REAL-TIME WALKTHROUGH OF OUTDOOR SCENES USING TRI-VIEW MORPHING

Qianqian Li, Yu Zhou, Yao Yu, Sidan Du, Ziqiang Wang

Nanjing University, 210023, China

ABSTRACT

In this paper, an image-based walkthrough system is presented for navigating real-world outdoor scenes based on only three uncalibrated images without reconstructing 3D model. An image-based rendering operation is performed on these sample images and generates photorealistic in-between views in real time. We extend the traditional two-step view morphing to real-time tri-view morphing based on epipolar constraint. Compared with other tri-view morphing methods, our scheme copes well with both complex outdoor scenes and wide baseline scenes, which also poses a qualified alternative to state-of-the-art dense 3D reconstruction. Putting on a head mount display (HMD) like Oculus Rift, user can experience realistic and immersive exploration of real-world scenes.

***Index Terms*—** Imaged-based rendering, Tri-view morphing, Virtual walkthrough, Real-time display, Oculus Rift

(a) Demo (b) A synthetic view

Fig. 1. Navigating real-word scenes wearing Oculus Rift.

1. INTRODUCTION

Image-based rendering (IBR) techniques have recently received much attention as a powerful tool for synthesizing novel views and realizing virtual walkthrough. Instead of geometric primitives, IBR addresses this problem by capturing a collection of images of the scene and then creating novel views by re-sampling the images. Shum *et al.* [1] classified various IBR techniques into three categories, namely rendering with no geometry, rendering with implicit geometry and rendering with explicit geometry.

Thanks to Grant No.BE2015152 from the Natural Science Foundation of Jiangsu Province and Grant No.61100111, 61300157, 61201425, 61271231 from the National Natural Science Foundation of China for funding.

In this paper, we demonstrate a new tri-view morphing method to reconstruct an interactive exploration of outdoor scenes. The contributions of this paper are mainly three folds. Firstly, we optimize the traditional two-step view morphing to tri-view morphing, which greatly reduces the computation time and achieves an interactive rate. Secondly, unlike previous methods, our system simply needs a small number of images and can generate photorealistic walk-through experiences. Finally, the rendering procedure can be explored in real time regardless of the complexity of sample images.

2. RELATED WORK

Previous work on IBR reveals a continuum of image-based representations based on the tradeoff between how many input images are needed and how much is known about the scene geometry. For example, given the internal and external parameters as well as the depth of a scene point (with respect to the camera), it is easy to obtain the correspondences between images and calculate the synthetic view [2–5].

Where no knowledge on the imaging device can be assumed, Light Field [6, 7] and Lumigraph [8, 9] work. However, Light Field has a tendency to rely on over-sampling to counter undesirable aliasing effects in output display. Lumigraph is similar to light field rendering but it applies approximated geometry to compensate for non-uniform sampling in order to improve rendering performance.

Uncalibrated point transfer techniques utilize image-to-image constrains to re-project pixels from the sample images to the synthetic image. Seitz and Dyer [10] calculated arbitrary view from two sample images. Manning and Dyer [11] extended Seitz's static view morphing to dynamic view morphing. Gurdan *et al.* [12] proposed a framework for two-image interpolation in space and time. However, user could only move along a straight line because there are only two sample images. Xiao *et al.* [13] relaxed this constraint and allowed an arbitrary motion. They extended this to three sample images and obtained photo-realistic results. Chatkaewmanee [14] extended Xiao's scheme to work with wide baseline imagery, but it can neither run in real time nor deal with outdoor scenes.

So far, no other work on IBR could simultaneously cope with casual camera setups and preserve image quality in the

interpolated images. Moreover, they are not able to navigate at an interactive rate due to expensive computations. Our framework is designed to deal with all these problems in a simple and efficient way.

