mosaicing new views: The crossed-slits projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):741– 754.