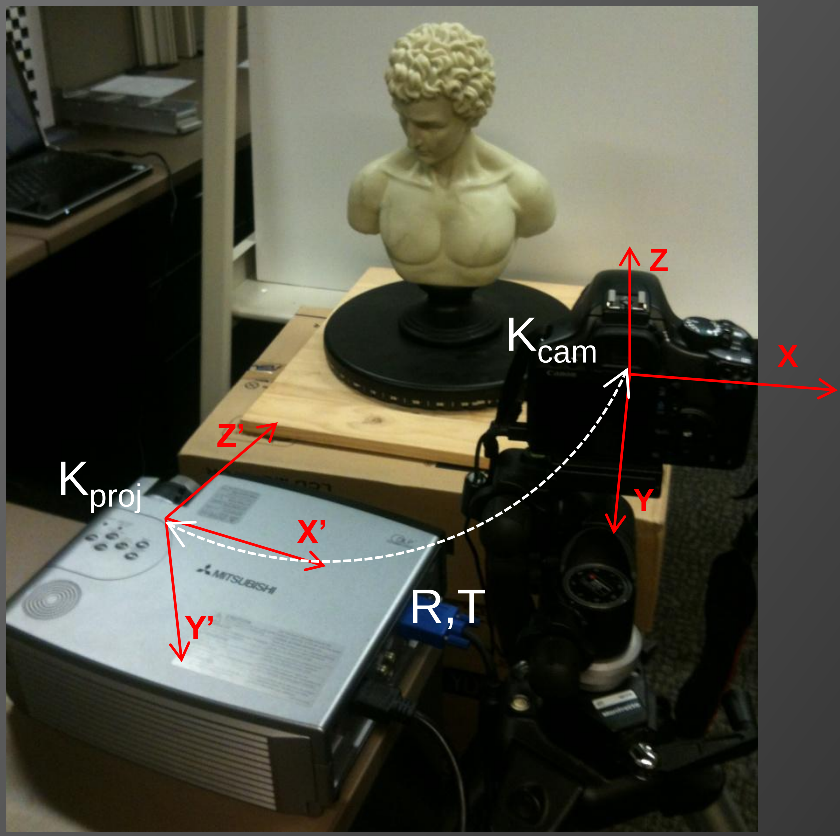
SIMPLE, ACCURATE, AND ROBUST PROJECTOR-CAMERA CALIBRATION

Daniel Moreno

October 2012

Overview



Geometric calibration

- Camera intrinsics: K_{cam}
- Projector intrinsics: K_{proj}

- Projector-Camera extrinsics:

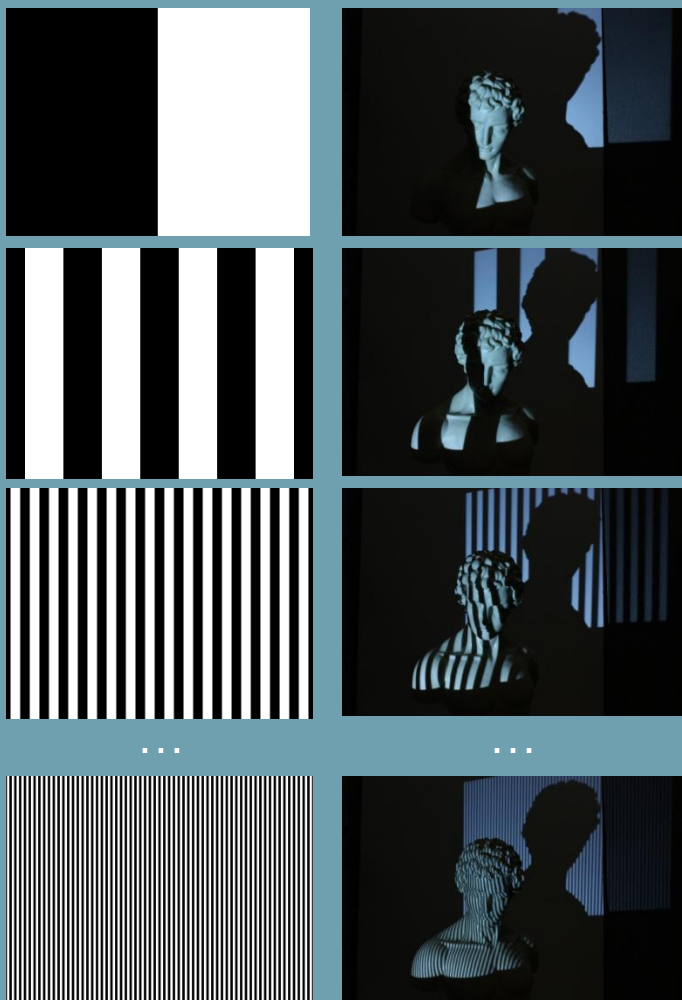
  Rotation and translation:

  R,T

The simplest structured-light system consists of a camera and a data projector.

Application: 3D scanning

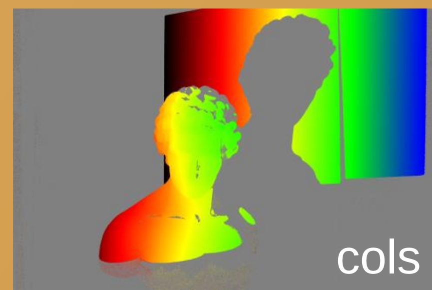
Projector | Camera

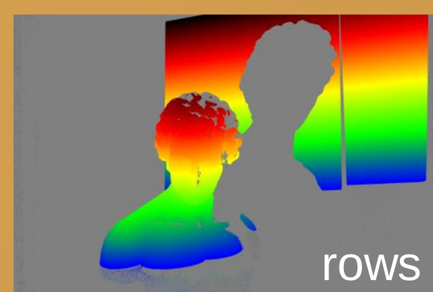


Data acquisition

1

Projector-Camera correspondences

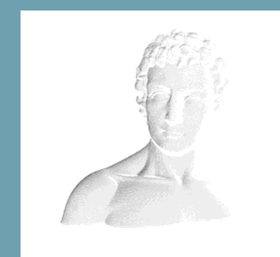




Decode

2

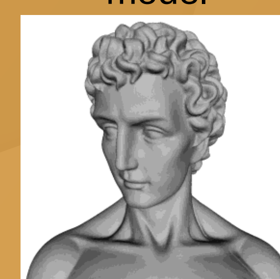
Correspondences
+
Calibration
=
Pointcloud



