A Free-viewpoint TV System

Yuichi Yaguchi, Takashi Matsuzaki, Toshimitsu Suzuki, Yukihiro Yoshida,
Yuichi Okuyama, Kazuaki Takahashi and Ryuichi Oka
University of Aizu
Tsuruga, Ikkimachi, Aizuwakamatsu, Fukushima, 965-8580 Japan.
{yaguchi, s1150212, s1150126, m5141146, okuyama}@u-aizu.ac.jp,
gabukazu@gmail.com, oka@u-aizu.ac.jp

Abstract

We propose an implementation of a model-based free-viewpoint TV (FTV) system using only three uncalibrated cameras. FTV is next-generation media that enables us to see a scene from any viewpoint. A model-based approach for realizing FTV requires real-time 3D object capture using multiple cameras. Here, we propose a system for reconstructing 3D object surfaces using the so-called 2D continuous dynamic programming (2DCDP) method with factorization. 2DCDP is a powerful technique for full-pixel optimal matching. It provides pixel correspondences between the images captured by the three cameras. The proposed system works well as a promising FTV system.

1 Introduction

Free-viewpoint television (FTV), which is different from traditional television, is a next-generation system that allows any user to change his or her viewpoint of an object. The current TV system is only a projected 2D surface; it can carry separate views for the left and right eyes to exploit the understanding of 3D depth in the human vision system. FTV allows people to change their viewpoint because captured objects are projected onto 3D space [1]. Thus, FTV is truly a next-generation system that we hope can be applied to education, entertainment, medical, and other areas.

Prior research on FTV can be classified into two types: view-based methods and model-based methods. A view-based method aims to construct a 2D image using interpolation techniques from many 2D images that are in a 3D space. The ray-based method [2] is the most popular technique among view-based methods; it uses a ray-space technique to create a new viewpoint image, mainly from neighbor images. A feature of the viewpoint-based method is its ability to create natural 2D images from many cameras that are calibrated precisely, and previously calculated epipolar geometry for two particular cameras. However, it requires at least several dozen and possibly hundreds of cameras to achieve precise images. Thus, it is difficult and expensive to prepare environment for it.

On the other hand, with the model-based method, it is easy to render free-viewpoint images if the model is able to reconstruct them precisely. There are many techniques for 3D reconstruction from images: the stereo method [3]; shapes from silhouettes [4]; the factorization method [5]; mixing factorization and epipolar geometry for the projected space [6]; shape from shading [7]; and photometric stereo [8]. Every technique has effective and weak points for developing FTV. Shape from shading and photometric stereo can achieve object shapes with very high precision, but it is difficult to create effective textures and it is difficult to calculate a precise lighting position. Shape from silhouette is easy to implement as a real-time system [9, 10], but it is difficult to set camera positions and parameters, and a large space like a studio is required. The stereo method and the epipolar geometry method are well-implemented in many computer vision research, but these methods need strict-calibrated cameras all have fixed positions and focuses. Thus, these methods are difficult to extend in physical resources. The factorization method does not require initial camera calibration and position information. This method can reconstruct target objects precisely from three or more cameras if the correspondence points in the views can be fixed precisely.

For extracting precise image correspondence, Scale-invariant Feature Tracker (SIFT) [11] is well known method for image matching. Another method, Kanade–Lucas–Tomasi Tracker with corner detection [12], is also used effectively to understand 3D surfaces in real-world augmented-reality applications. However, these image matching or tracking techniques can track only sparse pixels of an image, rather than the complete image. Thus, they can reconstruct rectilinear artifacts, but it is difficult to reconstruct soft, curved bodies like those of humans or animals.

We propose to develop a real-time FTV object capturing system using a consumer PC. Our main contribution is to use no calibrated camera, blue screen, special sensor device or special hardware, but we use a multicore CPU, three synchronous cameras and an image-matching algorithm called 2D Continuous dynamic programming (2DCDP), which gives dense pixel-wise correspondences [13], with a factorization technique for reconstructing a 3D surface from the dense pixel-wise correspondence.

Section 2 gives an overview of our system and Section 3 briefly describes key features such as 2DCDP, factorization, and a morphing technique using 2DCDP. Section 4 explains the result of FTV implementation, and considers the precision of reconstructed objects. Section 5 concludes this paper and describes our future work.

