社団法人　情報処理学会　研究報告
IPSJ SIG Technical Reports

2004－AVM－47（2）
2004／12／9

幾何変換を用いた効率的なFTV画像圧縮

山本 健詞† 圓道 知博‡ 藤井 俊彰‡ 谷本 正幸‡

† ‡名古屋大学大学院 工学研究科 電子情報システム専攻 464-8603 名古屋市千種区不老町

E-mail †yamamoto@tanimoto.nuee.nagoya-u.ac.jp,

‡{yendo, fujii, tanimoto}@nuee.nagoya-u.ac.jp

あらまし 多数のカメラを用いたシステムとして自由視点テレビがある．自由視点テレビを構築する際の課題の一つに，カメラを理想通りに配置できないためにおこる画像圧縮率の低下が挙げられる．本論文では各カメラの画像を“理想の位置にカメラが配置されている場合の画像”に射影変換（マルチカメラ画像変換）し，画像圧縮率を改善させることを提案する．さらに，エピポーラ幾何を用いて射影変換行列を算出する手法を提案する．本手法により，ほぼすべてのビットレートで画質が3.0dB改善されることを実験で確認した．

キーワード 自由視点テレビ，マルチカメラ，射影変換，エピポーラ幾何

Projective transformation to align multi-camera images for efficient FTV compression

Kenji YAMAMOTO† Tomohiro ENDO‡ Toshiaki FUJII‡ and Masayuki TANIMOTO‡

† ‡Department of Electrical Engineering and Computer Science, Graduate school of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan

E-mail †yamamoto@tanimoto.nuee.nagoya-u.ac.jp,

‡{yendo, fujii, tanimoto}@nuee.nagoya-u.ac.jp

Abstract This paper introduces a preprocessing method for efficient compression of multi-camera images in FTV (Free viewpoint TeleVision). In every multi-camera system, it is difficult to precisely align cameras at the desired place. In this paper, we propose a preprocessing method that transforms the captured multi-camera images by projective transformation matrices to compensate for this misalignment. This transformation increases the efficiency of image compression because multi-camera images become more correlative. Furthermore, we propose a method for finding the projective transformation matrices automatically in consideration of the epipolar geometry. Experimental results show that PSNR gains up to 3.0 dB in comparison with the compression without any processing when MPEG-2 is used in the condition ranging from 1.6 to 3.0 Mbps.

Keyword FTV (Free viewpoint TeleVision), multi-camera, projective transformation, epipolar geometry

1. Introduction

Even though television is considered to be a good visual tool in a visual communication system, it still possesses some limitations. Many studies are aiming to improve and develop new television systems that overcome present limitations.

We proposed a new television system with multi-cameras and named it FTV [1]. The aim of this system is to overcome the problem that users cannot change their viewpoint position. With this system, users can freely control their viewpoint position for any dynamic real world scene (Figure 1). This feature introduces new applications ranging from next-generation television systems, to security and environmental monitoring systems[2].

To implement FTV, there are several considerable technical hurdles. Because of the large size of raw multi-camera images, compression is one of the challenges faced in building FTV. Therefore there have

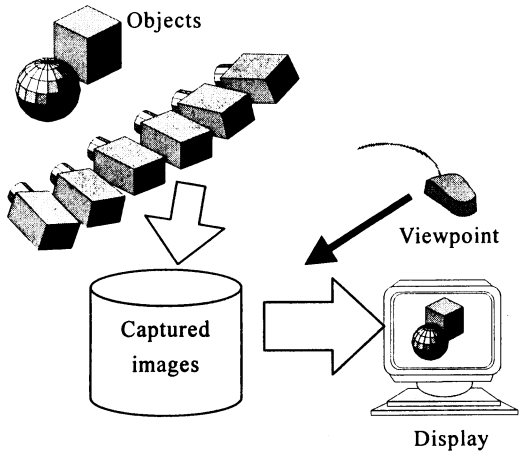

Figure 1: Free Viewpoint Television.

been some solutions proposed for compressing multi-camera images in[3][4][5], which use the correlation between each camera's images in addition to the traditional video encoding methods like MPEG[6][7].

In these methods[4][5], all cameras are assumed to be precisely aligned at the desired place; however, this is difficult in practice. As a result of this misalignment, the compression efficiency sometimes falls short of our expectations. Hence, we propose a method for transforming the captured images into images that could be captured at desired places as a preprocessing scheme before compression. This method can be used for every application of a multi-camera system like FTV, not only for the purpose of image compression, but also for that of image interpolation.