3. VIRTUAL WALKTHROUGH SYSTEM BASED ON TRI-VIEW MORPHING

3.1. Algorithm Overview

Suppose we are given a stack of n images $\{I_k\}_{k=1}^{n}$ where each image $I_k = \{p\}$ (p indicates the point in the image) captures the same real-world scene at different positions. Every image is represented with a point and we use the Delaunay method [15] to triangulate these points. A triple of images are the minimum unit for our algorithm. Fig. 2 presents the architecture of our proposed algorithm, which is implemented as following steps : *pre-warping*, *morphing* and *post-warping*.

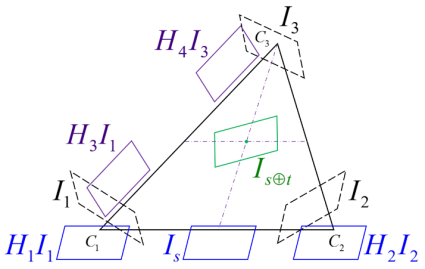


Fig. 2. Architecture of our algorithm.

3.2. Pre-warping and Morphing

As Seitz [10] declared, morphing is a *shape-distorting* transformation, and a particularly disturbing effect of image morphing is its tendency to bend straight lines. Some previous works ignored this process or required user assistance to help [10, 16, 17]. We form a new method to simultaneously rectify three images with no user assistance, geometric knowledge of the scene or calibration information of the camera.

3.2.1. Two-step View Morphing

In the first step, three sample images I_1, I_2 and I_3 are divided into two image paires I_1, I_2 and I_1, I_3 (regard I_1 as reference image). Then the normal rectifying algorithm [18] is employed on each pair to figure out the projective transformation pairs $\mathbf{H}_1$, $\mathbf{H}_2$ and $\mathbf{H}_3$, $\mathbf{H}_4$. Finally, two projective transformation pairs are applied to sample image pairs respectively to form parallel view pairs $\mathbf{H}_1I_1$, $\mathbf{H}_2I_2$ and $\mathbf{H}_3I_1$, $\mathbf{H}_4I_3$.

After forming parallel view pairs, whose corresponding points are in the same scanline, three dense mapping matrices $\mathbf{D}_{1\to2} : \mathbf{H}_1I_1 \Rightarrow \mathbf{H}_2I_2$, $\mathbf{D}_{1\to3} : \mathbf{H}_3I_1 \Rightarrow \mathbf{H}_4I_3$ and $\mathbf{D}_{3\to1} : \mathbf{H}_4I_3 \Rightarrow \mathbf{H}_3I_1$ are determined for two image pairs using Mozerov's stereo matching method [19]. Suppose I_s is the in-between view of $\mathbf{H}_1I_1$ and $\mathbf{H}_2I_2$ subject to the morphing parameter s as shown in Fig. 3. Following Xiao's theory [13], any linear combination of two parallel views satisfies the perspective geometry property, we figure out I_s by

$$\begin{aligned} I_s &= (1-s)\mathbf{H}_1I_1 + s\mathbf{H}_2I_2 \\ &= (1-s)\mathbf{H}_1I_1 + s\mathbf{D}_{1\to2}(\mathbf{H}_1I_1) \end{aligned} \quad (1)$$

s.t. $0 \le s \le 1$.

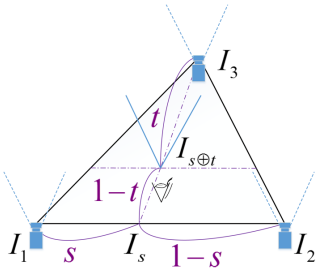


Fig. 3. Morphing parameters

In the second step, we form a dense mapping matrix $\mathbf{D}_{3\to s} : I_3 \Rightarrow I_s$ and use this to rectify I_3 and I_s.

