Multimedia indexing

Norbert Fuhr

representation of non-textual media

- audio
- images
1 Audio

1.1 Sound retrieval


Levels of audio retrieval

1. exact match of sound samples
2. inexact match of sounds, irrespective of sample rate, quantization, compression, . . .
3. inexact match of acoustic features / perceptual properties of sound
4. content-based match (for speech, musical content)

here: inexact match of acoustic features and perceptual properties

N. Fuhr
**Acoustic features**

aspects of sound considered:

**loudness** root-mean-square of audio signal (in decibels)

**pitch** greatest common divisor of peaks in Fourier spectra

**brightness** centroid of short-time Fourier magnitude spectra
  (higher frequency content of signal)

**bandwidth** magnitude-weighted average of differences between spectral components and the centroid
  (variation of frequencies, e.g. sine wave vs. white noise)

**harmonicity** deviation of the sound’s spectrum from a harmonic spectrum
  (i.e. harmonic spectra vs. inharmonic spectra vs. noise)
variation of aspects over time:

1. compute aspect values at certain time intervals

2. derive features from sequences:
   - average value
   - variance
   - autocorrelation

   (feature values weighted by amplitude)
sound example

N. Fuhr
<table>
<thead>
<tr>
<th>Property</th>
<th>Mean</th>
<th>Variance</th>
<th>Autocorrelation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loudness</td>
<td>-54.4112</td>
<td>221.451</td>
<td>0.938929</td>
</tr>
<tr>
<td>Pitch</td>
<td>4.21221</td>
<td>0.151228</td>
<td>0.524042</td>
</tr>
<tr>
<td>Brightness</td>
<td>5.78007</td>
<td>0.0817046</td>
<td>0.690073</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>0.272099</td>
<td>0.0169697</td>
<td>0.519198</td>
</tr>
</tbody>
</table>
Indexing and retrieval

Indexing of a sound:
compute and store feature vector \( a \)
(mean, variance and autocorrelation for loudness, pitch, brightness, bandwidth and harmonicity)

Retrieval:

1. conditions w.r.t. feature values
2. similarity of sounds: weighted Euclidian distance

\( M \) – # sounds considered

mean: \( \mu = \frac{1}{M} \sum_{j=1}^{M} a_j \)

covariance \( R = \frac{1}{M} \sum_{j=1}^{M} (a_j - \mu)(a_j - \mu)^T \)

distance \( D = \sqrt{(a - b)^T R^{-1} (a - b)} \)

N. Fuhr
Property-based training and classification

training:
based on set of training sounds for a property
(e.g. scratchiness)
compute property-specific mean and covariance
importance of feature: mean divided by standard deviation

classification
compute distances to means of all classes,
select class with minimum distance
likelihood:

\[ L = \exp \left( \frac{D^2}{2} \right) \]
**Example:**

class model for laughter

<table>
<thead>
<tr>
<th>Feature</th>
<th>Mean</th>
<th>Variance</th>
<th>Importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>2.71982</td>
<td>0.191312</td>
<td>6.21826</td>
</tr>
<tr>
<td>Loudness: Mean</td>
<td>-45.0014</td>
<td>18.9212</td>
<td>10.3455</td>
</tr>
<tr>
<td>– Variance</td>
<td>200.109</td>
<td>1334.99</td>
<td>5.47681</td>
</tr>
<tr>
<td>– Autocorrelation</td>
<td>0.955071</td>
<td>7.71106e-05</td>
<td>108.762</td>
</tr>
<tr>
<td>Brightness: Mean</td>
<td>6.16071</td>
<td>0.0204748</td>
<td>43.0547</td>
</tr>
<tr>
<td>– Variance</td>
<td>0.0288125</td>
<td>0.000113187</td>
<td>2.70821</td>
</tr>
<tr>
<td>– Autocorrelation</td>
<td>0.715438</td>
<td>0.0108014</td>
<td>6.88386</td>
</tr>
<tr>
<td>Bandwidth: Mean</td>
<td>0.363269</td>
<td>0.000434929</td>
<td>17.4188</td>
</tr>
<tr>
<td>– Variance</td>
<td>0.00759914</td>
<td>3.57604e-05</td>
<td>1.27076</td>
</tr>
<tr>
<td>– Autocorrelation</td>
<td>0.664325</td>
<td>0.0122108</td>
<td>6.01186</td>
</tr>
<tr>
<td>Pitch: Mean</td>
<td>4.48992</td>
<td>0.39131</td>
<td>7.17758</td>
</tr>
<tr>
<td>– Variance</td>
<td>0.207667</td>
<td>0.0443153</td>
<td>0.986485</td>
</tr>
<tr>
<td>– Autocorrelation</td>
<td>0.562178</td>
<td>0.00857394</td>
<td>6.07133</td>
</tr>
</tbody>
</table>

\[
\text{importance} = \frac{|\text{mean}|}{\sqrt{\text{variance}}}
\]

N. Fuhr
2 Images

- syntax vs. semantics ps. pragmatics

- syntactic features: color, texture, contour

- semantic image retrieval: IRIS
2.1 Introduction

2.1.1 Semantic vs. syntactic indexing and retrieval

**syntactic** image features:

- color
- texture
- contour

**semantic** image features:

- objects
  
  *(humans, animals, buildings, art works)*

- topics
  
  *(pollution, demonstration, political visit)*

most image indexing methods support syntactic features only
2.1.2 Aboutness vs. ofness

ofness:
objects shown in the image
→ semantics

aboutness:
topic which is illustrated by the image
→ pragmatics

aboutness is very much user-dependent

*e.g.* image showing water pollution
2.2 Basic techniques

- color frequency
- spatial color
- texture
- contour
2.2.1 Color frequency

- histograms
- moments of distribution
Color histograms

color models: RGB, YUV, HSV, Munsell
→ 3-dimensional color space

color histogram:
\( b_i \) bins for \( i \)th dimension, \( i = 1, 2, 3 \)
→ \( N \)-dimensional vector with \( N = b_1 \cdot b_2 \cdot b_3 \)
Similarity measures

simple similarity measure for comparing image histograms \( I \) and \( J \):

\[
D(I, J) = \frac{\sum_{i=1}^{N} \min(I_i, J_i)}{\sum_{i=1}^{N} I_i}
\]

does not consider similarity of different colors!

color similarity matrix:

\[
A = [a_{ij}], \quad i = 1, \ldots, n, \quad j = 1, \ldots, n
\]

improved color histogram similarity:

\[
D'(I, J) = (I - J)^T A (I - J) = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} (I_i - J_i)(I_j - J_j)
\]
Quantization of color space
8 bits per color $= 2^{24}$ possible colors
→ quantization necessary (for reducing # bins)

**uniform:** divides each axis into intervals of equal length

**LGB:** minimize mean-squared error resulting from quantization
  subdivide 3D color space into $N$ subspaces s.th. resulting error is minimized
  (high processing costs, preprocessing of database necessary)

**product vector:** minimize mean-squared error for each dimension
  (lower processing costs, but still preprocessing necessary)
**Statistical moments**

compute statistical moments for each color channel (according to underlying color model)

$N \ # \ pixels \ in \ image$

$j \ pixel \ number, \ j \in [1, N]$

$r \ # \ color \ channels$

$i \ color \ channel, \ i \in [1, r]$

$p_{ij} \ intensity \ of \ j^{th} \ pixel \ for \ channel \ i$

\[
\text{mean} \quad E_i = \frac{1}{N} \sum_{j=1}^{N} p_{ij}
\]

\[
\text{variance} \quad \sigma_i = \left( \frac{1}{N} \sum_{j=1}^{N} (p_{ij} - E_i)^2 \right)^{\frac{1}{2}}
\]
skewness \( s_i = \left( \frac{1}{N} \sum_{j=1}^{N} (p_{ij} - E_i)^3 \right)^{\frac{1}{3}} \)

distance metrics for two images \( I, I' \):
\[
d(I, I') = \sum_{i=1}^{r} w_{i1} |E_i - E_i'| + w_{i2} |\sigma_i - \sigma_i'| + w_{i3} |s_i - s_i'|
\]

\( w_{kl} \) user-specific weights
2.2.2 Spatial color distribution

Sample spots
[Rickman/Stonham SPIE 4]

- consider color at predefined spots only
- each spot consists of small # pixels
- represent spot by medium/median hue
$S_k$ spot

$m$ # spots

$H_{ki}$ hue at $i$th pixel of spot $S_k$

$n$ # pixels per spot

feature vector $F = (F_1, \ldots, F_m)$ with $F_k = \frac{1}{n} \sum_{i=1}^{n} H_{ki}$

distance metrics for two images $I, I'$: $d(I, I') = \sum_{k=1}^{m} cmp(F_k, F'_k)$

$cmp$ – compare function, to be defined, e.g.

$cmp(a, b) = \begin{cases} 0 & \text{if } |a - b| < \varepsilon \\ 1 & \text{otherwise} \end{cases}$
A visual perception model
[Lai/Tait 98, SIGIR WS MM IR]

- based on 10 color groups
- browsing based on hierarchical classification of images
- similarity search based on spatial color distribution

human perception:

- color more important than shape
- languages have few words to describe colors

psychological experiment:

1. generate 125 color samples
2. subjects label samples
3. sort samples according to hue value
4. identify boundaries between groups with different labels
result color groups:

<table>
<thead>
<tr>
<th>Colour Descriptor</th>
<th>Perceptual Colour Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Uncertain Colours: “very dark” or “very bright”</td>
</tr>
<tr>
<td>1</td>
<td>White</td>
</tr>
<tr>
<td>2</td>
<td>Grey</td>
</tr>
<tr>
<td>3</td>
<td>Black</td>
</tr>
<tr>
<td>4</td>
<td>Red, Pink</td>
</tr>
<tr>
<td>5</td>
<td>Brown, Dark Yellow, Olive</td>
</tr>
<tr>
<td>6</td>
<td>Yellow, Orange, Light Yellow</td>
</tr>
<tr>
<td>7</td>
<td>Green, Lime</td>
</tr>
<tr>
<td>8</td>
<td>Blue, Cyan, Aqua, Turquoise</td>
</tr>
<tr>
<td>9</td>
<td>Purple, Violet, Magenta</td>
</tr>
</tbody>
</table>
Image indexing

1. map image colors onto color groups
2. compute color histogram
3. classify according to increasing frequency of color histogram

example classification:
hierarchy has max. $10! \approx 3.6 \cdot 10^6$ clusters

browsing based on this hierarchy
Similarity matching
“fuzzy images”: reduce resolution

similarity matching based on $15 \times 15$ grid

1. map image colors onto color groups
2. reduce resolution to $15 \times 15$ grid
3. scale to square pattern
4. construct fuzzy pattern: matrix of color group numbers
Interactive retrieval
Application
Navigation by hierarchical color
problems with shadows
→ filter out shadow areas
(do not use color group 0 for classification)
Query by sketch
Query by example
### List of Images

<table>
<thead>
<tr>
<th>No#</th>
<th>Class Key</th>
<th>Filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>131</td>
<td>580123</td>
<td>F:\SoftKey\BUILDING\3258008.TIF</td>
</tr>
<tr>
<td>323</td>
<td>58021347</td>
<td>F:\SoftKey\PLACES\3258006.TIF</td>
</tr>
<tr>
<td>458</td>
<td>580173</td>
<td>F:\SoftKey\STRUCTS\3258005.TIF</td>
</tr>
<tr>
<td>248</td>
<td>580371</td>
<td>F:\SoftKey\MONUMENT\3259007.TIF</td>
</tr>
<tr>
<td>233</td>
<td>58013</td>
<td>F:\SoftKey\MISC\3259009.TIF</td>
</tr>
<tr>
<td>032</td>
<td>50816249</td>
<td>F:\SoftKey\ANIMALS\3260028.TIF</td>
</tr>
<tr>
<td>025</td>
<td>508317</td>
<td>F:\SoftKey\ANIMALS\3260021.TIF</td>
</tr>
<tr>
<td>414</td>
<td>850714</td>
<td>F:\SoftKey\SCENIC\3260081.TIF</td>
</tr>
<tr>
<td>424</td>
<td>587031</td>
<td>F:\SoftKey\SCENIC\3260092.TIF</td>
</tr>
<tr>
<td>493</td>
<td>57013</td>
<td>F:\SoftKey\URBAN\IMG0016.TIF</td>
</tr>
<tr>
<td>317</td>
<td>5803716</td>
<td>F:\SoftKey\PLACES\3257049.TIF</td>
</tr>
<tr>
<td>028</td>
<td>507413</td>
<td>F:\SoftKey\ANIMALS\3260024.TIF</td>
</tr>
<tr>
<td>048</td>
<td>50731</td>
<td>F:\SoftKey\ANIMALS\3260061.TIF</td>
</tr>
<tr>
<td>204</td>
<td>5071643</td>
<td>F:\SoftKey\FOOD\3259060.TIF</td>
</tr>
<tr>
<td>307</td>
<td>5071643</td>
<td>F:\SoftKey\PLANTS\3261281.TIF</td>
</tr>
</tbody>
</table>

N. Fuhr
Color cooccurrence descriptors

Image representation

\( c_i \) – color of pixel \( i \)

\( d_{ij} \) – euclidian distance between pixel \( i \) and \( j \)

represent image as matrix

\[
W(c_i, c_j, d_{ij})
\]

\( W \) frequency of cooccurrence of colors \( c_i \) and \( c_j \) at distance \( d_{ij} \)

representation is invariant to rotation, reflection and translation!

matrix stored as set of elements:

\[
E_k \in \{(i_k, w_k) | \exists w_k = W(c_i, c_j, d_{ij}) \neq 0 \land i_k = f(c_i, c_j, d_{ij})\}
\]

\( i_k \) – element index

N. Fuhr
Image retrieval

$T_q$ – query image descriptor
$T_t$ – target image descriptor

dissimilarity measure:

$$D(T_q, T_t) = \frac{\sum_{E_k \in T_q \cap T_t} |w^q_k - w^t_k|}{\sum_{E_k \in T_q} w_k + \sum_{E_k \in T_t} w_k}$$

similarity measure:

$$S(T_q, T_t) = 1 - D(T_q, T_t)$$
Analytical tests

1. noise:
   white noise ranging from 0 to $\pm 0.5$ added to each pixel
2. subimages:

arbitrary positioned subimages of the relative size from 0 to 1 of the original image
Retrieval examples
2.2.3 Textures

patterns in luminance band (greylevel image)
structural and/or statistical properties
Cooccurrence matrices

1. compute normalized co-occurrence matrix $P$ for specified direction(s):
   (e.g. $0^\circ$, $90^\circ$, $45^\circ$, $135^\circ$)

   example: with 2 grey values,
   sequence of values (in one direction):
   111100001 vs. 110011001 vs. 101010101
   unnormalized matrices $\hat{P}$:

   $\begin{array}{c|cc}
   0 & 1 \\
   \hline
   0 & 3 & 1 \\
   1 & 1 & 3 \\
   \end{array}$

   $\begin{array}{c|cc}
   0 & 1 \\
   \hline
   0 & 2 & 2 \\
   1 & 2 & 2 \\
   \end{array}$

   $\begin{array}{c|cc}
   0 & 1 \\
   \hline
   0 & 0 & 4 \\
   1 & 4 & 0 \\
   \end{array}$
$P$ cooccurrence matrix

$P = \{p(i, j)\}, \ i = 1, \ldots Ng, \ j = 1, \ldots Ng$

$p(i, j)$ probability of pair $(i, j)$

$Ng$ # values of the grey scale

$\mu$ mean of the values $p(i, j)$

$\mu_x (\mu_y)$ mean of marginal probabilities in $x (y)$ direction

$\sigma_x (\sigma_y)$ standard deviation of marginal probabilities in $x (y)$ direction
2. compute the following features from $P$:

- angular second moment (homogeneity of the image)
  \[ f_1 = \sum_i \sum_j p(i, j)^2 \]

- contrast (local variations)
  \[ f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \right\} , \text{ with } |i - j| = n \]

- correlation (linear relationship between pixel values)
  \[ f_3 = \frac{\sum_i \sum_j (ij)p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y} \]

- variance (deviation from the average)
  \[ f_4 = \sum_i \sum_j (p(i, j) - \mu)^2 \]
• entropy

\[ f_5 = \sum_i \sum_j p(i, j) \log p(i, j) \]
2.2.4 Contours

Edge-based search

[Hirata/Kato]

Edge detection

input: full color image

1. reduction $\rightarrow$ regular-sized image
2. global edge detection $\rightarrow$ edge image
3. local edge detection $\rightarrow$ refined edge image
4. thinning and shrinking $\rightarrow$ abstract image
$|I_{ij}|$: intensity power,

$$|I_{ij}| = \left\{ \frac{1}{9} \sum_{r=i-1}^{i+1} \sum_{s=j-1}^{j+1} p_{rs}^2 \right\}^{1/2}$$

**gradient in RGB space:**

$$^1 \partial_{ij} = \frac{1}{|I_{ij}|} \frac{1}{3} \left\{ (p_{i-1j-1} + p_{ij-1} + p_{i+1j-1}) - (p_{i-1j+1} + p_{ij+1} + p_{i+1j+1}) \right\}$$

$$^2 \partial_{ij} = \frac{1}{|I_{ij}|} \frac{1}{3} \left\{ (p_{i-1j-1} + p_{i-1j} + p_{i-1j+1}) - (p_{i+1j-1} + p_{i+1j} + p_{i+1j+1}) \right\}$$

N. Fuhr
\[ 3 \partial_{ij} = \frac{1}{|I_{ij}|} \frac{1}{3} \left\{ (p_{i-1j-1} + p_{ij-1} + p_{i-1j}) - (p_{i+1j} + p_{ij+1} + p_{i+1j+1}) \right\} \]

\[ 4 \partial_{ij} = \frac{1}{|I_{ij}|} \frac{1}{3} \left\{ (p_{ij-1} + p_{i+1j-1} + p_{i+1j}) - (p_{i-1j} + p_{i-1j+1} + p_{1j+1}) \right\} \]

\[ |\partial_{ij}| = \max(|1 \partial_{ij}|, \cdots, |4 \partial_{ij}|) \]

**global edge candidates:**

calculate average and deviation for the gradients

\[ \mu = \frac{1}{MN} \frac{1}{4} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{k=1}^{4} |k \partial_{ij}| \]
\[ \sigma = \left\{ \frac{1}{MN} \frac{1}{4} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{k=1}^{4} (|k \partial_{ij}| - \mu)^2 \right\}^{1/2} \]

global edge candidate if \( |\partial_{ij}| \geq \mu + \sigma \)
local edge candidates:

local average and local deviation for the gradient values

\( (m, n: \text{window size, e.g. } n = m = 3) \)

\[
\mu_{ij} = \frac{1}{(2m + 1)(2n + 1)} \frac{1}{4} \sum_{r=i-m}^{i+m} \sum_{s=j-n}^{j+n} \sum_{k=1}^{4} |k \partial_{rs}|
\]

\[
\sigma_{ij} = \left\{ \frac{1}{(2m + 1)(2n + 1)} \frac{1}{4} \sum_{r=i-m}^{i+m} \sum_{s=j-n}^{j+n} \sum_{k=1}^{4} (|k \partial_{rs}| - \mu_{ij})^2 \right\}^{1/2}
\]

local edge candidate if \( |\partial_{ij}| \geq \mu_{ij} + \sigma_{ij} \)
Contour matching

\( P_t = \{p_{ij}\} \): abstract (monochrome) image, \( i, j = 0, \ldots, 63 \)

\( Q = \{q_{ij}\} \) query image (sketch), \( i, j = 0, \ldots, 63 \)

8 × 8 blocks per image

\( m \times n \) pixels per block

1. divide abstract image \( P_t \) and linear sketch \( Q \) into 8 × 8 local blocks

2. compute local correlation \( C_{\delta\varepsilon} \) between local blocks \( P_t \) and \( Q \), with shifting \( \delta, \varepsilon \):

\[
C_{\delta\varepsilon}^{ab} = \sum_{r=ma}^{m(a+1)-1} \sum_{s=nb}^{n(b+1)-1} \alpha \cdot p_{rs} \cdot q_{r+\delta, s+\varepsilon} + \beta \cdot \bar{p}_{rs} \cdot \bar{q}_{r+\delta, s+\varepsilon} + \gamma \cdot p_{rs} \oplus q_{r+\delta, s+\varepsilon}
\]

\( \alpha, \beta, \gamma \): weighting factors (edge/edge, blank/blank, different)

N. Fuhr
local correlation:

\[ a^b C = \max(a^b C_{\delta \varepsilon}), \quad -m/2 \leq \delta \leq m/2, \]
\[ -n/2 \leq \varepsilon \leq n/2 \]

3. compute global correlation:

\[ C_t = \sum_{a=0}^{7} \sum_{b=0}^{7} a^b C \]

4. rank images according to decreasing global correlation values
Local scope

Shifted position on an abstract image

Corresponding position on an abstract image

$\delta$ $\varepsilon$

$m \times n$ pixels

$2m \times 2n$ pixels

$M \times N$ blocks
2.3 **IRIS**

semantic indexing of images

1. image analysis
   - color
   - contour
   - texture

2. object recognition
   (a) basic objects:
      - clouds, snow, water, sky, forest, grass, sand, stone
   (b) high-level objects:
      - forestscene, skyscene, mountainscene, landscapescene, ...
2.3.1 System overview

- **colour-based segmentation**: 1) RGB/HLS-colour model, 2) colour of single segments, 3) segment grouping
- **texture-based segmentation**: 1) 2. order statistics, 2) neural network, 3) texture of single segments, 4) segment grouping
- **contour-based shape analysis**: 1) edge detection, 2) contour connection, 3) shape analysis

Annotations for image xxx:
- colour:
- texture:
- contour:
- objects:

Parsing of the hypothesis

Hypothesis construction

Construction of the neighborhood graph

Thesaurus
2.3.2 Image Analysis

Color

color model: HLS
R-G-B

H-L-S

N. Fuhr
IRIS subdivides color space into about 20 different colors

1. subdivide image into nonoverlapping tiles
2. compute color histogram for each tile
3. most frequent color =: color of tile
4. join tiles with similar colors and compute circumscribing rectangle
5. compute attributes of color rectangles:
   - position
   - size
   - color
   - color density
     (\# tiles with color / \# tiles in rectangle)
   - color evidence
Original image
color-based segmentation:

... 

colour2  HOR=mid, VER=up, SIZ=XL, SHP=Rect, COL=BLUE,  
UL=0—1, LR=44—11, DEN=415—495

colour3  HOR=mid, VER=mid, SIZ=M, SHP=Rect, COL=BLUE,  
UL=15—10, LR=44—17, DEN=136—240

colour4  HOR=left, VER=mid, SIZ=XS, SHP=Quad, COL=BLUE, 

N. Fuhr
UL=1—11, LR=1—11, DEN=1—1

colour5 HOR=left, VER=mid, SIZ=XS, SHP=Rect, COL=BLUE,

UL=3—11, LR=14—12, DEN=13—24

...
Texture

consider local distribution and variation of grey values

1. compute normalized co-occurrence matrix \( p \) for 4 directions: 0°, 90°, 45°, 135°

2. for each of the four directions, compute the following features from \( C \):
   - angular second moment
   - contrast (local variations)
   - correlation (linear relationship between pixel values)
   - variance (deviation from the average)
   - entropy

3. for each of the five parameters, compute the average from the values for the 4 directions
   (\( \rightarrow \) invariance against rotation)
4. feed average values into neural network
5. NN yields texture for each tile

6. join tiles with identical textures and compute circumscribing rectangles

7. compute attributes of texture rectangles:
   - position
   - size
   - texture
   - texture density (\# tiles with texture / \# tiles in rectangle)
texture3  HOR=mid,VER=mid,SIZ=L,SHP=Rect,TEX=ice,
       UL=2—2,LR=10—3,DEN=11—18

texture4  HOR=left,VER=mid,SIZ=S,SHP=Path,TEX=clouds,
       UL=0—3,LR=3—3,DEN=4—4

texture5  HOR=left,VER=mid,SIZ=S,SHP=Quad,TEX=stone,
       UL=4—3,LR=5—4,DEN=3—4

texture6  HOR=mid,VER=mid,SIZ=S,SHP=Rect,TEX=clouds,
       UL=5—3,LR=8—4,DEN=5—8
Contour

based on grey level image

1. gradient-based edge detection
   based on two convolution kernels
   \( I(x, y) \) image function
   \* convolution operator

\[
\nabla f(x, y) = \nabla G(x, y) \ast I(x, y)
\]

gives direction and magnitude of image gradient,
\rightarrow pixels with steepest slope along gradient = edge pixels
2. determination of object contours
   start with pixels with gradient magnitude exceeding threshold:
   if pixel is
     • isolated: no edge
     • termination pixel of a contour connection: search its neighbours
     • within a contour: do nothing
   method gives pixel connections which circumscribe image regions

3. shape analysis: compute
   • position of centroid
   • size of region
   • bound coordinates of region
extracted contour points:

extracted regions:

contour0 MID=24—7, SOP=45,
UL=0—0, LR=44—17, SHP=UND

contour1 MID=20—23, SOP=26,
UL=0—14, LR=44—28, SHP=UND

contour2 MID=21—17, SOP=19,
UL=0—11, LR=44—22, SHP=UND
2.3.3 Object Recognition

1. step from syntactical to semantical features:
   identification of primitive objects

2. derivation of higher-level semantical features

1. identification of primitive objects

   • basis: color, texture and contour features

   • for each feature, consider corresponding region

   • form graph describing topological relationships between feature regions:
     – node = feature
     – edge = topological relationship: overlaps, meets, contains
• formulate graph grammar rules for detecting primitive objects
Conditions of "Clouds"

\[
\text{predicate}((\text{valcompeq}(*\text{self}(2,\text{"colorseg"},\text{"COL"}),\text{"blue"})) \text{ ||} \\
\text{valcompeq}(*\text{self}(2,\text{"colorseg"},\text{"COL"}),\text{"white"})) \text{ &&} \\
\text{valcompeq}(*\text{self}(2,\text{"colorseg"},\text{"VER"}),\text{"up"})));
\]

\[
\text{predicate}(\text{nrkind}(*\text{self}(1,\text{"contourseg"}),\text{"contains"},*\text{self}(1,\text{"colorseg"}))) \text{ &&} \\
\text{nrkind}(*\text{self}(1,\text{"contourseg"}),\text{"contains"},*\text{self}(1,\text{"textureseg"})))
\]
2. derivation of higher-level semantical features

based on knowledge representation: