RAY-SPACE BASED CAMERA SPACING CORRECTION VIA CONVEX OPTIMIZATION

Lingfeng XU, Oscar C. AU, Wenxiu SUN, Jiali LI, Wei HU, Rui MA,

The Hong Kong University of Science and Technology, Hong Kong
{lingfengxu, eeau, eeshine, jiali, huwei, rmaaa}@ust.hk

ABSTRACT

3D technologies such like three-dimensional television and free viewpoint television have caught enormous attentions in the consumer market recently. However, because of the inaccurate camera configuration and environmental constraint, there are errors in the assumed equally-spaced camera intervals. In this paper, we propose a novel camera spacing correction algorithm to detect the spacing errors among the multiple cameras by making the corresponding points co-linear in the epipolar plane images. Experimental results show that the proposed algorithms are robust and can achieve good performance even if the corresponding pixels are not well detected. Meanwhile, our algorithm can be solved by convex optimization with an extremely low complexity.

Index Terms— multi-view images, ray space interpolation, camera spacing correction, epipolar plane image

1. INTRODUCTION

Multi-view technologies are becoming more and more popular in recent years. It has many applications such like three-dimensional television (3DTV) [1] and free viewpoint television (FTV) [2], which provide people the opportunity to get the 3D perception from different viewpoints. The multi-view images are captured by the camera array which usually contains several or dozens of cameras. Since cameras cannot be placed continuously in space because of its bulky size, only a finite number of views (called real views) can be captured. Many free viewpoint generating algorithms are proposed to synthesize the virtual views inbetween the real views. And they can be classified into image domain based algorithms[3][4], ray-space based algorithms[5], surface light field based algorithms[6], etc.

Most of those algorithms take the multiple real views captured by the camera array as input images. And the most common used camera configuration is the linear arrangement, in which the cameras are placed along a straight line in parallel and equally spaced. The focal length and other intrinsic parameters are the same for all of the cameras under the assumption of pinhole camera model. In this idea case, the view captured are parallel with the corresponding epipolar line along the same horizontal line [7]. For the same object, it only has horizontal movement (disparity) among different views.

However, in the real case, both of the intrinsic and extrinsic parameters contain some errors due to the imperfect camera model and environmental constraint. Therefore pre-processing is needed to correct those errors. Existing pre-processing algorithms include camera calibration [8], radial distortion correction [9], image rectification [10], and color correction [11]. The former two are related with the intrinsic parameters of each single camera while the latter two are about the "relationship" between different cameras. The real cameras are not exactly within the assumption of pinhole camera model. In this case, camera calibration is performed to find the exact intrinsic camera parameters while radial distortion correction is applied to compensate the geometric distortion from the lens. As for the extrinsic parameters, the cameras are not exactly in parallel in the real world because of the small rotations and displacement existed. Image rectification is then utilized to rectify the multiple view images such that there is only horizontal motion in adjacent views. And the color difference for the same object in the multi-view images are corrected by color correction algorithms.

All of those pre-processing algorithms make the assumption that the cameras are equally spaced, which can simplify the problems of multi-view processing, such like ray-spaced interpolations [5], multi-view depth estimation [12], etc. However, in the real situation, the cameras may not be equally spaced due to the inaccurate camera configuration or the environmental constraint. Inaccurate camera spacing can further affect the depth estimation and view interpolation. To solve this problem, we propose an algorithm named camera spacing correction, which can derive the exact camera intervals by finding the corresponding pixels within different views. The proposed camera spacing correction is implemented in the ray-space of the epipolar plane images to make the corresponding pixels co-linear while keeping a small displacement from the initial camera positions. The rest of the paper is organized as follows: in Section 2, we review some basic properties of camera model and ray-space representation for multi-view images. Proposed camera spacing correction algorithm will be presented in Section 3 and experimental results are shown in Section 4, followed by conclusion in Section 5.