For every point in $\mathbf{H}_1I_1$, we calculate corresponding point in I_s using Eq. 1 and form a dense mapping matrix $\mathbf{D}_{1\to s} : \mathbf{H}_1I_1 \Rightarrow I_s$. For every point p in I_3, following the path of $I_3 \to \mathbf{H}_4I_3 \to \mathbf{H}_3I_1 \to I_1 \to \mathbf{H}_1I_1 \to I_s$ as presented in Fig. 2, we figure out the corresponding point of p in I_s as $\mathbf{D}_{1\to s}(\mathbf{H}_1{\mathbf{H}_3}^{-1}\mathbf{D}_{3\to1}(\mathbf{H}_4p))$. Hence, a dense mapping matrix $\mathbf{D}_{3\to s} : I_3 \Rightarrow I_s$ is formed. Following the normal rectifying algorithm [18], two projective transformations ${\mathbf{H}_5}^s$ and ${\mathbf{H}_6}^s$ are formed to rectify I_s and I_3. Suppose $I_{s\oplus t}$ is the in-between view of ${\mathbf{H}_5}^sI_s$ and ${\mathbf{H}_6}^sI_3$ subject to the morphing parameter t as shown in Fig. 3. We figure out $I_{s\oplus t}$ by linearly combining the parallel image pair ${\mathbf{H}_5}^sI_s$ and ${\mathbf{H}_6}^sI_3$

$$I_{s\oplus t} = (1-t){\mathbf{H}_6}^sI_3 + t{\mathbf{H}_5}^sI_s \quad (2)$$

where $I_3 = \mathbf{H}_4^{-1}\mathbf{D}_{1\to3}(\mathbf{H}_3I_1)$ and I_s is as shown in Eq. 1.

We formulate $I_{s\oplus t}$ integrallty as

$$\begin{aligned} I_{s\oplus t} = {} & t(1-s){\mathbf{H}_5}^s\mathbf{H}_1I_1 \\ & + st{\mathbf{H}_5}^s\mathbf{D}_{1\to2}(\mathbf{H}_1I_1) \\ & + (1-t){\mathbf{H}_6}^s\mathbf{H}_4^{-1}\mathbf{D}_{1\to3}(\mathbf{H}_3I_1) \end{aligned} \quad (3)$$

s.t. $0 \le s,t \le 1$.

Eq. 3 demonstrates the complete formulation of $I_{s\oplus t}$. However, it is scarcely possible to figure out it in real time because it costs too much computation especially on ${\mathbf{H}_5}^s$ and ${\mathbf{H}_6}^s$ which are based on s.

3.2.2. Tri-view Morphing

On the basis of epipolar geometry, in the case of two image planes are parallel to each other (e.g. $\mathbf{H}_1I_1$ and $\mathbf{H}_2I_2$) as

shown in Fig. 4, the epipolar lines of two images are horizontal. As a result, the corresponding points in them are in the same scanline.

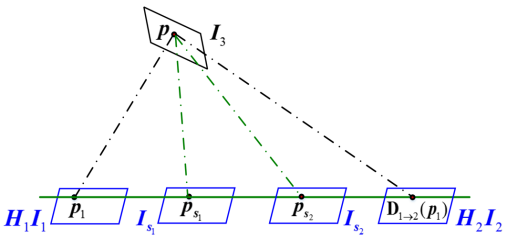


Fig. 4. Parallel image planes

Suppose p is an arbitrary point in I_3, p_1 is the corresponding point of p in $\mathbf{H}_1 I_1$. p_{s_1} and p_{s_2} are corresponding points of p_1 in I_{s_1} for $s = s_1$ and I_{s_2} for $s = s_2$ respectively. As illustrated in Eq. 1, $p_{s_2} = p_{s_1} + (s_2 - s_1)(\mathbf{D}_{1\to2}(p_1) - p_1)$.

Following parallel-image epipolar constraint, we obtain

$$p^T \mathbf{F} p_{s_1} = 0 \tag{4}$$

$$\begin{aligned} p^T \mathbf{F} p_{s_2} &= p^T \mathbf{F}(p_{s_1} + (s_2 - s_1)(\mathbf{D}_{1\to2}(p_1) - p_1)) \\ &= (s_2 - s_1) p^T \mathbf{F} \begin{bmatrix} \sigma & 0 & 0 \end{bmatrix}^T = 0 \end{aligned} \tag{5}$$

where $\mathbf{F}$ is the fundamental matrix between I_3 and I_{s_1}, and $\begin{bmatrix} \sigma & 0 & 0 \end{bmatrix}^T = \mathbf{D}_{1\to2}(p_1) - p_1$.