In Section 2, we describe the concept of multi-camera alignment. Section 3 shows one method of finding projective transformation matrices in consideration of the epipolar geometry, and Section 4 presents the experimental results. Finally, Section 5 concludes this paper and provides an outlook for future work.

2. Transformation of multi-camera images

We propose applying a projective transformation to the captured images before compression as a method of multi-camera alignment to achieve efficient FTV compression (Figure 2). The considerable fundamental distortions caused by camera misalignment are translation, rotation, zoom and keystone. However, it is possible to compensate for distortions by a projective transformation[8]; an additional advantage is that the computation time for a projective transformation is low.

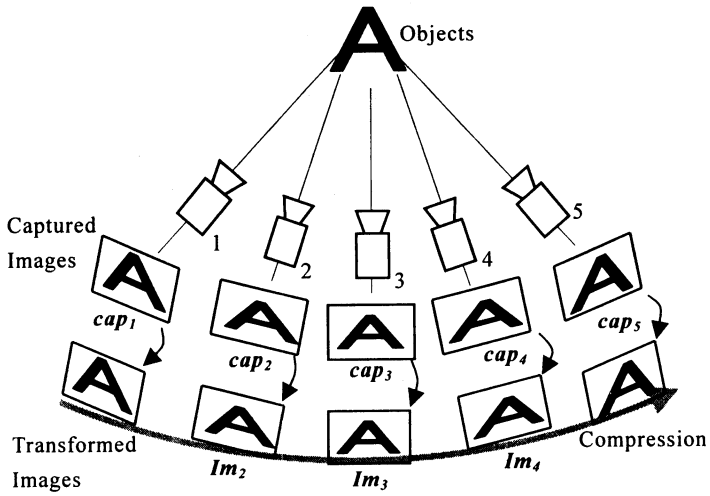

Figure 2: Multi-camera Alignment in the case of 5-camera system.

Hence we adopt projective transformation for multi-camera alignment.

The features of multi-camera alignment proposed in this paper are as follows:

- A projective transformation is applied to each captured image to compensate for the misalignment of each camera.
- A 3x3 matrix in homogeneous coordinates is used for the projective transformation.
- The projective transformation matrices are found for each camera. They are not found again over time, because the camera postures are generally not changed. If the postures are changed, they must be found again.

The projective transformation matrices are not unique; any projective transformation matrix can be used for efficient compression if it can reduce the transmission rate and improve PSNR. Furthermore it is desirable that the matrices be found automatically for all sorts of multi-camera image sequences.

3. Projective Transformation Matrices with Epipolar Geometry

In this section, we present one method of searching the projective transformation matrices in consideration of the epipolar geometry. By this method, the matrices are found automatically for all sorts of multi-camera image sequences.

In subsections 3.1 and 3.2 we show the preparations for 3.3, where an actual calculation method is shown.

3.1. Epipolar Geometry

What is the relationship between two images that are captured by two close cameras at the same time? The epipolar geometry answers this question[8][9].

Let C_{i-1} and C_i be the first and the second camera centers in 3-D (three-dimensional) space, respectively, and let x_{i-1} and x_i be the points in camera images, respectively, which correspond to X in 3-D space. As Figure 3 demonstrates, C_{i-1}, C_i, X, x_{i-1}, and x_i must be in the same plane, called the epipolar plane. This

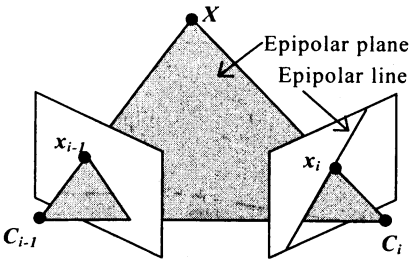

Figure 3: Epipolar Geometry.

constraint is called the epipolar geometry, and x_i is called the corresponding point to x_{i-1}.

According to this theory, if C_{i-1}, C_i and x_{i-1} are given but X and x_i are not, x_i is constrained to lie on a line called the epipolar line.

3.2. Evaluation of Transformed Image

A projective transformation is applied to multi-camera alignment, thus the transformed image is shown as

$$Im_i = H_i \circ Cap_i, \qquad (1)$$

where i = 1,2... is the camera number, Im_i is a transformed image, H_i is a 3x3 projective transformation matrix, Cap_i is a captured image, and operator (∘) denotes the projective transformation.

