

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

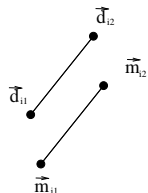
Geometric Verification

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

For each model-data line pair, do 3 tests:

Test 1: Are model and data lines ?

(For simplicity, use \vec{m}_i in notation instead of $\sigma R\vec{m}_i + \vec{t}$)

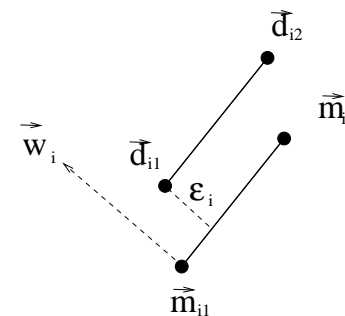


If

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

then OK (threshold = 0.9?)

Test 2: Are model and data lines ?

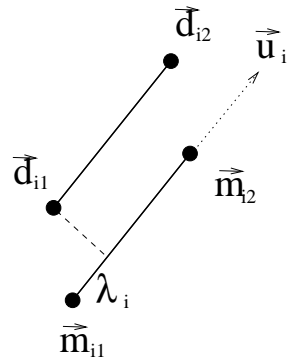


Let $(r, s) = \frac{\vec{m}_{i1} - \vec{m}_{i2}}{\|\vec{m}_{i1} - \vec{m}_{i2}\|}$ and $\vec{w}_i = (-s, r)$

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?)

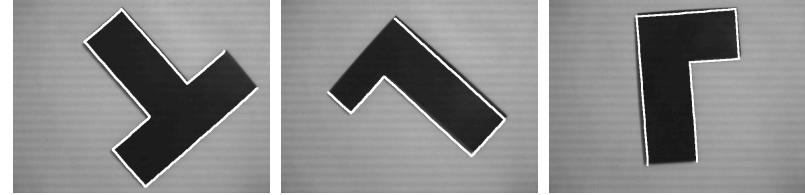
Test 3: Do model and data lines ?



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

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

Verified Position Result Examples



Limit = number of model lines - 1

Confusion Matrix

	Est	Est	Est	No
	Tee	Thin L	Thick L	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 to allow this.

What Have We Learned?

Introduction to

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