Auto-calibration project report

Table of Contents

Camera model

We use a so-called pinhole camera model. In this model, a scene view is formed by projecting 3D points into the image plane using a perspective transformation.

$$s\,m = K[R|T]M$$

or

$$s\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}f_x & s & c_x\\0 & f_y & c_x\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_1\\r_{21} & r_{22} & r_{23} & t_y\\r_{31} & r_{32} & r_{33} & t_3\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

where:

- (u,v) are the coordinates of the projection point in pixels (i.e. image coordinates),
- K is a matrix of intrinsic parameters (intrinsics),
- f_x, f_y are the focal lengths expressed in pixel-related units,
- (c_x, c_y) is a principal point that is usually at the image center,
- s is the skew coefficient between the x and the y axis,
- (X,Y,Z) are the coordinates of a 3D point in the world coordinate space.

When we talk about a single pinhole camera we'll call it a mono camera sometimes – to highlight contrast with a stereo camera. Under a stereo camera we understand a pair of pinhole cameras:

$$s\,m_1 = K[I|0]M$$

$$s\, m_2 = K [R|T] M$$

where:

- m_1, m_2 are the coordinates of the projection point in pixel in the first and the second images respectively,
- R, T are relative rotation and translation between the first and the second camera coordinate systems,
- note that the intrinsic parameters for each camera in the pair are equal.

Problem statement

Auto-calibration is the process of determining internal camera parameters directly from multiple uncalibrated images.

Here is the table of parameters to auto-calibrate in cases of mono and stereo cameras:

Camera type	**Parameters to auto-calibrate**
Mono camera	K
Stereo camera	K, R, T

Rotating camera auto-calibration

In this work we investigate a problem of mono camera auto-calibration which doesn't undergo translations, i.e. it's fixed at a point and can be rotated only. It's assumed that the camera matrix K remains constant during the whole image sequence.

Stereo camera auto-calibration

Also we investigate a problem of stereo camera auto-calibration which undergoes any movements, but the relative rotation R and translation T between the cameras in the pair are fixed over the whole images sequence. The camera matrix K is assumed to be constant during the sequence too.

Investigated approaches

Rotating camera auto-calibration

Here is the pseudo-code of the approach that was implemented to solve the problem of rotating camera auto-calibration:

1. Find keypoints and descriptors in all images
2. Match all image pairs
3. Estimate pairwise 2D projective transformations between images using the algorithm proposed in *[1] "4 Estimation – 2D Projective Transformations"*
4. Compute pairwise match confidences using the algorithm proposed in *[2] "3.2 Probabilistic Model for Image Match Verification"*
5. Build a weighted graph, where vertices are images, edges are matches weighted with its confidences
6. Remove edges with confidences lower than the given threshold
7. Find the maximum spanning tree and remove the rest part of the graph
8. Run the auto-calibration algorithm proposed in *[1] "19.6 Calibration from rotating cameras"* on the remaining images and matches with the iterative improvement step turned on. See the *figure 1*.

Objective

Given $m \geq 2$ views acquired by a camera rotating about its centre with fixed or varying internal parameters, compute the parameters of each camera. It is assumed that the rotations are not all about the same axis.

Algorithm

(i) **Inter-image homographies:** Compute the homography $\mathtt{H}^i$ between each view i and a reference view such that $\mathbf{x}^i = \mathtt{H}^i\mathbf{x}$ using, for example, algorithm 4.6(p123). Normalize the matrices such that $\det \mathtt{H}^i = 1$.

(ii) **Compute ω:**

- In the case of constant calibration: rewrite the equations $\boldsymbol{\omega} = (\mathtt{H}^i)^{-\mathsf{T}}\boldsymbol{\omega}(\mathtt{H}^i)^{-1}$, $i = 1, \ldots, m$ as $\mathtt{A}\mathbf{c} = \mathbf{0}$ where $\mathtt{A}$ is a $6m \times 6$ matrix, and $\mathbf{c}$ the elements of the conic $\boldsymbol{\omega}$ arranged as a 6-vector, **or**
- For variable calibration parameters, use the equation $\boldsymbol{\omega}^i = (\mathtt{H}^i)^{-\mathsf{T}}\boldsymbol{\omega}(\mathtt{H}^i)^{-1}$ to express linear constraints on entries of $\boldsymbol{\omega}^i$ in table 19.4 (e.g. unit aspect ratio) as linear equations in the entries of $\boldsymbol{\omega}$.