To evaluate Im_i, we consider the following two restrictions:

- Corresponding points should be on their epipolar lines.
- The average distance of corresponding points along an epipolar line should be similar on every pair of neighbor images. In other words, it should be roughly predictable if the cameras are aligned precisely.

We define a Lagrangian cost function J_i as

$$J_i = J_{iv} + \lambda J_{ih}, \quad J_{iv} = \sum_{S_i} \left(distance\_ep(x_i)\right)^2$$
$$J_{ih} = \sum_{S_i} \left|h_i - \hat{h}\right|^2, \quad h_i = (x_i - x_{i-1}) \cdot e_i, \qquad (2)$$

where $\lambda \geq 0$ is the Lagrange parameter, $distance\_ep(x_i)$ denotes the distance between x_i and

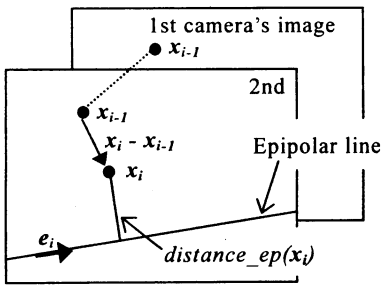

Figure 4: Evaluation of Transformed Image.

the epipolar line for x_i, S_i represents all corresponding points in the set, $\hat{h}$ is the expected value of h_i, x_{i-1} is a point on the first camera's image, x_i denotes the corresponding point to x_{i-1} on the second camera's image, operator $(\cdot)$ means the inner product, and e_i is the unit vector in parallel with that epipolar line (Figure 4). The epipolar line is calculated by the relation between first and second cameras' designated positions. If J_i is smaller, it means a better Im_i.

J_{iv} represents the restriction that corresponding points should be on their epipolar lines, while J_{ih} represents the restriction that the average distance between x_{i-1} and x_i should be similar on every pair of neighbor images.

This method searches for a corresponding point for each 16 x 16 block. A set of corresponding points S is produced by gathering these points. To search corresponding points this method uses block matching, which is popular in MPEG and so on.

In the case of many applications, a set of corresponding points is formed carefully because any error will seriously affect subsequent process. In this method, however, the error of corresponding points affects only J_{iv} and J_{ih}, which consist of all corresponding points. Because the number of errors is quite small in comparison with that of the correct one, the error does not affect J_i seriously, which is why this method employs block matching.

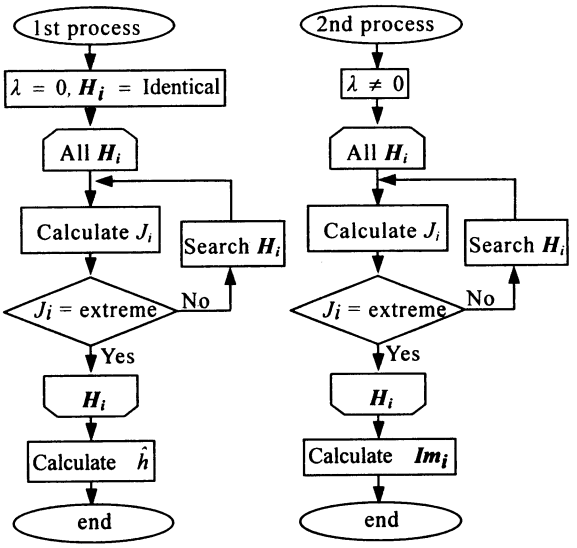

Figure 5: Calculation of the Matrices and Transformed Images.

3.3. Calculation Flowchart

We can divide the search procedure into two processes: the former process is predicting $\hat{h}$, and the latter is searching of H_i (Figure 5).

We assume that the cameras are aligned from left to right in order as shown in Figure 2. At each process this method finds all H_i and Im_i matrices in sequence, except the far-left ones; H_1 is an identity matrix and Im_1 is the same as Cap_1 because Im_{n-1} does not exist to calculate them. H_2, the matrix for the neighbor of the far-left camera, is produced at first through the use of Im_1, the captured image of the far-left camera. After transforming the image Im_2 through the use of H_2, H_3 is made with Im_2. In this manner H_2, Im_2, H_3, Im_3, H_4, Im_4 ... are calculated in this sequence (Figure 6).

Needless to say, it is no problem with searching Im_i from the far-right side or the center instead of from the far-left side, as mentioned above.

At each process this method uses the Steepest Descent method (the SD method), in which all elements in 3 x 3 matrix except that at row 3, column 3 are changed slightly and costs are calculated respectively to find the gradient direction. After that, the step is decided with

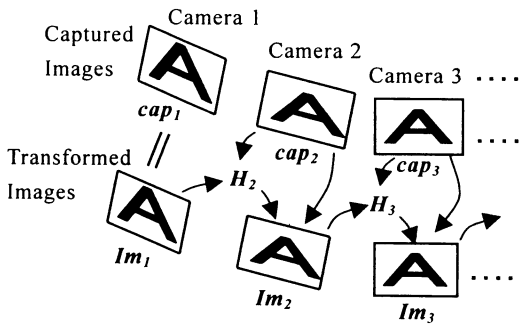

Figure 6: Calculation Diagram.

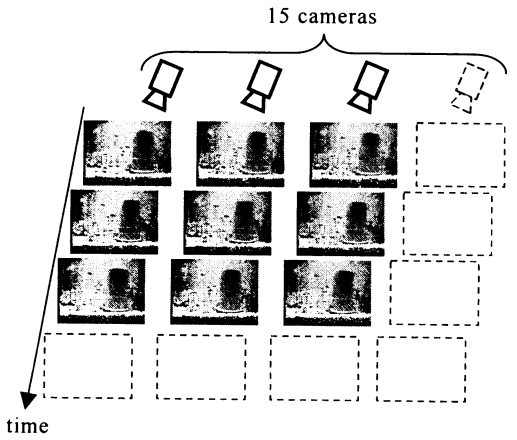

Figure 7: *Aquarium* Sequence.

respect to the result of the costs.

For the former process, this method employs the SD method to calculate $\boldsymbol{H}_i$ and $\boldsymbol{Im}_i$ from Camera 2 to the last camera in sequence. In the SD method, the cost function is J_i with $\lambda = 0$ and the initial $\boldsymbol{H}_i$ is an identity matrix. $\boldsymbol{S}_i$ and J_i are calculated through the use of $\boldsymbol{Im}_{i-1}$ and $\boldsymbol{H}_i \circ \boldsymbol{Cap}_i$. Here, $\boldsymbol{H}_i$ is modified iteratively until J_i reaches its local extremum. After all $\boldsymbol{Im}_i$ matrices are obtained, $\hat{h}$ can be calculated as the average of all h_i of $\boldsymbol{Im}_i$.

For the latter process, this method uses a calculation similar to that used in the first process to search the final $\boldsymbol{H}_i$ and the final $\boldsymbol{Im}_i$ matrices. The calculation in the second process differs from that of the first in the following ways: First, λ is not equal to zero. Second, the initial $\boldsymbol{H}_i$ is not necessary for identity matrices; one

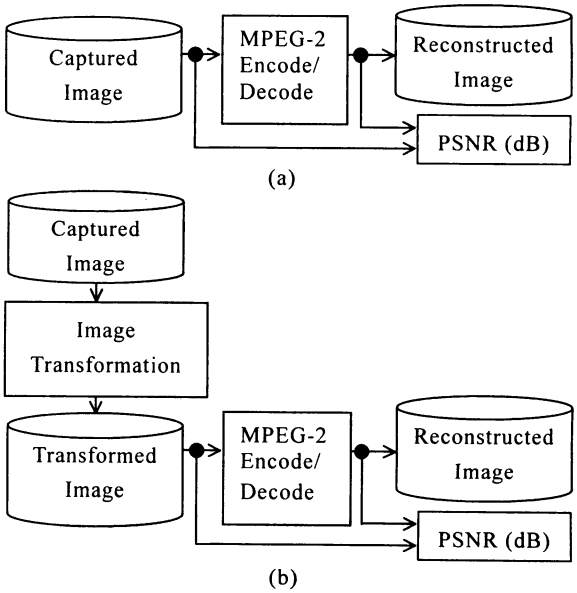

Figure 8: PSNR calculation between images. (a) In the case without preprocessing, (b) in the case with preprocessing.

option is to use the result of the first process. Third, $\hat{h}$ is the result of the first process. The $\boldsymbol{Im}_i$ matrices gained after the second process are the outputs of this method. Since they are more correlative than the captured images, image compression is expected to become more efficient.

4. Experiment

To confirm the efficiency of our approach, we conducted experiments with the proposed method mentioned in Section 3, using *Aquarium* (Figure 7) [10] as a sequence and MPEG-2_TM5 as an encoder[11]. All experiments were carried out on the luminance component only. The reconstruction quality of the compressed images with and without preprocessing is measured in terms of PSNR (peak signal to noise ratio) (Figure 8).

