



<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper presented at *2017 International Conference on 3D Immersion (IC3D 2017), Brussels, Belgium, 11th-12th December 2017.*

Citation for the original published paper:

Dima, E., Sjöström, M., Olsson, R. (2017)

Modeling Depth Uncertainty of Desynchronized Multi-Camera Systems.

In: IEEE Signal Processing Society

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:miun:diva-31841>

This paper is published in the open archive of Mid Sweden University DIVA <http://miun.diva-portal.org> to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

Dima, E.; Sjöström, M.; Olsson, R., "Modeling Depth Uncertainty of Desynchronized Multi-Camera Systems", in International Conference on 3D Immersion (IC3D), 11-12 December 2017, Brussels, Belgium.

©2017 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

MODELING DEPTH UNCERTAINTY OF DESYNCHRONIZED MULTI-CAMERA SYSTEMS

Elijs Dima, Mårten Sjöström, Roger Olsson

Dept. of Information Systems and Technologies, Mid Sweden University
SE-851 70 Sundsvall, Sweden

ABSTRACT

Accurately recording motion from multiple perspectives is relevant for recording and processing immersive multi-media and virtual reality content. However, synchronization errors between multiple cameras limit the precision of scene depth reconstruction and rendering. In order to quantify this limit, a relation between camera de-synchronization, camera parameters, and scene element motion has to be identified. In this paper, a parametric ray model describing depth uncertainty is derived and adapted for the pinhole camera model. A two-camera scenario is simulated to investigate the model behavior and how camera synchronization delay, scene element speed, and camera positions affect the system's depth uncertainty. Results reveal a linear relation between synchronization error, element speed, and depth uncertainty. View convergence is shown to affect mean depth uncertainty up to a factor of 10. Results also show that depth uncertainty must be assessed on the full set of camera rays instead of a central subset.

Index Terms— Camera synchronization, Synchronization error, Depth estimation error, Multi-camera system

1. INTRODUCTION

Using multiple cameras to record video is increasingly relevant for motion capture [1], multi-media production and post-processing [2, 3], surveillance and computer vision [4], 3D-TV applications [5], 360-degree video [6], and virtual reality video [7]. In all such applications, accurate depth and 3D information is desirable, as it increases the objective and subjective quality of any depth-dependent composition, rendering and post-processing results.

Multi-camera capture is complicated due to synchronization concerns. Camera sensor shutters can be unsynchronized (e.g. in mobile, low-cost or drone-mounted multi-camera networks, or systems using Time-of-Flight depth sensors), or be synchronized only up to a certain precision via nearest-frame video sequence alignment [1]. This lack of accurate synchronization leads to an imprecise recording and reconstruction of moving element positions in 3D space. The importance of synchronization has been noted [1, 2, 5], however, this 3D position imprecision - which we call depth uncertainty (Δd) - has not been modeled as a direct consequence of the camera system properties and applications.

Although numerous single-camera models exist (surveyed in [8, 9, 10]), there are few models describing multi-camera systems [3, 11, 12, 13, 14, 15, 16]. These models describe the spatial relations between cameras, but do not treat camera synchronization error (desynchronization) as a system parameter affecting depth uncertainty. Parametrizing depth uncertainty from multi-camera system and scene parameters enables determining whether a given camera system can retain accurate scene depth while recording scenes with moving elements.

The purpose of this paper is to investigate how desynchronization and convergence between cameras influence the system's depth uncertainty. We introduce a parametric model of depth uncertainty, and use it to answer the following two research questions:

- 1) *How do changes in camera-to-camera desynchronization (Δt) and maximum scene element speed (v) influence overall Δd ?*
- 2) *Do parallel-oriented cameras have lower overall Δd compared to inward-rotated (toed-in) cameras?*

Because the model's computational cost scales with the number of rays in the system, we also address a third question:

- 3) *Does calculating Δd for all rays of one camera and only the principal ray of another camera produce results equivalent to calculating Δd for all rays of both cameras?*

The novelties of this paper are: 1) we introduce a parametric model that relates depth uncertainties to synchronization errors and camera properties, 2) we combine our ray-based model with the pinhole camera model, and 3) we use our model to show how camera orientation and desynchronization affect the overall depth uncertainty of a multi-camera system. The article is organized as follows: Section 2 discusses camera synchronization, and defines depth uncertainty. Section 3 introduces our model and its extension with the pinhole camera model. Section 4 describes simulation scenarios. Section 5 contains simulation results that address our research questions. We present our conclusions in section 6.