Eq. 4-5 indicate that the fundamental matrix $\mathbf{F}$ between I_3 and I_s is independent of s. Following Seitz's theory [10], two projective transformations ${\mathbf{H}_5}^s$ and ${\mathbf{H}_6}^s$, which are determined by $\mathbf{F}$, will also remain unchanged while s changes. This property is very important for the simplification of our algorithm because both ${\mathbf{H}_5}^s$ and ${\mathbf{H}_6}^s$ become constant. We note ${\mathbf{H}_5}^s$ as $\mathbf{H}_5$ and ${\mathbf{H}_6}^s$ as $\mathbf{H}_6$ to highlight their invariance (s is 0.5 in our implementation). Now Eq. 3 can be written as

$$I_{s\oplus t} = t(1-s)\hat{I}_1 + st\hat{I}_2 + (1-t)\hat{I}_3 \tag{6}$$

$$\begin{aligned} \text{where} \quad \hat{I}_1 &= \mathbf{H}_5 \mathbf{H}_1 I_1 \\ \hat{I}_2 &= \mathbf{H}_5 \mathbf{D}_{1\to2}(\mathbf{H}_1 I_1) \\ \hat{I}_3 &= \mathbf{H}_6 \mathbf{H}_4^{-1} \mathbf{D}_{1\to3}(\mathbf{H}_3 I_1) \end{aligned}$$

As above, $\hat{I}_1$, $\hat{I}_2$, and $\hat{I}_3$ remain constant in our implementation.

3.3. Post-warping

To fully determine a view synthesis, a post-warp matrix must be provided for each in-between image to re-project it to the final position to obtain an image interpolating the positions and color of the three sample images. Fragneto *et al.* [20] formed a two-image post-warp algorithm. We expend their algorithm to work with three images by

$$\mathbf{H}_s = \mathbf{H}_5 \mathbf{H}_1 [(\mathbf{H}_5 \mathbf{H}_1)^{-1} (\mathbf{H}_5 \mathbf{H}_2)]^s \tag{7}$$

$$\mathbf{H}_t = \mathbf{H}_6 ({\mathbf{H}_6}^{-1} \mathbf{H}_s)^t \tag{8}$$

where $\mathbf{H}^s = \exp(s \log(\mathbf{H}))$, $\log(\mathbf{H}) = -\sum_{k=1}^{\infty} \frac{(\mathbf{E}-\mathbf{H})^k}{k}$, $\exp(\mathbf{H}) = \mathbf{E} + \sum_{k=1}^{\infty} \frac{\mathbf{H}^k}{k!}$, and $\mathbf{E}$ is a $3 * 3$ identity matrix.

However, $\mathbf{H}_s$ and $\mathbf{H}_t$ may not convergence in the case of wide baseline sample images. We form a simple yet efficient linear post-warp homography for wide baseline case.

$$\mathbf{H}_s = (1-s)\mathbf{H}_5 \mathbf{H}_1 + s\mathbf{H}_5 \mathbf{H}_2 \tag{9}$$

$$\mathbf{H}_t = (1-t)\mathbf{H}_6 + t\mathbf{H}_s \tag{10}$$

We apply $\mathbf{H}_t^{-1}$ to $I_{s\oplus t}$ and yield $I_{s\bullet t}$, which is a normal view between the three sample images with respect to the user's view and position.

$$I_{s\bullet t} = \mathbf{H}_t^{-1}[t(1-s)\hat{I}_1 + st\hat{I}_2 + (1-t)\hat{I}_3] \tag{11}$$

Since $\hat{I}_1$, $\hat{I}_2$ and $\hat{I}_3$ remain unchanged, we divide our work into *off-line* and *on-line* part. In the off-line part, we calculate $\hat{I}_1$, $\hat{I}_2$ and $\hat{I}_3$ which remain constant for future use. As the saltation of $\mathbf{D}_{1\to2}$ and $\mathbf{D}_{1\to3}$ can generate black areas in the morphing view, we linearly interpolate the place where they saltate to get dense and continious mappings. This process greatly cuts down the time for online interpolation. In the on-line part, we just need to linearly combine them and apply a post-warp projection. The actual image rendering procedure can be explored in 0.05s for 480p sample images with assistance of Graphics Processing Unit (GPU), while Chatkaewmanee's Tri-view Morphing method [14] takes nearly 0.5s for only 240p sample images.