(iii) **Compute $\mathtt{K}$:** Determine the Cholesky decomposition of $\boldsymbol{\omega}$ as $\boldsymbol{\omega} = \mathtt{U}\mathtt{U}^{\mathsf{T}}$, and thence $\mathtt{K} = \mathtt{U}^{-\mathsf{T}}$.

(iv) **Iterative improvement:** (Optional) Refine the linear estimate of $\mathtt{K}$ by minimizing

$$\sum_{i=2,m;\ j=1,n} d(\mathbf{x}_j^i, \mathtt{K}\mathtt{R}^i\mathtt{K}^{-1}\mathbf{x}_j)^2$$

over $\mathtt{K}$ and $\mathtt{R}^i$, where $\mathbf{x}_j, \mathbf{x}_j^i$ are the position of the j-th point measured in the first and i-th images respectively. Initial estimates for the minimization are obtained from $\mathtt{K}$ and $\mathtt{R}^i = \mathtt{K}^{-1}\mathtt{H}^i\mathtt{K}$.

Figure 1. Calibration of a camera rotating about its centre

Stereo camera auto-calibration

Here is the pseudo-code for the approach that was implemented to solve the problem of stereo camera auto-calibration:

1. Find keypoints and descriptors in all images
2. Find matches between left images of all stereo pairs
3. For each stereo pair:
    1. Find matches between left and right images of the current stereo pair
4. Find the fundamental matrix for each matched image pair using the algorithm proposed in *[1] "11 Computation of the Fundamental Matrix F"*
5. For each stereo pair:
    1. Retrieve camera matrices from the respective fundamental matrix up to a projective transform using the algorithm proposed in *[1] "9.5 Retrieving the camera matrices"*
6. For each two stereo pairs matched via the left images:
    1. Find the plane-at-infinity using the algorithm proposed in *[1] "Auto-calibration of a stereo rig"*, see *figure 2*
    2. Upgrade the projective reconstruction and camera matrices to affine reconstruction and camera matrices using found plane-at-infinity
7. Estimate infinity homographies (*[1] "Auto-calibration of a stereo rig"*)
8. For each stereo pair upgrade the affine reconstructions and camera matrices to metric reconstructions and camera matrices using the initial guess of intrinsic parameters
9. Perform non-linear minimization of the overall point-to-epipolar distance varying camera intrinsics and relative R and T for each stereo pair

Objective

Given two (or more) stereo pairs of images acquired by a fixed stereo rig undergoing general motions (i.e. both $\mathtt{R}$ and $\mathbf{t}$ are non-zero, and $\mathbf{t}$ not perpendicular to the axis of $\mathtt{R}$), compute an affine reconstruction.

Algorithm

(i) **Compute an initial projective reconstruction $\mathbf{X}$:** Using the first stereo pair compute a projective reconstruction $(\mathtt{P}^{\mathrm{L}}, \mathtt{P}^{\mathrm{R}}, \{\mathbf{X}_j\})$ as described in chapter 10. This involves computing the fundamental matrix $\mathtt{F}$ and point correspondences between the images of the first pair $x_j^{\mathrm{L}} \leftrightarrow x_j^{\mathrm{R}}$, e.g. use algorithm 11.4(p291).

(ii) **Compute a projective reconstruction $\mathbf{X}'$ after the motion:** Compute correspondences between the images of the second stereo pair $x_j'^{\mathrm{L}} \leftrightarrow x_j'^{\mathrm{R}}$. Since both the internal and relative external parameters of the cameras are fixed, the second stereo pair has the same fundamental matrix $\mathtt{F}$ as the first. The same camera matrices $\mathtt{P}^{\mathrm{L}}, \mathtt{P}^{\mathrm{R}}$ are used for triangulating points $\mathbf{X}_j'$ in 3-space from the computed correspondences $x_j'^{\mathrm{L}} \leftrightarrow x_j'^{\mathrm{R}}$ in the second stereo pair.

(iii) **Compute the 4×4 matrix $\mathtt{H}_{\mathrm{P}}$ which relates $\mathbf{X}$ to $\mathbf{X}'$:** Compute correspondences between the left images of the two stereo pairs $x_j^{\mathrm{L}} \leftrightarrow x_j'^{\mathrm{L}}$ (e.g. again use algorithm 11.4(p291)). This establishes correspondences between the space points $\mathbf{X}_j \leftrightarrow \mathbf{X}_j'$. The homography $\mathtt{H}_{\mathrm{P}}$ may be estimated linearly from five or more of these 3-space point correspondences, and then the estimate refined by minimizing a suitable cost function over $\mathtt{H}_{\mathrm{P}}$. For example, minimizing $\sum_j (d(x_j^{\mathrm{L}}, \mathtt{P}^{\mathrm{L}}\mathtt{H}\mathbf{X}_j')^2 + d(x_j^{\mathrm{r}}, \mathtt{P}^{\mathrm{R}}\mathtt{H}\mathbf{X}_j')^2)$ minimizes the distance between the measured and reprojected image points.