2. SYNCHRONIZATION AND DEPTH UNCERTAINTY

The extent of desynchronization in multi-camera systems varies based on whether hardware trigger synchronization or software-based synchronization [17] is used. The ability to use hardware synchronization is limited by cost [18], but provides much higher synchronization accuracy than any other method [19]. Several multi-camera systems without hardware synchronization have had rendering artifacts caused by lack of accurate synchronization [20, 21, 22, 23].

Synchronization of multiple cameras is commonly treated as a video alignment problem. Sequences are aligned via image feature correspondences [18, 24, 25, 26, 27], intensity-matching [28, 29], or even time-stamp based stream buffering [17, 19]. However, all these approaches estimate synchronization error and align videos to the nearest-frame at the cost of post-processing computation time. None of these papers have investigated exactly how large of a synchronization improvement would be necessary to record sufficiently accurate scene depth information.

Depth uncertainty exists whenever moving elements in the captured scene are recorded with imperfectly synchronized cameras. Each recorded frame represents a point in time, where a moving element's position is well-known in two (transverse) directions relative to the camera. The position in the third direction - depth - is only defined by triangulation from frames of at least one other camera. If

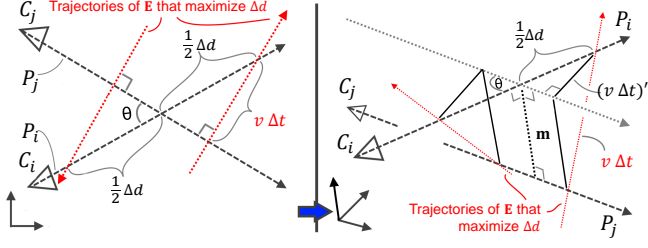


Fig. 1. Depth uncertainty Δd in desynchronized cameras i, j at positions C_i, C_j . Left: Maximum Δd of moving element \mathbf{E} , captured in different positions by co-planar camera rays P_i, P_j at time instants separated by Δt . Maximum movement speed is v . Movement direction of \mathbf{E} is unknown. θ = is the smallest angle between rays P_i, P_j . Right: Same scenario in 3D space with P_i, P_j at different planes and \mathbf{m} as the shortest path (distance) from P_i to P_j .

the frames are captured at different times, the element's distance to each camera is uncertain, since each camera may have captured the element at a different position. Depth uncertainty describes the maximum error between the true positions of scene elements at the time of capture, and the positions recovered without considering desynchronization. Thus, we define depth uncertainty as the maximum difference of a moving element's possible distances from each camera, when observed by desynchronized cameras.

3. DEPTH UNCERTAINTY MODEL

3.1. General depth uncertainty model

In the general case, we treat *depth uncertainty* as an output parameter, given some *a priori* knowledge about the camera system (e.g. camera positions, sensor & lens properties, expected synchronization accuracy) and the typical observed scene (e.g. speed of scene elements that the camera system must record).

Notation: The Cartesian ("z-axis") depth of any element \mathbf{E} from the i -th camera's viewpoint is directly related to the distance d between \mathbf{E} and the center C_i of the i -th camera, along the ray $\overrightarrow{C_i\mathbf{E}}$ (i.e. $d = \|\mathbf{E} - C_i\|$). For convenience, hereafter we denote such rays by P , e.g. $\overrightarrow{C_i\mathbf{E}} = P_i$. Further, the vector of the ray P_i is denoted by \mathbf{p}_i . We use two *a priori* parameters: v as the maximum speed of \mathbf{E} , and Δt as the synchronization offset (error) between cameras i, j .

Depth Uncertainty: Determining d is possible by triangulation with at least one other camera j observing \mathbf{E} from a different viewpoint. As long as the cameras are synchronized, the intersection of rays P_i, P_j connecting \mathbf{E}, C_i and C_j is at a specific position of \mathbf{E} . Thus, the depth uncertainty Δd is 0. Note that the depth is, in this work, calculated along the ray to the camera principal point, not to the camera sensor plane.

If \mathbf{E} is moving and the cameras i, j capture the scene at times separated by Δt , we cannot determine a precise d at either camera's capture times. Instead we can have a range of possible d values $\Delta d = \max(d) - \min(d)$ around the intersection of P_i and P_j .