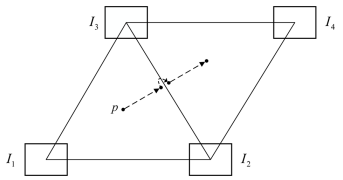
4. EXPERIMENTS

Results at edge points: At edge points like Fig. 5 (b) and (c), Eq. 11 demonstrates that both synthetic views based on two image triples approach $\mathbf{H}_t^{-1}(t\hat{I}_2 + (1-t)\hat{I}_3)$. This ensures automatic transition between triples to obtain smooth morph sequences incorporating with more than three images.

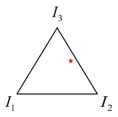
We evaluated the framework on both small and wide baseline cases as well as images from the web. User can interactively navigate the real-world scenes while putting on a HMD such as Oculus Rift as presented in Fig. 1 (a)[1]. Oculus Rift captures user's head position and orientation and we convert the information into s and t as illustrated in Fig. 3.

Results of real-world scenes: The top row of Fig. 6 presents four sample images of a museum taken with a digital camera from different positions. The scene is quite complicated which is very difficult to model by 3D textured model employing only these four images. We grouped the four sample images into two triples as illustarted in Fig. 5 (a) and generated photorealistic novel views (bottom row of Fig. 6).

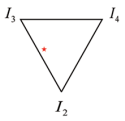
[1]This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. This includes a video demo about user navigates through the outdoor scenes in real time putting on Oculus Rift and a readme file. This material is 4.29 MB in size.



(a) Jumping from one triple to another.



(b) Rendered view based on I_1, I_2 and I_3.



(c) Rendered view based on I_2, I_3 and I_4.

Fig. 5. Example results of automatic transition between triples to create a long smooth walkthrough.

Fig. 6. Example of images taken with a hand-held digital camera. Top : four uncalibrated sample images. Bottom : a series of synthesized virtual views.

Results of images from the web: Some images were cut from Google Street View [21] website as presented in Fig.7, which were not as clear as the images taken from digital cameras. We regarded these images as input images for our system and rendered novel views between them. Despite of the low image quality, we achieved photorealistic results and even the characters on the billboard were clear enough to identify.

Results of wide baseline images: Our system managed to do well even on wide baseline camera setups. The middle row of Fig. 8 shows Chatkaewmanee's [14] results as a baseline and the bottom row shows our results. Chatkaewmanee's method is more edge-preserving than our method because they need boundary detection and request user assistance to estimate the occluded points. This method certainly works well on this simple and homochromous background. Nevertheless, it will definitely not work on complex or outdoor scenes like in Fig. 6 and Fig. 7.

Fig. 7. Example of images cut from Google Street View. Top : three uncalibrated sample images. Bottom : a series of synthesized virtual views.

Fig. 8. Example of wide baseline images. Top : three wide baseline sample images. Middle : Chatkaewmanee's Adaptive Tri-View Morphing results. Bottom: our results.

5. CONCLUSION

In this paper, we proposed and evaluated a method for real-time walkthrough of real-world outdoor environments based on only three uncalibrated images without any priori knowledge on the camera or the scene. We avoided computationally intensive by dividing the work into off-line and on-line part. With the epipolar constraint, we had most of the work be processed off-line and just needed linear combination and post-warp projection in the on-line part. This routine achieves an interactive rate and poses a qualified alternative to state-of-the-art dense 3D reconstruction. User can interactively navigate through the scene with a HMD like Oculus Rift. In the future, we hope to work on several issues, e.g. allowing all three sample images to be reference images and using this information to refine the mapping matrices $\mathbf{D}_{1\to2}$ and $\mathbf{D}_{1\to3}$ to avoid ghosting effects especially in the occluded regions.

6. REFERENCES

[1] Harry Shum and Sing B Kang, "Review of image-based rendering techniques," in *Visual Communications and Image Processing 2000*. International Society for Optics and Photonics, 2000, pp. 2–13.

[2] Christoph Fehn, "Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv," in *Electronic Imaging 2004*. International Society for Optics and Photonics, 2004, pp. 93–104.

[3] Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis, "Depth synthesis and local warps for plausible image-based navigation," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 3, pp. 30, 2013.

[4] Konstantinos Rematas, Tobias Ritschel, Mario Fritz, and Tinne Tuytelaars, "Image-based synthesis and re-synthesis of viewpoints guided by 3d models," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 3898–3905.

[5] Chong Wang, Zhen-Yu Zhu, Shing-Chow Chan, and Heung-Yeung Shum, "Real-time depth image acquisition and restoration for image based rendering and processing systems," *Journal of Signal Processing Systems*, vol. 79, no. 1, pp. 1–18, 2015.

[6] Marc Levoy and Pat Hanrahan, "Light field rendering," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 31–42.

[7] Fu-Chung Huang, Kevin Chen, and Gordon Wetzstein, "The light field stereoscope: immersive computer graphics via factored near-eye light field displays with focus cues," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 60, 2015.

[8] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen, "The lumigraph," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 43–54.

[9] Seth Berrier, Michael Tetzlaff, Michael Ludwig, and Gary Meyer, "Improved appearance rendering for photogrammetrically acquired 3d models," in *2015 Digital Heritage*. IEEE, 2015, vol. 1, pp. 255–262.

[10] Seitz, M. Steven, Dyer, and R. Charles, "View morphing," *In Proceedings of Computer Graphics (SIGGRAPH 96*, pp. 21–30, 1999.

[11] Russell A Manning and Charles R Dyer, "Interpolating view and scene motion by dynamic view morphing," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* IEEE, 1999, vol. 1.

[12] Tobias Gurdan, Martin R Oswald, Daniel Gurdan, and Daniel Cremers, "Spatial and temporal interpolation of multi-view image sequences," in *German Conference on Pattern Recognition*. Springer, 2014, pp. 305–316.

[13] Jiangjian Xiao and Mubarak Shah, "Tri-view morphing," *Computer Vision and Image Understanding*, vol. 96, no. 3, pp. 345–366, 2004.

[14] Pin Chatkaewmanee and Matthew Nelson Dailey, "Object virtual viewing using adaptive tri-view morphing," *IET Image Processing*, vol. 7, no. 6, pp. 586–595, 2013.

[15] Jonathan Richard Shewchuk, "Triangle: Engineering a 2d quality mesh generator and delaunay triangulator," in *Applied computational geometry towards geometric engineering*, pp. 203–222. Springer, 1996.

[16] Kaname Tomite, Kazumasa Yamazawa, and Naokazu Yokoya, "Arbitrary viewpoint rendering from multiple omnidirectional images for interactive walkthroughs," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. IEEE, 2002, vol. 3, pp. 987–990.

[17] Hideo Saito, Makoto Kimura, Satoshi Yaguchi, and Naho Inamoto, "View interpolation of multiple cameras based on projective geometry," in *International Workshop on Pattern Recognition and Understanding for Visual Information Media*, 2002.

[18] Charles Loop and Zhengyou Zhang, "Computing rectifying homographies for stereo vision," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on*, 1999, p. 1125.

[19] Mikhail G Mozerov and Joost van de Weijer, "Accurate stereo matching by two-step energy minimization," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 1153–1163, 2015.

[20] Pasqualina Fragneto, Andrea Fusiello, Beatrice Rossi, Luca Magri, and Matteo Ruffini, "Uncalibrated view synthesis with homography interpolation," in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*. IEEE, 2012, pp. 270–277.

[21] Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver, "Google street view: Capturing the world at street level," 2010.