2. REVIEW OF RAY-SPACE REPRESENTATION

In ray-space representation, each ray in the 3D world is represented by a corresponding point in the ray-space domain. The propagation path of a ray going through space can be uniquely parameterized by the location of a point and a direction [13]. For the linear camera arrangement, people proposed the idea of epipolar plane image (EPI) in the ray-space domain. An EPI is constructed by collating the horizontal rows with the same y coordinate in the multi-view images one by one. If the multi-view images have H rows each, the number of EPIs constructed will be H. An example is shown in Fig. 1, in which the multi-view images are captured by cameras which are equally-spaced along the baseline and indexed accordingly by an integer u. In Fig. 1, the EPI shown is generated by collating the horizontal rows with $y = 287$ from 101 views of the first frame of the "Xmas" multi-view test sequences. The top row of EPI corresponds to the leftmost view ($u = 0$) while the bottom row corresponds to the rightmost view ($u = 100$). An important property of EPI is that if the camera interval (the distance between adjacent cameras) is constant, the pixels corresponding to the same scene point in different multi-view images would form a straight line in the EPI with a slope

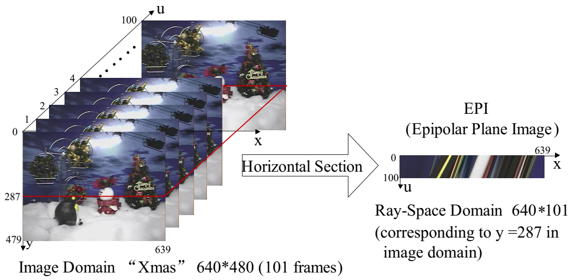


Fig. 1. EPI generation for test sequence “Xmas”

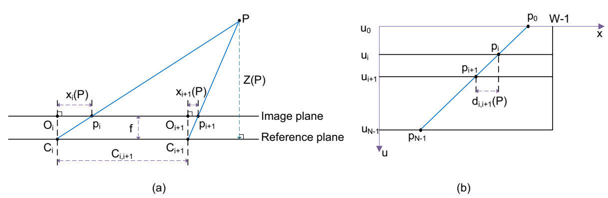


Fig. 2. (a)Pin-hole camera model, (b)The corresponding EPI

proportional to the depth of that scene point. The depth is the perpendicular distance between the scene point and the reference plane. This relationship can be easily obtained from the projective camera model as shown in Fig. 2.

In Fig. 2(a), P is a scene point of interest in the world. All the cameras are placed equally along a straight line in the reference plane, with C_i being the optical center of camera i, and O_i being the intersection point between its optical axis and the image plane which is parallel to the reference plane and is at a distance of f from the reference plane, for $i = 0, 1, ..., N-1$. $C_{i,i+1}$ is the camera spacing between C_i and C_{i+1}. Here p_i and p_{i+1} are the two pixels corresponding to P in two adjacent multi-view images ($u = i$ and $u = i+1$). $Z(P)$ is the perpendicular distance between P and the reference plane. Based on triangle similarity, we can obtain:

$$\frac{C_{i,i+1} - (x_i(P) - x_{i+1}(P))}{C_{i,i+1}} = \frac{Z(P) - f}{Z(P)} \tag{1}$$

which can be simplified to give

$$d_{i,i+1}(P) = x_i(P) - x_{i+1}(P) = \frac{f \cdot C_{i,i+1}}{Z(P)} \tag{2}$$

And $d_{i,i+1}(P)$ is called the *disparity* (or relative displacement in local coordinate) between the pixels p_i and p_{i+1}. As $Z(P)$ and focal length f are fixed, the disparity $d_{i,i+1}(P)$ depends only on the camera interval $C_{i,i+1}$. As the cameras are equally spaced, the $C_{i,i+1} = C$ is constant for all i and thus the disparity $d_{i,i+1}(P) = d(P)$ is constant for all i. In Fig. 2(b), $u_{i,i+1}$ is the distance between neighboring rows in the u direction, $i = 0, 1, 2, ..., N-2$. Thus the slope between p_i and p_{i+1} is

$$s_{i,i+1}(P) = \frac{u_{i,i+1}}{d_{i,i+1}(P)} \tag{3}$$

Since we pack the rows one by one in the EPI, the distance between adjacent rows $u_{i,i+1}$ is constant (actually, $u_{i,i+1} = 1$, $i = 0, 1, 2, ..., N-2$). In this case, the slopes of the line segments in any two adjacent rows ($s_{i,i+1}(P)$) are the same. Therefore, pixels corresponding to any scene point P from different views form a straight line in the EPI. Considering that all pixels in any image correspond to some scene points, an EPI contains multiple straight lines each of which corresponds to one scene point.