The specifications of *Aquarium* are as follows:

- 320 x 240 pixels per image.
- 15-camera system.

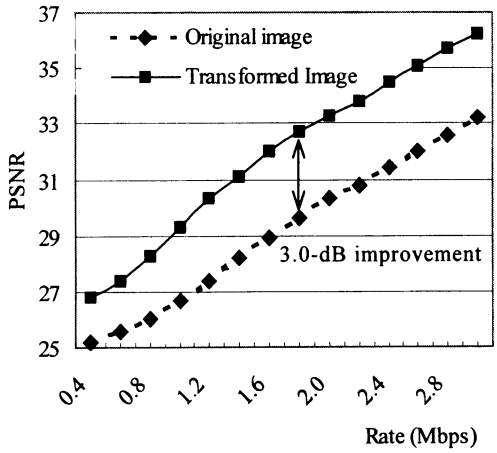

Figure 9: Experimental Results. The transformed image is more efficient than the original image. The improvement is 3.0 dB on the condition from 1.6 to 3.0 Mbps.

- 5° between each camera.
- 350 mm between the cameras and the aquarium.
- 4-mm focal length.
- 3.65 x 2.74-mm image detection area.

Because the epipolar lines for each 16x16 block are not parallel, it is necessary to calculate all of them. However, since they are approximately parallel in the case of *Aquarium*, we treat them as parallel here.

Figure 9 shows the result at the rate from 0.4 to 3.0 Mbps (bit per second). PSNR gains up to 3.0 dB in comparison with compression without any processing at the rate from 1.6 to 3.0 Mbps. At all rates PSNR improves just as it does in this figure.

5. Conclusion

We proposed applying a projective transformation to captured multi-camera images for efficient FTV compression. Furthermore we proposed one method that searches the projective matrices automatically in consideration of the epipolar geometry. Experimental results revealed that PSNR gained up to 3.0 dB in comparison with compression without any processing when MPEG-2 is used in the condition ranging from 1.6 to 3.0 Mbps.

In contrast, the SD method spends too much time searching the projective transformation matrices and has the risk of local minima. Our future works include first, further development to save searching time, second, confirmation of whether local minima exist, and third, research to avoid the local minima if they do exist.

REFERENCES

[1] T. Fujii and M. Tanimoto, "Free-viewpoint Television based on the Ray-Space representation," Proc. SPIE ITCom 2002, pp. 175-189, Boston, Massachusetts USA, August 2002.

[2] M. P. Tehrani, T. Fujii, and M. Tanimoto, "Optimization of Multiuser Camera Sensor Network for Arbitrary View Generation," Proc. 7th International Workshop on Advanced Image Technology (IWAIT2004), pp. 311-315, Singapore, January 2004.

[3] M. Magnor and B. Girod, "Data Compression for Light-Field Rendering," IEEE Transactions on Circuits and Systems for Video Technology, vol. 10, No. 3, pp. 338-343, April 2000.

[4] P. Na. Bangchang, T. Fujii, and M. Tanimoto, "Ray-Space Data Compression Using Spatial and Temporal Disparity Compensation," Proc. 7th International Workshop on Advanced Image Technology (IWAIT2004), pp. 311-315, Singapore, January 2004.

[5] Y. Hayashi, T. Fujii, and M. Tanimoto, "Coding for Ray Space based on SPECK," Proc. 23rd Picture Coding Symposium (PCS2003), pp. 339-342, Saint Malo, France, April 2003.

[6] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, and Ajay Luthra, "Overview of the H.264 / AVCVideo Coding Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, No. 7, pp. 560-576, July 2003.

[7] ITU-T Recommendation H.264 "Advanced Video Coding for Generic Audiovisual Services," May 2003.

[8] Richard Hartley and Andrew Zisserman, Multiple View Geometry in computer vision, Cambridge University Press, Cambridge, 2000.

[9] Charles Loop and Zhengyou Zhang, "Computing rectifying homographies for stereo vision," Proc. IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 125-131, Fort Collins, Colorado USA, July 1999.

[10] http://www.tanimoto.nuee.nagoya-u.ac.jp/

[11] http://isotc.iso.ch/livelink/livelink/fetch/2000/2489/Ittf_Home/PubliclyAvailableStandards.htm