Simultaneous Camera Calibration and Temporal Alignment of 2D and 3D Trajectories

Joni Herttuainen, Tuomas Eerola, Lasse Lensu and Heikki Kälviäinen
Machine Vision and Pattern Recognition Laboratory, School of Engineering Science, Lappeenranta University of Technology, P.O. Box 20, 53851, Lappeenranta, Finland
joni.herttuainen@student.lut.fi, {tuomas.eerola, lasse.lensu, heikki.kalviainen}@lut.fi

Keywords: Trajectory Alignment, Camera Calibration.

Abstract: In this paper, we present an automatic method that given the 2D and 3D motion trajectories recorded with a camera and 3D sensor, automatically calibrates the camera with respect to the 3D sensor coordinates and aligns the trajectories with respect to time. The method utilizes a modified Random Sample Consensus (RANSAC) procedure that iteratively selects two points from both trajectories, uses them to calculate the scale and translation parameters for the temporal alignment, computes point correspondences, and estimates the camera matrix. We demonstrate the approach with a setup consisting of a standard web camera and Leap Motion sensor. We further propose necessary object tracking and trajectory filtering procedures to produce proper trajectories with the setup. The result showed that the proposed method achieves over 96% success rate with a test set of complex trajectories.

1 INTRODUCTION

The motivation for this work comes from the human-computer interaction (HCI) research where exists a need to accurately record natural hand and finger movements of test subjects in various HCI tasks. Advances in gesture interfaces, touchscreens, augmented and virtual reality bring new usability concerns that need to be studied when using them in natural environment and in an unobtrusive way. Several robust approaches for hand tracking exist, such as data gloves with electromechanical or magnetic sensors that can measure the hand and finger location with high accuracy. However, such devices affect the natural hand motion and cannot be considered a feasible solution when pursuing natural HCI. As a consequence, there is a need for image-based solutions that provide an unobtrusive way to study and track human movement and enable natural interaction with technology.

Modern digital cameras make it possible to study object trajectories with high accuracy and also with high frame rate, and state-of-the-art object trackers provide robust and fast tools to construct the motion trajectories automatically from the videos. For example, in (Hiltunen et al., 2014), several object tracking methods were compared with high-speed videos and the top methods were found to be suitable for the problem of measuring HCI. In (Kuronen et al., 2015), the tracking was supplemented with filtering techniques to provide a methodology to measure and study 2D hand motion of test subjects performing various HCI tasks.

Typically, the real motion trajectories are in 3D, but recording accurate 3D trajectories would require multiple high-speed cameras. Such measurement setup is both expensive and difficult to build. On the other hand, 3D sensors, such as Leap Motion or Kinect, do not allow high frame rates and lack the versatility and debuggability of a camera based system. A setup consisting of a single high-speed camera accompanied with a separate 3D sensor to capture depth information provides an affordable alternative that produces reasonably accurate trajectories.

In this work, we propose a method to automatically calibrate the camera with respect to 3D sensor coordinates and temporally align 2D and 3D trajectories, i.e., given a point in time, the location of the object can be obtained in both image and 3D sensor coordinates (see Fig. 1). By camera calibration we mean determining the mapping from 3D sensor coordinates to 2D image coordinates, i.e., estimating the camera matrix. The method requires only the tracked 2D and 3D trajectories, and no temporal synchronization of the devices is needed as long as the trajectories are at least partially overlapping with respect to time.

Herttuainen J., Eerola T., Lensu L. and KÃď lviÃď inen H.
Simultaneous Camera Calibration and Temporal Alignment of 2D and 3D Trajectories.
DOI: 10.5220/0006126304450450
In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*, pages 445-450
ISBN: 978-989-758-227-1

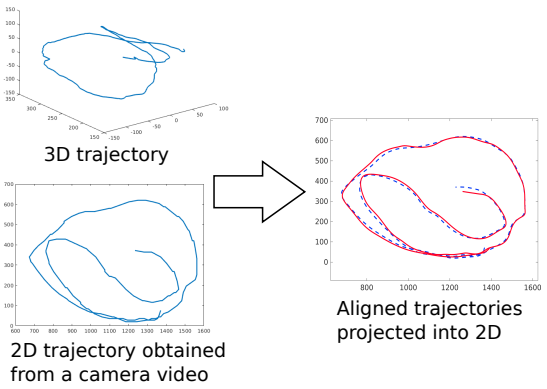

Figure 1: Alignment of 2D and 3D trajectories.