Triangulation

3

Pointclouds from several viewpoints can be merged into a single one and used to build a 3D model



Mesh

4

Camera calibration: well-known problem

Pinhole model + radial distortion

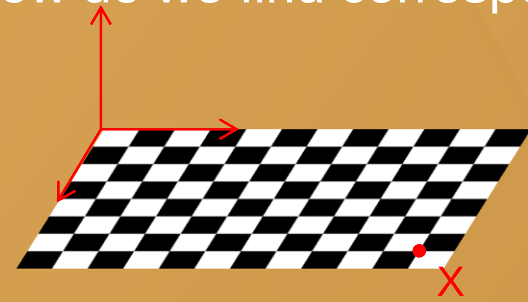
$$K = \begin{bmatrix} \mathrm{fx} & \mathrm{s} & \mathrm{cx} \\ 0 & \mathrm{fy} & \mathrm{cy} \\ 0 & 0 & 1 \end{bmatrix}$$

$$x = K \cdot L(X; k_1, k_2, k_3, k_4)$$

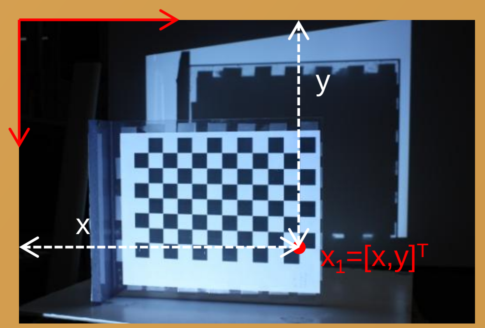
X: 3D point
$k_1,\ldots,k_4$: distortion coefficients
K: camera intrinsics
x: projection of X into the image plane

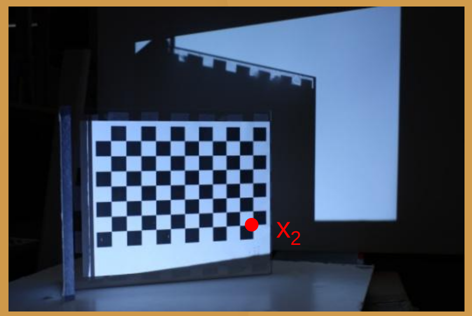
If we have enough X↔x point correspondences we can solve for all the unknowns

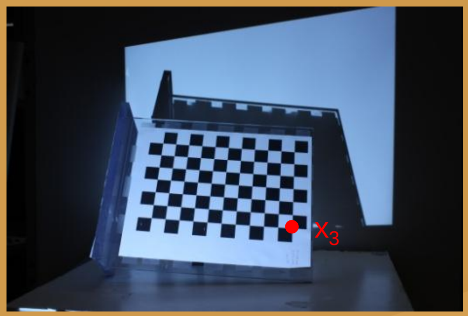
How do we find correspondences?



Object of known dimensions







$$x_1 = K \cdot L(R_1 X + T_1; k_1, k_2, k_3, k_4)$$

$$x_2 = K \cdot L(R_2 X + T_2; k_1, k_2, k_3, k_4)$$

$$x_3 = K \cdot L(R_3 X + T_3; k_1, k_2, k_3, k_4)$$

…

Images from different viewpoints

Projector calibration: ?

Use the pinhole model to describe the projector:

- Projectors work as an <u>inverse camera</u>

$$K_{proj} = \begin{bmatrix} \mathrm{fx} & \mathrm{s} & \mathrm{cx} \\ 0 & \mathrm{fy} & \mathrm{cy} \\ 0 & 0 & 1 \end{bmatrix}$$

$$x = K_{proj} \cdot L(X; k_1, k_2, k_3, k_4)$$

If we model the projector the same as our camera, we would like to calibrate the projector just as we do for the camera:

- We need correspondences between 3D world points and projector image plane points: X↔x
- The projector <u>cannot capture images</u>

Challenge: How do we find point correspondences?

Related works

There have been proposed several projector calibration methods*, they can be divided in three groups:

1. Rely on camera calibration
   - First the camera is calibrated, then, camera calibration is used to find the 3D world coordinates of the projected pattern
   - <u>Inaccuracies in the camera calibration</u> translates into errors in the projector calibration
2. Find projector correspondences using homographies between planes
   - Cannot model projector lens distortion because of the linearity of the transformation
3. Too difficult to perform
   - Required special equipments or calibration artifacts
   - Required color calibration
   - …

(*) See the paper for references

Existing methods were <u>not accurate</u> enough or <u>not practical</u>

Proposed method: overview

Features:

Simple to perform:

