Verifying 2D Shape Matching

Robert B. Fisher
School of Informatics
University of Edinburgh

Verification Goal

Ensure that we have a good match between model and data

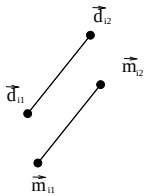
Ensure that we have a good pose estimation

Use geometric shape properties to eliminate bad matches & poses

Geometric Verification

Transform model lines into place: for each $\vec{m}_i$ compute $\sigma \mathrm{R}\vec{m}_i + \vec{t}$

For each model-data line pair, do 3 tests:
Test 1: Are model and data lines parallel?
(For simplicity, use $\vec{m}_i$ in notation instead of $\sigma \mathrm{R}\vec{m}_i + \vec{t}$)

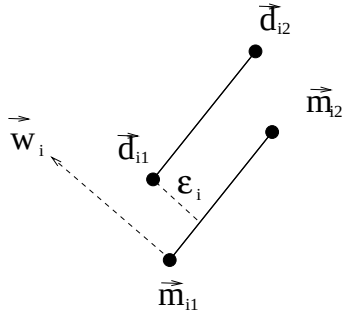


If

$$| \frac{\vec{m}_{i1} - \vec{m}_{i2}}{|| \vec{m}_{i1} - \vec{m}_{i2} ||} \cdot \frac{\vec{d}_{i1} - \vec{d}_{i2}}{|| \vec{d}_{i1} - \vec{d}_{i2} ||} | > threshold$$

then OK (threshold = 0.9?)

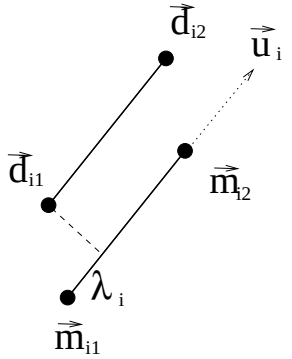
Test 2: Are model and data lines close?



Let $(r, s) = \frac{\vec{m}_{i1} - \vec{m}_{i2}}{||\vec{m}_{i1} - \vec{m}_{i2}||}$ and $\vec{w}_i = (-s, r)$

For $k = i1, i2$, compute $\epsilon_i = (\vec{d}_k - \vec{m}_{i1})' \vec{w}_i$
If $| \epsilon_i | < threshold$ then OK (threshold = 15 pixels?)

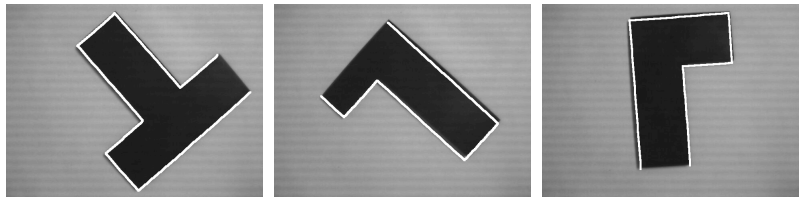
Test 3: Do model and data lines overlap?



For $k = i1, i2$, compute $\lambda_k = (\vec{d_k} - \vec{m}_{i1})'\vec{u}_i$

If $-tolerance \mid\mid \vec{m}_{i1} - \vec{m}_{i2} \mid\mid \leq \lambda_k \leq (1 + tolerance) \mid\mid \vec{m}_{i1} - \vec{m}_{i2} \mid\mid$, then OK (tolerance = 0.3?)

Verified Position Result Examples



Limit = number of model lines - 1

Confusion Matrix

	Est Tee	Est Thin L	Est Thick L	No Est
True Tee	4	0	0	0
True Thin L	0	3	0	1
True Thick L	0	0	4	0

Image 8 had Thin L model flipped over.
Matching process can be extended to allow this.

What Have We Learned?

Introduction to

- Geometric Model-based Object Recognition
- 2D Geometric Verification Algorithm
- Similar techniques for shapes other than straight lines: circular arcs, corners, holes, ...
- Extendable to 3D