2 System Overview

Our system is based on the 3D reconstruction system developed by Yaguchi et al. [14], which can extract dense pixel correspondences between input images and reconstruct a trusted 3D surface. The key feature of our system is 2DCDP image matching, which can give precise dense correspondences between a reference image and a target image. 2DCDP has a high calculation

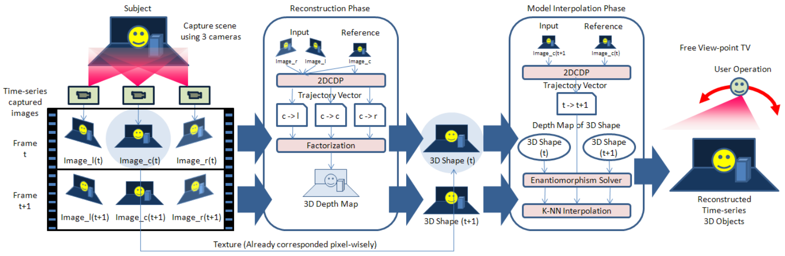


Figure 1. Schematic diagram of Free-viewpoint TV capturing system using 2DCDP and factorization

burden, thus our system uses a faster implementation called weak spotting. An overview of this system is shown in Figure 1. After acquiring a frame of three images, the system finds the correspondences between the center and left or right images and reconstructs 3D points from the pixel correspondences. 2DCDP can save pixel alignments on each image, thus the texture of a 3D surface can be added directly to the reconstructed 3D points via factorization method [15]. For smooth movement between frames, this system applies a morphing technique to interpolate several frames of 3D surfaces between the current and previous frame by 2DCDP matching. Finally, the user can view the 3D image using a 3D surface viewer that can change viewpoint easily.

Weak Spotting The current 2DCDP implementation was proposed by Yaguchi et al. in 2010 [13] and in its optimal implementation [16], it can achieve precise pixel-wise matching with an arbitrarily shaped reference image. However, our system does not require arbitrary shapes and large pixel movements because the frame interval is at most 0.2 to 0.3 seconds. Thus, our implementation applies an alignment window to avoid the accumulation calculation. The alignment window limits pixel movement between the reference image and the target image.

With the measurement equation $D(\hat{i}, \hat{i}, m, n)$, a pixel $(\hat{i}, \hat{j})$ optimally matches the target image S with a pixel (m, n) on the reference image R, as follows:

$$D(\hat{i}, \hat{j}, m, n) = \min_{\xi,\eta}\{\sum_{m=1}^{M}\sum_{n=1}^{N} d(\xi(m,n), \eta(m,n), m, n)\}, \tag{1}$$

where for a pixel in the target image set $(i, j) \in S$ and a pixel in the reference image set $(m, n) \in R$, each i, j, m, n has a limit value I, J, M, N such as $i, j, m, n \subset \mathbb{N}$, $i \leq I, j \leq J, m \leq M, n \leq N$ and a local distance set $d(i, j, m, n)$. In the accumulation calculation, the calculated correspondence between $(m, n) \in R$ and $(\xi(m, n), \eta(m, n)) \in S$ is almost moving only small distance in continuous motion images. Then, suppose $M = I$ and $N = J$ for each image, and the correspondence pixel of (m, n) in the target image is close enough so that $(\xi(m, n), \eta(m, n))$, $\xi(m, n) = m, \eta(m, n) = n$ in the target image S. The scope of the accumulation calculation on position (m, n) in the target image can then be defined as:

$$(\xi(m,n), \eta(m,n)) \in S, m - \alpha \leq \xi(m,n) \leq m + \alpha,$$
$$n - \beta \leq \eta(m,n) \leq n + \beta, \tag{2}$$

and the coverage Φ of the calculation plane as:

$$\Phi = \frac{(2\alpha + 1) * (2\beta + 1)}{I * J}, \tag{3}$$

which directly contributes to the reduced calculation costs.

Optimization for the 3D Model For 3D reconstruction, this system uses the factorization technique proposed by Tomasi and Kanade [15]. This method is not precise but is stable in the presence of noise, and does not require the distances between cameras and objects. A C implementation of the factorization technique is used in Kanatani and Sugaya's implementation of orthogonal factorization [17]. After creating a 3D model from the pixel trajectory of three views, the reconstructed object has depth errors, disarrangement of x and y coordination and two solutions that are enantiomorphic to each other. Thus, this system optimizes each object to provide compatible time-series 3D motion images.

Let the αth three-dimensional point be $(x_{t\alpha}, y_{t\alpha}, z'_{t\alpha})$ $(i = 1, \ldots, N)$ and its enantiomorph $(x_{t\alpha}, y_{t\alpha}, z''_{t\alpha})$ $(i = 1, \ldots, N, z''_{t\alpha} = z'_{t\alpha})$ in a model S_t, which is taken at time t. We also have a three-dimensional point N, and this point has a two-dimensional texture anchor $(m'_{t\alpha}, n'_{t\alpha})(0 \leq x_{t\alpha} \leq 1, 0 \leq n_{t\alpha} \leq 1)$. In this implementation, S_t is the 3D object extracted using the center camera coordination, and correspondences between texture and the center camera image, which is a reference image for 2DCDP matching, are expressed as $m' \simeq \frac{m}{M}, n' \simeq \frac{n}{N}$. This texture coordination can compensate for the distorted 3D shape S_t. The depth map $z'_{t\alpha}$ and $z''_{t\alpha}$of S_t is not decided strictly, so this system normalizes the depth map in the range of $0 \leq z'_{t\alpha}, z''_{t\alpha} \leq 1$. Next, to solve the enantiomorph problem, we determine the correct depth map $z_{t\alpha}$ as follows:

$$z_{t\alpha} \triangleq \underset{z'_{t\alpha}, z''_{t\alpha}}{\operatorname{argmin}} \begin{cases} \displaystyle\sum_{k=1}^{N} \sqrt{z'_{tk} - z_{(t-1)k}} \\ \displaystyle\sum_{k=1}^{N} \sqrt{z''_{tk} - z_{(t-1)k}}. \end{cases} \tag{4}$$

Finally, the model S_t is uniquely determined and the system can continue to reconstruct 3D objects.

Interpolation on Each Frame using a Morphing Technique People can extract motion from pictures at rates of 7∼13 frames per second (fps) [18]

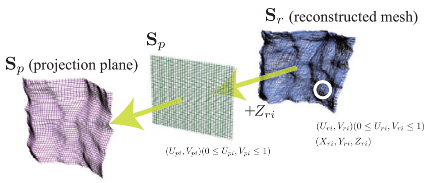

$\mathbf{S}_r$ (reconstructed mesh)
$\mathbf{S}_p$
$\mathbf{S}_p$ (projection plane)
$+Z_{ri}$
$(U_{ri}, V_{ri})(0 \le U_{ri}, V_{ri} \le 1)$
(X_{ri}, Y_{ri}, Z_{ri})
$(U_{pi}, V_{pi})(0 \le U_{pi}, V_{pi} \le 1)$

Figure 2. Fixing a reconstructed object on the x and y axes

Table 1. Specification sheet for the experiment

Machine	Dell Precision Workstation T7400
CPU	2 × Intel Xeon X5472 3.0 GHz quad-core processors
Memory	64 GB DDR2 SDRAM
Graphics	NVidia GeForce
Camera	3 × Pointgray Flea2 IEEE1394b
Image size	320×240 QVGA, RGB888 Color
Camera separation	From 10 cm to 50 cm, optional
Object distance	From 2 m to 4 m, optional

and above, and traditional movies have around 24 fps recorded to film. Our system also tries to create a 24 fps movie but it is difficult to create a high-frame-rate movie because the processing time for each frame is around 0.1 msec, which exceeds the 0.04 msec available at 24 fps. Therefore, we plan to interpolate several frames of 3D surfaces to achieve 24 fps using 2DCDP and the nearest neighbor method.

Let the texture image of the current frame be the target image S and let the texture image of the previous frame be the reference image R. 2DCDP describes the motion between R and S as a pixel correspondence $(m, n) \to (i, j)$. Now, let the number of interpolation frames be F, and let the number of an interpolation frame be f. Then the position (m'_f, n'_f) of frame f is $(m'_f, n'_f) \to (m+(i-1)/f, n+(j-n)/f)$. Next, let the color value $S_{pf}(m_f, n_f)$ of pixel (m_f, n_f) and the depth value $S_{df}(m_f, n_f)$ in interpolation frame f be defined by the color value $S_{pk}(i, j), R_{pk}(m, n)$ and depth value $S_{dk}(i, j, R_{dk}(m, n)$, respectively, and let the distance from the center of pixel $d_k = \sqrt{(m_f - m'_f)^2 + (n_f - n'_f)^2}$ of three neighbor points be (m'_f, n'_f):