Figure 1 (left) shows how Δd is found from coplanar rays P_i, P_j (a 2D case). To maximize Δd along ray P_i , we set a right-triangle relation such that P_i contains the triangle's hypotenuse. The distance covered by \mathbf{E} is the edge $v\Delta t$ in a right-angled triangle. This edge is opposite to angle θ between rays P_i, P_j . Any other placement of the trajectory of \mathbf{E} would produce a lower Δd value. By constructing the right triangle as shown in Figure 1, we make sure that we obtain

the maximum Δd for a given distance $v\Delta t$. The hypotenuse along P_i corresponds to half of Δd , since the true trajectory is unknown and \mathbf{E} can reach P_i from P_j at either side of the ray intersection. In the 2D case, Δd can be determined via the following expression:

$$\Delta d = 2 \frac{v\Delta t}{\sin(\theta)}. \quad (1)$$

The same principle applies for the 3D scenario where P_i, P_j may be non-coplanar, as shown in Figure 1, right. To find Δd along P_i , we project P_j to a ray P'_j in P_i 's plane via a shortest-distance vector \mathbf{m} , which is perpendicular to both P_i and P_j . The maximum distance $(v\Delta t)'$ covered by \mathbf{E} in the plane of P_i, P'_j can be found by another right-angled triangle relation between \mathbf{m} , $(v\Delta t)'$ and $v\Delta t$. Therefore, $(v\Delta t)'$ is determined by using the Pythagoras theorem:

$$(v\Delta t)' = \sqrt{(v\Delta t)^2 - \|\mathbf{m}\|^2}. \quad (2)$$

In the 3D scenario, θ is still determined between vectors $\mathbf{p}_i, \mathbf{p}_j$, since the vector \mathbf{p}_j of ray P_j equals the vector of ray P'_j :

$$\theta = \arccos\left(\frac{\mathbf{p}_i \cdot \mathbf{p}_j}{\|\mathbf{p}_i\| \|\mathbf{p}_j\|}\right). \quad (3)$$

Combining (1) and (2), and checking whether the rays ever come close enough, gives a model for the depth uncertainty Δd :

$$\Delta d = \begin{cases} \frac{2\sqrt{(v\Delta t)^2 - \|\mathbf{m}\|^2}}{\sin(\theta)} & , \text{if } (v\Delta t)^2 > \|\mathbf{m}\|^2, \\ \text{-undefined-} & \text{otherwise.} \end{cases} \quad (4)$$

The "undefined" case in (4) occurs when \mathbf{E} cannot traverse between P_i and P_j (i.e. P_i, P_j cannot be the same \mathbf{E} ; this implies a wrong prior assumption for Δt or v , or a false-positive correspondence matching conclusion).

In case the rays P_i, P_j are co-planar, $\|\mathbf{m}\| = 0$ if P_i, P_j are convergent (otherwise $\|\mathbf{m}\| = \|\mathbf{C}_j - \mathbf{C}_i\|$). This case is in fact the 2D-case and Eq. (1) holds. If, on the other hand, P_i, P_j are not co-planar, the nearest distance between two non-intersecting rays in 3D space can be calculated from the dot product, as the nearest distance between non-parallel rays. The vector \mathbf{m} is simultaneously perpendicular to the direction vectors of the two rays. By defining the vector between two camera origins $\mathbf{p}_o = \overrightarrow{C_j C_i}$, we get:

$$\mathbf{m} = \mathbf{p}_o + \frac{(\hat{b}\hat{e} - \hat{c}\hat{d})\mathbf{p}_i - (\hat{a}\hat{e} - \hat{b}\hat{d})\mathbf{p}_j}{\hat{a}\hat{c} - \hat{b}^2}, \quad (5)$$

where $\hat{a} = \mathbf{p}_i \cdot \mathbf{p}_i$, $\hat{b} = \mathbf{p}_i \cdot \mathbf{p}_j$, $\hat{c} = \mathbf{p}_j \cdot \mathbf{p}_j$, $\hat{d} = \mathbf{p}_i \cdot \mathbf{p}_o$, $\hat{e} = \mathbf{p}_j \cdot \mathbf{p}_o$, as shown in [30]. To ensure that \mathbf{m} in (5) connects P_i, P_j and not just their extended lines "behind the camera", the following conditions must be satisfied:

$$\frac{\hat{b}\hat{e} - \hat{c}\hat{d}}{\hat{a}\hat{c} - \hat{b}^2} \geq 0 \text{ and } \frac{\hat{a}\hat{e} - \hat{b}\hat{d}}{\hat{a}\hat{c} - \hat{b}^2} \geq 0. \quad (6)$$

If (6) does not hold, the rays are at their closest at (or very near) the origin, i.e. $\|\mathbf{m}\| = \|\mathbf{p}_o\|$.

