Descriptions Using Moments

Robert B. Fisher
School of Informatics
University of Edinburgh

Moments

Family of stable binary (and grey level) shape descriptions

Can be made invariant to translation, rotation, [?]

Let $\{p_{rc}\}$ be the binary (0,1) image pixels for row r and col c where 1 pixels are the object

Moments II

Area $A = \sum_r \sum_c p_{rc}$
Center of mass
$(\hat{r}, \hat{c}) = (\frac{1}{A} \sum_r \sum_c r p_{rc}, \frac{1}{A} \sum_r \sum_c c p_{rc})$

A family of 'central' ([?] invariant) moments (for any u and v):

$$m_{uv} = \sum_r \sum_c (r - \hat{r})^u (c - \hat{c})^v p_{rc}$$

Subtracting center of mass makes it translation invariant

Scale invariant moments

If double in dimensions, then moment m_{uv} increases by $2^u 2^v$ for weightings and 4 for the number of pixels.
Similarly, area A increases by 4, and thus $A^{(u+v)/2+1}$ increases by $4 \times 2^u 2^v$

So, the ratio:

$$\mu_{uv} = \frac{m_{uv}}{A^{(u+v)/2+1}}$$

is invariant to [?].

Rotation invariant moments

Moment invariant theory has identified methods to generate various orders of moments invariant to rotation.

6 functions ci_i with rescaling applied to get into similar [?] ranges

Area $A = \sum_r \sum_c p_{rc}$
Center of mass $(\hat{r}, \hat{c})$

Define complex uv central moment:
$c_{uv} = \sum_r \sum_c ((r - \hat{r}) + i(c - \hat{c}))^u ((r - \hat{r}) - i(c - \hat{c}))^v p_{rc}$

 Slide credit: Bob Fisher

[?] invariance

Get specific scale invariant moments:
$s_{11} = c_{11}/(A^2)$
$s_{20} = c_{20}/(A^2)$
$s_{21} = c_{21}/(A^{2.5})$
$s_{12} = c_{12}/(A^{2.5})$
$s_{30} = c_{30}/(A^{2.5})$

 Slide credit: Bob Fisher

Rotation invariant moments II

Rescaled (so values in similar range) rotation invariants:
$ci_1 = real(s_{11})$
$ci_2 = real(1000 * s_{21} * s_{12})$
$ci_3 = 10000 * real(s_{20} * s_{12} * s_{12})$
$ci_4 = 10000 * imag(s_{20} * s_{12} * s_{12})$
$ci_5 = 1000000 * real(s_{30} * s_{12} * s_{12} * s_{12})$
$ci_6 = 1000000 * imag(s_{30} * s_{12} * s_{12} * s_{12})$

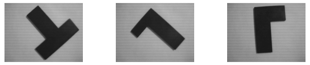
 Slide credit: Bob Fisher

[?] Moment matlab code

```
function vec = getproperties(Image)
    area = bwarea(Image);
    perim = bwarea(bwperim(Image,4));
    compactness = perim*perim/(4*pi*area);
    s11 = complexmoment(Image,1,1) / (area^2);
    s20 = complexmoment(Image,2,0) / (area^2);
    ...
    ci1 = real(s11);
    ci2 = real(1000*s21*s12);
    ci3 = 10000*real(s20*s12*s12);
    ...
```

 Slide credit: Bob Fisher

Example ? property values



compactness	1.93	1.81	1.90
ci_1	0.23	0.27	0.25
ci_2	0.18	0.37	0.45
ci_3	0.08	-0.50	0.11
ci_4	-0.00	0.37	-0.64
ci_5	0.23	-0.47	0.09
ci_6	-0.00	0.07	-0.63

Feature Vector

Standard description for many visual processes: form a ? from set of descriptions:

$$\vec{x} = (compactness, ci_1, ci_2, ci_3, ci_4, ci_5, ci_6)'$$

Multiple vectors if several structures or locations to describe

These vectors are then used in next processes, eg. recognition

Lecture Overview

1. Moments: an infinite family of shape descriptions
2. A way to make them ? to rotation, translation, and scale