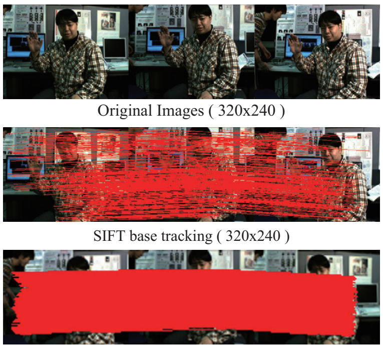
$$S_{pf}(m_f, n_f) = \frac{\sum_{k=1}^{3}[(1-d_k)\{(1-\frac{f}{F})S_{pk}(m,n)+\frac{f}{F}R_{pk}(i,j)]}{\sum_{k=1}^{3}(1-d_k)} \quad (5)$$

$$S_{df}(m_f, n_f) = \frac{\sum_{k=1}^{3}[(1-d_k)\{(1-\frac{f}{F})S_{dk}(m,n)+\frac{f}{F}R_{dk}(i,j)]}{\sum_{k=1}^{3}(1-d_k)}. \quad (6)$$

3 Experiment

In this section, we present three experimental results: a comparison study of 3D model accuracy using orthogonal factorization with 2DCDP and SIFT matching, and performance evaluation of our implementation for frame refresh rate and its stability. The evaluation environment is shown in Table 1. In 2DCDP matching, the target image is cut by 10% at the top, bottom, left and right side to define the occlusive area.

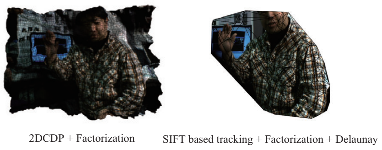
Comparison between SIFT and 2DCDP on FTV System Figure 3 compares pixel tracking precision between SIFT and 2DCDP. SIFT is good for tracking if there is enough texture on the surface and 2DCDP can include the same positions as SIFT and can also find more suitable points between two images. 2DCDP is forcibly matching on occlusions because an occlusive pixel should not match anywhere. However, 3D reconstruction using factorization could not find the segmentation area of each object; thus, the results of 2DCDP and SIFT have almost the same errors with occlusion areas such as an extended rubber-like surface 4. In texture rendering, 2DCDP can readily determine the 3D surface because all pixels are structured clearly, but SIFT matching must create a Delaunay triangle to determine the plane for texture rendering. This factor is a big advantage in real-time 3D reconstruction.


Original Images (320x240)

SIFT base tracking (320x240)

2DCDP base tracking (100x75)

Figure 3. Pixel tracking comparison between SIFT and 2DCDP on FTV system


2DCDP + Factorization

SIFT based tracking + Factorization + Delaunay

Figure 4. Comparison of the 3D reconstruction result between SIFT and 2DCDP with orthogonal factorization

Performance Evaluation for Processing Speed and Image Size Figure 5 shows the processing time for constructing a frame of a 3D surface. The image-matching and fixing part require three 2DCDP processes. In short, the cost of 2DCDP is almost 90% of the total cost in this system. If we use this system in real time on 5 fps (to create 25 fps with four frames interpolated after each key frame), we must restrict the image to 80 × 80 pixels.

Performance Evaluation for Precision and Stability Figure 7 compares another viewpoint of the actual scene and a reconstructed 3D view. From this result, the precision of 3D reconstruction is not adequate, because many occlusive areas create gaps in objects, but the reconstruction can express the relative position of the surfaces in 3D space. 2DCDP stores the value of the color distance between matched pixels, thus this visual event error can easily be reduced

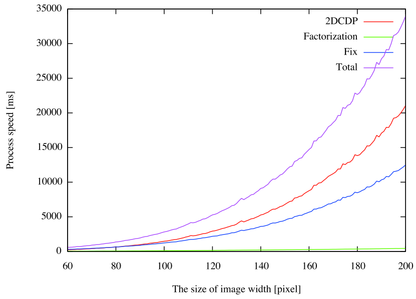


Figure 5. Processing time for a frame of a 3D surface: the vertical axis shows processing time and the horizontal axis shows image size

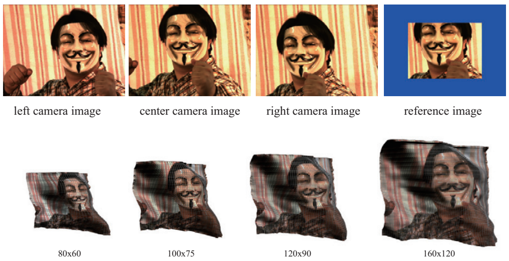


Figure 6. Top: Comparison between reconstructed 3D surface and another actual viewpoint. Below: Several other viewpoints of the 3D surface

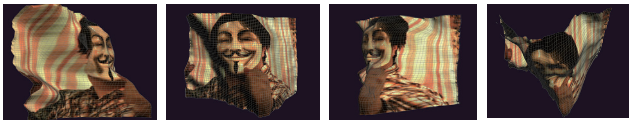


Figure 7. Comparison of precision of reconstructed images between input image vs size

from that information. In this experiment, we created 5 min of a 3D surface movie. In this time, there is no time to flip to another enantiomorphic surface, so our system has enough stability to extract a 3D surface continuously. We have also demonstrated real-time 3D reconstruction many times; sometimes the surface was turned over because it had such strong noise in z values for one pixel that the value was set to zero in the normalization process, but that case is very rare. This system can change camera position while recording 3D objects, and this function is not available in other image-based rendering techniques that do not use factorization. Thus, the proposed system has enough stability to capture 3D surfaces.

