Point/feature matching methods

George Stockman
Computer Science and Engineering
Michigan State University
Why match points across images?

- For fusing information from different sensors or different times
- To detect changes in a scene from time $t_1$ to time $t_2$
- To mosaic overlapping images of a scene to form one large image
- To recognize an object by registering its image to a map or model
High level view: match what?

- image points \( I[x,y] \) to map or model \( M[u,v] \)
- image points \( I_1[x,y] \) to image points \( I_2[u,v] \)
- model points \( M_1[x,y] \) to model points \( M_2[u,v] \)
Problems

- How to find landmarks to match across two images?
- How to distinguish one landmark from another?
- How to match N landmarks from I1 to M landmarks from I2, assuming that I2 has more and/or less than I1?
More problems

- The mapping between the two image spaces may be nonlinear/complicated
- There may be errors in the locations of detected point or area features
- There may be extra features in one image or the other
- There may be missed features in one image or the other
Some salient face points

Points identified by surface curvature in neighborhood and location relative to other salient points.

No R eye or R ear

No L eye
Salient points useful for matching sensed and model surfaces

2.5D images of face from Minolta Scanner

3D model made by mosaicking multiple 2.5D views
Why match faces?

- Image to model for recognition or verification
- Image to model for cloning/animation (e.g. animating a jackass in Shrek)
- Image to general model for detecting a face
- Image to image for recognition or verification
Matching landmarks via cross correlation

Treat image neighborhoods N1 and N2 as functions or vectors and match them using the L2 norm
How to determine landmarks

- By unique intensity/color structure of neighborhood
- By geometry or topology of neighborhood (part 2 below)

Vectors show how intensity neighborhoods moved: similar vectors are colored the same

Do these two images meet the definition of “registration”
(1) Determine salient points
(2) Match neighborhoods
Vector mathematics

signal vector: \[ S = [s_1, s_2, \ldots, s_n] \]

signal energy: \[ S \circ S = s_1^2 + s_2^2 + \ldots + s_n^2 \]

signal norm: \[ \langle S \rangle = \sqrt{\sum S \circ S} \]

cross energy: \[ S \circ T = s_1 t_1 + s_2 t_2 + \ldots s_n t_n \]

distance: \[ d(S, T) = \langle S - T \rangle = \langle T - S \rangle = d(T, S) \]

normalized cross correlation: \[ S \circ T / \langle S \rangle \langle T \rangle, \]

which is between \(-1\) and \(+1\) by Cauchy–Schwartz.

For 2D vectors that are \(r\) rows by \(c\) columns, just concatenate the rows to make a single vector of size \(r \times c\).
let $f$ and $g$ be continuous on $[a,b]$

can define $f \circ g = \int_a^b f(x)g(x)dx$

So, $\langle f \rangle = \sqrt{\int_a^b f^2(x)dx}$

d($f,g$) = $\sqrt{\int_a^b (f(x) - g(x))^2 dx} = \langle f - g \rangle = \langle g - f \rangle = d(g,f)$
Cross correlate model (or mask) $M$ with digital image $I$ on neighborhood of $m$ rows and $n$ columns

$$\eta(x, y) = \sum_{r=0}^{m-1} \sum_{c=0}^{n-1} I[x + c, y + r] M[c, r]$$

- The model is translated to location $[x, y]$
- $m \times n$ multiplications performed; resulting scalar can be large
- doubly nested for-loops to compute a single correlation
- quadruply nested for-loops to create an image of cross correlations at “all” image locations $[x, y]$
- sometimes, mean is subtracted before correlation done
Salient point selection

- perhaps has high energy: sum of the squares of all intensities is high
- perhaps variance of intensities is high
- perhaps variance in vertical, horizontal, and diagonal directions is high (Moravec interest operator example below)
Interesting neighborhood detection 1 (scan NHBDs)

```
procedure detect_corner_points(I, V);
{
    "$I[r, c]\$ is an input image of MaxRow rows and MaxCol columns"
    "$V$ is an output set of interesting points from $I$.
    "$\tau$ is a threshold on the interest operator output"
    "$w$ is the halfwidth of the neighborhood for the interest operator"
    for $r := 0$ to $\text{MaxRow} - 1$
        for $c := 0$ to $\text{MaxCol} - 1$
        {
            if $I[r, c]$ is a border pixel then break;
            else if (interest_operator($I$, $r$, $c$, $w$) $\geq \tau_1$) then add $[(r, c), (r, c)]$ to set $V$;
            "The second $(r, c)$ is a place holder in case vector tip found later."
        }
}
```
Interesting neighborhood detection 2 (rate interest)

```plaintext

real procedure interest_operator (I, r, c, w)
{
    "w is the halfwidth of operator window"
    "See alternate texture-based interest operator in the exercises."
    v1 := variance of intensity of horizontal pixels I_1[r, c - w] ... I_1[r, c + w];
    v2 := variance of intensity of vertical pixels I_1[r - w, c] ... I_1[r + w, c];
    v3 := variance of intensity of diagonal pixels I_1[r - w, c - w] ... I_1[r + w, c + w];
    v4 := variance of intensity of diagonal pixels I_1[r - w, c + w] ... I_1[r + w, c - w];
    return minimum \{v1, v2, v3, v4\};
}
```

