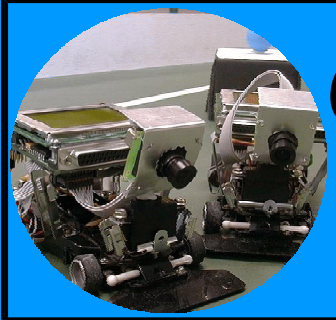
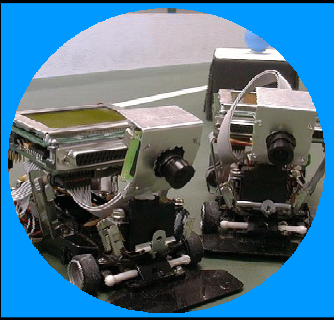


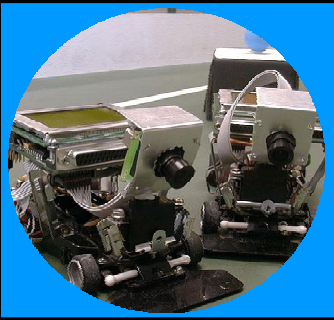
Camera Calibration Using Rectangular Textures

Jacky Baltes
Centre for Imaging Technology and Robotics
University of Auckland
Email: j.baltes@auckland.ac.nz
WWW: http://www.citr.auckland.ac.nz/~jacky



Introduction

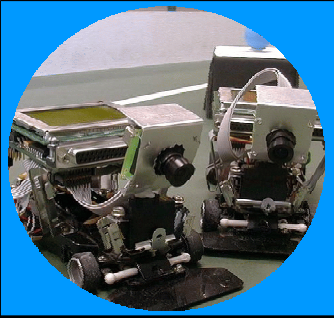
- Camera Calibration
- Problem Specification
- Rectangular Textures
- Algorithm: Compute Initial Matrix
- Iterative algorithm to sort calibration points
- Results
- Conclusion



Camera Calibration

- Position, orientation, focal length, lens distortion
- Camera parameters are used to compute world from image coordinates and vice versa
- Popular area of computer vision
- Pinhole Camera Model: Homogeneous Coordinates

$$\begin{bmatrix} x \\ y \\ z \\ k \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



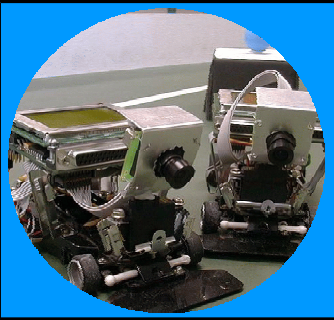
Problem Statement

Camera calibration of buildings, cities, skyscrapers, ...

Cover a building with a calibration carpet?