- no special equipment required
- reuse existing components

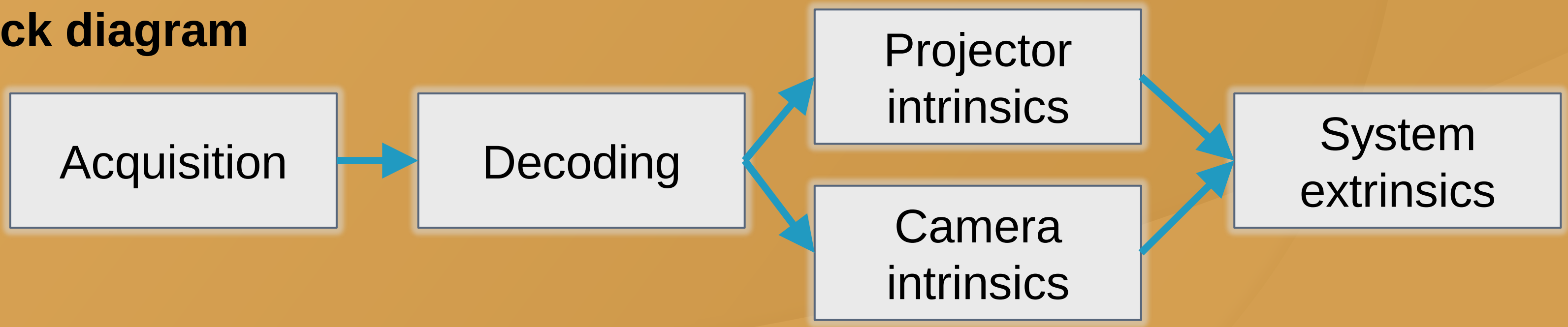
Accurate:

- there are no constrains for the mathematical model used to describe the projector
- we use the full pinhole model with radial distortion (as for cameras)

Robust:

- can handle small decoding errors

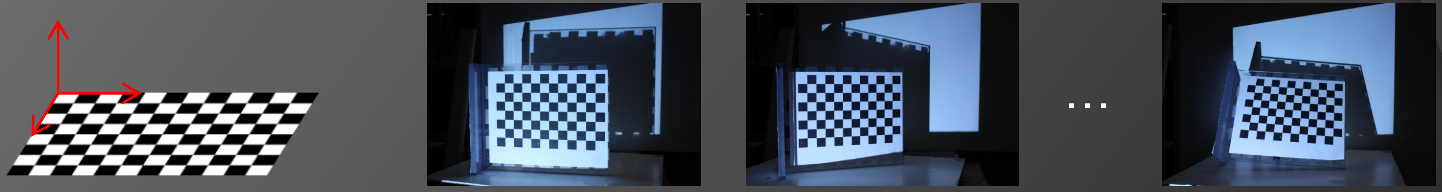
Block diagram



Proposed method: acquisition

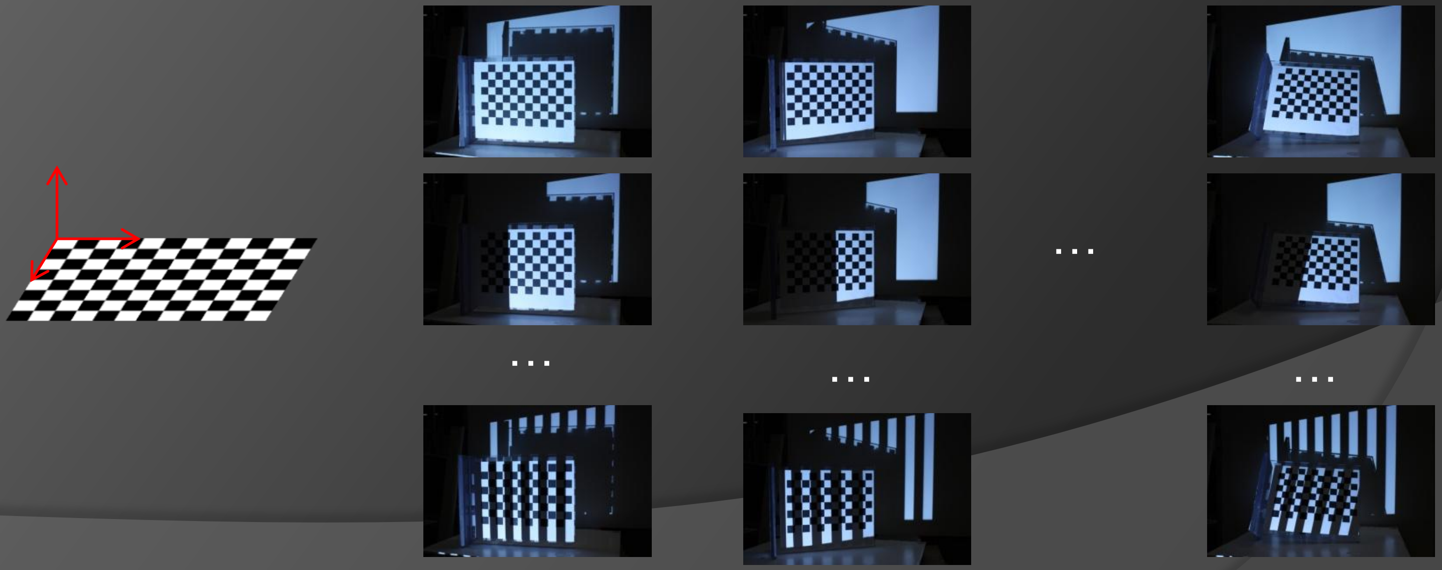
Traditional camera calibration

- requires a planar checkerboard (easy to make with a printer)
- capture pictures of the checkerboard from several viewpoints



Structured-light system calibration

- use a planar checkerboard
- capture structured-light sequences of the checkerboard from several viewpoints



Proposed method: decoding

Decoding depends on the projected pattern

- The method does not rely on any specific pattern

Our implementation uses complementary gray code patterns

- Robust to light conditions and different object colors (notice that we used the standard B&W checkerboard)
- Does not required photometric calibration (as phase-shifting does)
- We prioritize calibration accuracy over acquisition speed
- Reasonable fast to project and capture: if the system is synchronized at 30fps, the 42 images used for each pose are acquired in 1.4 seconds

Our implementation decodes the pattern using “robust pixel classification”(*)

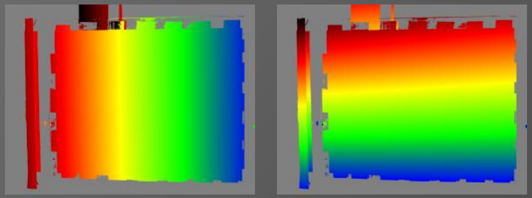
- High-frequency patterns are used to separate direct and global light components for each pixel
- Once direct and global components are known each pixel is classified as ON, OFF, or UNCERTAIN using a simple set of rules