3. PROPOSED RAY-SPACE BASED CAMERA SPACING CORRECTION

The cameras are assumed to be equally spaced with constant distance of C so far. But in the real situations, the camera distance may not be the same due to inaccurate camera configuration or environmental constraint, which leads to the unequally space or wrong view order. When this happens, pixels corresponding to the same scene point will not form a straight line in the EPI, as shown in the example of Fig. 3(a). In this section, we propose a camera spacing correction algorithm to ensure the planes are ordered accurately and the unequal camera spacing is detected to preserve the straight line structures in the EPI. The other pre-processing algorithms we mentioned in Section 1 are supposed to be performed before our algorithm such that all the cameras have the same focal length f and are placed along a horizontal baseline and all the optical axes are perpendicular to the reference plane. Notice that our method does not require the calibration parameters of the cameras, such as f, C_i, and camera intrinsic parameters. We assume that the locations of the corresponding pixels in each of the multi-view images are known for at least one scene point P. In our algorithm, the corresponding pixels can be marked manually or identified using some pixel matching algorithms.

As discussed in Section 2, when the cameras are equally spaced without any error, the pixels corresponding to the same scene point will form a straight line in the EPI. However, in the real situation, the camera space contains some errors, while the corresponding EPIs are still equally spaced at a distance of 1 between adjacent rows, as shown in Fig. 3(a). In this case, the line structure are not preserved. Note that in the example of Fig. 3(a), view 2 and 3 have a wrong order and all of the views are not equally spaced. Let u_i be the “true” location of row i in the EPI, the vector $\mathbf{u} = [u_0, u_1, ..., u_{N-1}]^T$ is defined to correct the unequal camera spacing. If the cameras are equally spaced, then $u_i = i$ which is called the nominal camera position. Let $\mathbf{n}_u = [0, 1, ..., N-1]^T$ as the vector of the nominal positions of $\mathbf{u}$. In the real case, if the cameras are unequally spaced, u_i can be any real number. Let $\Delta u_i = u_i - i$ which is the deviation of row i from its nominal location , $\Delta\mathbf{u} = [\Delta u_0, \Delta u_1, ..., \Delta u_{N-1}]^T$. Actually we can move each row i by Δu_i to form a Corrected EPI (CEPI) as shown in Fig. 3(b), the corresponding points in the CEPI (in our example: $p_1, p_2, ..., p_5$) will line up and form a straight line. The camera spacing correction problem is now reduced to finding the “optimal” $\Delta\mathbf{u}$ with minimum displacement such that the corresponding points are co-linear. The optimal $\hat{\mathbf{u}}$ can be derived by solving the following convex optimization problem:

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \|\mathbf{u} - \mathbf{n}_u\|_{l_2}^2 \tag{4}$$

$$\text{subject to} \quad \mathbf{u} = \mathbf{X} \cdot \mathbf{a}$$

where $\mathbf{x} = [x_0, x_1, .., x_{N-1}]^T$, in which x_i is the horizontal coordinate of pixel p_i. $\mathbf{X} = [\mathbf{x}\ \mathbf{I}_N]$, and $\mathbf{I}_N$ is a vector of size $N \times 1$ with all elements being 1. And $\mathbf{a} = [\alpha, \beta]^T$, where α and β are the slope and y-intercept of the feature line. Note that the constraint here are the affine formation of a line function and $\mathbf{a}$ is also a variable in the optimization. The constraint of this optimization problem guarantees the corresponding points (in our example: $p_1, p_2, ..., p_5$) of a

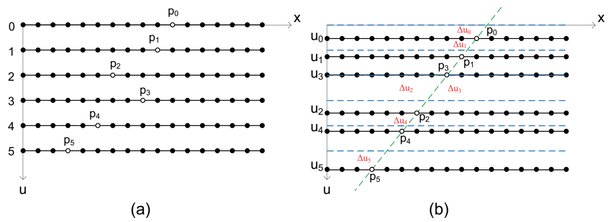


Fig. 3. Ray-space based spacing correction: (a)Equally-spaced EPI with non-equally-spaced cameras, (b)Space-corrected EPI

scene point, say P_0, are co-linear. We note that many $\mathbf{u}$ can make the points align into a straight line, because if $\mathbf{u}$ is a solution such that $\mathbf{u} = \alpha\mathbf{x} + \beta\mathbf{I}_N$, then $\mathbf{u} + \gamma\mathbf{I}_N$ will also be a solution because $\mathbf{u} + \gamma\mathbf{I}_N = \alpha\mathbf{x} + (\beta + \gamma)\mathbf{I}_N$. In this paper, we want to find the $\mathbf{u}$ with minimum deviation, $\|\mathbf{u} - \mathbf{n}_u\|_{l_2}^2$, which is unique since the cost function is convex.

