Real-Time Textureless-Region Tolerant High-Resolution Depth Estimation System

Bilal Demir, Jean-Philippe Thiran and Yusuf Leblebici
Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
name.surname@epfl.ch

***Abstract*—This study presents a real-time depth estimation hardware system aiming to provide high-resolution depth data and to eliminate its noise on textureless regions without causing any interference problem by utilizing artificial pattern projection. The system generates up to 2K resolution depth data and reaches up to 256 disparity range which are configurable by the end-user owing to its parameterized design. It is capable of streaming depth data with 21 frames per second (fps) with 2K resolution and 128 pixel disparity range, and its throughput performance changes depending on configuration of the output resolution and the disparity range.**

***Index Terms*—Real-time, trinocular disparity estimation, depth estimation, textureless-region depth noise, stereo vision, hardware system, embedded, FPGA.**

I. Introduction

With the recent trend of stereo camera systems, visual depth data is becoming more and more exciting topic to generate real-time, high-resolution and noise free depth data. It is highly requested by various video processing applications such as 3D modelling, virtual reality, robotics, autonomous vehicles, drones etc.

Various depth estimation methods are actively studied to satisfy this demand [1]–[9]. These methods are mainly categorized as local and global methods. Global depth estimation methods suffer from lack of real-time output streaming, while local depth estimation methods fail to eliminate noisy depth result especially on textureless or poorly-textured regions because local methods mostly rely on correlation and matching of the pixels captured from separate camera frames. On the other hand, Time of Flight (TOF) depth image sensors [10] produce accurate depth map. However they suffer from low resolution and interference problem.

Researchers examined various methods to eliminate textureless region depth noise while sustaining real-time and high-resolution performance. Many studies have been conducted to create dynamic or adaptive window size and shape, however, they could not provide decent noise-free output results [11]–[13]. Another approach is to model the textureless region of a roadway [14], however, this method is designed specifically for the roadway depth result instead of addressing the general case scenario. Some other techniques like [7]–[9] are incapable of providing real-time depth stream.

Regarding this problem, this paper presents novel approach to eliminate depth noise on textureless region while sustaining real-time and high-resolution performance. The illustration of proposed depth estimation system is presented in figure 1. This study is enhanced version of the previously published binocular disparity estimation algorithm which provides XGA 60 fps binocular depth output with XGA resolution and 128 disparity range [15].

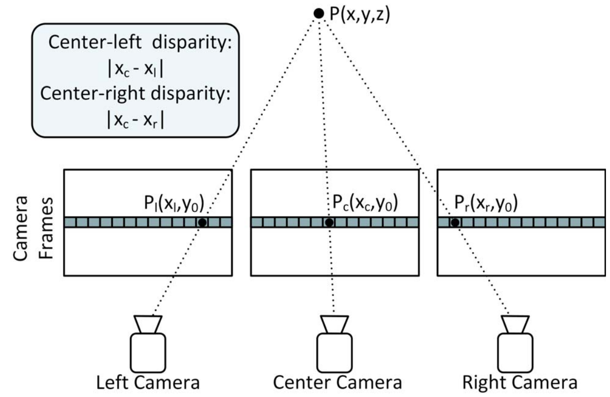


Fig. 1: Illustration of the proposed trinocular disparity estimation system

II. The Proposed Algorithm and Hardware

The block diagram of the proposed system is given in figure 2. All high performance real-time video processing is implemented in hardware. The system captures RGB image from 3 parallel positioned cameras and generates synchronized RGB (24-bits) pixel values with their computed depth (8-bits) values (RGB+D).

The cameras are initialized and configured by MicroBlaze soft processor core via I^2C interface. The three cameras are precisely synchronized through common clock source and common I^2C module which are provided and controlled by the FPGA. This synchronization is crucial to capture the identical frame rates, and thereby obtaining correct disparity matches between separate camera frames. The captured image frames are processed by Camera Interface unit to convert Bayer format (8-bits) to YCbCr format (24-bits). These images are stored in FIFO to maintain synchronization and pipelining, then they are transferred to Rectification unit. The internal and external camera calibration parameters are calculated offline through OpenCV toolbox [16]. These parameters are transferred to the FPGA via UART interface and stored in software accessible registers.

