

Multimedia-Datenbanken

Kapitel 4: Multimedia-Daten – Bild (Vektor- und Raster-)

Friedrich-Alexander-Universität Erlangen-Nürnberg
Technische Fakultät, Institut für Informatik
Lehrstuhl für Informatik 6 (Datenbanksysteme)

Prof. Dr. Klaus Meyer-Wegener

Wintersemester 2002 / 2003

Technische Universität Kaiserslautern
Fachbereich Informatik
AG Datenbanken und Informationssysteme

Dr. Ulrich Marder

Wintersemester 2003 / 2004

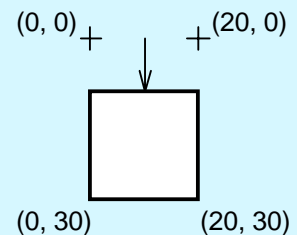
4.1 Graphik

- **Rohdaten:**
 - Menge (!)
von Linien und Flächen
mit Koordinatenangaben
und Attributen (Strichbreite, Farbe)
- **Registrierungsdaten:**
 - verwendetes Koordinatensystem
(kartesisch, polar)
 - Definition von Farben und
Texturen
- **Beschreibungsdaten:**
 - Gruppierung
von zusammenhängenden Linien
und Flächen
zu geometrischen Objekten
höherer Ordnung:
Quadrate, Quader,
Projektionen dreidimensionaler
Objekte

Anfangspunkt Endpunkt Strichstärke

Anfangspunkt	Endpunkt	Strichstärke
0	30	2
9	8	1
20	10	2
0	10	2
10	0	1
11	8	1
20	10	2

a) Menge von Liniendefinitionen



b) Dargestellte Graphik

Graphik (2)

□ Operationen:

- vorrangig ist die Ausgabe!
- für das Manipulieren spezielle Editoren, aber nicht im DBS
 - z. B. Zwischenspeicherung von technischen Zeichnungen; in CAD-System durch Projektion aus 3D-Modell abgeleitet
- „einfache“ Änderungen (Hinzufügen einer Linie) sollen jedoch möglich sein, ohne dass ganze Graphik überschrieben werden muss

□ Eingabe:

- Einlesen von Graphik-Dateien (z. B. GKS-Metafile)
- oder sukzessive:
 - Hinzufügen einer Linie zwischen zwei Punkten

Graphik (3)

□ Ausgabe (analog):

- in Datei (etwa zum Editieren)
- auf Bildschirm oder Plotter
- Linie für Linie (numberOfLines, getAllLines)

□ Modifikation:

- Hinzufügen einer Linie zwischen zwei Punkten
- Löschen der Linie mit angegebenen Endpunkten
- Verschieben der ganzen Graphik (Translation)
- Drehen um einen Punkt im bestimmten Winkel (Rotation)
- Vergrößern oder verkleinern

□ Auswerten, aggregieren, ableiten:

- rechtwinkligen Ausschnitt bilden (Clipping)
- außerdem Rekonstruktion geometrischer Objekte (soweit möglich)

Graphik (4)

- **Vergleich (für die Suche):**
 - auf Beschreibungsdaten wie bei Text
 - auf Rohdaten auch möglich:
Ähnlichkeit geometrischer Figuren
aber nur für sehr einfache Graphiken sinnvoll
- **wesentliche Erweiterungen notwendig für:**
 - gekrümmte Linien
 - Flächen
 - Segmente (Zusammenfassung beliebiger Elemente unter einem Namen)
- **Subtypen:**
 - technische Zeichnung
 - Balkendiagramm
 - Flussdiagramm
 - Landkarte
 - Geschäftsgraphik

4.2 Rasterbild

(vgl. Einleitung)

- **Rohdaten:**
 - Matrix von Bildpunkten (**Pixel** = Picture Element, manchmal auch „Pel“ genannt)
- **Registrierungsdaten:**
 - Anzahl Bits pro Pixel („Pixeltiefe“, Zahl der „Farbebenen“),
 - typisch: 1, 8, 24
 - aber auch andere Werte möglich!
 - Anzahl Pixel pro Zeile (Breite des Bildes)
 - Anzahl der Zeilen (Höhe des Bildes)
 - Art der linearen Abspeicherung:
 - zeilenweise, spaltenweise, nach Farbebenen
 - Bedeutung eines Pixels:
 - Grauwert
 - Farbdefinition
 - Index einer Farbtabelle (Colormap)

Rasterbild (2)

□ Registrierungsdaten (Forts.):

- ggf. Farbtabelle mit bestimmter Anzahl von Einträgen und Länge dieser Einträge (meist 24 Bit)
- ggf. Art der Farbdefinition:
 - RGB, IHS, YIQ,
- Verhältnis Höhe zu Breite bei Pixeln ("Aspect ratio")

□ Beschreibungsdaten:

- Text, Schlagworte, Wissensrepräsentation wie gehabt
- Graphik: erkannte Linien und Flächen
- darauf aufbauend zweidimensionale Objekte wie Kreise, Rechtecke usw.