(iv) **Affine reconstruction:** Compute $\boldsymbol{\pi}_\infty$ from the real eigenvector of $\mathtt{H}_{\mathrm{P}}^{\mathsf{T}}$ and thence an affine reconstruction.

Figure 2. Affine calibration of a stereo rig

Algorithms evaluation

Rotating camera auto-calibration

We tested the implemented algorithm on two cameras:

Camera	**Resolution**
Nokia 6303C mobile phone	1536x2048
Logitech QuickCam Pro 9000	1600x1200

Nokia 6303C mobile phone camera results

OpenCV circles-based calibration results				
fx	fy	cx	cy	s
2212	2208	820	969	0

Dataset	Auto-calibration results				
	fx	fy	cx	cy	s
1 (~0.5m far, 8 images)	2890	3005	792	649	0
2 (~1.5m far, 13 images)	2161	2266	815	1066	0
3 (~1.5m far, 9 images)	2169	2248	836	1029	0
4 (~30m far, 14 images)	2350	2363	802	1013	0

Dataset	Auto-calibration errors				
	fx, rel. err.	fy, rel. err.	cx, rel. err.	cy, rel. err.	s, abs. err.
1 (~0.5m far, 8 images)	30.65%	36.10%	3.41%	33.02%	0
2 (~1.5m far, 13 images)	2.31%	2.63%	0.61%	10.01%	0
3 (~1.5m far, 9 images)	1.94%	1.81%	1.95%	6.19%	0
4 (~30m far, 14 images)	6.24%	7.02%	2.20%	4.54%	0

Logitech QuickCam Pro 9000 camera results

OpenCV circles-based calibration results				
fx	fy	cx	cy	s
1327	1334	801	586	0

Dataset	Auto-calibration results				
	fx	fy	cx	cy	s
1 (~2m far, 10 images)	1333.62	1404.35	829.563	584.244	0
1 (~2m far, 30 images)	1342.91	1392.13	806.214	597.732	0
1 (~2m far, 57 images)	1330.43	1374.83	812.832	604.807	0
2 (~2m far, 10 images)	1303.56	1343.51	781.173	593.703	0
2 (~2m far, 30 images)	1351.81	1413.5	798.371	636.249	0
2 (~2m far, 74 images)	1328.74	1391.36	802.535	630.771	0

Dataset	Auto-calibration errors				
	fx, rel. err.	fy, rel. err.	cx, rel. err.	cy, rel. err.	s, abs. err.
1 (~2m far, 10 images)	0.50%	5.27%	3.57%	0.30%	0
1 (~2m far, 30 images)	1.20%	4.36%	0.65%	2.00%	0
1 (~2m far, 57 images)	0.26%	3.06%	1.48%	3.21%	0
2 (~2m far, 10 images)	1.77%	0.71%	2.48%	1.31%	0
2 (~2m far, 30 images)	1.87%	5.96%	0.33%	8.57%	0
2 (~2m far, 74 images)	0.13%	4.30%	0.19%	7.64%	0

Stereo camera auto-calibration

We tested the algorithm on two stereo cameras:

Camera	**Resolution**
Videre	640x480
LG-P920 phone	1600x1200

Videre stereo camera results

Ground truth intrinsics and relative rotation, translation between left and right cameras were obtained via the camera API.

Ground truth intrinsic parameters				
fx	fy	cx	cy	s
425	425	340	245	0

Ground truth rotation vector			Ground truth normalized translation		
rx	ry	rz	tx	ty	tz
-0.005	-0.0002	0.001	1	-0.004	-0.012

To perform evaluation of the stereo auto-calibration algorithm we generated a set of intrinsics initial guesses with relative errors from the [-50%, 50%] range, except skew, which was set to zero in all runs. 34 runs totally were performed on each dataset, 1 run on the first dataset was filtered out because of high final overall point-to-epipolar-line error (the threshold was set empirically). Results are below:

Dataset	Metric	Auto-calibrated intrinsics				
		fx	fy	cx	cy	s
1 (~30cm far, 5 pairs)	average	438.922	437.226	341.829	252.496	0.000
	stddev	5.907	10.578	8.422	23.699	0.000
2 (~30cm far, 5 pairs)	average	410.690	408.695	335.306	251.242	0.000
	stddev	0.126	0.102	0.141	0.300	0.000