In case of more than two cameras, the overall depth uncertainty Δd for an element \mathbf{E} is retrieved by investigating all depth uncertainties between pairwise cameras ($\Delta d_{i,j}$). The smallest $\Delta d_{i,j}$ represents the best available constraint on \mathbf{E} 's true position. Therefore, the two-camera case scales to the multi-camera case by computing the minimum of all pairwise depth uncertainties:

$$\Delta d = \min_{i,j}(\Delta d_{i,j}), \text{ where } i, j \in \{1, 2, \dots, n\}. \quad (7)$$

Table 1. Fixed Experiment Parameters

| | | |
|-----------------------------------|----------------|----|
| C_1^T | (-250, 0, 0) | mm |
| C_2^T | (250, 0, 0) | mm |
| K_1, K_2 | 773 0 320 | px |
| | 0 773 240 | |
| | 0 0 1 | |
| sensor resolution | 640x480 | px |
| sensor width | 22.3 | mm |
| focal length (35mm equivalent) | 26.9 (42.3) | mm |

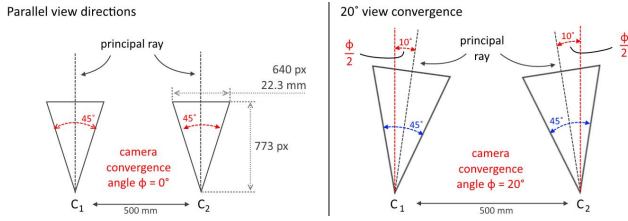


Fig. 2. Camera layout, showing parallel view directions (all parameters as shown in Table 1), and $\phi = 20^\circ$ convergence scenario.

3.2. Adaptation for the pinhole camera model

Adapting (4) to use the pinhole camera model [31] requires a mapping from camera and pixel coordinates to rays with origin and direction, and ensuring that both v and the camera extrinsic parameters are set in the same frame of reference.

We use the pinhole model’s 2D-to-3D back-projection method to describe a ray P_i of camera i as:

$$P_i = C_i + \lambda R_i^{-1} K_i^{-1} c_i, \quad (8)$$

where $C_i = (C_x, C_y, C_z)^T$ is the center of camera i , R_i and K_i are the rotation and intrinsic matrices of the camera i , c_i is the $(x, y, 1)^T$ coordinate in the image plane for the pixel intersected by P_i , and λ is a non-negative scaling factor defining a point along the ray P_i (and thereby setting the reference scale). Using (8) and setting $\lambda = \{0, 1\}$ for P_i start and end points, we can now describe the vectors $\mathbf{p}_i, \mathbf{p}_j$ of rays P_i, P_j as:

$$\mathbf{p}_i = R_i^{-1} K_i^{-1} c_i, \quad (9)$$

$$\mathbf{p}_j = R_j^{-1} K_j^{-1} c_j. \quad (10)$$

The vector \mathbf{p}_o remains $\overrightarrow{C_j C_i}$. From here, we can find $\|\mathbf{m}\|, \theta$ and Δd by substituting (9) and (10) into (3), (4) and (5).

4. EXPERIMENTAL SETUP

To answer the three research questions from section 1, three experiments were carried out. Several parameters (listed in Table 1) were kept constant in all experiments. To represent a typical computer-vision scenario, cameras were defined with 45° horizontal angle of view and set 50 cm apart. Sensor width was set to 22.3 mm, equivalent to commercial APS-C sensors. Synchronization error was set to half of a frame interval, to represent a worst-case desynchronization corrected only by nearest-frame alignment between recorded video sequences. Sensor integration times or effects of a rolling shutter

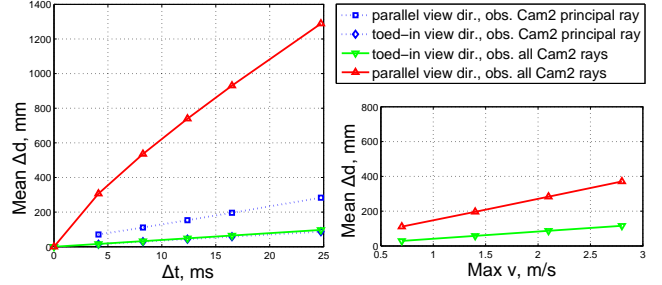


Fig. 3. Depth uncertainty Δd , given varying camera desynchronization (left), and varying maximum speed of scene elements (right), for parallel and $\phi = 20^\circ$ -convergent view directions.

were not considered, in order to reduce the amount of free variables in the model and experiments.

Experiment 1 addressed the research question 1): *How do changes in camera-to-camera desynchronization (Δt) and maximum scene element speed (v) influence overall Δd ?* The parameter Δt was varied from 4.125 ms to 25 ms, corresponding to a desynchronization of up to half a frame interval at 120 to 20 frames per second (fps), respectively. The “perfect synchronization” case $\Delta t = 0$ was also included. For $\Delta t = 16.5$ ms (half frame interval at 30 fps), the parameter v was varied from 0.7 m/s to 2.8 m/s, equivalent to half and double of average human walking speed. The experiment was done for parallel camera view directions (Figure 2, left), and for camera convergence angle $\phi = 20^\circ$ (Figure 2, right). “Overall depth uncertainty” from research question 1 was calculated as the mean of $\Delta d_{1,2}$ for all possible rays P_1 of camera 1, and all rays P_2 of camera 2. Additionally, for varying Δt , mean of $\Delta d_{1,2}$ was calculated for all possible P_1 and only the principal ray of camera 2 as P_2 .

Experiment 2 was set to answer research question 2) (*do parallel-oriented cameras have lower overall Δd compared to inwards-rotated (toed-in) cameras*). The camera convergence angle ϕ , encoded by rotation matrices R_1, R_2 , was varied between 0° and 40° . Parameters Δt and v were set to 16.5 ms and 1.4 m/s, respectively. Other parameters were kept as shown in Table 1. Overall depth uncertainty was calculated as the mean of $\Delta d_{1,2}$ for all possible rays P_1 of camera 1, and all rays P_2 of camera 2.

Experiment 3 addressed research question 3) by mapping Δd for each ray P_1 of camera 1, using two Δd estimations. Parameters Δt and ϕ were varied using same steps as in *experiment 1* and *experiment 2*, respectively. In the first estimation, Δd of each P_1 was calculated using only the principal ray of camera 2 as P_2 . In the second estimation, Δd of each P_1 was calculated as mean Δd of all combinations of P_1 and P_2 , for every ray of camera 2 as P_2 . P_1, P_2 combinations where $\Delta d = \infty$ or $\Delta d = \text{“undefined”}$ were excluded. The number of possible P_1, P_2 combinations was also tracked, in order to assess the model’s computational cost, and to investigate the need to use the first estimation of Δd instead of the second estimation.

5. RESULTS AND ANALYSIS

The results discussed in this section are from simulations of a scenario with cameras “1” and “2”, following the experiments given in section 4. All results show depth uncertainty with respect to camera “1”, since the camera “2” results are symmetrically equivalent.

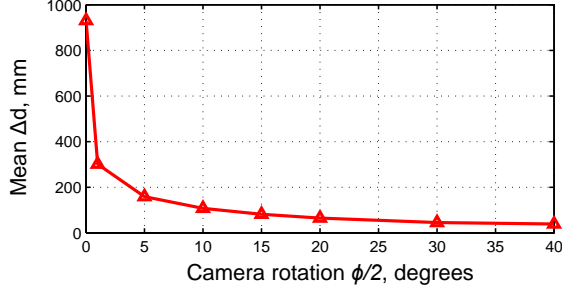


Fig. 4. Mean Δd along all rays of camera 1, for varying convergence ϕ of both cameras (indicated rotation $\phi/2$ for camera 1, with simultaneous negative rotation $-\phi/2$ on camera 2).

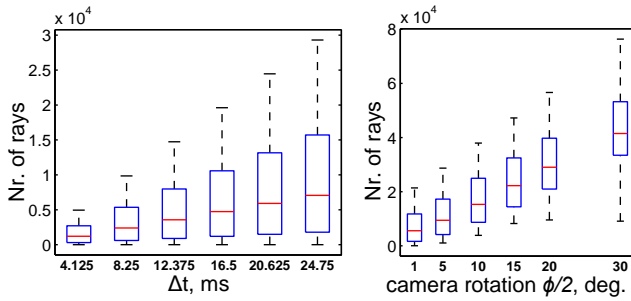


Fig. 5. Number of rays from camera 2 that satisfy non-infinite Δd estimation prerequisites from Eq. (4), for each ray of camera 1. Box plots show mean (red line inside the box), 25th and 75th percentile (box bottom & top), minimum and maximum (top and bottom whisker).

Experiment 1: At 0 ms desynchronization between cameras, the depth uncertainty is, as expected, 0. For non-zero delays between cameras, the relation between Δt and Δd is nearly linear (see Figure 3). While increased camera view convergence significantly decreases Δd (by about a factor of 10, with 20° convergence), the relation still exhibits a linearity. A similar behavior is seen when altering v instead of Δt . At near-0 Δt , Figure 3, (left) does not show a linear behavior for parallel cameras - this may have been caused by a decrease in ray combinations satisfying the prerequisite condition in eq. (4).

Experiment 2: Depth uncertainty peaks for parallel camera orientations, and decreases by a factor of 4 at a $\phi = 2^\circ$ camera convergence, as shown in Figure 4. Depth uncertainty reaches the lowest values when the cameras are observing the scene at right angles to each other ($\phi = 90^\circ$), as that maximizes the average θ between rays of both cameras. However, going from $\phi = 30^\circ$ to $\phi = 90^\circ$ does not significantly change Δd , whereas it does limit the observable scene dimensions more and more due to decreasing view overlap volume. Moreover, as shown in Figure 3 and in Figure 7, in parallel-oriented cameras Δd increases at a faster rate compared to toed-in cameras, given the same increases in Δt and v . Thus, a parallel-oriented multi-camera configuration suffers significantly more from imperfect synchronization, and has a larger Δd than even slightly toed-in camera arrays.

Experiment 3: Figure 5 shows how many rays of camera 2 satisfy the condition in eq. (4), for each ray in camera 1 (i.e. how many

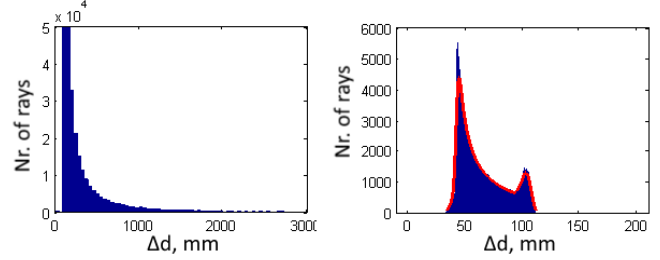


Fig. 6. 1000-bin histograms of Δd values for individual camera 1 rays. $\phi = 20^\circ$ convergent view directions. **Left** graph depicts the case when only the principal ray of camera 2 is considered; **right** graph depicts the case when all rays of camera 2 are considered.

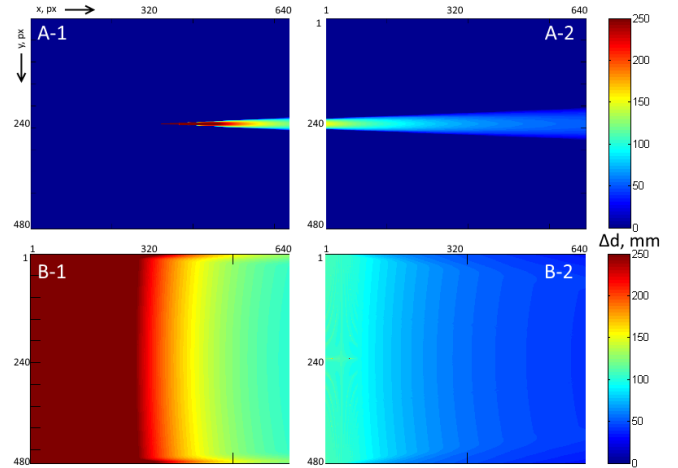


Fig. 7. Mean Δd on camera sensor in 4 configurations. Each pixel (x,y) describes mean Δd of a ray going from the camera center through (x,y) . Δd scale clipped at 250 mm for visibility. Configurations: the mean Δd was calculated with **top row** only the principal ray of camera 2; **bottom row** all rays of camera 2; **left column** $\phi = 0^\circ$ view convergence (Figure 2, left); **right column** $\phi = 20^\circ$ convergence (Figure 2, right).

possible interactions a ray from camera 1 has with rays from camera 2). The influence of Δt is linear for mean, minimum and maximum ray counts, and therefore has a linear effect on computation requirements. Camera convergence also has a linear effect on mean and maximum ray counts, however the minimum number of interactions increases until a $\phi = 45^\circ$ convergence. Both sides indicate that, for each ray of camera 1, there is likely to be a significant number of rays from camera 2 that contribute to the overall depth uncertainty, with a directly proportional computational cost.

The Δd of camera 1 rays show different distributions in Figure 6, when estimated only the principal ray of camera 2 (“first estimation” in section 4, experiment 3) instead of using all rays of camera 2 (“second estimation”). Using only the principal ray to estimate Δd is not an acceptable abstraction of the general depth uncertainty model, as indicated by differences between the first and second estimations in sensor utilization and Δd values shown in Figure 7, in the distributions seen in Figure 6, and in the Δd values plotted in Figure 3.

6. CONCLUSIONS

In this paper, we highlighted the lack of a parametric relation between camera desynchronization, camera properties, scene element depth capture, and motion. We defined a parametric model that describes this relation, and combined it with the pinhole camera model. We used our model to investigate how desynchronization and convergence between cameras affects depth uncertainty.

Simulations indicate that both desynchronization and scene element speed have a linear relation with the system depth uncertainty. In desynchronized camera systems, depth uncertainty is affected by the angle of convergence between the involved cameras. In particular, parallel-oriented camera arrays have significantly worse depth accuracy than toed-in cameras. Depth uncertainty has to be assessed by considering not just the principal rays of the involved cameras, but all rays on both cameras involved in the depth determination. We also showed that, for overall system depth uncertainty estimations, our model has a computational cost linear with both desynchronization intervals and camera convergence.

The proposed model can be improved by extending the supported parameter set (e.g. sensor integration time, line-by-line time offset due to rolling-shutter), and by considering depth to a camera plane instead of camera center point. Furthermore, the model can be included in a design or cost-analysis process for constructing or evaluating a multi-camera system with given requirements on depth recovery in moving scenes.

7. ACKNOWLEDGMENT

This work has been supported by the LIFE project grant 20140200 of the Knowledge Foundation, Sweden.