Theorem 1: As long as the corresponding points of one scene point are co-linear in the CEPI, for the corresponding points of any other scene points visible will also be co-linear.

Proof. From Eqn. (2) and Eqn. (3), we can compute the slope of the line segment connecting p_i and p_{i+1} in the CEPI for the points corresponding to any P by

$$s_{i,i+1}(P) = \frac{u_{i,i+1} \cdot Z(P)}{f \cdot C_{i,i+1}}$$

Suppose we make the corresponding points of one scene point, say "P_0", co-linear. Thus, there exists a constant κ such that $\kappa = u_{i,i+1}/C_{i,i+1}$ for all i. Since κ is only related with camera distance and row interval in the CEPI, for any other scene point P', $s_{i,i+1}(P') = \kappa\, Z(P')/f$ which is the same for all i. In other words, the corresponding points for any scene point are co-linear as long as one set of points are co-linear. □

As explained in ***Theorem 1***, just one set of points corresponding to a scene point are enough for camera spacing correction, which can be solved directly by Eqn. 4. However, in reality, the values of x_i may contain small errors due to the finite precision of locations (integer precision only) or the marking error of corresponding points locations. There is no need to force such set of points to form a straight line. Instead, we only need the set of points corresponding to a scene point to be close to some straight line. In order to increase the robustness of our correction algorithm, we assume M available scene points leading to M set of points in the EPIs are marked. The problem is relaxed into

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \|\mathbf{u} - \mathbf{n}_u\|_{l_2}^2 \tag{5}$$

$$\text{subject to} \quad \frac{1}{M}\sum_{k=0}^{M-1} \|\mathbf{u} - \mathbf{X}_k \cdot \mathbf{a}_k\|_{l_2}^2 < \varepsilon$$

where α_k and β_k are the slope and y-intercept of the k^{th} line in the EPI. $\mathbf{a}_k = [\alpha_k, \beta_k]^T$ and $\mathbf{X}_k$ are the corresponding quantities. Note that $\mathbf{u}$ is the same for all k. By using the KKT conditions [14] of the convex problem, Eqn. 5 can be transformed into minimizing the following unconstrained cost function:

$$E(\mathbf{u}, \mathbf{a}_0, \mathbf{a}_1, \cdots, \mathbf{a}_{M-1}) = \|\mathbf{u} - \mathbf{n}_u\|_{l_2}^2 + \frac{\lambda}{M}\sum_{k=1}^{M} \|\mathbf{u} - \mathbf{X}_k\mathbf{a}_k\|_{l_2}^2 \tag{6}$$

Where λ is the Lagrange parameter related with ε. For any $\mathbf{u}$, we take the derivative of the cost function with respect to $\mathbf{a}_k$

$$\begin{aligned}
\frac{\partial E}{\partial \mathbf{a}_k} &= \frac{\partial}{\partial \mathbf{a}_k}\left[\frac{\lambda}{M}(\mathbf{u} - \mathbf{X}_k\mathbf{a}_k)^T(\mathbf{u} - \mathbf{X}_k\mathbf{a}_k)\right] \\
&= \frac{\lambda}{M}\frac{\partial}{\partial \mathbf{a}_k}(\mathbf{u}^T\mathbf{u} - \mathbf{a}_k^T\mathbf{X}_k^T\mathbf{u} - \mathbf{u}^T\mathbf{X}_k\mathbf{a}_k + \mathbf{a}_k^T\mathbf{X}_k^T\mathbf{X}_k\mathbf{a}_k) \\
&= -\frac{\lambda}{2M}(\mathbf{X}_k^T\mathbf{u} - \mathbf{X}_k^T\mathbf{X}_k\mathbf{a}_k)
\end{aligned}$$