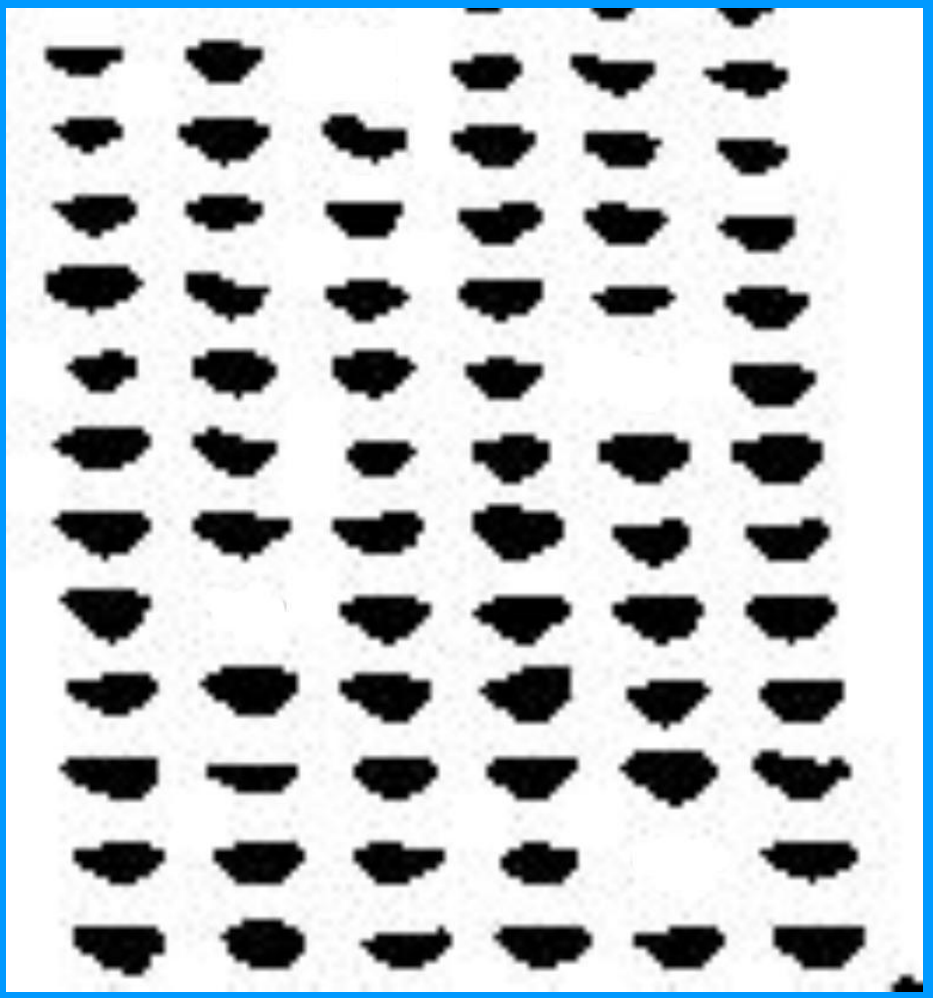
Rectangular textures (width and height) occur in a lot of man−made objects

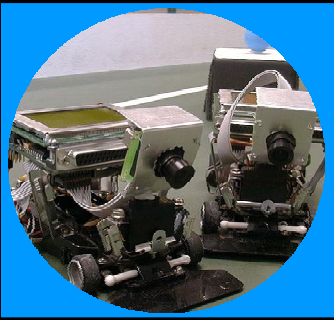
Use these feature points as calibration points



Preprocessing

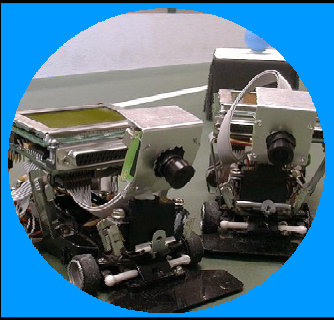
- Extract feature points using color segmentation
- Noisy input images.
- Detection based on sufficient conditions
- Some of the features are missing





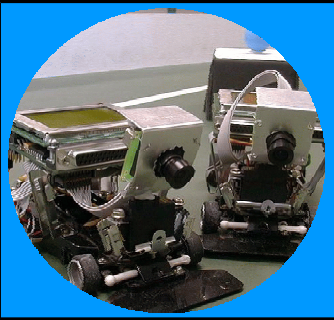
Compute Initial Matrix

- Sort feature points automatically.
- Chicken and Egg problem:
  If we know the orientation and position, we can easily cope with missing features, arbitrary orientations, ...
- Previously: sort by going through the image, maintain guess of distance to next feature. Strong assumptions.
- Compute a guess of the calibration matrix



Approximate Initial Matrix

- Position and orientation can be represented as a 3x4 transformation matrix.
- Requires a minimum of 5 points (Z=0)
- Extract seed points (5 points with minimum distance in the middle of the picture)
- Try both orientations (width and height) and choose the one with the best match (smallest error)