4 Conclusion

We have shown the feasibility of a model-based FTV capturing system using 2DCDP and the factorization method. 2DCDP has enough accuracy to determine a pixel trajectory by pixel-wise matching, and orthogonal factorization can reconstruct 3D surfaces.

At present, our system does not include any learning techniques. Their incorporation will enable the system to improve the precision of object shapes, and to use motion recognition from matching between frames for frame interpolation. For our next step, we plan to implement a faster 2DCDP algorithm using a coarse-to-fine strategy, install a GPGPU for speedup, and solve conflicts of occlusive areas in postprocessing. We should then be able to reconstruct objects precisely and add textures to high-definition images.

References

[1] K. Sohn, H. Kim and Y. Kim, “3-D video processing for 3-D TV,” Three-Dimensional Imaging, Visualization, and Display, Springer, pp. 251–278, 2009.

[2] T. Fujii and M. Tanimoto, “Free viewpoint TV system based on ray-space representation,” in Proceedings of SPIE, vol. 4864, p. 175, 2002.

[3] S. Barnard and M. Fischler, “Computational stereo,” ACM Computing Surveys, vol. 14, no. 4, pp. 553–572, 1982.

[4] E. Aganj, J-P. Pons, F. Segonne and R. Keriven, “Spatio-temporal shape from silhouette using four-dimensional Delaunay meshing,” ICCV 2007, pp. 1–8, 2007.

[5] C. Tomasi and T. Kanade, “Shape and motion from image streams under orthography: a factorization method,” IJCV, vol. 9, no. 2, pp. 137–154, 1992.

[6] C. Rother, “Multi-view reconstruction and camera recovery using a real or virtual reference plane,” PhD thesis, Computational Vision and Active Perception Laboratory, Kungl Teknisk H ogskolan, 2003.

[7] R. Zhang, P.S. Tsai, J.E. Cryer and M. Shah, “Shape-from-shading: a survey,” IEEE Trans. on PAMI, vol. 21, no. 8, pp. 690–706, 2002.

[8] R. Woodham, “Photometric method for determining surface orientation from multiple images,” Optical Engineering, vol. 19, no. 1, pp. 139–144, 1980.

[9] T. Matsuyama, X. Wu, T. Takai and S. Nobuhara, “Real-time 3D shape reconstruction, dynamic 3D mesh deformation, and high fidelity visualization for 3D video,” CVIU, vol. 96, no. 3, pp. 393–434, 2004.

[10] Y. Takai and T. Matsuyama, “High fidelity visualization algorithm and 3D editing system for 3D video,” Journal of the IIITE, vol. 56, no. 4, pp. 593–602, 2002 (in Japanese).

[11] D. Lowe, “Distinctive image features from scale invariant keypoints,” IJCV, vol. 60, no. 2, pp. 91–110, 2004.

[12] C. Tomasi and T. Kanade, “Detection and tracking of point features,” CMU Technical Report CMU-CS-91-132, April, 1991.

[13] Y. Yaguchi, K. Iseki and R. Oka, “Full pixel matching between images for non-linear registration of objects,” IPSJ Trans. on CVA, Vol. 2, pp. 1–14, 2010.

[14] Y. Yaguchi, K. Iseki and R. Oka, “3D object reconstruction using full pixel matching,” CAIP 2009, pp. 873–880, 2009.

[15] C. Tomasi and T. Kanade, “Factoring image sequences into shape and motion”, in IEEE Workshop on Visual Motion, pp. 21–28, 1991.

[16] Y. Yoshida, K. Yamaguchi, Y. Yaguchi, Y. Okuyama, K. Kuroda and R. Oka, “Accelerate two-dimensional continuous dynamic programming by memory reduction and parallel processing,” IADIS Applied Computing 2010, pp. 61–68, 2010.

[17] K. Kanatani and Y. Sugaya, “Complete recipe for factorization,” IEICE technical report. Neurocomputing, PRMU2003-118, pp. 12–24, 2003.

[18] K. Uchikawa and S. Shioiri, “Science of Sense and Perceptivity, Vision 2,” Asakura-Shoten, 2007 (in Japanese).