978-1-7281-2862-7/19/$31.00 ©2019 IEEE
DOI 10.1109/DSD.2019.00020

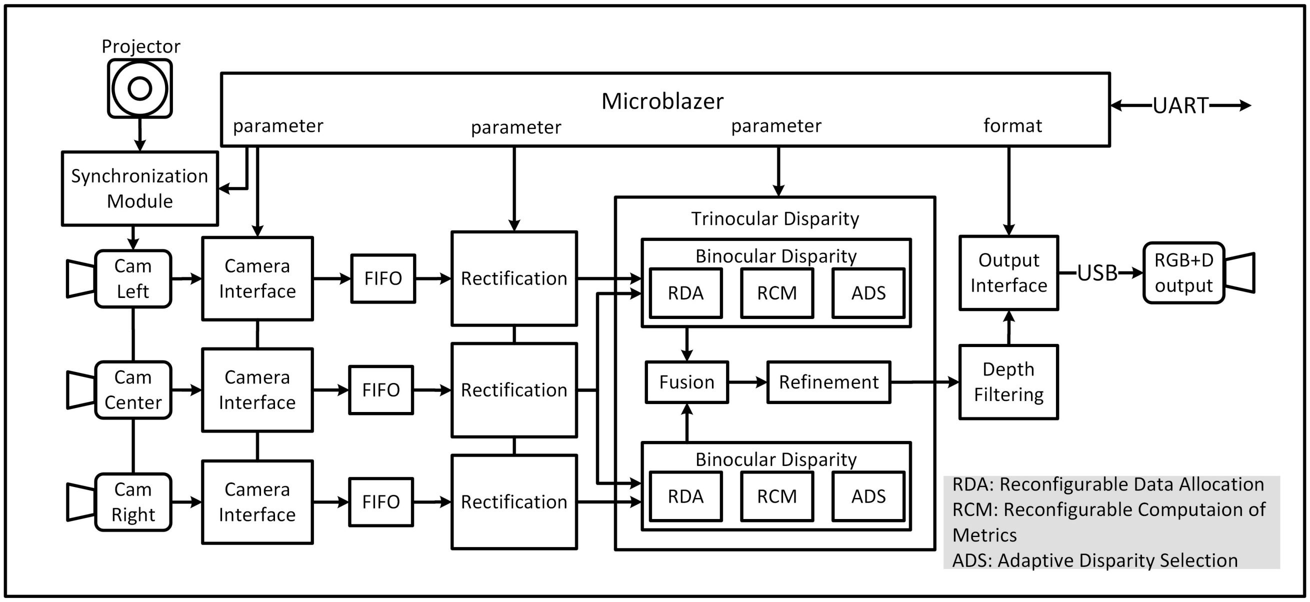


Fig. 2: Block diagram of the proposed system

Caltech rectification algorithm [17] is applied to align image frames coming from three cameras. YCbCr images captured from FIFOs and 64 rows of each image are buffered in on-chip BRAMs to enable high-speed pixel access during rectification process. These images synchronously processed in parallel by the Rectification unit and the rectified image pairs (center-right and center-left) are transferred to two binocular disparity estimation units.

Binocular disparity estimation unit consists of three sub-modules: Reconfigurable Data Allocation (RDA), Reconfigurable Computation of Metrics (RCM) and Adaptive Disparity Selection (ADS). The RDA unit buffers 39 rows of each rectified image in single-port BRAMs to perform window-based matching scheme. The windows size is dynamically selected based on the texture feature around the each pixel during disparity matching process as either 7×7, 13×13, 25×25 pixels. Window size is adaptively determined depending on Mean absolute deviation (MAD) and threshold values. Hardware complexity remains constant because evenly distributed 49 contributor pixels are included to matching calculation in each window size scheme. The RCM unit computes Census and binary-window sum of absolute difference (BW-SAD) matrixes from the 49 contributor pixels. They are combined to form the Hybrid Cost (HC) values which are used as metrics in disparity voting process. Moreover, Confidence value is computed for each pixel from its best and second best HC values and this Confidence value is stored and used in Refinement unit. ADS unit receives these values from RCM unit and performs disparity matching by selecting the pixel with minimum HC value. Further details about Binocular Disparity unit are presented in [15].

The binocular disparities of center-right and center-left images are transferred to Fusion unit. The Fusion unit merges these two binocular disparity maps into one trinocular disparity map. The Fusion unit compares two binocular disparity values and chooses the one with lower HC value by applying winner-take-all approach. In this way, Fusion unit eliminates noisy and incorrect depth data caused by occlusion. The trinocular disparity map produced by Fusion unit is smoothen by Refinement unit. It identifies the outliers of the disparity results in the neighboring pixels by omitting the pixels with low confidence values and replacing them with the most frequent disparity value. Bilateral filtering is applied to refined disparity map and the end-user can choose to display filtered or unfiltered 32-bits of RGB+D output data via USB interface. The user is also capable of monitoring only RGB, Depth data or RGB+D data. The synchronized 32-bits RGB+D data generation provides great convenience for practical video processing applications like foreground detection and background subtraction. The hardware is configurable from the PC to adjust resolution and disparity range. The Rectification and Trinocular Disparity Estimation units avoid DDR3 memory utilization in this algorithm to achieve an efficient and high-performance video processing.

In addition to refinement and bilateral filtering, the proposed system contains pico-projector to generate artificial pattern projection. It serves to further eliminate the depth noise on textureless region. First of all, synchronization between the projector and the cameras is ensured so as to eliminate Moiré effect. Regarding this, it was disassembled and its scanning signal, which is adjusting position of the projection laser, was extracted with the help oscilloscope by reverse engineering. The projector scanning signal ($proj_{scan}$) is essential for synchronization between the cameras and the Picopro. Since $proj_{scan}$ signal is so noisy, it was filtered via low pass filter and FPGA glitch elimination to generate projector signal

($proj_{sync}$). The camera synchronization signal (cam_{sync}) is produced from $proj_{sync}$ signal to trigger the camera shutter and to determine camera readout scan. These synchronization signals are illustrated in figure 3. The cam_{sync} and $proj_{sync}$ signals have following specifications:

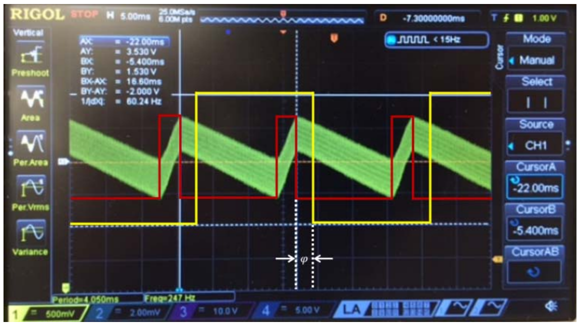
- cam_{sync} signal should have even duty cycle unlike $proj_{sync}$ signal as camera frame rates do not change over time.
- Amplitude of the cam_{sync} signal needs to be converted from $proj_{sync}$ signal by voltage converter since voltage range of the FPGA (3.3V) and the cameras (5V) are different.
- A period of cam_{sync} signal needs to be positive integer multiple of period of $proj_{sync}$ signal to sustain synchronization between the projector and the cameras.
- cam_{sync} signal has phase shift (φ) with respect to $proj_{sync}$ signal due to the hard-to-measure signal transmission delay between the projector to the cameras through the FPGA and the PCB.

In the light of the specifications explained below, the relationship between cam_{sync} and $proj_{sync}$ signals can be formulated as:

$$cam_{sync} = A \times proj_{sync}(\frac{\omega}{N} + \varphi) \text{ with 50\% duty cycle} \quad (1)$$

where:

A**:** Amplitude constant.
N**:** Positive integer constant.
ω**:** Signal angular frequency.
φ**:** Phase shift.



```
AX:     = -22.00ms
AY:     = 3.530 V
BX:     = -5.400ms
BY:     = 1.530 V
BX-AX:  = 16.60ms
BY-AY:  = -2.000 V
1/|dX|: = 60.24 Hz
```

Fig. 3: Illustration of the synchronization signals: $proj_{scan}$, $proj_{sync}$, cam_{sync}.

The parameter A can be easily computed ($1.51 = 5V/3.3V$) as we know voltage ranges of the FPGA and the cameras. However, it is difficult to accurately calculate parameters of φ and N because:

- Phase shift is determined by the hard-to-measure signal transmission delay from the projector to the cameras through the FPGA and the PCB.
- Parameter N needs to be adjusted such that the camera frame rate and the projector display rate match.
- The camera frame rate depends on not only shutter width but also resolution.

These parameters are manually tuned until all undesired interferences between the cameras and the projector like Moiré interference disappear. Considering the camera configuration side of the synchronization, Electronic Rolling Shutter (ERS) snapshot mode is preferred because this scheme does not only diminish shearing affect but also gives digital tuning flexibility for camera synchronization.

The proposed architecture is different than the hardware presented in [15], as the proposed system: 1) utilizes trinocular disparity estimation, 2) includes projector integration and synchronization, 3) generates higher resolution, 4) searches for higher disparity range, 5) provides real-time hardware filtering, 6) enables larger bandwidth data transfer and 7) is fully configurable in terms of resolution and disparity without changing hardware architecture.

III. Implementation Results

The proposed system is implemented in XILINX Virtex-707 FPGA with 190 MHz clock frequency. It consumes 121k Look-Up-Tables (LUT), 3483 (LUTRAM), 119k DFFs and 438 BRAMS and 92 DSP resources. MT9P031 digital image sensor with 5 megapixels resolution and Celluon PicoPro portable projector with HD resolution are used. The system is capable of producing real-time depth data up to 2K resolution and up to 256 disparity range. The resolution and disparity range are fully configurable by the end-user. The correlation among resolution, disparity range and output frame rate is given in table I.

Resolution/ Disparity	XGA	Full HD	2K
64 pixels	87 fps	33 fps	31 fps
128 pixels	60 fps	23 fps	21 fps
256 pixels	37 fps	14 fps	13 fps

TABLE I: The throughput performance of the system with respect to resolution and disparity range configuration.

The captured image and its depth map is presented in figure 4a-b. The white rectangular box is taken as a region of interest to test the performance of the artificial pattern projection as the box has no texture on its surface. Various textures are projected upon the region of interest, and their noise reduction performances in the filtered and the unfiltered depth maps are examined.

The performance comparison of the projected artificial patterns are conducted considering the following error metrics:

$$err_{peak} = \frac{100}{256} max(|p(i,j) - g(i,j)|) \quad (2)$$

$$err_{average} = \frac{100}{256} (\sum_{i,j \in S} |p(i,j) - g(i,j)|)/w \quad (3)$$

$$\begin{aligned} err_{spatial} = \frac{100}{255} (\sum_{i,j \in S} |4p(i,j) - p(i,j+1) - p(i,j-1) \\ - p(i+1,j) - p(i-1,j)|)/w \quad (4) \end{aligned}$$

where:

S: Region of interest (white rectangular box).
w: Total number of pixels over S.
$p(i,j)$: Depth pixel value at (i,j).
$g(i,j)$: Ground truth depth pixel value at (i,j)
err_{peak}: Peak error rate in S.
$err_{spatial}$: Spatial error rate in S.
$err_{average}$: Average error rate in S.

Every error rate is multiplied by $100/256$ in the formulas to calculate percentage error rate. 256 and 100 stand for [0,255] pixel range in grayscale image and percentage error respectively.

These error metrics are chosen to measure the noise elimination performance of the artificial pattern projection.err_{peak} defines maximum depth error rate with respect to ground truth depth value. $err_{average}$ defines average depth error rate in the region of interest. $err_{spatial}$ defines an average depth error rate with respect to four neighbor pixels. In other words, it indicates how much depth pixel deviates compared to its neighbor pixels. $err_{spatial}$ can be calculated in the given formula since the rectangular box is flat and it is positioned diagonally to central camera direction, hence it is supposed to have the same depth value on its surface. Ground truth depth values $g(i,j)$ are manually calculated by averaging the depth values with high confidence and low HC values. The textureless box with artificial patterns projected on it, their raw depth results and their bilateral filtered depth results are presented in figure 4. Their corresponding error metrics are given in table II.

Pattern/ Error	#0	#1	#2	#3	#4	#5
err_{peak}	72.46	72.16	72.16	70.59	**61.69**	72.16
$err_{spatial}$	4.96	3.53	2.01	4.84	**0.38**	2.56
$err_{average}$	14.90	9.43	4.15	11.81	**0.91**	4.81
err_{peak} filtered	70.59	72.16	60.39	69.02	**47.06**	70.20
$err_{average}$ filtered	13.35	8.68	2.52	9.38	**0.59**	1.14
$err_{spatial}$ filtered	0.92	0.89	0.39	1.34	**0.08**	0.36

TABLE II: Overall depth error summary.

Analyzing the results from figure 4 and error metrics summary from the table II, we observe:

- Spatially repetitive patterns like Pattern#1 and Pattern#3 provides slight improvement since the repetitive texture does not help to distinguish pixels during disparity detection process.
- Denser random textures provide better accuracy as they provide more distinctive features.
- Randomly distributed and various sized shapes like Pattern#4 and Pattern#5 provides prevailing results since they are more likely to create typical and unique features.
- Artificial pattern projection also helps the bilateral filter for further noise reduction.

IV. Conclusion

In this paper, we proposed real-time 2K resolution trinocular depth estimation system with 256 disparity range. The system provides up to 76 fps performance depending on the selected resolution and disparity range as indicated in table I. The system utilizes Fusion, Refinement, bilateral depth filtering units accompanied with the artificial pattern projection in order to efficiently eliminate depth noise including but not limited to textureless regions. The end-user can easily configure the system without hardware modification in terms of the resolution, disparity range, filtering option and projected pattern by using the PC. The cameras and the pico-projector are synchronized to remove undesired Moiré interference effect. The projected pattern brings about no interference issue to nearby systems contrary to structured light based depth estimation systems.

Several artificial patterns are evaluated considering different error metrics and the one providing the best result is chosen. The experimental results verified that the proposed system successfully improve the accuracy of the disparity in terms of resolution, throughput and noise reduction performances. The comparison between the proposed system and other similar state-of-art depth estimation systems is presented in table III.

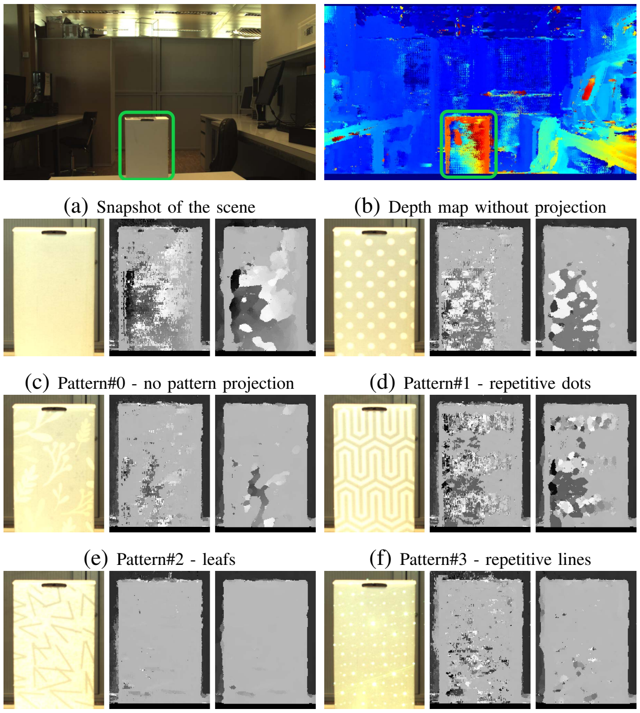

(a) Snapshot of the scene
(b) Depth map without projection
(c) Pattern#0 - no pattern projection
(d) Pattern#1 - repetitive dots
(e) Pattern#2 - leafs
(f) Pattern#3 - repetitive lines
(g) Pattern#4 - randomly distributed and sized lines
(h) Pattern#5 - randomly distributed and sized dots

Fig. 4: Depth map results with several pattern projections on region of interest (marked by green rectangle). The region of interest, depth output and bilateral filtered depth results are presented respectively from left to right (c-g).

System/ Property	Hierarchical design [1]	RGBD imager [2]	This work
Stereo type	3 views	3 views	2 or 3 views
Disparity range	32 pixels	64 pixels	up to 256 pixels
Resolution	$640x480$	$320x240$	up to 2K
Frame rate	52 fps	30 fps	(12-76 fps)[1]
TRT[2]	yes	no	yes
Depth filtering	no	no	yes
Configurability	no	no	yes
Platform	Cyclone-IV	Virtex-4	Virtex-7

[1] It depends on the resolution and the disparity, refer to table I.
[2] Textureless Region Tolerant.

TABLE III: The performance comparison with other similar state-of-art depth estimation systems.

References

[1] A. Motten, L. Claesen, and Y. Pan, "Trinocular stereo vision using a multi level hierarchical classification structure," in *VLSI-SoC: From Algorithms to Circuits and System-on-Chip Design*, A. Burg, A. Coskun, M. Guthaus, S. Katkoori, and R. Reis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 45–63.

[2] L. Chen, Y. Jia, and M. Li, "An fpga-based rgbd imager," *Machine Vision and Applications*, vol. 23, no. 3, pp. 513–525, May 2012. [Online]. Available: https://doi.org/10.1007/s00138-011-0334-z

[3] N. Y. Chang, T. Tsai, B. Hsu, Y. Chen, and T. Chang, "Algorithm and architecture of disparity estimation with mini-census adaptive support weight," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 6, pp. 792–805, June 2010.

[4] S. Hsiao, W. Wang, and P. Wu, "Vlsi implementations of stereo matching using dynamic programming," in *Technical Papers of 2014 International Symposium on VLSI Design, Automation and Test*, April 2014, pp. 1–4.

[5] C. Ttofis, S. Hadjitheophanous, A. S. Georghiades, and T. Theocharides, "Edge-directed hardware architecture for real-time disparity map computation," *IEEE Transactions on Computers*, vol. 62, no. 4, pp. 690–704, April 2013.

[6] M. Mozerov, J. Gonzàlez, X. Roca, and J. J. Villanueva, "Trinocular stereo matching with composite disparity space image," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, Nov 2009, pp. 2089–2092.

[7] Y. Du, D. Tian, P. He, X. Han, and Z. Yuan, "Belief propagation and self-adaptive voting dense stereo disparity estimation for textureless scene 3d reconstruction," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, Oct 2016, pp. 459–463.

[8] B. Raman, S. Nagarajan, R. Bhargava, M. Kumar, and S. Kumar, "Depth recovery of complex surfaces from texture-less pair of stereo images," in *ELCVIA. Electronic letters on computer vision and image analysis*, vol. 8, 2009, pp. 44–56.

[9] A. Saouli and M. C. Babahenini, "Towards a stochastic depth maps estimation for textureless and quite specular surfaces," in *SIGGRAPH Posters*, 2018.

[10] T. Liao, N. Lee, and C. Hsieh, "A cmos time of flight (tof) depth image sensor with in-pixel background cancellation and sensitivity improvement using phase shifting readout technique," in *2017 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Nov 2017, pp. 133–136.

[11] O. Veksler, "Fast variable window for stereo correspondence using integral images," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 1, June 2003, pp. I–I.

[12] Y. Boykov, O. Veksler, and R. Zabih, "A variable window approach to early vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1283–1294, Dec 1998.

[13] A. Akin, R. Capoccia, J. Narinx, I. Baz, A. Schmid, and Y. Leblebici, "Trinocular adaptive window size disparity estimation algorithm and its real-time hardware," *VLSI Design, Automation and Test(VLSI-DAT)*, pp. 1–4, 2015.

[14] J. Kim, K. Kim, and K. Jung, "Reliable estimation of disparity map in textureless region of roadway," in *2017 19th International Conference on Advanced Communication Technology (ICACT)*, Feb 2017, pp. 399–402.

[15] A. Akin, I. Baz, A. Schmid, and Y. Leblebici, "Dynamically adaptive real-time disparity estimation hardware using iterative refinement," *Integration, the VLSI Journal*, vol. 47, no. 3, pp. 365–376, 2014.

[16] OpenCV. (2019) Camera calibration with opencv. [Online]. Available: https://docs.opencv.org/3.0-beta/doc/tutorials/calib3d/camera_calibration/camera_calibration.html

[17] P. F. Sturm and S. J. Maybank, "On plane-based camera calibration: A general algorithm, singularities, applications," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, vol. 1, June 1999, pp. 432–437 Vol. 1.