CVPR04 Stockman (27 June 04)
Interest point detection

- summary of matches between two images is shown
- endpoints are from image 1 and image 2
- endpoints represent high interest values
- endpoints also have similar neighborhood structure measured by cross correlation (see below)

Change is so large that this would not be regarded as image registration.
Matching by cross correlation

- best matching neighborhood is that which has smallest energy in the squared difference image (N1 - N2)
- can normalize to benefit from the Cauchy-Schwartz theorem
- don’t do it unless both N1 and N2 have magnitude significantly above noise level
Cross-correlation matching

- Correspond pixels of the 2 neighborhoods
- Sum the squares of the differences of pixel values
- Best match has least sum
- \((L2\ norm)\)

\[
v = (a-e)^2 + (b-f)^2 + (c-g)^2 + (d-h)^2
\]

Best match has the lowest \(v\) value. Can search Image 2 for neighborhood that best matches the one in Image 1. Usually, neighborhoods are much larger than 2 x 2. 16 x 16 in MPEG
MPEG motion compression

Video frames N and N+1 show slight movement: most pixels are same, just in different locations. 16 x 16 block A in image 2 has best match to 16 x 16 block B in image 1.
Best matching blocks between video frames N+1 to N ("motion" vectors)

16x16 blocks are matched from I1 to I2. The bulk of the vectors show the true motion of the airplane taking the pictures. The long vectors are incorrect motion vectors, but they do work well for image compression!

Best matches from 2\textsuperscript{nd} to first image shown as vectors overlaid on the 2\textsuperscript{nd} image. (Work by Dina Eldin.)
Radial mass transform

Useful for
(a) detecting interesting points and
(b) matching corresponding points
Radial mass transform

- rotation invariant characterization of the area about an image point
- transforms a 2D or 3D neighborhood into a 1D signal
- easy to compare two 1D RMT signals
- easy to train a SVM to distinguish them
Definition of RMT

\[ f(r) = \int_{\rho=r-\delta}^{\rho=r+\delta} \int_{\theta=0}^{\theta=2\pi} I(\rho, \theta) \, \rho \, d\rho \, d\theta \]

Mathematical definition for the neighborhood about the image origin.

\( f(r) \) is total “mass” inside ring of radius \( r \).
RMT at any image point: translate origin to any point

\[ f(r)_{(x,y)} = \int \int I(s-x, t-y) \rho \phi d\phi \theta \quad \text{where} \quad (s-x)^2 + (t-y)^2 = r^2 \]

• characterize neighborhood of \( I[x,y] \) by shifting origin to \([x,y]\) to get \( f(r) \)
• can examine \( f(r) \) to determine if \([x,y]\) is an interesting point
• can match \( f(r) \) to other 1D signals to classify it
• for 3D image \( I[x,y,z] \) integrate mass in a spherical shell of radius \( r \) instead of a circular ring
Examples of RMT

- RMT of disk
- RMT of square (next slide)
- RMT of soil
Discrete RMT rep. of binary segmented objects

For square at top right (max radius is 14)
1 7 11 20 25 31 40 43 49 48 43 24 8 3 0 0

For square at top left (max radius is 13)
1 6 12 21 25 31 37 43 52 40 34 18 6 0 0

For top “pacman” (max radius is 23)
1 7 13 18 26 31 38 42 47 54 56 67 68 77 82 84 93 94 100 71 25 14 6 0 0 0 0

For “pacman” at left middle (max radius is 23)
0 7 12 18 26 29 39 41 48 56 54 68 74 83 83 94 92 102 74 32 16 2 0 0 0 0
2D discrete RMT

- Assume each object pixel is labeled L and bounding box and centroid \([X_c, Y_c]\) of object known
- Initialize histogram \(RMT[r] = 0\) all \(r\)
- For each pixel \([x, y]\) of object L
- Compute rounded \(r\) as distance between \([x, y]\) and \([X_c, Y_c]\)
- \(RMT[r] = RMT[r] + 1;\) (if unit mass)
comments on discrete RMT

- Pixelization of continuous object
- Causes variation of centroid
- Causes variation in area
- Causes variation in RMT
- Enables “inverse” computation of RMT
  For each pixel \([x, y]\), compute radius \(r\)
  and add pixel mass to RMT[\(r\)]
- Enables templates to be made offline to map \([x, y]\) or
  \([x, y, z]\) to radius \(r\)
- Easy opportunities for parallel computation
Interesting points on jack

• Solid 3D jack phantom created
• Sample interesting points selected
• Sample uninteresting point selected
• SVM trained on those samples
• SVM then used to classify all points of the 3D jack image
Point interest from SVM

Green points are most positive examples; about +2.0 from boundary.

Black points are most negative examples; about -2.0 from boundary.

White points are also far from the boundary on the positive side.

A distributed set of these interest points can be used for registration: rigid or deformable.
Summary of Matching Part 1

- Showed how to find interesting points
- Showed how to match using cross correlation
- Showed how to match using cross correlation of radial mass transform
- Matching was brute force: now need smarter methods for making point correspondences - go to Part 2
references