Although the motivation for the work comes partly from high-speed imaging, in the experimental part of the work, the method is demonstrated using standard frame rate videos mainly due to computational issues. However, the method does not make any assumptions about frame rates and can be straightforwardly generalized to high-speed videos with increase in computation time being the only downside.

2 RELATED WORK

Camera auto-calibration is a widely studied topic (Triggs, 1998; Sturm, 2000; Zhao and Lv, 2012; Liu et al., 2003; El Akkad et al., 2014). The aim is to calibrate the camera, i.e., determine the camera parameters from multiple images of an arbitrary scene without any calibration target or object. The basic idea is to use image point correspondences between the images with different views to estimate the intrinsic and extrinsic camera parameters, and to reconstruct 3D structure of a scene. In addition, single image auto-calibration techniques do exist. For example, in (Wu et al., 2007), a method to estimate camera parameters from a single image using vanishing points and RANSAC algorithm was proposed. In (Rahimi et al., 2004), a method to automatically calibrate cameras in a multi-camera system based on object trajectories was presented. However, the method requires synced cameras and the problem of temporal alignment of the trajectories was not considered.

Various methods are available for temporal alignment and fusion of 2D and 3D trajectories. In (Rangarajan et al., 1993), a method to match 2D trajectories with 3D trajectories using scale-space representations was presented. In (Knoop et al., 2009), an iterative closest point (ICP) based method was proposed to fuse 2D and 3D sensor data for human motion capture. In (Caspi and Irani, 2000), an approach to align two image sequences using both spatial and temporal information available within the sequence was presented. Besides the last method that only considers 2D trajectories, all the methods require a precalibrated setup. In (Noguchi and Kato, 2007), a method that simultaneously finds the lag in shutter timing between unsynchronized cameras and calibrates the cameras was proposed. However, the method assumes that the sensors (cameras) have the same frame rate which often is not the case in systems consisting of multiple types of sensors (e.g. a camera and 3D sensor). To the best of our knowledge, no method exist to simultanously align 2D and 3D trajectories with an arbitrary delay and frame rates and to calibrate the camera with respect to the 3D coordinates.

3 PROPOSED METHOD

Our method is inspired by the Random Sample Consensus (RANSAC) algorithm (Fischler and Bolles, 1981). Given 2D and 3D trajectories consisting of sets of points ($T_{2D} = \{\mathbf{p_1}, \mathbf{p_2}, ..., \mathbf{p_M}\}$, $T_{3D} = \{\mathbf{q_1}, \mathbf{q_2}, ..., \mathbf{q_L}\}$), the method starts by selecting two random points from both trajectories ($\{\mathbf{p_{k1}}, \mathbf{p_{k2}}\}$ and $\{\mathbf{q_{l1}}, \mathbf{q_{l2}}\}$, respectively). These points are assumed to represent the same moment in time, and based on this assumption, a temporal alignment between the two trajectories is made. That is, the delay (t_t) and ratio of frame rates (s_t) are computed as

$$\begin{aligned} s_t &= \frac{k_2 - k_1}{l_2 - l_1} \\ t_t &= k_1 - s_t l_1, \end{aligned} \tag{1}$$

where k_1 and k_2 are the indices of the selected random points in the 2D trajectory and l_1 and l_2 are the point indices in the 3D trajectory.

Based on the alignment parameters ($\{t_t, s_t\}$), a corresponding point for each point in the 2D trajectory (T_{2D}) is computed from the 3D trajectory (T_{3D}) using linear interpolation resulting a new 3D trajectory ($\tilde{T}_{3D} = \{\tilde{\mathbf{q}}_\mathbf{1}, \tilde{\mathbf{q}}_\mathbf{2}, ..., \tilde{\mathbf{q}}_\mathbf{L}\}$) with the same number of points as in the 2D trajectory. N ($N \geq 6$) random point correspondences, i.e., $\{\mathbf{p_k}, \tilde{\mathbf{q}}_\mathbf{k}\}$ pairs are then selected from the trajectories. These point correspondences are used to estimate the camera matrix

$$M = \begin{pmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} \end{pmatrix} \tag{2}$$

using a linear least squares algorithm with an extra restriction of $m_{1,1} = 1$ to avoid the trivial solutions

($m_{i,j} = 0$ for all $i = 1,2,3$ and $j = 1,2,3,4$) (Abdel-Aziz, 1971; Heikkila and Silvén, 1997). Also Newton's method was tested to further optimize the solution, but since it increased the computation time and did not significantly improve the results, it was not used in the experimental part of the work.