Dataset	Metric	Auto-calibrated rotation vector			Auto-calibrated normalized translation		
		rx	ry	rz	tx	ty	tz
1 (~30cm far, 5 pairs)	average	0.021	0.007	0.006	1.000	0.020	-0.028
	stddev	0.001	0.001	0.000	0.000	0.002	0.005
2 (~30cm far, 5 pairs)	average	0.021	0.007	0.006	1.000	-0.001	-0.053
	stddev	0.001	0.001	0.000	0.000	0.000	0.000

LG-P920 phone stereo camera results

OpenCV chess-based calibration				
fx	fy	cx	cy	s
2592.36	2592.36	1071.67	742.54	0

Rotation vector (OpenCV)			Normalized translation (OpenCV)		
rx	ry	rz	tx	ty	tz
0.00	0.04	0.00	1.00	-0.04	-0.12

To perform evaluation of the stereo auto-calibration algorithm we generated a set of intrinsics initial guesses with relative errors from the [-30%, 30%] range, except skew, which was set to zero in all runs. 6 runs were performed on the first dataset, and 28 runs were performed on the second dataset. 3 runs on the second dataset were filtered out because of high final overall point-to-epipolar-line error (the threshold was set empirically). Results are below:

Dataset	Metric	Auto-calibrated intrinsics				
		fx	fy	cx	cy	s
1 (~30cm far, 6 pairs, 6 runs)	average	2651.153	2654.990	970.189	749.188	-6.178
	stddev	135.919	126.533	57.037	54.550	6.441
2 (~30cm far, 26 pairs, 28-3 runs)	average	2663.181	2663.563	956.484	721.895	-0.296
	stddev	27.991	29.230	18.952	28.368	1.780

Dataset	Metric	Auto-calibrated rotation vector			Auto-calibrated normalized translation		
		rx	ry	rz	tx	ty	tz
1 (~30cm far, 6 pairs, 6 runs)	average	0.000	0.003	-0.003	1.000	-0.005	-0.170
	stddev	0.000	0.000	0.000	0.000	0.002	0.016
2 (~30cm far, 26 pairs, 28-3 runs)	average	0.001	0.002	-0.003	1.000	-0.015	-0.133
	stddev	0.000	0.000	0.000	0.000	0.002	0.002

Conclusion

During that project we were investigating two approaches of solving the auto-calibration problem proposed in [1]. Those algorithms were extended by taking into account some ideas from [2]. Also we modified the algorithms to select the best images and matches subset for further processing. To estimate absolute rotations and translations from a set of relative ones (for further refinement) we proposed a scheme using the maximum spanning tree which is build from the graph consisting of images as vertices and weighted with the confidence matches as edges. The whole pipeline was implemented to allow user to perform experiments and vary parameters as easy as possible.

It's possible to do calibration of cameras without any patterns, but the main point is that quality of data is very important for achieving high quality of auto-calibration. If a dataset is chosen properly then result would rather be good, if the dataset is poorly conditioned (camera undergoes only subset of available motions that can be handled by an algorithm, or all objects are low textured) results would be poor. Of course, all that requirements affects on how the auto-calibration algorithms can be used in practice.

There is one additional requirement in case of rotating cameras – translation must be relatively small in contrast with the distance from camera to objects. And there is one additional requirement in case of stereo camera – camera must be close enough to objects in contrast with the baseline (distance between left and right cameras in a stereo pair).

In fact, the investigated auto-calibration algorithms are quite sensitive to datasets and parameters, for instance, it took a long period of time to create the datasets and tune parameters to make stereo auto-calibration work with the LG camera, while it's very easy to find a dataset where it doesn't work well.

In a few rare cases the final step of the auto-calibration algorithms didn't converge closely enough to the ground truth, but it is always possible to detect such kind of failures by analyzing final error (the reprojection error in case of rotating camera and the point-to-epipolar-line error in case of stereo camera) – if it's greater than a threshold (selected experimentally) then it's probably a failure. Moreover, when we're dealing with auto-calibration and we know something a-priory about camera parameters, say, we know that skew is zero – which is quite common in practice, we can allow algorithm to vary it (like it was done in experiments with the LG stereo camera) and then we can use the difference between the value refined and the expected value as the measure of success.

All codes, datasets, and results are available here: http://code.google.com/p/autocalib/.

References

1. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision, 2nd edition*. Cambridge University Press, 2003
2. M. Brown and D. Lowe. *Automatic Panoramic Image Stitching using Invariant Features*. International Journal of Computer Vision, 74(1), pages 59-73, 2007