(*) Y. Xu and D. G. Aliaga, “Robust pixel classification for 3D modeling with structured light”

Proposed method: projector calibration

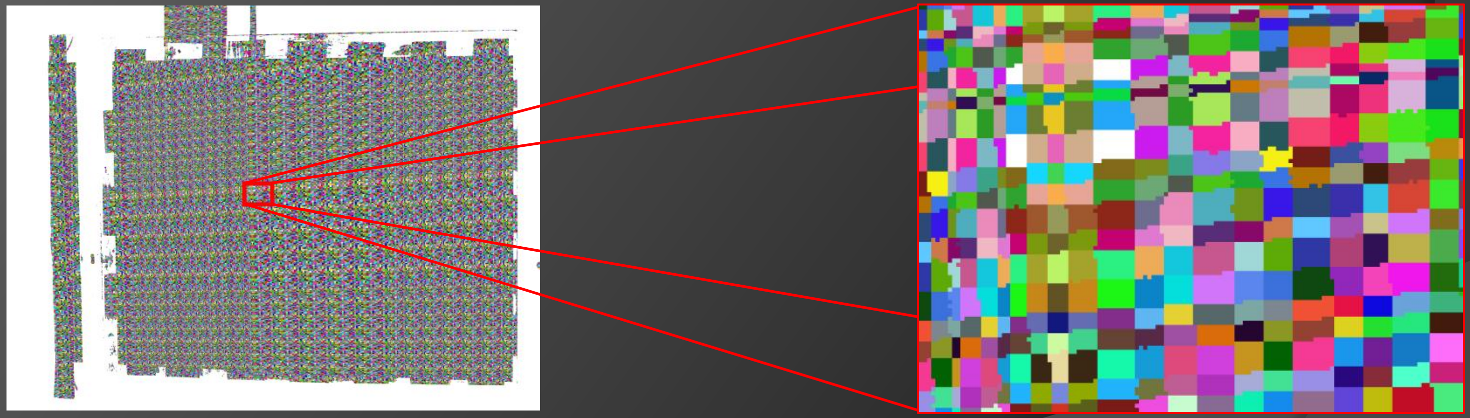
Once the structured-light pattern is decoded we have a mapping between projector and camera pixels:

1) Each camera pixel is associated to a projector row and column, or set to UNCERTAIN



For each (x, y): Map(x, y) = (row, col) or UNCERTAIN

2) The map is not bijective: many camera pixels corresponds to the same projector pixel

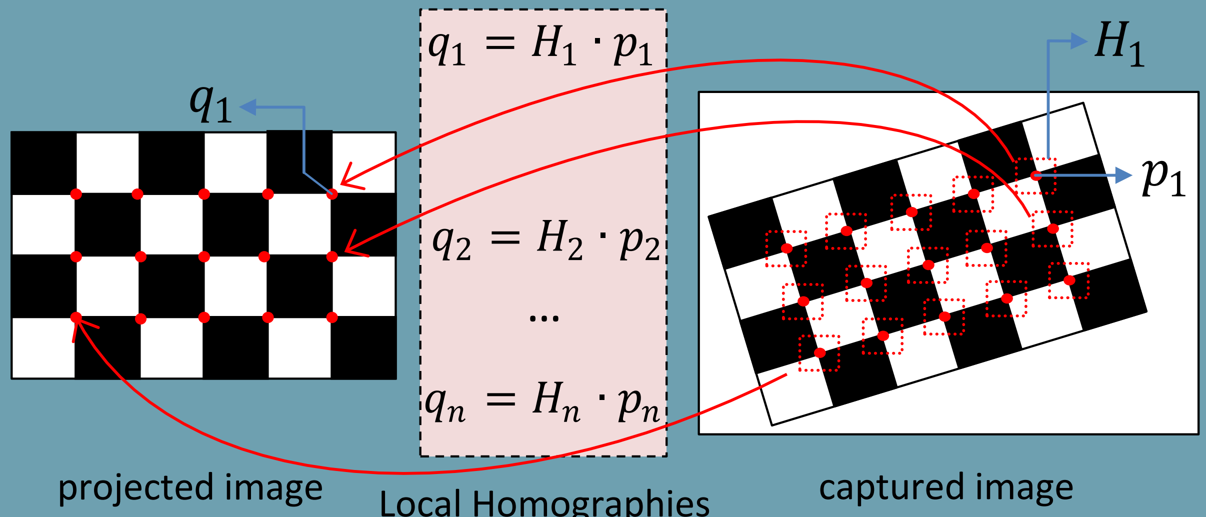


3) Checkerboard corners are not located at integer pixel locations

Proposed method: projector calibration

Solution: local homographies

1. Surface is locally planar: actually the complete checkerboard is a plane
2. Radial distortion is negligible in a small neighborhood
3. Radial distortion is significant in the complete image:
   - a single global homography is not enough



For each checkerboard corner solve:

$$\hat{H} = \underset{H}{argmin} \sum_{\forall p} \|q - Hp\|^2, \qquad \bar{q} = \hat{H} \cdot p$$

$$\hat{H} \in \mathbb{R}^{3\times3}, p = [x, y, 1]^T, q = [col, row, 1]^T$$

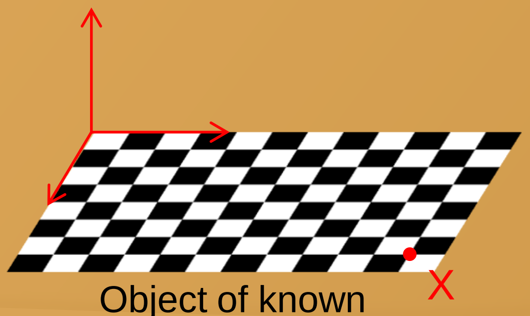
Proposed method: projector calibration

Summary:

1. Decode the structured-light pattern: camera $\leftrightarrow$ projector map
2. Find checkerboard corner locations in camera image coordinates
3. Compute a local homography H for each corner
4. Translate each corner from image coordinates x to projector coordinates x' applying the corresponding local homography H

$$x' = H \cdot x$$

5. Using the correspondences between the projector corner coordinates and 3D world corner locations, $X \leftrightarrow x'$, find projector intrinsic parameters



$$x'_1 = K_{proj} \cdot L(R_1 X + T_1; k_1, k_2, k_3, k_4)$$

$$x'_2 = K_{proj} \cdot L(R_2 X + T_2; k_1, k_2, k_3, k_4)$$

$$x'_3 = K_{proj} \cdot L(R_3 X + T_3; k_1, k_2, k_3, k_4)$$

...

No difference with camera calibration!!

Camera calibration and system extrinsics

Camera intrinsics

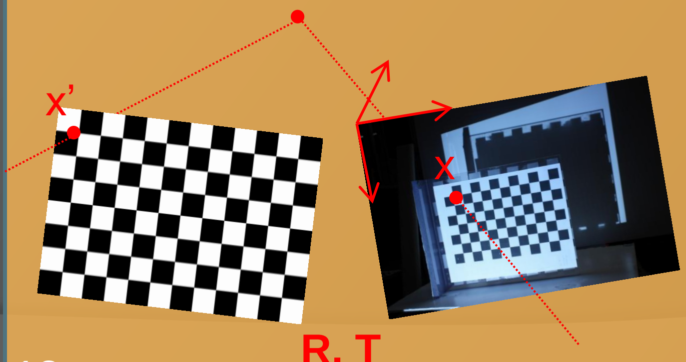
Using the corner locations in image coordinates and their 3D world coordinates, we calibrate the camera as usual

- Note that no extra images are required

System extrinsics

Once projector and camera intrinsics are known we calibrate the extrinsics (R and T) parameters as is done for camera-camera systems

Using the previous correspondences, $x \leftrightarrow x'$, we fix the coordinate system at the camera and we solve for R and T:



$$\tilde{x}_1 = L^{-1}(K_{cam}^{-1} \cdot x_1; k_1, k_2, k_3, k_4) \qquad x'_1 = K_{proj} \cdot L(R \cdot \tilde{x}_1 + T; k'_1, k'_2, k'_3, k'_4)$$

$$\tilde{x}_2 = L^{-1}(K_{cam}^{-1} \cdot x_2; k_1, k_2, k_3, k_4) \qquad x'_2 = K_{proj} \cdot L(R \cdot \tilde{x}_2 + T; k'_1, k'_2, k'_3, k'_4)$$

$$\tilde{x}_3 = L^{-1}(K_{cam}^{-1} \cdot x_3; k_1, k_2, k_3, k_4) \qquad x'_3 = K_{proj} \cdot L(R \cdot \tilde{x}_3 + T; k'_1, k'_2, k'_3, k'_4)$$

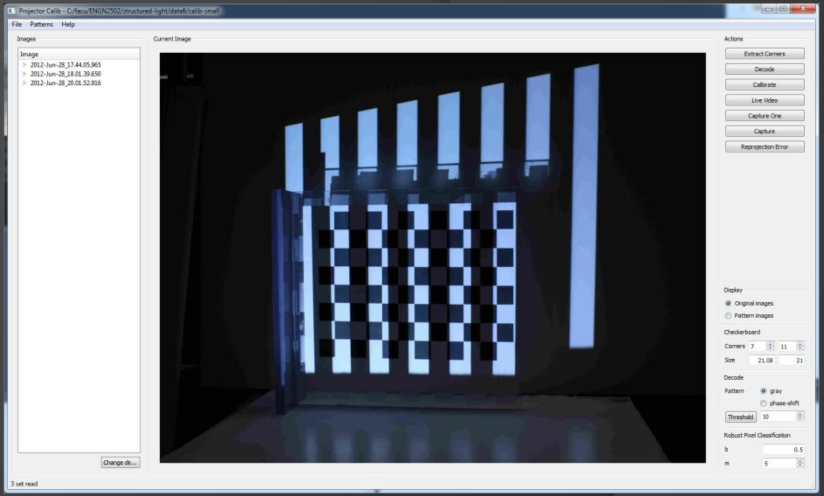
$$\ldots \qquad \ldots$$

Calibration software

Software

The proposed calibration method can be implemented fully automatic:

- The user provides a folder with all the images
- Press “calibrate” and the software automatically extracts the checkerboard corners, decode the structured-light pattern, and calibrates the system



Algorithm

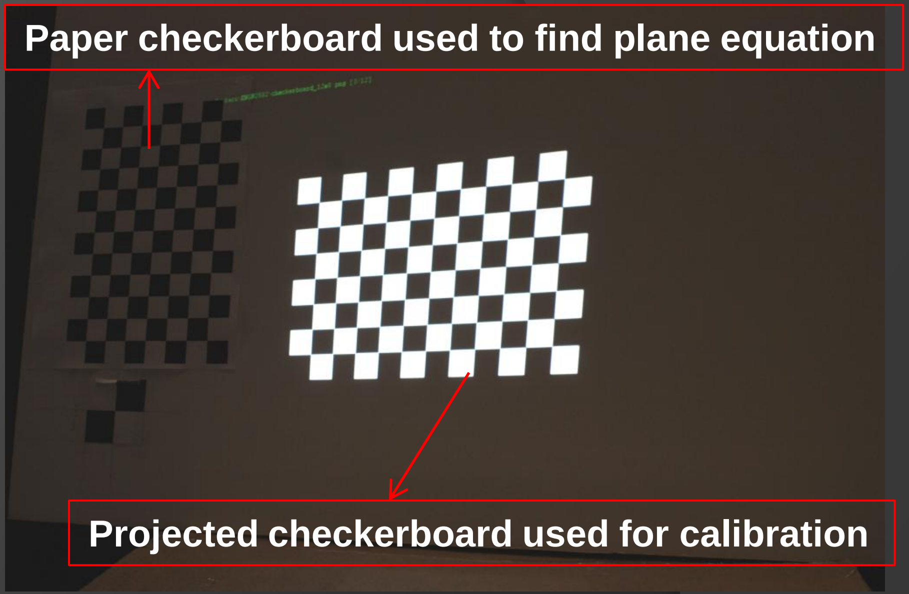
1. Detect checkerboard corner locations for each plane orientation
2. Estimate global and direct light components
3. Decode structured-light patterns
4. Compute a local homography for each checkerboard corner
5. Translate corner locations into projector coordinates using local homographies
6. Calibrate camera intrinsics using image corner locations
7. Calibrate projector intrinsics using projector corner locations
8. Fix projector and camera intrinsics and calibrate system extrinsic parameters
9. Optionally, all the parameters, intrinsic and extrinsic, can be optimized together

Results

Comparison with existing software:

procamcalib

- Projector-Camera Calibration Toolbox
- http://code.google.com/p/procamcalib/



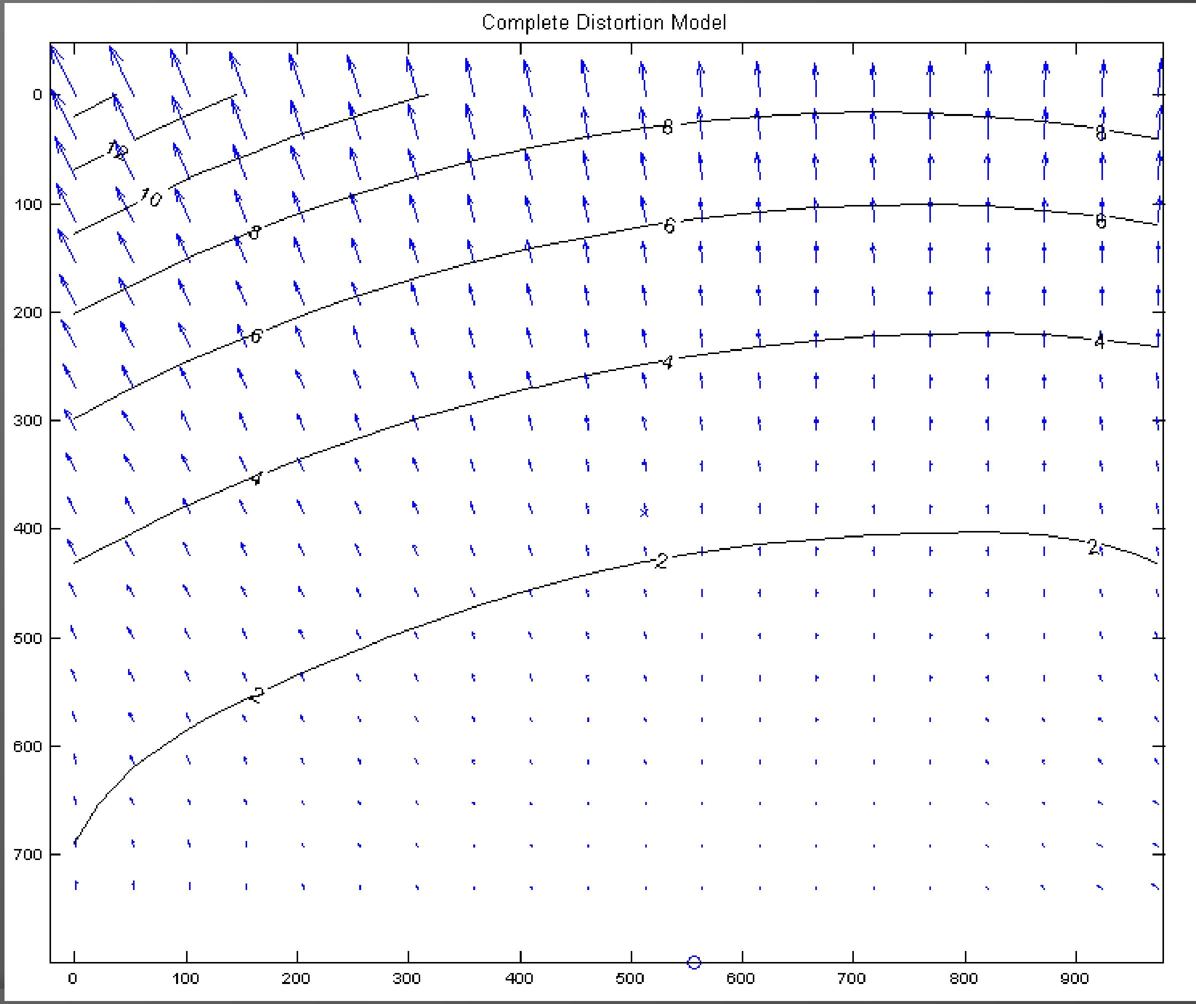
Reprojection error comparison

Method	Camera	Projector
Proposed		0.1447
With global homography	0.3288	0.2176
Procamcalib		0.8671

- Only projector calibration is compared
- Same camera intrinsics is used for all methods
- Global homography means that a single homography is used to translate all corners

Results

Example of projector lens distortion



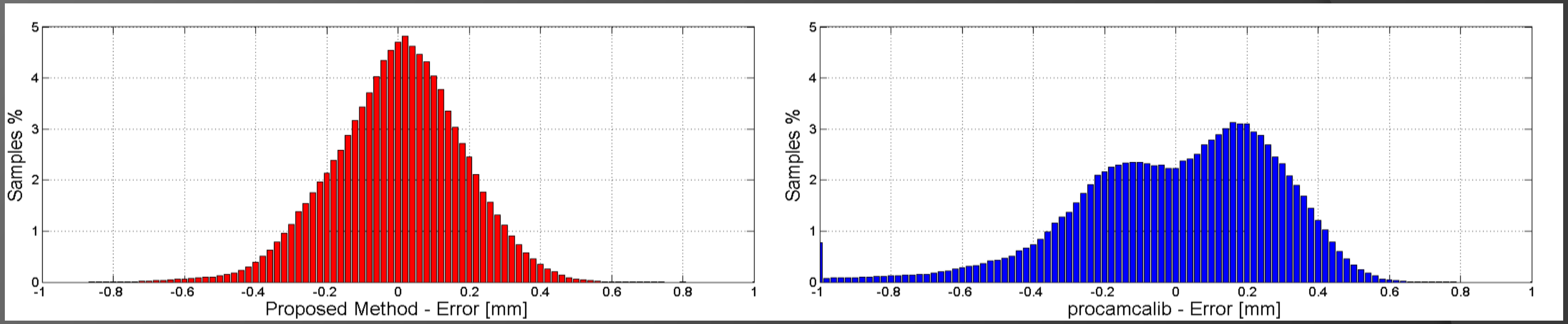
Distortion coefficients

k_1	k_2	k_3	k_4
-0.0888	0.3365	-0.0126	-0.0023

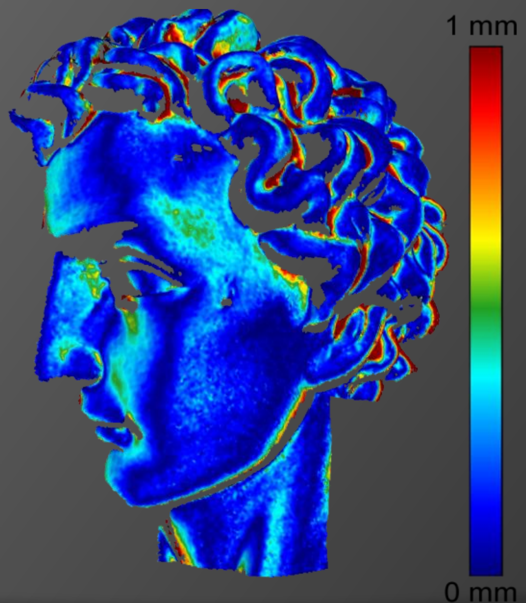
Non trivial distortion!

Results

Error distribution on a scanned 3D plane model:

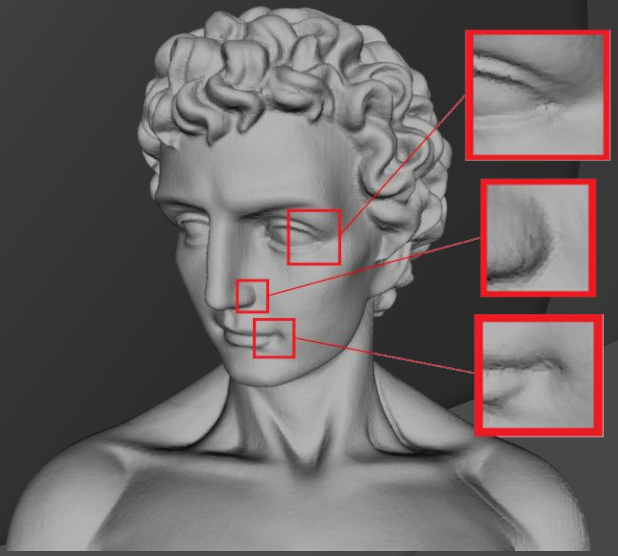


Laser scanner comparison



Hausdorff distance

3D Model



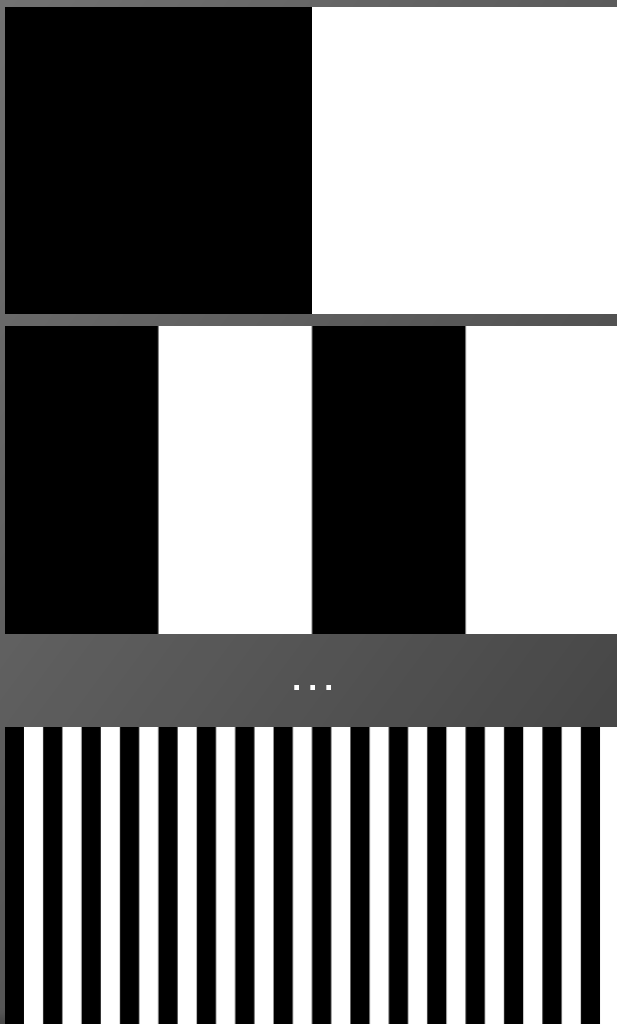
Model with small details reconstructed using SSD

Conclusions

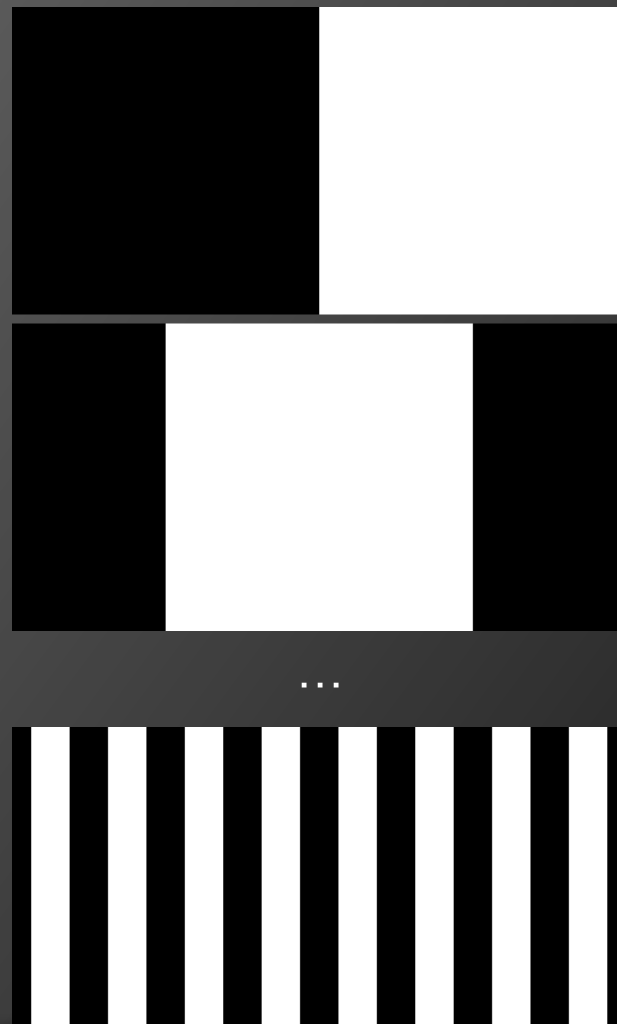
- It works ☺
- No special setup or materials required
- Very similar to standard stereo camera calibration
- Reuse existing software components
  - Camera calibration software
  - Structured-light projection, capture, and decoding software
- Local homographies effectively handle projector lens distortion
- Adding projector distortion model improves calibration accuracy
- Well-calibrated structured-light systems have a precision comparable to some laser scanners

Gray vs. binary codes

Binary



Gray



Dec	Bin	Gray
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101

Direct/Global light components

$$L^{+} = L_d + \alpha L_g + b(1-\alpha)L_g \qquad L^{-} = bL_d + (1-\alpha)L_g + \alpha b L_g$$

$$L_d = \frac{L^{+} - L^{-}}{1-b} \qquad L_g = 2\frac{L^{-} - bL^{+}}{1-b^2} \qquad \hat{L}^{+} = \max_{0<i<K} I_i \qquad \hat{L}^{-} = \min_{0<i<K} I_i$$

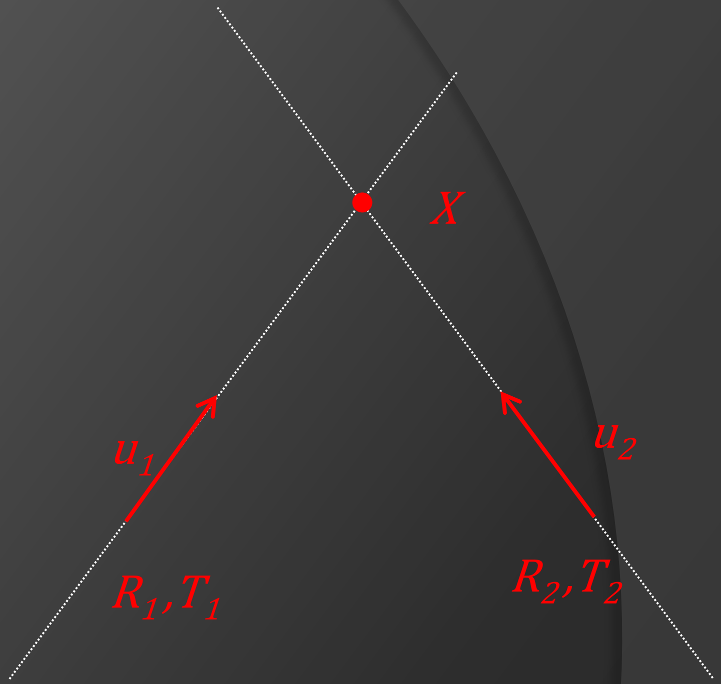
Robust pixel classification

$$\begin{cases} L_d < m \rightarrow \text{UNCERTAIN} \\ L_d > L_g \wedge p > \overline{p} \rightarrow \text{ON} \\ L_d > L_g \wedge p < \overline{p} \rightarrow \text{OFF} \\ p < L_d \wedge \overline{p} > L_g \rightarrow \text{OFF} \\ p > L_g \wedge \overline{p} < L_d \rightarrow \text{ON} \\ otherwise \rightarrow \text{UNCERTAIN} \end{cases}$$

Triangulation

$$\lambda_1 u_1 = R_1 X + T_1$$
$$\lambda_2 u_2 = R_2 X + T_2$$

$$\hat{u}_1 \lambda_1 u_1 = \hat{u}_1 R_1 X + \hat{u}_1 T_1 = 0$$
$$\hat{u}_2 \lambda_2 u_2 = \hat{u}_2 R_2 X + \hat{u}_2 T_2 = 0$$



In homogeneous coordinates:

$$\begin{bmatrix} \hat{u}_1 R_1 & \hat{u}_1 T_1 \\ \hat{u}_2 R_2 & \hat{u}_2 T_2 \end{bmatrix} X = 0$$