The University of Auckland

CBIR: Shape Descriptors / Image Indexing / Dimensionality Reduction

COMPSCI.708.S1.C

A/P Georgy Gimel'farb

The University of Auckland

Shape Feature Extraction

- Object shape → a clue to object recognition: shape carries **semantic information**
  - Other low-level features (colour, texture, or motion) do not directly reveal object identity
- But shape features are less developed than their colour and texture counterparts
- It is hardly possible to precisely segment an image with low-level features into meaningful regions related to objects of interest

---

The University of Auckland

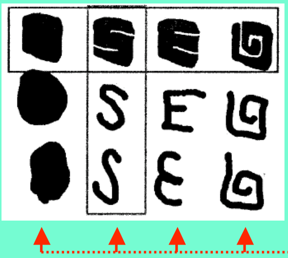
Shape Feature Extraction

- The notion of object shape has many meanings
  - Most of the real-world objects have 3D shapes
  - Images / videos deal with 2D projections of 3D objects having 2D shapes
- No mathematical description is able to fully capture all aspects of visually perceived shapes
- Shape comparison is also a difficult problem
- Today's CBIR exploits two groups of shape descriptors:
  - **Contour-based** descriptors representing an outer boundary (or contour)
  - **Region-based** descriptors representing an entire region

---

The University of Auckland

Contour- / Region-based Similarity



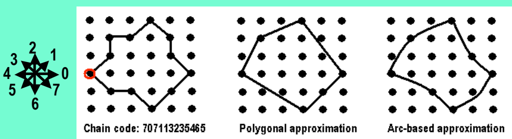
Region-based similarity

Contour-based similarity

The University of Auckland

Shape Representation by Boundary

- **Boundary** – a closed curve around a given shape
  - Curve can be specified by a chain code, polygon, a sequence of circular arcs or splines, etc
- A query by a graphical sketch: *an "elastic" template*



The University of Auckland

Shape Representation by Region

- Interior, or "body" descriptions with moment invariants or primitives: sets of points, rectangles, quadrics, disks, deformable templates, skeletons, …
- **Skeleton** → the axis of symmetry between the borders:
  - *Medial axis*: locus of inscribed maximum-size circles
  - *Shock set*: "grassfire" propagation from boundaries (i.e. singularities of collisions of propagating fronts)
- "**Blobworld**": an elliptic shape (the object's centroid + the scatter matrix) + texture + two dominant colours

The University of Auckland

Spatial Relationships of Objects

- Additional shape features: spatial *topological* and *directional* relationships of objects
- **Topological relationships** between object boundaries: "near to", "within", "adjacent to", etc.
- **Directional relationships**: relative positions of objects w.r.t. each other: "in front of", "on the left of", "on top of", etc.
- Spatial relationships are frequently described with the **attributed-relational graph** (**ARG**):
  - a *node* ⇒ an object
  - an *arc* between two nodes ⇒ a certain relationship between the objects

The University of Auckland

MPEG 7: Shape Descriptors

- Goals of selection:
  - Compactness
  - Invariance to scaling, rotation, translation
  - Invariance to shape distortions due to imaging conditions (e.g. perspective transformations of a 2D shape if viewing angle is changing)
- Selected shape descriptors:
  - 3D shape descriptor
  - Region-based shape descriptor
  - Contour-based shape descriptor
  - 2D/3D shape descriptor

The University of Auckland

3D Shape Descriptor

- **Shape spectrum**: the histogram of the shape index over the entire 3D surface
  - **Shape index** of an oriented 3D surface $S(x,y)$ at point (x,y):

    $$I_S(x,y) = 0.5 - \frac{1}{\pi}\tan^{-1}\frac{k_1(x,y)+k_2(x,y)}{k_1(x,y)-k_2(x,y)} \in [0,1]$$

    where $k_1(x,y) \geq k_2(x,y)$ are the principal curvatures at (x,y)
  - *Principal curvatures* – eigen-values of the 2x2 Hessian at (x,y)

    $$\left\| \begin{matrix} \frac{\partial^2 S(x,y)}{\partial x^2} & \frac{\partial^2 S(x,y)}{\partial x \partial y} \\ \frac{\partial^2 S(x,y)}{\partial y \partial x} & \frac{\partial^2 S(x,y)}{\partial y^2} \end{matrix} \right\| \mathbf{u}_i = k_i(x,y)\mathbf{u}_i; \quad i = 1,2;\ \mathbf{u}_1 \perp \mathbf{u}_2$$

The University of Auckland

3D Shape Descriptor

- Shape index describes the **local convexity** of a surface
  - For 3D meshes, it is computed for each vertex of the mesh
- **Shape histogram**: 100 bins, 12 bits/bin
- **Two additional variables**:
  - **Relative area of planar surface regions** of the mesh, w.r.t. the entire area of the mesh
  - **Relative area of all polygonal components** where reliable estimation of the shape index is impossible, w.r.t. the entire area of the 3D mesh

The University of Auckland

Region-based Shape Descriptor

- **Angular Radial Transformation** (ART): $c_0 = 1$ and $c_n = 2$; $n > 0$

  $$G_{nm} = \frac{c_n}{\pi}\int_0^{2\pi}\int_0^1 g(\rho,\theta)\cos(\pi n\rho)e^{-jm\theta}\,d\rho\, d\theta$$

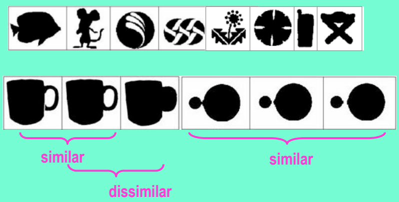
  image intensity

  basis function
- **Moment–based** description of spatial pixel distribution within a connected or disconnected 2D object region:
  - 35 ART coefficients (polar moments); $n = 10$; $m = 10$

The University of Auckland

Region-based Shape Descriptor

- The default descriptor: 140 bits
  - 35 coefficients quantised to 4 bits per coefficient



The University of Auckland

Contour-based Shape Descriptor

- Curvature scale-space (CSS) contour representation
- Descriptor: in average – 112 bits per contour
  - Eccentricity and circularity of the original and filtered contour (each 6 bits)
  - Number of peaks in the CSS image (6 bits)
  - Height of the highest peak (7 bits)
  - (x,y)-positions of the remaining peaks (9 bits per peak)
- N equi-distant points on the contour → an arbitrary start clockwise; grouped X and Y series of coordinates

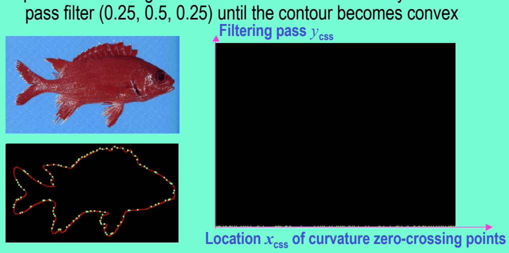
Semester 1, 2006 Lecture G9 13

The University of Auckland

From: SQUID system website; Centre for Vision, Speech and Signal Processing; Dept. EEE, University of Surrey, UK

Contour-based Shape Descriptor

Repetitive smoothing of X and Y contour coordinates by the low-pass filter (0.25, 0.5, 0.25) until the contour becomes convex



Semester 1, 2006 Lecture G9 14

The University of Auckland

Contour-based Shape Descriptor

- Due to smoothing, the contour evolves and its concave parts gradually flatten-out, until it becomes convex
  Contour evolution is given by a **CSS image** of zero-crossing points of curvature along each smoothed contour:
  x_{css} – the index of a point along the contour, i.e. its relative position along the contour of unit length;
  y_{css} – the number of passes of the coordinate filtering to smooth this particular contour
- The CSS image helps to explain the descriptor but does not have to be explicitly used for describing a shape

Semester 1, 2006 Lecture G9 15

The University of Auckland

Contour-based Shape Descriptor

- Each zero crossing along a contour separates concave and convex parts
- "Black" x_{css} in the row y_{css} of the CSS image indicates the relative zero crossing position w.r.t. the contour of the unit length obtained after y_{css} smoothing passes
- **Shape description**: quantised eccentricity and circularity plus ordered by decreasing values y_{css}, non-linearly transformed, and quantised coordinates (x_{css}, y_{css}) of prominent peaks in the CSS image

Semester 1, 2006 Lecture G9 16

The University of Auckland

2D / 3D Shape Descriptor

- Combining 2D descriptors representing a visual feature of a 3D object seen from different view angles
- A complete 3D view-based representation of the object
- Any 2D visual descriptor, such as contour shape, region shape, colour, or texture can be used.
- Supporting integration of the 2D descriptors used in the image plane to describe the 3D (real-world) objects
  - Experiments with 2D/3D descriptor and contour-based shape descriptor → good performance in multi-view 3D description

Semester 1, 2006 Lecture G9 17

The University of Auckland

Image Indexing

- Indexing accelerates the queries and overcomes the "curse of dimensionality" in the content-based search
- ***Whole image match***: the query template is an entire image and the similar images have to be retrieved
  - a single feature vector for indexing and retrieval
- ***Subimage match***: the query template is a portion of an image, and the images with similar portions or portions of images with desired objects have to be retrieved

Semester 1, 2006 Lecture G9 18

Similarity Between Features

- CBIR: low-level colour, texture, and shape descriptors
- Typically descriptors: multidimensional vectors
- Similarity of two images in the vector feature space:
  - the *range query:* all the points within a hyper-rectangle aligned with the coordinate axes
  - the *nearest-neighbour* or *within-distance* (α–cut) *query:* a particular metric in the feature space
  - dissimilarity between statistical distributions: the same metrics or specific measures

Semester 1, 2006 Lecture G9 19

Vector Space Distances

Euclidean (Cartesian)	$D_{[2]}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
Chebyshev	$D_{[\infty]}(\mathbf{x},\mathbf{y}) = \max_{i=1}^{n} \lvert x_i - y_i \rvert$
Manhattan (city-block)	$D_{[1]}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} \lvert x_i - y_i \rvert$
Minkowsky	$D_{[p]}(\mathbf{x},\mathbf{y}) = \left[\sum_{i=1}^{n} \lvert x_i - y_i \rvert^p\right]^{\frac{1}{p}}$
Weighted Minkowsky	$D_{[p,\mathbf{w}]}(\mathbf{x},\mathbf{y}) = \left[\sum_{i=1}^{n} w_i \lvert x_i - y_i \rvert^p\right]^{\frac{1}{p}}$
Mahalanobis	$D(\mathbf{x},\mathbf{y}) = \lvert \det \mathbf{C} \rvert^{1/n} (\mathbf{x}-\mathbf{y})^{\mathrm{T}} \mathbf{C}^{-1} (\mathbf{x}-\mathbf{y})$

Semester 1, 2006 Lecture G9 20

Distances / Similarity Measures

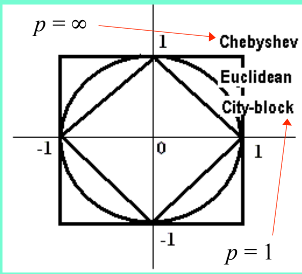
Generalised Euclidean (quadratic)	$D(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^{\mathrm{T}} \mathbf{K} (\mathbf{x}-\mathbf{y})$
Correlation	$\rho(\mathbf{x},\mathbf{y}) = \dfrac{\sum_{i=1}^{n}(x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x}_i)^2 \sum_{i=1}^{n}(y_i-\bar{y}_i)^2}}$
Relative entropy (Kullback-Leibler divergence)	$D(\mathbf{x} \Vert \mathbf{y}) = \sum_{i=1}^{n} x_i \log \frac{x_i}{y_i}$ when $\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i = 1$
χ^2-*Distance*	$D_{\chi^2}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{y_i}$ when $\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i = 1$

Semester 1, 2006 Lecture G9 21

Minkowski Distance

Chebyshev, Euclidean, city-block distances are particular cases of the Minkowski distance:

$$D_{[p]}(x,y) = \sum_{i=1}^{n} \lvert x_i - y_i \rvert^p$$



Semester 1, 2006 Lecture G9 22

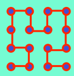
Similarity-based Search

- ***Vector space indexes****:* regions of the feature space
- ***Metric space indexes****:* distances between the vectors
- **Algorithmic indexing structures**:
  - ***Nonhierachical indexing***: the feature space is split into regions to be found in a fixed number of steps
  - ***Recursive indexing***: the space as a tree to optimise computational efficiency of the retrieval
  - ***Projection based indexing***: projections of vectors onto a subspace to reduce the dimensionality

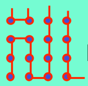
Semester 1, 2006 Lecture G9 23

Nonhierarchical Indexing

- The n-dimensional vectors are mapped onto the real line using a space-filling (e.g., Peano or Hilbert) curve and the mapped records are indexed with a 1D indexing structure



Hilbert curve



Peano curve

- Because space-filling curves preserve to some extent the neighbourhood relations between initial vectors, range, nearest-neighbour, and α-cut queries are rather closely approximated along the linear mapping

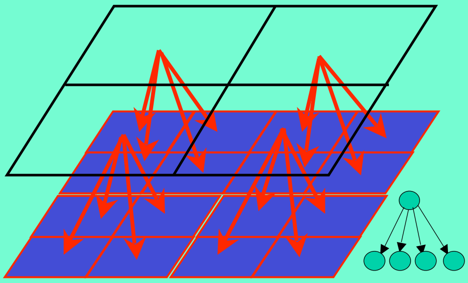
Semester 1, 2006 Lecture G9 24

Recursive Indexing

- The decomposition of the space by a tree:
- Most popular: quad-trees, *k–n–*trees and R–trees
  - ***Quad-trees***: in the *n*-dimensional space, each non-terminal node has 2^n children (correspong to the hyperrectangles aligned with the coordinate axes and splitting each axis into two parts)
  - ***k-n-trees***: the (*n*-1)-dimensional hyperplanes perpendicular to a coordinate axis selected by the data in the node divide the space; each nonterminal node has at least two children

Semester 1, 2006 Lecture G9 25

2D Quad-trees



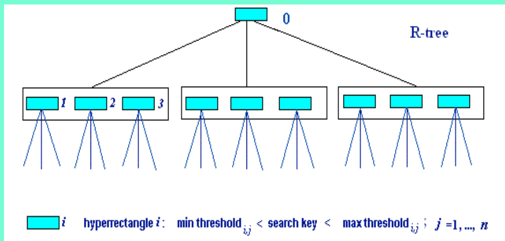
Semester 1, 2006 Lecture G9 26

Recursive Indexing

- **R-trees** generalise multi-way B-trees allowing an extremely efficient search for a scalar search key
- R-tree and modifications: the best for multidimensional indexing
  - Each internal node: a *k*-dimensional hyper-rectangle rather than a scalar range
  - The hyper-rectangle of the node contains all the (overlapping) hyper-rectangles of the children
  - To improve a performance, the R*-tree is proposed that minimises the overlap among the nodes

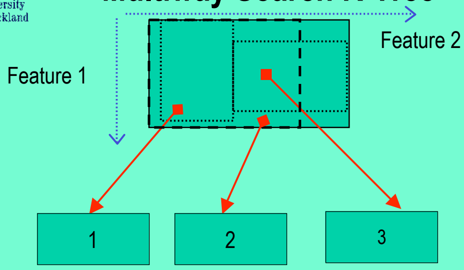
Semester 1, 2006 Lecture G9 27

Multiway Search R-Tree



Semester 1, 2006 Lecture G9 28

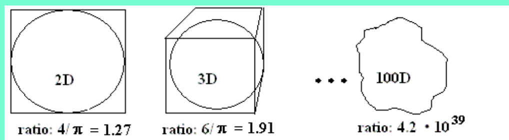
Multiway Search R-Tree



Semester 1, 2006 Lecture G9 29

Hypercube vs. Hypersphere

- R- and R*-trees work well until the dimension is < 20
- R-trees are extremely inefficient for an α-cut query with the Euclidean distance because the search is actually based on the minimum bounding hyperrectangle



Semester 1, 2006 Lecture G9 30

"Curse of Dimensionality"

- No tight data clusters in the high-dimensional cases
  - ***Example***: normal distribution of n-vectors $\mathbf{x}=[x_1, ..., x_n]$ with independent zero-mean components and the standard deviation s
  - Euclidean distance d^2 between $\mathbf{x}$ and $\mathbf{y}$: $d^2 = (x_1 - y_1)^2 + ... + (x_n - y_n)^2$ has math expectation $2ns^2$ and variance $4ns^4$
  - $s=1, n=1$ $\Rightarrow$ most of the distances $0.0 \le d \le 2.8$
  - $s=1, n=100 \Rightarrow$ most of the distances $11.4 \le d \le 16.1$
  - No points "close" to or "far" from the query: the α-cut and the nearest-neighbour search are meaningless

Dimensionality Reduction

- In practice the feature space has often a local structure that makes the close neighbourhood of a query image still meaningful
- Interdependent features can be approximated by their projections onto an appropriate lower-dimensional space, where the distance- or similarity-based indexing behaves well
- The mapping from a higher-dimensional to a lower-dimensional space is called ***dimensionality reduction*** and performed by selecting a subset of variables, or multidimensional scaling, or geometric hashing

Select a Subset of Variables

- Minimum error of approximating the vectors with lower-dimensional projections after a linear transformation of the feature space
  - Uncorrelated projections: the Karhunen-Loeve transform (**KLT**), principal component analysis (**PCA**), or singular value decomposition (**SVD**)
  - All these methods are equivalent, data-dependent, computationally expensive, and suited well for only static databases
  - Dynamic databases: special (and computationally very expensive) KLT/PCA/SVD techniques

Multidimensional Scaling

- Nonlinear mapping of the n-dimensional feature space into m-dimensional one ($m < n$)
- No general theory or precise definition of this approach
  - ***Metric multidimensial scaling***: minimum changes of the distances between the pairs of points
  - Numerous other statements of the problem exist
- Better reduces dimensionality than linear methods
  - But much heavier computations
  - Data-dependent approach poorly suited for dynamic databases

Geometric Hashing

- Data-independent mapping of the n-dimensional feature space into the 1D real line or the 2D real plane
- Ideally, hashing spreads the database uniformly across the range of the low-dimensional space, so that the metric properties of the hashed space differ significantly from those of the original feature space
- Difficulties in designing a good hashing function grow with the dimensionality of the original space

Special Indexing Structures

- Particular classes of queries: more efficient indexing
- CSVD (Clustering with Singular Value Decomposition)
  - Partitioning the data into homogeneous clusters and reducing the dimensionality of each cluster
  - the index is a tree: each node $\Rightarrow$ cluster parameters and dimensionality reduction data
  - Non-leaf nodes: to assign a query to its cluster
  - Terminal nodes (leaves): an indexing structure that supports nearest-neighbour queries