By setting it to zero we obtain $\hat{\mathbf{a}}_k = (\mathbf{X}_k^T\mathbf{X}_k)^{-1}\mathbf{X}_k^T\mathbf{u}$. Substituting it into Eqn. (6), we derive

$$E(\mathbf{u}, \hat{\mathbf{a}}_0, \hat{\mathbf{a}}_1, \cdots, \hat{\mathbf{a}}_{M-1}) = \|\mathbf{u} - \mathbf{n}_u\|_{l_2}^2 + \frac{\lambda}{M}\sum_{k=1}^{M} \|\mathbf{u} - \mathbf{A}_k\mathbf{u}\|_{l_2}^2 \tag{7}$$

where $\mathbf{A}_k = \mathbf{X}_k(\mathbf{X}_k^T\mathbf{X}_k)^{-1}\mathbf{X}_k^T$ which is an $N \times N$ matrix. By taking the derivative of Eqn. (7) with respect to $\mathbf{u}$ and setting it to zero, we can derive the close-form solution:

$$\hat{\mathbf{u}} = [(\lambda + 1)\mathbf{I}_{N\times N} - \frac{\lambda}{M}\sum_{k=1}^{M}\mathbf{A}_k]^{-1}\mathbf{n}_u \tag{8}$$

Here $\mathbf{I}_{N\times N}$ is an identity matrix of size $N \times N$.

4. EXPERIMENTAL RESULTS

The proposed spacing correction algorithm is simulated on the test sequence "Xmas" which contains 101 views, each of size 640×480. In our experiment, 11 unequally spaced views are chosen with the ground truth (original) camera locations $\mathbf{l}_o$ = [1, 16, 18, 35, 37, 62, 51, 75, 80, 95, 99] which are used to form 480 EPIs each of 640x11. In our experiment, we assume that the "real" unequal spacing are unknown and the cameras are equally spaced. Therefore in the initial stage, in each EPI, row i corresponds to a nominal camera location $(N-1)i$, for $i = 0, 1, ..., 10$. Note that the incorrect camera order are simulated by placing View 62 in front of View 51.

To order to measure the accuracy of the spacing correction algorithm, the Camera Spacing Error (CSE) is defined for the corrected row position $\mathbf{u}$ as

$$CSE(\mathbf{u}) = \|\mathbf{u} - \mathbf{u}_o\|_{l_1} \tag{9}$$

where $\mathbf{u}_o = \mathbf{l}_o/(N-1)$ is the normalized ground truth (original) camera location. Obviously, a small CSE suggests a good camera spacing correction algorithm.

Fig. 4 shows the relationship of CSE corresponding to the scene points number M and the weighting coefficient λ of our proposed algorithm in Eqn. (6). Here $M = 1, 2, ..., 10$ and $\lambda = 1, 2, ..., 50$. Note that the curve is the average behavior of lots of experiments. We apply the algorithm to two EPIs corresponding to Row 287 and Row 319 of the multi-view images. A total of 10 sets of points corresponding to 10 scene points are marked manually in our experiment. When $M = 1$, there are 10 ways to choose 1 set out of the 10. The CSE are computed for each case with the average shown in Fig. 4. For $M > 1$, all the possible combinations of M sets of points are used to compute the average CSE. For each M, CSE is computed with respect to λ which is from 1 to 50 with step 1.

As shown in Fig. 4, CSE appears to decrease as M increases, as expected since more data tend to give more reliable estimations. For any M, small λ tends to generate large CSE, because small λ does not emphasize enough the collinearity of the corresponding points.

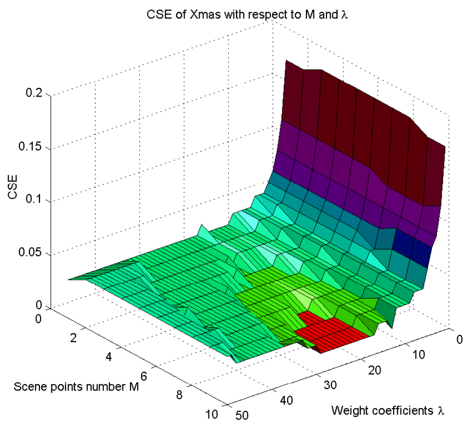


Fig. 4. CSE of Xmas with respect to M and λ

On the other hand, if λ is too large (more than 40), we emphasize too much on the collinearity of the corresponding pixels, which is not reasonable in the case that errors exist from the quantization errors or marking errors for those pixels, the CSE will also become large when we force them co-linear. The combination of $M = 10$ and $\lambda = 25$ gives the minimum CSE in our experiment. And we observe that even the combination of $M = 2$ (low complexity) and $\lambda = 25$ works quite well with a very small CSE.

Fig. 5(a) shows the EPI corresponding to Row 287 with equally spaced rows before camera spacing correction. We apply our proposed camera spacing correction with $M = 2$ and $\lambda = 25$. The two sets of points corresponding to $M = 2$ are marked manually as bright green points, which do not form straight lines before spacing correction. By using the two marked sets of points, the proposed camera spacing correction algorithm is applied to the EPI with the result shown in Fig. 5(b). The optimal corrected row positions are

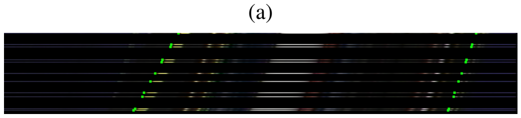
$$\mathbf{u}^* = [0.1,\ 1.6,\ 1.8,\ 3.5,\ 3.7,\ 6.1,\ 5.1,\ 7.4,\ 8.0,\ 9.5,\ 9.8]$$

with a small CSE of 0.027. The $\mathbf{u}^*$ corresponds to real camera locations [1, 16, 18, 35, 37, 61, 51, 74, 80, 95, 98], which are very similar to the original locations $\mathbf{l}_o$. As expected, each sets of points form straight lines in the CEPI.

5. CONCLUSION

In this paper, to solve the problem that cameras are not equally spaced, a novel ray-space based camera spacing correction algorithm is proposed to find the real camera intervals by adjusting the EPIs such that pixels corresponding to the same scene point form a straight line. Experimental results suggest that proposed method can achieve good performance, even in the case that very few corresponding points sets are provided.

(a)



(b)

Fig. 5. Results of ray-space based spacing correction. (a) EPI with incorrect camera order and distance, (b) EPI with camera spacing correction.

6. ACKNOWLEDGEMENT

This work has been supported in part by the Research Grants Council (RGC) of the Hong Kong Special Administrative Region, China. (GRF Project no. 610210).

References

[1] C. Zhang, "Multiview imaging and 3DTV," in *IEEE Signal Processing Magazine*, 2007.

[2] M. Tanimoto, "Overview of free viewpoint television," *Signal Processing: Image Communication*, 2006.

[3] W. SUN, L. XU, O. C. AU, S. CHUI, and C. KWOK, "An overview of free viewpoint depth-image-based rendering (DIBR)," in *APSIPA Annual Summit and Conference*, 2010.

[4] L. XU, O. C. AU, W. SUN, Y. LI, and J. LI, "Hybrid plane fitting for depth estimation," in *APSIPA Annual Summit and Conference*, 2012.

[5] L. Hou, O.C. Au, M. Ma, L. Guo, and X. Fan, "Free view generation in ray-space via the radon transform," in *International Conference on Neural Networks and Signal Processing*, 2008.

[6] W.C. Chen, J.Y. Bouguet, M.H. Chu, and R. Grzeszczuk, "Light field mapping: efficient representation and hardware rendering of surface light fields," *ACM Trans. Graphic.*, 2002.

[7] L. Xu, O.C. Au, W. Sun, Y. Li, S. Chui, and C. Kwok, "Image rectification for single camera stereo system," in *Image Processing. IEEE International Conference on*, 2011.

[8] Z. Zhang, "A flexible new technique for camera calibration," in *Pattern Analysis and Machine Intelligence*, 2000.

[9] R. Hartley and S.B. Kang, "Parameter-free radial distortion correction with center of distortion estimation," in *Pattern Analysis and Machine Intelligence*, 2007.

[10] DV Papadimitriou and TJ Dennis, "Epipolar line estimation and rectification for stereo image pairs," in *Image Processing, IEEE Transactions on*, 1996.

[11] K. Yamamoto, M. Kitahara, H. Kimata, etc, "Multiview video coding using view interpolation and color correction," in *Circuits and Systems for Video Technology*, 2007.

[12] T. Masayuki, F. Toshiaki, and S. Kazuyoshi, "Depth estimation reference software (DERS) with image segmentation and block matching," in *ISO/IEC JTC1/SC29/WG11, MPEG2008/M16092*, 2009.

[13] R. Bolles, H. Baker, and D. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," *International Journal of Computer Vision*, 1987.

[14] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *Proceedings of 2nd Berkeley Symposium. Berkeley: University of California Press*, 1951.