Rasterbild (3)

□ Eingabe:

- von Datei
 - SUN Rasterfile, GIF, TIFF, JPEG,
- aus Hauptspeicherstruktur
 - Matrix, z. B. Ximage
- direkt vom Gerät
 - Scanner, Kamera

□ Ausgabe:

- auf Datei (in bestimmtem Format)
- auf Bildschirm oder Drucker
- an ein Programm als Hauptspeicherstruktur

Rasterbild (4)

- **Modifikation:**
 - Setzen einzelner Pixel
 - Ändern der Farbtabelle
 - bei künstlichen Farben (Tomogramme)
 - zur Analyse (Kontrastverstärkung)
 - zur Animation
 - Bitmap-Operationen: Überlagern anderer Bilder
 - (oft unerwünscht aus dokumentarischen Gründen)
- **Auswerten, aggregieren, ableiten:**
 - Kontrastverstärkung
 - Linienerkennung
 - Vergrößern oder verkleinern (Zoom)
 - Falschfarben
 - Umsetzung Farbe nach Grauwert
 - Ausschnittbildung (window)

Rasterbild (5)

- **Vergleich (Suche):**
 - Mustererkennung (Pattern Matching):
 - wird unterstützt durch Zugriffspfade (Iconic Indexing)
 - wichtig für manche Bildanalysetechniken
 - allgemein aber nicht sehr aussagekräftig
 - Bild-Bild-Vergleich
 - etwa Phantombild oder Photo mit Archiv zur Identifikation von Personen
 - Benutzung von Ähnlichkeitsmaßen?
 - ungelöstes Problem
 - Benutzung der Beschreibungsdaten:
 - mehr Semantik
 - etwa Graphik oder Text (s. dort)
- **Subtypen**
 - Satellitenfoto
 - Tomogramm
 - Röntgenaufnahme

4.3 Bildsuche

- ❑ **mehr Forschung als bei den anderen Medien**
 - weniger Erfahrung, großer aktueller Bedarf
- ❑ **etliche Techniken und Systeme verfügbar**
- ❑ **vier Klassen von Ansätzen:**
 - attributbasiert
 - Merkmalsextraktion und Objekterkennung („high-level“)
 - Textbeschreibung (Annotation)
 - elementare Bildmerkmale wie Farbe und Textur („low-level“)
- ❑ **davon inhaltsorientiert:**
 - Objekterkennung - aber noch nicht praktikabel
 - elementare Merkmale (syntaktisch, statistisch)
- ❑ **deshalb hier:**
 - Textbeschreibung und elementare Merkmale

Textbasierte Bildsuche

- ❑ **Bildbeschreibung**
 - uneingeschränkter (freier) Text
- ❑ **Anfragen**
 - Schlagworte oder freier Text
 - mit oder ohne Boolesche Operatoren
- ❑ **Suche**
 - konventionelle IR-Technik (siehe oben)
- ❑ **Unterschiede zu Textsuche:**
 - Beschreibung muss manuell erstellt werden: **Annotation**
(falls nicht eine Bildunterschrift genutzt werden kann)
 - effizient, vollständig, konsistent
 - Anwendungswissen, Thesaurus
 - Beschreibung unvollständig und/oder subjektiv
 - Wissensbasis und Relevanzrückkopplung benutzen

Textbasierte Suche (2)

□ Vorteile:

- Abstraktionen und Konzepte nutzbar ("Lächeln", "Glück")
 - mit den anderen Techniken sehr schwierig

□ Nachteile:

- die elementaren Merkmale (Textur, unregelmäßige Form) in Textform nur umständlich darzustellen
- Beispielbilder in Anfragen nicht unterstützt

Farbbasierte Suche

□ am weitesten verbreitet

- einfache Konzepte, leicht zu implementieren

□ zentrale Idee:

- Bilder finden mit ähnlicher Farbwahrnehmung wie in einem Beispielbild oder einer Beschreibung
- drei Primärfarben oder Farbkanäle, z. B. RGB
- jeder Kanal diskretisiert in m Intervalle
- Anzahl verschiedener Farbkombinationen („bins“) also m^3

□ Farbhistogramm

- $H(M)$ für Bild M
- Vektor $(h_1, h_2, \dots, h_j, \dots, h_n)$ mit n Anzahl der Bins und h_j Anzahl der Pixel von Bild M , die in den Bin j fallen

Farbhistogramm

□ Anfrage

- ebenfalls in Histogramm abbilden
 - aus Beispielfeld berechnen oder aus Beschreibung schätzen

□ Suche

- Distanz berechnen zwischen Histogrammen der Anfrage und der gespeicherten Bilder
- Bilder zurückgeben
 - entweder Distanz geringer als gegebener Schwellenwert
 - oder aufsteigend nach Distanz ordnen und die ersten k nehmen

□ Metriken

- viele vorgeschlagen
- am einfachsten: L-1

$$d(H_1, H_2) = \sum_{l=1}^n |h_{1,l} - h_{2,l}|$$

Farbhistogramm – Erweiterungen

□ Beschränkungen des einfachen Ansatzes:

- Ähnlichkeit zwischen Farben (und Bins) wird ignoriert
- Annahme: alle Bilder haben N Pixel
 - falls nicht, werden sie auf N Pixel normalisiert
- maximale Distanz zwischen zwei Bildern dann $2N$
 - in allen Bins, in denen das eine Bild Pixel hat, hat das andere gar keine und umgekehrt
- haben Bilder, die zwar ähnlich wahrnehmbare, aber keine gemeinsamen Farben aufweisen
- nicht ausreichend:
 - Anfragen geben nicht genau die gewünschten Farben vor, sondern nur annähernd
 - Farben verschieben sich leicht durch Rauschen oder Lichtverhältnisse
- Bins verschärfen das Problem durch harte Schnitte an ihren Rändern

Farbhistogramm – Erweiterungen (2)

□ Beiträge ähnlich wahrnehmbarer Farben einbeziehen in Distanzberechnung

- Methode von Niblack
- X Anfragehistogramm, Y Histogramm eines Bildes in der Datenbank
- Z Bin-für-Bin-Histogramm der Ähnlichkeit: $Z = |X - Y|$
- dann Ähnlichkeit von X und Y berechnen mit

$$\|Z\| = Z^T A Z$$

- A symmetrische Farbähnlichkeitsmatrix mit
$$a(i, j) = 1 - d(c_i, c_j) / d_{max}$$
- c_i und c_j sind der i -te und j -te Farb-Bin im Histogramm, $d(c_i, c_j)$ ist die Distanz ihrer Farben (in einer Transformation in den Munsell-Farbraum) und d_{max} ist die maximale Distanz aller Farben des Farbraums
- sind zwei Farben sehr verschieden, ist $d(c_i, c_j)$ sehr nah an d_{max} , also ist $a(i, j)$ sehr klein, also nicht einbeziehen in Ähnlichkeitsberechnung
- und umgekehrt

Farbhistogramm – Erweiterungen (3)

□ kumulatives Histogramm:

- $CH(M) = (ch_1, ch_2, \dots, ch_n)$ auf der Basis eines Farbhistogramms $H(M)$ für das Bild M mit

$$ch_i = \sum_{j \leq i} h_j$$

- Distanzen zwischen kumulativen Histogrammen wieder mit L-1 oder Euklidisch berechnen
- einfach
- bezieht allerdings Ähnlichkeiten in der Wahrnehmung nicht ein
 - gerade bei großen i werden sehr viele und sehr unterschiedliche Farben zusammengeworfen

Farbhistogramm – Erweiterungen (4)

- **wahrnehmungsgewichtetes Histogramm**
 - „perceptually weighted histogram“ (PWH)
 - repräsentative Farben des Farbraums bestimmen
 - Anzahl entspricht der der erforderlichen Bins
 - gleichmäßig über den Farbraum verteilt
 - Berechnung von Histogrammen:
 - zu jedem Pixel die zehn ähnlichsten repräsentativen Farben ermitteln
 - Distanz zu ihnen ermitteln
 - Gewichte umgekehrt proportional zur Distanz zuteilen
 - Pixel zählen also bei mehreren Bins mit
 - wichtigster Unterschied also: schon Histogramm-Berechnung berücksichtigt Farb-Ähnlichkeiten
 - soll bessere Leistung bieten als die anderen Methoden

Farbhistogramm – Erweiterungen (5)

- **räumliche Anordnung einbeziehen**
- **Segmentierung der Bilder in feste Zahl von Regionen**
 - dann Histogramm für jede Region
- **Trennung Hintergrund und Vordergrund**
 - flächiger Hintergrund dominiert oft das Histogramm
 - Vordergrund aber oft wichtiger für die Suche
 - daher getrennte Histogramme
 - Segmentierung muss nicht sehr genau sein, minimales umfassendes Rechteck genügt
 - kann ermittelt werden über Pixel-Variationen in horizontaler und vertikaler Richtung
 - oder auch manuell erzeugt beim Ablegen der Bilder

Farbhistogramm – Erweiterungen (6)

- **Trennung Hintergrund und Vordergrund (Forts.)**
 - Suche: Anfrage entscheidet sich für eine Option
 - nur ein Histogramm für Anfrage und Bild (wie gehabt)
 - Anfrage liefert nur Vordergrund-Histogramm
 - Anfrage liefert nur Hintergrund-Histogramm
 - beide Histogramme werden genutzt, und die beiden Distanzen werden gewichtet
 - Experimente zeigen deutliche Verbesserung

Farbhistogramm – Erweiterungen (7)

- **Farbverteilung**
 - bisher Farbräume gleichmäßig unterteilt
 - berücksichtigt nicht, welche Farben tatsächlich verwendet werden
 - Vorschläge zur ungleichmäßigen Unterteilung
 - Bereiche, denen viele Pixel zugeordnet werden, feiner unterteilen als andere
- **bessere Farbdarstellung**
 - welche Farbräume sind am besten geeignet für die Distanzberechnung?
 - wann bedeuten Pixel verschiedener Bilder das gleiche?
 - