ARAVIND
2019102014

Lecture 22

Image Matching

DCT has lesser RMS error than WHT, DFT.

↳ 8×8 subimages ⇒ 64 coefficients

↳ When we attenuate frequencies, we attenuate high frequency coefficients while retaining lower

Sub-image size select

Comparing graphs of RMS (vs) SubImage Size

→ DCT has 2 times the periodicity than DFT

→ Minimises blocking artifacts

JPEG steps:

→ Divide image into 8x8 subimages (blocks

↳ Do a shift from grey value then apply DCT

→ Optimize coefficients

↳ Quantisation Table

$1 \leq$ quality ≤ 25

↳ worst - high compression

best - low compression

→ Encode coefficients using variable length encoding

Fine details are better preserved without blocky artifacts using WSQ compression

JPEG exploits

- → Compress in color-dominant space
- → Changes in image intensity

Support modes

- → Seq.
- → Progressive: Through image, multiple iteration

COMPARING IMAGES

Compare Images by pixel comparison [works only when no transforms etc]

Noise, quantisation etc introduce differences

Better approach: Template matching

↳ Identify similar sub-images

Used in match left and right pic for stereo image

Template Matching

- → Move pattern over search image
- → Measure diff b/w template sub-image
- → Record position where highest similarity

Temp matching in Intensity Images:

$$R_{r,s}(u,v) = R(u-r, v-s)$$

Type of differences:

→ Sum of absolute diff b/w template & subimages → Euclidean

→ Sum of squared dist etc

$$SSD = \sum \left(f(i,j) - g(i,j) \right)^2 \quad \text{\} Sum of sq. diff}$$

$$C_{fg} = \sum f(i,j) \cdot g(i,j) \quad \text{\} Correlation}$$

Cross Correlation

$$(I \oplus R)(r,s) = \sum\sum I(r+i, s+j) \cdot R(i,j)$$

Problem with cross-correlation:

→ For twice as bright image → larger score

∴ Brighter parts of image will almost always have higher score → incorrect match

→ Can possibly subtract off mean value of template

→ Still doesn't give exact match always

SSD or block matching

$$\sum [f(i,j) - g(i,j)]^2 \quad \text{\} have cross correlation term}$$

SSD implicitly includes cross correlation term

↳ Tries to minimise cross correlation term

$$d_e^2(r,s) = \underbrace{\sum I^2(r+i, s+j)}_{A(r,s)} + \underbrace{\sum R^2(i,j)}_{B(\ )} - \underbrace{2\sum I(r+i,s+j) \cdot R(i,j)}_{C(r,s)}$$

→ SSD gives a correct match

→ Normalise intensity for both search & query image

$$\hat{f} = \frac{f-\bar{f}}{\sqrt{\Sigma(f-\bar{f})^2}} \qquad \hat{g} = \frac{g-\bar{g}}{\sqrt{\Sigma(g-\bar{g})}}$$

Normalised
Cross correlation:

$$N(c(f,g)) = C_{fg}(\hat{f},\hat{g}) \text{ (or) } \frac{f \cdot g}{\|f\|\,\|g\|} = \sum \hat{f}(i,j)\cdot\hat{g}(i,j)$$

Correlation coefficient $\in [-1,1]$

Shape of Template

→ Need not be rectangular

→ Can be circular, elliptical

Matching under Rot, Scaling

→ Simple approach:

→ Store multiple rotated & scaled version

→ Computationally prohibitive

→ Alternate approachs:

→ Matching in logarithmic polar space

→ Affine matching : Use local SIFT methods

Matching binary images:

→ Direct comparision: Count pixels

→ Detection shapes without texture : Small shifts make big difference when only edges preserved

Distance transform:

$$D(p) = \min_{p' \in FG(I)} dist.(p, p')$$

→ Euc dist: $\sqrt{(u-u')^2 + (v-v')^2}$

→ Manhattan dist: $|u-u'| + |v-v'|$

Chamfer Matching

Find template in distance transformed image

→ Edge model translated over dist. image

→ Avg of distance values that edge hits gives chamfer dist.

RMS chamfer: $\frac{1}{3}\sqrt{\frac{1}{n}\sum_{i=1}^{n} V_i^2}$

→ Provides smooth cost fn

→ Affected on rotation, translation

→ Sensitive to small shape changes

→ Need large no. of template shape

↳ They are robust, computationally inexpensive