A MATCHING-BASED VIEW INTERPOLATION SCHEME

Xiaoyong Sun, Eric Dubois

School of Information Technology and Engineering,
University of Ottawa,
Ottawa, Ontario, K1N 6N5,
xsun@site.uottawa.ca , edubois@site.uottawa.ca

ABSTRACT

A view interpolation algorithm for image-based rendering applications is proposed. In the proposed method, the view change is specified through the motions of feature points, which serve as control points. The triangulation of the images, combined with an affine transformation model, has been applied for the texture mapping. Simulation results show that the new synthesized views are of good quality and can well represent the view change due to the virtual camera movement.

1. INTRODUCTION

Image based rendering (IBR) provides a new class of approaches to construct virtual environments. The essential idea is to represent a 3D scene using a set of pre-captured 2D images. Then, any novel view can be generated based on the pre-captured images through view interpolation. The ways to obtain the pre-captured images and the methods for view interpolation are the distinguishing features in the different image-based rendering approaches [1].

According to the basic elements involved in interpolation, the view interpolation methods for novel view generation can be classified as pixel-based, column-based and block-based view interpolation. The pixel-based view interpolation methods such as light field rendering [2] rely on the precise control of the camera positions in two dimension. In the column-based view interpolation methods such as Concentric Mosaics [3], the camera positions are only precisely controlled in one dimension. As a consequence, depth distortion is usually unavoidable. Triangulations have also been used in view interpolation where the images are segmented into triangular patches, or irregular blocks. However, the method in [4] does not directly use feature points for segmentation and the algorithm is very complex.

In this paper, a novel view-consistent model to constrain the view change with the motion of the virtual camera is first proposed for the IBR application. After converting this general model to the corresponding feature-position-consistent model, we describe our matching-based view interpolation algorithm in accordance to the proposed model. Based on this approach, our IBR strategy is to divide the whole navigation area into small sub-areas and to interpolate the novel views from virtual camera positions within the sub-areas using neighboring pre-captured images. The sub-area used in this paper is a triangle. In this way, we avoid the precise control of the camera positions where the pre-captured images should be taken.

Traditionally, view interpolation is based on dense disparity maps. It is well-known that precise dense disparity maps are very difficult to obtain in practice [5]. On the other hand, the feature or corner-based sparse matching techniques have made great progress. The new view interpolation algorithm proposed in this paper is based on triangulation together with the affine transformation for each triangular patch instead of dense disparities. As a consequence, it requires a large number of approximately uniformly distributed feature matchings. We develop a multi-view matching detection algorithm based on a new view-matching relationship constraint, together with some other traditional epipolar constraints used in computer vision. This is another contribution in this paper. In addition, we propose the method to fill in the texture in the boundary area where there are no feature matchings.

In section 2, the view-consistent model and the proposed algorithm that follows this model are described together with the scenario used in this paper. The view interpolation method is addressed in section 3. The simulation results follow in section 4, with our conclusion and future work in section 5.

2. THE VIEW INTERPOLATION MODEL AND THE PROPOSED ALGORITHM

There are many different applications for view interpolation, such as frame rate conversion, stereo pair adjustment, IBR, etc. For different applications, the objectives are different and result in different models. In this section, we give our interpolation model for the IBR application.

2.1. A view-consistent model for IBR application

Assume Π denotes a specified navigation area, and $I_{\boldsymbol{s}(k)}$, $k = 1, 2, .., K$, are pre-captured images with the camera projection center at positions $\boldsymbol{s}(1), \boldsymbol{s}(2), ...\boldsymbol{s}(K)$, ($\boldsymbol{s}(1), \boldsymbol{s}(2)$, $...\boldsymbol{s}(K) \in \Pi$). These images have similar, but not necessarily identical, viewing directions of the scene. For the IBR application, the objective is to synthesize an arbitrary view at some position $\boldsymbol{s_i}$ with the similar viewing direction, and satisfy the following consistency conditions:

$$I_{\boldsymbol{s_i}} = I_{\boldsymbol{s}(k)}, \text{if } \boldsymbol{s_i} = \boldsymbol{s}(k), \text{for } k = 1, 2, ...K, \quad (1)$$

and for any of the two views $I_{\boldsymbol{s}_i}$ and $I_{\boldsymbol{s}_j}$ within Π,

$$\lim_{\boldsymbol{s_i} \to \boldsymbol{s_j}} |I_{\boldsymbol{s_i}} - I_{\boldsymbol{s_j}}| = 0. \quad (2)$$

where, $I_{\boldsymbol{s}_i}$ and $I_{\boldsymbol{s}_j}$ can be pre-captured or interpolated views.

Based on this view-consistent interpolation model, the synthesized virtual views may be different from the real physical ones, but will look reasonable during the navigation.

2.2. The correspondent feature-position-consistent interpolation model

In this paper, a matching-based view interpolation algorithm is proposed. Feature matchings among pre-captured images are first detected and used as control points. The above texture-consistent interpolation model is converted to the feature-position-consistent interpolation model.

Assume that N feature points have been found in all of the K pre-captured images and there is a unique corresponding relationship of these N feature points among the K pre-captured images. Further assume that the matchings of these N feature points will appear in any of the views that will be synthesized from these K pre-captured images.

The matchings between views $I_{\boldsymbol{s}_i}$ and $I_{\boldsymbol{s}_j}$ are represented as two sets of feature points $\text{MP}_i = \{\boldsymbol{x}_i(\boldsymbol{x}_{s(1),n}) | n = 1, 2, ..., N\}$ and $\text{MP}_j = \{\boldsymbol{x}_j(\boldsymbol{x}_{s(1),n}) | n = 1, 2, ..., N\}$ in $I_{\boldsymbol{s}_i}$ and $I_{\boldsymbol{s}_j}$, respectively. For any feature point $\boldsymbol{x}_i(\boldsymbol{x}_{s(1),n}) \in \text{MP}_i$, its matching point in $I_{\boldsymbol{s}_j}$ is denoted as $\boldsymbol{x}_j(\boldsymbol{x}_{s(1),n}) \in \text{MP}_j$. Here, we use the feature points $\boldsymbol{x}_{s(1),n}, (n = 1, 2...N)$ in $I_{\boldsymbol{s}(1)}$ as the reference points. Then the correspondent feature-position-consistent interpolation model can be represented as,

$$\boldsymbol{x}_i(\boldsymbol{x}_{s(1),n}) = \boldsymbol{x}_{s(k)}(\boldsymbol{x}_{s(1),n}) \text{if } \boldsymbol{s_i} = \boldsymbol{s}(k) \quad (3)$$

for $k = 1, 2, ...K, n = 1, 2, ...N$. $\boldsymbol{x}_{s(k)}(\boldsymbol{x}_{s(1),n})$ denotes the matchings in the pre-captured image $I_{\boldsymbol{s}(k)}$. And

$$\lim_{\boldsymbol{s_i} \to \boldsymbol{s_j}} |\boldsymbol{x}_i(\boldsymbol{x}_{s(1),n}) - \boldsymbol{x}_j(\boldsymbol{x}_{s(1),n})| = 0, n = 1, 2, ...N. \quad (4)$$

It is obvious that the feature-position-consistent interpolation model will converge with the texture-consistent interpolation model when the number N of the total matchings becomes large enough.

2.3. The scenario considered in this paper

In this paper, we will address the scenario that the sub-area is a triangle, and that $K = 3$. The triangular area is specified by its vertices, the positions where three pre-captured images are taken. Assume that the three pre-captured images with similar viewing directions are $I_{\boldsymbol{s}_A}$, $I_{\boldsymbol{s}_B}$, and $I_{\boldsymbol{s}_C}$, taken at positions $\boldsymbol{s}_A$, $\boldsymbol{s}_B$, and $\boldsymbol{s}_C$. The matchings in these three images form the sets MP_A, MP_B, and MP_C. We will describe the procedure to generate an arbitrary view $I_{\boldsymbol{s}_i}$ with the similar viewing direction at position $\boldsymbol{s}_i$, which is within the triangle with vertices $\boldsymbol{s}_A$, $\boldsymbol{s}_B$ and $\boldsymbol{s}_C$.

Following the feature-position-consistent view interpolation model, the positions of the feature matchings in $I_{\boldsymbol{s}_i}$ can be calculated as

$$\boldsymbol{x}_i(\boldsymbol{x}_{A,n}) = \eta_A \cdot \boldsymbol{x}_{A,n} + \eta_B \cdot \boldsymbol{x}_B(\boldsymbol{x}_{A,n}) + \eta_C \cdot \boldsymbol{x}_C(\boldsymbol{x}_{A,n}) \quad (5)$$

for $n = 1, 2, ...N$. Thus, all points $\boldsymbol{x}_i(\boldsymbol{x}_{A,n})$ form the matching set MP_i for the new image $I_{\boldsymbol{s}_i}$.

Different ways to obtain the weights η_A, η_B and η_C may exist in order to satisfy the feature-position-consistent interpolation model. One way to calculate the weights that is used in this paper is

$$\begin{aligned} \eta_A &= \frac{l_B \cdot l_C}{l_A \cdot l_B + l_B \cdot l_C + l_A \cdot l_C} \\ \eta_B &= \frac{l_A \cdot l_C}{l_A \cdot l_B + l_B \cdot l_C + l_A \cdot l_C} \\ \eta_C &= \frac{l_B \cdot l_A}{l_A \cdot l_B + l_B \cdot l_C + l_A \cdot l_C} \end{aligned} \quad (6)$$

where, $l_A = |\boldsymbol{s}_A - \boldsymbol{s}_i|$, $l_B = |\boldsymbol{s}_B - \boldsymbol{s}_i|$ and $l_C = |\boldsymbol{s}_C - \boldsymbol{s}_i|$.

3. THE VIEW INTERPOLATION METHOD

In this section, the triangulation-based view interpolation method is described. We will start with the multiple view matching detection. Then the relationship among correspondent triangular patches in different views will be determined.

An affine transformation will be used for the texture mapping from the pre-captured images to the new view. The affine transformation for texture mapping is a good model under the following conditions: 1) the triangular patch is physically located in a plane in the scene; or 2) the triangular patch is small enough; or 3) the separations between the camera positions where pre-captured images are taken are small enough. Then the correspondent texture of the triangular patch in the new view can be mapped from the three pre-captured images. The affine transformation can be determined from the geometric relationship between the positions of the three correspondent vertices. In addition, the rendering strategy for the boundary area of the image where there are no matchings is given.

3.1. Tri-view feature matching

The multi-view matching problem can be solved through various feature tracking techniques. In this paper, the matchings from three pre-captured views will be detected. Recently, an epipolar-gradient-based matching detection algorithm has been proposed based on the camera's calibration information [6]. However, the camera's calibration information is usually obtained from some pre-obtained matchings, and it is very sensitive to the accuracy of the pre-obtained matchings. In this paper, a large number of approximately uniformly distributed matchings are required for our affine transformation based view interpolation. We first obtain a set of good matchings through the view-matching relationship, which we call the ABCA law.

The Harris corners are first detected in the images $I_{\boldsymbol{s}_A}$, $I_{\boldsymbol{s}_B}$ and $I_{\boldsymbol{s}_C}$. Then, the two-view matchings between $I_{\boldsymbol{s}_A}$ and $I_{\boldsymbol{s}_B}$, $I_{\boldsymbol{s}_B}$ and $I_{\boldsymbol{s}_C}$, and $I_{\boldsymbol{s}_C}$ and $I_{\boldsymbol{s}_A}$ are found from the detected corners through normalized correlation and refined through the fundamental matrices. From the matchings between $I_{\boldsymbol{s}_A}$ and $I_{\boldsymbol{s}_B}$, and $I_{\boldsymbol{s}_B}$ and $I_{\boldsymbol{s}_C}$, we can set up the tri-view matching relationship among $I_{\boldsymbol{s}_A}$, $I_{\boldsymbol{s}_B}$ and $I_{\boldsymbol{s}_C}$, related through the common feature points in image $I_{\boldsymbol{s}_B}$. Finally, we use the matchings between $I_{\boldsymbol{s}_C}$ and $I_{\boldsymbol{s}_A}$ to check the validity of the above tri-view matchings. Experiments show that a set of good matchings can be obtained through the above methods. It is obvious that the proposed ABCA law can easily be extended to multi-view (more than three) matching detection.

In order to increase the number of the matchings, we calculate the fundamental matrices between each two-view pair and the tri-view tensor from the above matchings. The matchings between each two-view pair will be checked with the correspondent new fundamental matrix and then more matchings can be obtained using tensor-based transferring from two-view matchings to the third view. In addition, the matchings that are inconsistent with their neighbors are removed.

3.2. Setting up relationship between triangular patches

We have obtained the feature matching relationship among the pre-captured views $I_{\boldsymbol{s}_A}$, $I_{\boldsymbol{s}_B}$, $I_{\boldsymbol{s}_C}$ and new view $I_{\boldsymbol{s}_i}$. Now we want to set up the relationship between triangular patches among these views.

The new view without texture is first partitioned using Delaunay triangulation through the point $\boldsymbol{x}_i(\boldsymbol{x}_{A,n})$. The corresponding triangular patches in pre-captured image $I_{\boldsymbol{s}_A}$, $I_{\boldsymbol{s}_B}$ and $I_{\boldsymbol{s}_C}$ can thus be obtained based on the Delaunay triangulation of the view $I_{\boldsymbol{s}_i}$. The corresponding matchings (in $I_{\boldsymbol{s}_A}$, $I_{\boldsymbol{s}_B}$, $I_{\boldsymbol{s}_C}$) of the three vertices of one triangular patch in $I_{\boldsymbol{s}_i}$ construct the correspondent triangular patches in $I_{\boldsymbol{s}_A}, I_{\boldsymbol{s}_B}, I_{\boldsymbol{s}_C}$. Obviously, the triangulations of pre-captured images generated in this way may not be exactly the Delaunay triangulations, but are approximate ones.

In this way, we set up the triangular patch relationships among the pre-captured images and the new view. Assume $\boldsymbol{T}_i^m(\boldsymbol{x}_{i,n_1}, \boldsymbol{x}_{i,n_2}, \boldsymbol{x}_{i,n_3})$ denotes a triangular patch in image $I_{\boldsymbol{s}_i}$ with three vertices $\boldsymbol{x}_{i,n_1}$, $\boldsymbol{x}_{i,n_2}$ and $\boldsymbol{x}_{i,n_3}$, $m = 1, 2, ..., M$ with M the total number of Delaunay triangles. Its correspondent triangular patch in $I_{\boldsymbol{s}_A}$ are $\boldsymbol{T}_A^m(\boldsymbol{x}_A(\boldsymbol{x}_{i,n_1}), \boldsymbol{x}_A(\boldsymbol{x}_{i,n_2}), \boldsymbol{x}_A(\boldsymbol{x}_{i,n_3}))$, and similarly for images $I_{\boldsymbol{s}_B}$ and $I_{\boldsymbol{s}_C}$. From now on, we will use $\boldsymbol{T}_i^m$, $\boldsymbol{T}_A^m$, $\boldsymbol{T}_B^m$ and $\boldsymbol{T}_C^m$ to denote a set of correspondent triangular patches.

3.3. Texture rendering methods for different categories of triangular patches

The affine transformation is used for texture mapping. A six- parameter affine transformation $\boldsymbol{t}_m^A$can be easily obtained from the geometric relationship between three correspondent vertices of triangular patches $\boldsymbol{T}_i^m$ and $\boldsymbol{T}_A^m$, or between $(\boldsymbol{x}_{i,n_1}, \boldsymbol{x}_{i,n_2}, \boldsymbol{x}_{i,n_3})$ and $(\boldsymbol{x}_A(\boldsymbol{x}_{i,n_1}), \boldsymbol{x}_A(\boldsymbol{x}_{i,n_2}), \boldsymbol{x}_A(\boldsymbol{x}_{i,n_3}))$.

If we use $E(\boldsymbol{T}_i^m)$ to denote the texture within the triangular patch $\boldsymbol{T}_i^m$, then $E_A(\boldsymbol{T}_i^m) = F(\boldsymbol{T}_i^m, I_{\boldsymbol{s}_A}, \boldsymbol{t}_m^A)$ represent obtaining the texture of $\boldsymbol{T}_i^m$ from $I_{\boldsymbol{s}_A}$ through affine transformation $\boldsymbol{t}_m^A$. Similarly, we can have $E_B(\boldsymbol{T}_i^m) = F(\boldsymbol{T}_i^m, I_{\boldsymbol{s}_B}, \boldsymbol{t}_m^B)$ and $E_C(\boldsymbol{T}_i^m) = F(\boldsymbol{T}_i^m, I_{\boldsymbol{s}_C}, \boldsymbol{t}_m^C)$. Theoretically, $E(\boldsymbol{T}_i^m)$ can be obtained from $E_A(\boldsymbol{T}_i^m)$, $E_B(\boldsymbol{T}_i^m)$, $E_C(\boldsymbol{T}_i^m)$ or the combination of them. In order to minimize the discontinuities between the triangle patches in the new view, the following rendering strategy is used,

$$E(\boldsymbol{T}_i^m) = \begin{cases} E_A(\boldsymbol{T}_i^m) & \text{if } d_A \le d_B, d_C \\ E_B(\boldsymbol{T}_i^m) & \text{if } d_B < d_A, d_C \\ E_C(\boldsymbol{T}_i^m) & \text{if } d_C < d_A, d_B \end{cases}$$

where, $d_A = (\sum_{n=1}^{N} |\boldsymbol{x}_i(\boldsymbol{x}_{A,n}) - \boldsymbol{x}_{A,n}|)/N$, $d_B = (\sum_{n=1}^{N} |\boldsymbol{x}_i(\boldsymbol{x}_{A,n}) - \boldsymbol{x}_B(\boldsymbol{x}_{A,n})|)/N$, and $d_C = (\sum_{n=1}^{N} |\boldsymbol{x}_i(\boldsymbol{x}_{A,n}) - \boldsymbol{x}_C(\boldsymbol{x}_{A,n})|)/N$.

3.4. Filling in the texture at the boundary area

One of the problems associated with triangle-based view interpolation is the boundary area of the image where there are no matchings detected, and thus no triangular patches. Often the texture is not rich in these areas.

In this paper, the distances between the positions where the pre-captured images are taken are not large. As a consequence, the matchings of the boundary feature points in $I_{\boldsymbol{s}_i}$ are also the boundary feature points in $I_{\boldsymbol{s}_A}$, $I_{\boldsymbol{s}_B}$ and $I_{\boldsymbol{s}_C}$.Thus, we use a global affine transformation to estimate the texture in these areas. The boundary matchings $\boldsymbol{x}_i(\boldsymbol{x}_{A,n}^b)$ within MP_i are selected and form a subset $\text{MP}_{i,b}$, or $\text{MP}_{i,b} = \{\boldsymbol{x}_i(\boldsymbol{x}_{A,n}^b) | n = 1, 2, ..., N^{'}\}$, and $\text{MP}_{i,b} \subset \text{MP}_i$. $\boldsymbol{x}_{A,n}^b$ denotes the boundary matchings in the pre-captured image $I_{\boldsymbol{s}_A}$, or $\text{MP}_{A,b} = \{\boldsymbol{x}_{A,n}^b | n = 1, 2, ..., N^{'}\}$. Similarly, $\text{MP}_{B,b} =$

$\{\boldsymbol{x}_B(\boldsymbol{x}^b_{A,n})|n = 1, 2, ..., N^{'}\}$ and $\text{MP}_{C,b} = \{\boldsymbol{x}_C(\boldsymbol{x}^b_{A,n})|n = 1, 2, ..., N^{'}\}$. $N^{'}$ is the total number of boundary matchings.