In each iteration, the goodness (optimization criterion) of the estimated camera matrix is computed. Two different optimization criteria were considered. The first one is similar to the RANSAC algorithm. All the points in the trajectory $\tilde{T}_{3D}$ are reprojected into 2D using the camera matrix M resulting a 2D trajectory $\tilde{T}_{2D} = \{\tilde{\mathbf{p_1}}, \tilde{\mathbf{p_2}}, ..., \tilde{\mathbf{p_M}}\}$. Next, the Euclidean distance between each point in the 2D trajectory (T_{2D}) and corresponding point in the reprojected 3D trajectory ($\tilde{T}_{2D}$) are computed as

$$d_i = \sqrt{(p_{i,1} - \tilde{p_{i,1}})^2 + (p_{i,2} - \tilde{p_{i,2}})^2} \quad (3)$$

The first optimization criterion (Criterion 1) is defined as the number of inliers, i.e., points for which the Euclidean distance is smaller than threshold τ:

$$g = \sum_{i=1}^{M} x_i, \quad (4)$$

where

$$x_i = \begin{cases} 0, & d_i > \tau \\ 1, & d_i < \tau \end{cases} \quad (5)$$

The second optimization criterion (Criterion 2) is computed as the mean Euclidean distance over all points in T_{2D}:

$$g = \frac{1}{M} \sum_{i=1}^{M} d_i. \quad (6)$$

The above steps are iteratively repeated for a predefined number of times, and finally the best camera matrix is selected based on the chosen optimization criterion with the corresponding alignment parameters. The whole algorithm is summarized in Algorithm 1.

Algorithm 1: Simultaneous camera calibration and temporal alignment.

1: **Input**: 2D trajectory (T_{2D}) and 3D trajectory (T_{3D})
2: **Output**: frame rate ratio s_t, delay t_t, and estimated camera matrix $\hat{M}$
3: **while** iteration $< k$ **do**
4: Randomly select point pairs $\{\mathbf{p_{k1}}, \mathbf{p_{k2}}\}$ and $\{\mathbf{q_{l1}}, \mathbf{q_{l2}}\}$ from the 2D and 3D trajectories T_{2D} and T_{3D}
5: Randomly select N points from the 2D trajectory.
6: Assuming $\mathbf{p_{k1}}$ corresponds to $\mathbf{q_{l1}}$, and $\mathbf{p_{k2}}$ to $\mathbf{q_{l2}}$, use interpolation to find point correspondences for the selected N points in the 3D trajectory.
7: Estimate the camera matrix M using the linear least squares algorithm.
8: Reproject the 3D trajectory points to 2D using M.
9: Compute the reprojection error for all points in the 2D trajectory using Euclidean distance.
10: Compute the goodness of the estimated camera matrix using Eq. 4 or 6.
11: If the goodness is higher than any previous one, update the best camera matrix ($\hat{M}$).
12: **end while**
13: Compute parameters s_t and t_t for temporal alignment using Eq. 1.
14: Recompute $\hat{M}$ using all inliers.

4 EXPERIMENTS

4.1 Experimental Arrangements

The 3D trajectories of the index finger of a human hand and a pencil were recorded using a Leap Motion sensor [1]. A fixed frame rate of 50 fps was used instead of the default varying frame rate. It was noted that, at times, the Leap Motion lost the track of the finger or pencil giving no coordinates for individual points. For those points, new coordinates were estimated by using the neighboring points and interpolation based on cubic splines. The Leap Motion software provides filtered coordinates for more robust gesture recognition. Both filtered and unfiltered coordinates were saved for further analysis.

The 2D videos were recorded using a standard web camera with 20 fps. The 2D trajectories of the finger and pencil were obtained by using the Kernelized Correlation Filters (KCF) tracking method (Henriques et al., 2015) that was found suitable for similar tracking tasks in (Kuronen et al., 2015). The trajectories produced by the tracker contained noise. A typical error was that when the initial point was set onto the tip of the finger, the tracker window would move closer to the joint between the intermediate phalange. This error was minimized by manually optimizing the size and aspect ratio of the tracking window individually for the videos, for which the tracking was erroneous.

The noisy trajectories were further smoothed using Local regression using weighted linear least squares (LOESS) (Cleveland, 1979) and a 2nd degree

[1] https://www.leapmotion.com/

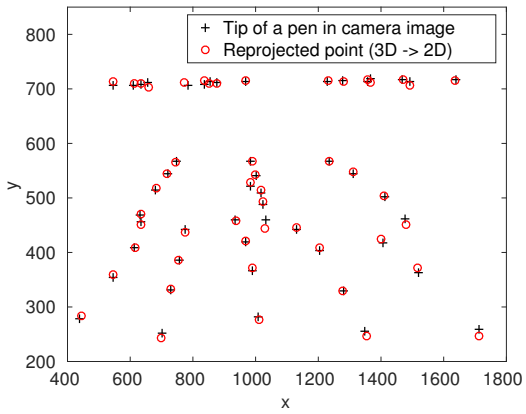

Figure 2: Ground truth for camera calibration.

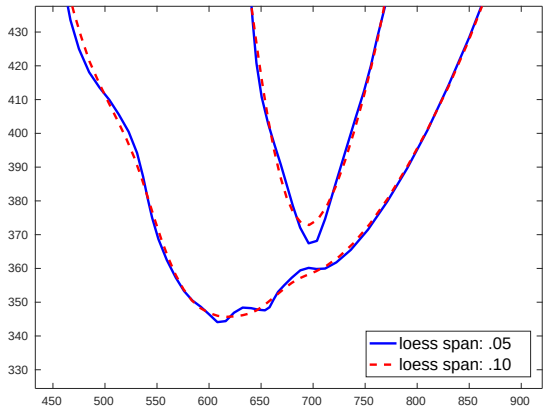

Figure 3: Effect of LOESS span.

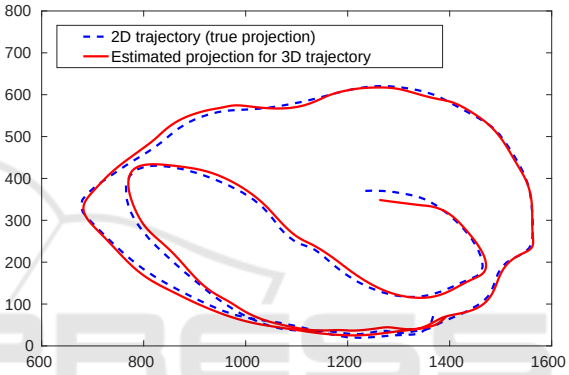

Figure 4: Example of successful camera calibration and temporal alignment.

polynomial model with varying spans (5% or 10% of data points). In total, 29 trajectories (15 trajectories for the finger and 14 for the pencil), each lasting 10 seconds, were recorded. The location of the camera with respect to the Leap Motion sensor was varied.

The ground truth for the camera calibration was acquired by capturing images of a pencil mounted on a stand. The location of the tip of the pencil in the Leap Motion coordinates was recorded simultaneously. The tip of the pencil was marked in the images manually. This was repeated 50 times with different locations of the pencil. The camera matrix was estimated using all the points and the linear least squares approach algorithm. There was one outlier for which the reprojection error was over 50 pixels. The outlier was left out and the camera matrix estimation was repeated with the remaining 49 points. For the ground truth measurements, the mean reprojection error was 5.6 pixels and 5.3 pixels for the unfiltered and filtered Leap Motion data, respectively (see Fig. 2).

4.2 Results

The proposed method was applied for both filtered and unfiltered Leap Motion data. The results are shown in Tables 1 and 2, respectively. The performance measures used were 1) success rate (%), 2) average distance between the true 2D trajectory and the reprojected 3D trajectory in pixels, and 3) the percentage of inliers (projected points for which the distance was smaller than the RANSAC inlier threshold). The experiment was repeated for different LOESS spans (5% or 10% of data points), number of points (N) used to estimate the camera matrix, and minimum distance (D) between the two selected random points. The other tunable parameters were set as follows: the number of iterations was 1000 and the inlier threshold for RANSAC was 10 pixels.

As can be seen from Table 1, the method did not work well with the filtered 3D data. A typical problem was that the trajectory was estimated to be only a part of the projection. Moreover, even in the cases were the method worked for filtered data, the reprojection error was smaller with the unfiltered data.

Using LOESS with the 10 percent span for filtering the 2D trajectories resulted in slightly better results than with the 5 percent span. However, it should be noted that when using a large span, the trajectories already begin to loose their shape as can be seen from Fig. 3. Therefore, a shorter span is recommended. The method performed better when the selected two random points were not allowed to be too close to each other. When no such restraint was set, the whole 3D trajectory was often projected onto a single very short line. The minimum of 6 points to estimate the camera matrix was enough to achieve high success rate. With a higher number of points, however, the reprojection error was smaller. Fig. 4 shows a typical successful camera calibration and temporal alignment result.

Out of the two optimization criteria, Criterion 2 (mean distance) outperformed Criterion 1 (number

Table 1: Results with filtered leap motion data and different combinations of method parameters (N is the number of points used to estimate the camera matrix and D is the minimum distance between the two selected random points).

Method parameters				Performance measures		
Optimization Criterion	LOESS span (%)	N	D	Average distance	Inliers (%)	Success rate (%)
Criterion 1	5	6	1	-	-	0.0
Criterion 1	5	10	1	-	-	0.0
Criterion 1	10	6	1	-	-	0.0
Criterion 1	10	10	1	17.77	92	3.4
Criterion 1	5	6	90	31.53	77	10.3
Criterion 1	5	10	90	23.24	77	31.0
Criterion 1	10	6	90	210.07	83	24.1
Criterion 1	10	10	90	20.99	79	48.3
Criterion 2	5	6	1	-	-	0.0
Criterion 2	5	10	1	-	-	0.0
Criterion 2	10	6	1	-	-	0.0
Criterion 2	10	10	1	32.07	44	3.4
Criterion 2	5	6	90	47.09	54	62.1
Criterion 2	5	10	90	27.65	67	58.6
Criterion 2	10	6	90	36.77	70	65.5
Criterion 2	10	10	90	31.06	67	58.6

Table 2: Results with unfiltered Leap Motion data.

Method parameters				Performance measures		
Optimization Criterion	LOESS span (%)	N	D	Average distance	Inliers (%)	Success rate (%)
Criterion 1	5	6	1	-	-	0.0
Criterion 1	5	10	1	-	-	0.0
Criterion 1	10	6	1	-	-	0.0
Criterion 1	10	10	1	17.20	94	3.4
Criterion 1	5	6	90	58.37	81	24.1
Criterion 1	5	10	90	23.80	74	72.4
Criterion 1	10	6	90	19.12	81	24.1
Criterion 1	10	10	90	19.51	79	72.4
Criterion 1	5	15	45	19.33	81	82.8
Criterion 1	5	15	90	21.50	79	82.8
Criterion 2	5	6	1	20.49	79	3.4
Criterion 2	5	10	1	18.65	86	10.3
Criterion 2	10	6	1	12.27	74	3.4
Criterion 2	10	10	1	21.79	72	13.8
Criterion 2	5	6	90	28.85	61	**96.6**
Criterion 2	5	10	90	23.11	69	93.1
Criterion 2	10	6	90	25.94	68	**96.6**
Criterion 2	10	10	90	21.54	71	**96.6**
Criterion 2	5	15	45	22.09	72	**96.6**
Criterion 2	5	15	90	21.50	73	**96.6**

of inliers). Criterion 1 failed to estimate the matrix correctly in approximately one-third of the cases, whereas Criterion 2 achieved over 96% success rate. However, in those cases where Criterion 1 worked, it outperformed Criterion 2 with respect to the accuracy. Fig. 5 shows a comparison of the two criteria with a single trajectory.

5 CONCLUSION

We proposed a method to simultaneously calibrate the camera and to temporally align 2D and 3D motion trajectories obtained by using a camera and Leap Motion sensor. The experiments showed that by properly tuning the method parameters, the approach

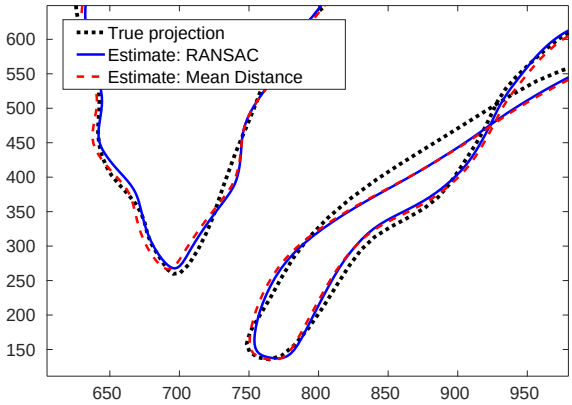

Figure 5: Comparison of the optimization criteria.

achieved a high success rate with a test set consisting of complex trajectories. The best results were acquired by using unfiltered Leap Motion data, LOESS smoothed 2D trajectory data obtained using the KCF tracker, mean distance based optimization criterion, 10 points for the camera matrix estimation, and by restraining the minimum distance between the two random points selected in each iteration. With the current nonoptimized, single-core MATLAB implementation, the camera calibration and temporal alignment takes about 30 minutes for trajectories of 10 seconds. Future work will include enhancing the computation performance in order to make the method efficient for high-speed imaging. Besides combining the trajectories recorded with a camera and 3D sensor, the method provides an intuitive way to perform the camera calibration and holds potential in other similar applications.

ACKNOWLEDGEMENTS

The research was carried out as part of the COPEX project (No. 264429) funded by the Academy of Finland.

REFERENCES

Abdel-Aziz, Y. (1971). Direct linear transformation from comparator coordinates in close-range photogrammetry. In *ASP Symposium on Close-Range Photogrammetry*.

Caspi, Y. and Irani, M. (2000). A step towards sequence-to-sequence alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 682–689.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836.

El Akkad, N., Merras, M., Saaidi, A., and Satori, K. (2014). Camera self-calibration with varying intrinsic parameters by an unknown three-dimensional scene. *The Visual Computer*, 30(5):519–530.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Heikkila, J. and Silvén, O. (1997). A four-step camera calibration procedure with implicit image correction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1112.

Henriques, J. F., Caseiro, R., Martins, P., and Batista, J. (2015). High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596.

Hiltunen, V., Eerola, T., Lensu, L., and Kalviainen, H. (2014). Comparison of general object trackers for hand tracking in high-speed videos. In *2014 22nd International Conference on Pattern Recognition (ICPR)*, pages 2215–2220. IEEE.

Knoop, S., Vacek, S., and Dillmann, R. (2009). Fusion of 2d and 3d sensor data for articulated body tracking. *Robotics and Autonomous Systems*, 57(3):321–329.

Kuronen, T., Eerola, T., Lensu, L., Takatalo, J., Häkkinen, J., and Kälviäinen, H. (2015). High-speed hand tracking for studying human-computer interaction. In *Scandinavian Conference on Image Analysis*, pages 130–141.

Liu, P., Shi, J., Zhou, J., and Jiang, L. (2003). Camera self-calibration using the geometric structure in real scenes. In *Computer Graphics International, 2003. Proceedings*, pages 262–265.

Noguchi, M. and Kato, T. (2007). Geometric and timing calibration for unsynchronized cameras using trajectories of a moving marker. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 20–20. IEEE.

Rahimi, A., Dunagan, B., and Darrell, T. (2004). Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–187.

Rangarajan, K., Allen, W., and Shah, M. (1993). Matching motion trajectories using scale-space. *Pattern recognition*, 26(4):595–610.

Sturm, P. (2000). A case against kruppa's equations for camera self-calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10):1199–1204.

Triggs, B. (1998). Autocalibration from planar scenes. In *European Conference on Computer Vision (ECCV)*, pages 89–105.

Wu, Q., Shao, T.-C., and Chen, T. (2007). Robust self-calibration from single image using ransac. In *Third International Symposium on Visual Computing*, pages 230–237.

Zhao, Y. and Lv, X. (2012). An approach for camera self-calibration using vanishing-line. *Information Technology Journal*, 11(2):276–282.