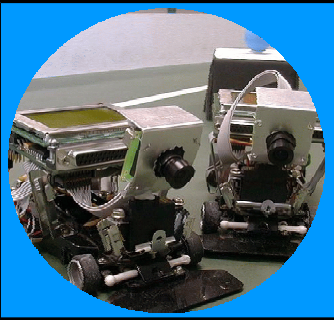


Initial Matrix Algorithm

Algorithm 1 Algorithm to Assign Real World Coordinates

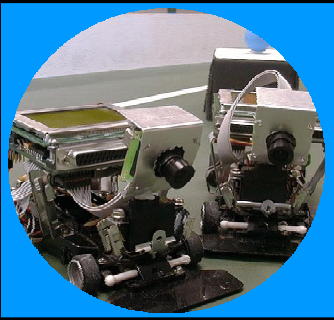
1: $S = \text{extractCentres}(Image)$
2: $P = \text{selectSeedPoints}(S)$
3: $M_{XY} = \text{computeTransformationMatrix}(S, \text{realWorldCoords}(S))$
4: $M_{YX} = \text{computeTransformationMatrix}(S, \text{swapCoord}(\text{realWorldCoords}(S)))$

5: **if** $\text{error}(M_{XY}) < \text{error}(M_{YX})$ **then**
6: $\quad M_{Initial} = M_{XY}$
7: **else**
8: $\quad M_{Initial} = M_{YX}$
9: $\quad \text{swapCoordinates}(S)$
10: **end if**
11: $C_S = \text{assignCentres}(S, M_{Initial})$



Iterative Part

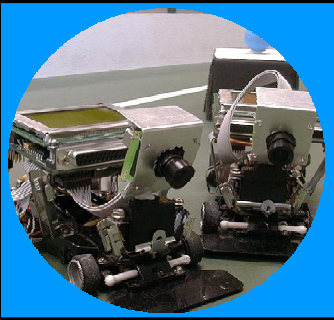
- Select closest feature point s that has not been assigned yet in S
- Assign world coordinates based on coordinates of the assigned neighbors N. Relative assignment
- Add point s to the assigned feature points N
- Compute a new transformation matrix M



Assign Real World Coordinates

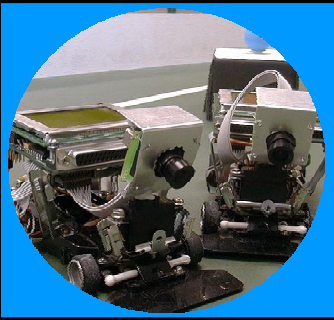
Algorithm 2 Algorithm to Assign Real World Coordinates($S, M_{Initial}$)

1: $M = M_{Initial}$
2: $N =$
3: **while** $S \neq$ **do**
4: **for all** $s \in S$ **do**
5: $s_{North}, s_{East}, s_{South}, s_{West} = \text{findNearestNeighbors}(s, M)$
6: **end for**
7: **for all** $n \in \{s_{North}, s_{East}, s_{South}, s_{West}\}$ **do**
8: $b_x = dist((s-n)/width, M), b_y = dist((s-n)/height, M)$
9: $n_x, n_y = s_x + bx * width, s_y = by * height$
10: $N = N + n, S = S - n$
11: $M = \text{computeTransformationMatrix}(N, \text{realWorldCoords}(S))$
12: **end for**
13: **end while**
14: return N

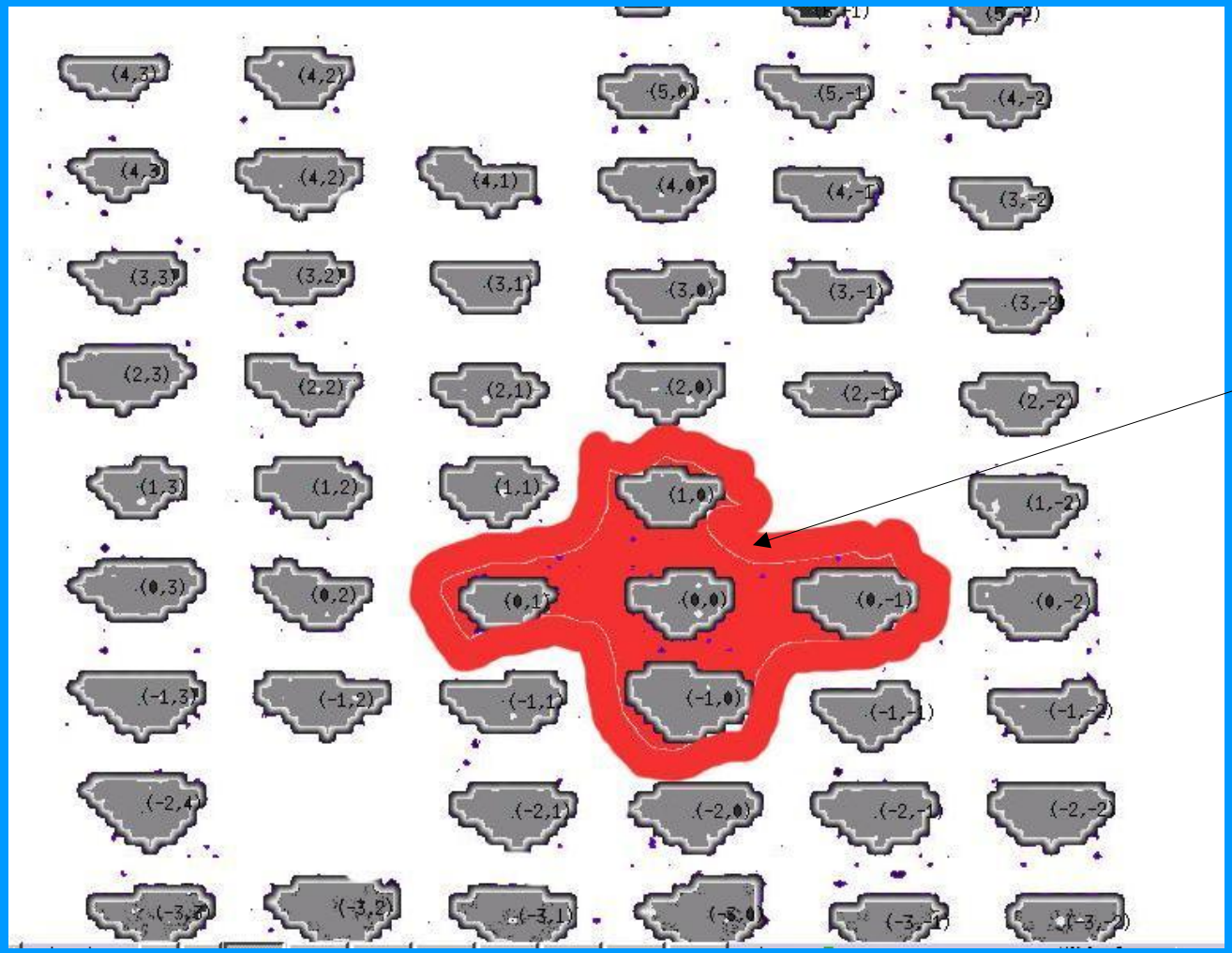


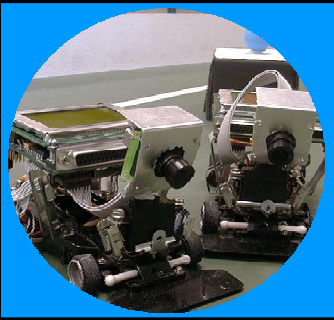
Tsai Calibration

- Roger Tsai developed a practical method for camera calibration (Radial Alignment Constraint)
- Gaussian Lens Model
- Compute 6 external and 5 internal parameters. Distortion parameters (k1,k2).
- Requires around 15 calibration points
- Calibration point: Image point (2D) and matching world coordinate (3D)



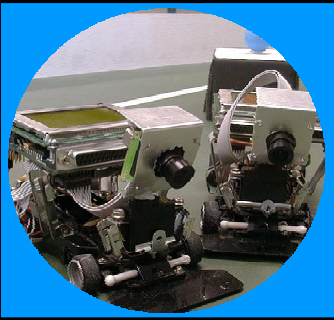
Result





Conclusion

- Method for assigning real world coordinates to features in an image
- Assume a rectangular texture. Constant width and height
- Iterative algorithm to compute position, orientation, and focal length
- Cope with missing feature points
- Arbitrary position and orientation



Future Work

- Tsai calibration can not adjust the Sx parameter (Horizontal uncertainty factor) given only co−planar calibration points
- Poor generalisation to 3D objects
- Batista [2000] proposes a calibration method for co−planar calibration data
- More complex tectures