The texture in the boundary area is mapped from the same pre-captured image as the one to render the triangular patches. For example, if the texture will be rendered from image $I_{\boldsymbol{s}_A}$, then an optimal global affine transformation will be determined by,

$$\hat{\boldsymbol{t}}^{\boldsymbol{A}}_{\boldsymbol{b}} = \arg\min_{t^A_b} \sum_{\boldsymbol{n}=1}^{N^{'}} |\widetilde{\boldsymbol{x}}^{'}_i(\boldsymbol{x}^b_{A,n}, \boldsymbol{t}^{\boldsymbol{A}}_{\boldsymbol{b}}) - \boldsymbol{x}_i(\boldsymbol{x}^b_{A,n})| \quad (7)$$

where, $\widetilde{\boldsymbol{x}}^{'}_i(\boldsymbol{x}^b_{A,n}, \boldsymbol{t}^{\boldsymbol{A}}_{\boldsymbol{b}})$ denotes the new value of $\boldsymbol{x}_i(\boldsymbol{x}_{A,n})$ transferred from points $\boldsymbol{x}_{A,n}$ through the affine transformation $\boldsymbol{t}^{\boldsymbol{A}}_{\boldsymbol{b}}$. Finally, the boundary matchings $\boldsymbol{x}_i(\boldsymbol{x}^b_{A,n})$ will be updated in MP_i through the optimal global affine transformation, and thus the texture within the triangular patches related to the boundary matchings, in order to minimize the potential texture discontinuities.

4. SIMULATION RESULTS

A simulation was performed on three pre-captured images of the size 1024×768, taken at three positions in the navigation area with similar viewing directions. 3595 matchings were detected and the images were segmented into 7893 triangle patches. Thus the area of each triangular patch is relatively small, which is suitable for the affine transformation model. One of the synthesized new views is shown in Fig. 1. From the figure, we can see that the synthesized view is

Fig. 1. One of the synthesized views using the proposed method

of good quality using the affine transformations instead of dense disparity maps.

5. CONCLUSION AND FUTURE WORK

In this paper, a matching-based view interpolation method is proposed for IBR application. The navigation area can be triangulated by a set of positions, where the pre-captured images are taken, and consequently, a view interpolation method based on three source images is proposed.

For view interpolation, the triangulation of the images and the affine transformation models are used. In this way, feature matchings instead of dense disparities are required because the feature matchings are usually more reliable. A multiple view matching detection strategy is described, which can obtain a large number of approximately uniformly distributed matchings.

Currently, there is no theoretical rule to follow in order to determine the optimal separations between the camera positions where the pre-captured images should be taken. This is a 3D scene sampling issue. In addition, if the separations between camera positions where the pre-captured images are taken are large, the affine transformation will not be a good model for texture mapping and discontinuities between adjacent triangular patches will appear. These issues are currently being studied.

6. ACKNOWLEDGEMENT

This work was supported by Natural Sciences and Engineering Research Council of Canada Strategic Grant STPGP 269997.

7. REFERENCES

[1] H.-Y. Shum, S. Kang, and S.-C. Chan, "Survey of image-based representations and compression techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 1020–1037, November 2003.

[2] M. Levoy and P. Hanrahan, "Light field rendering," *Computer Graphics (SIGGRAPH'96)*, pp. 31–42, August 1996.

[3] H.-Y. Shum and L. He, "Rendering with concentric mosaics," *Computer Graphics (SIGGRAPH'99)*, pp. 299–306, January 1999.

[4] M. Lhuillier and L. Quan, "Image-based rendering by joint view triangulation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, 2003.

[5] S. Baker, T. Sim, and T. Kanade, "When is the shape of a scene unique given its light-field: A fundamental theorem of 3D vision?," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, pp. 100 – 109, 2003.

[6] E. Vincent and R. Laganière, "Matching with epipolar gradient features and edge transfer," *Proc. IEEE Int. Conf. Image Processing*, pp. 277–280, September 2003.