8. REFERENCES

- [1] Nils Hasler, Bodo Rosenhahn, Thorsten Thormahlen, Michael Wand, Jürgen Gall, and Hans-Peter Seidel, “Markerless motion capture with unsynchronized moving cameras,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 224–231.
- [2] Matthias Ziegler, Andreas Engelhardt, Stefan Müller, Joachim Keinert, Frederik Zilly, Siegfried Foessel, and Katja Schmid, “Multi-camera system for depth based visual effects and compositing,” in *Proceedings of the 12th European Conference on Visual Media Production*. ACM, 2015, p. 3.
- [3] Ginni Grover, Ram Narayanswamy, and Ram Nalla, “Simulating multi-camera imaging systems for depth estimation, enhanced photography and video effects,” in *Imaging Systems and Applications*. Optical Society of America, 2015, pp. IT3A–2.
- [4] Xiaogang Wang, “Intelligent multi-camera video surveillance: A review,” *Pattern Recognition Letters*, vol. 34, no. 1, pp. 3–19, 2013.
- [5] Marek Domański, Adrian Dziembowski, Dawid Mieloch, Adam Łuczak, Olgierd Stankiewicz, and Krzysztof Wegner, “A practical approach to acquisition and processing of free viewpoint video,” in *Picture Coding Symposium (PCS), 2015*. IEEE, 2015, pp. 10–14.
- [6] Oliver Schreer, Ingo Feldmann, Christian Weissig, Peter Kauff, and Ralf Schafer, “Ultrahigh-resolution panoramic imaging for format-agnostic video production,” *Proceedings of the IEEE*, vol. 101, no. 1, pp. 99–114, 2013.
- [7] Robert Anderson, David Gallup, Jonathan T Barron, Janne Kontkanen, Noah Snavely, Carlos Hernández, Sameer Agarwal, and Steven M Seitz, “Jump: virtual reality video,” *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 198, 2016.
- [8] Steffen Urban, Sven Wursthorn, Jens Leitloff, and Stefan Hinz, “Multicol bundle adjustment: A generic method for pose estimation, simultaneous self-calibration and reconstruction for arbitrary multi-camera systems,” *International Journal of Computer Vision*, pp. 1–19, 2016.
- [9] Thomas Luhmann, Clive Fraser, and Hans-Gerd Maas, “Sensor modelling and camera calibration for close-range photogrammetry,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 115, pp. 37–46, 2016.
- [10] Luis Puig, Jesús Bermúdez, Peter Sturm, and José Jesús Guerrero, “Calibration of omnidirectional cameras in practice: A comparison of methods,” *Computer Vision and Image Understanding*, vol. 116, no. 1, pp. 120–137, 2012.
- [11] Xinzhao Li, Yuehu Liu, Shaozhuo Zhai, and Zhichao Cui, “A structural constraint based dual camera model,” in *Chinese Conference on Pattern Recognition*. Springer, 2014, pp. 293–304.
- [12] Hooman Shidanshidi, Farzad Safaei, and Wanqing Li, “A method for calculating the minimum number of cameras in a light field based free viewpoint video system,” in *2013 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2013, pp. 1–6.
- [13] Chris Sweeney, Victor Fragoso, Tobias Hollerer, and Matthew Turk, “Large scale SFM with the distributed camera model,” *arXiv preprint arXiv:1607.03949*, 2016.
- [14] Junbin Liu, Sridha Sridharan, Clinton Fookes, and Tim Wark, “Optimal camera planning under versatile user constraints in multi-camera image processing systems,” *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 171–184, 2014.
- [15] Huogen Wang, Jiachen Wang, Zhiyong Ding, and Fei Guo, “Self-converging camera arrays: Models and realization,” in *2013 Ninth International Conference on Natural Computation (ICNC)*. IEEE, 2013, pp. 338–342.
- [16] Robert Pless, “Using many cameras as one,” in *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2003, vol. 2, pp. II–587.
- [17] Georgios Litos, Xenophon Zabulis, and Georgios Triantafyllidis, “Synchronous image acquisition based on network synchronization,” in *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW’06)*. IEEE, 2006, pp. 167–167.
- [18] Dmitry Pundik and Yael Moses, “Video synchronization using temporal signals from epipolar lines,” in *European Conference on Computer Vision*. Springer, 2010, pp. 15–28.
- [19] Richard Latimer, Jason Holloway, Ashok Veeraraghavan, and Ashutosh Sabharwal, “Socialsync: Sub-frame synchronization in a smartphone camera network,” in *European Conference on Computer Vision*. Springer, 2014, pp. 561–575.
- [20] Yuichi Taguchi, Keita Takahashi, and Takeshi Naemura, “Real-time all-in-focus video-based rendering using a network camera array,” in *2008 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video*. IEEE, 2008, pp. 241–244.

- [21] Si Ying Hu, James Baldwin, Armand Niederberger, and David Fattal, "I3. 2: Invited paper: A multiview 3d holochat system," in *SID Symposium Digest of Technical Papers*. Wiley Online Library, 2015, vol. 46, pp. 286–289.
- [22] Jason C Yang, Matthew Everett, Chris Buehler, and Leonard McMillan, "A real-time distributed light field camera," *Rendering Techniques*, vol. 2002, pp. 77–86, 2002.
- [23] Yuichi Taguchi, Takafumi Koike, Keita Takahashi, and Takeshi Naemura, "Transcaip: A live 3d tv system using a camera array and an integral photography display with interactive control of viewing parameters," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 5, pp. 841–852, 2009.
- [24] Cheng Lei and Yee-Hong Yang, "Tri-focal tensor-based multiple video synchronization with subframe optimization," *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2473–2480, 2006.
- [25] Tinne Tuytelaars and Luc Van Gool, "Synchronizing video sequences," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2004, vol. 1, pp. I–I.
- [26] Cheng Lu and Mrinal Mandal, "A robust technique for motion-based video sequences temporal alignment," *IEEE Transactions on Multimedia*, vol. 15, no. 1, pp. 70–82, 2013.
- [27] Georgios D Evangelidis and Christian Bauckhage, "Efficient subframe video alignment using short descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 10, pp. 2371–2386, 2013.
- [28] Ferran Diego, Daniel Ponsa, Joan Serrat, and Antonio M López, "Video alignment for change detection," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1858–1869, 2011.
- [29] Yaron Caspi and Michal Irani, "Spatio-temporal alignment of sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1409–1424, 2002.
- [30] Vladimir J Lumelsky, "On fast computation of distance between line segments," *Information Processing Letters*, vol. 21, no. 2, pp. 55–61, 1985.
- [31] Richard Hartley and Andrew Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, 2003.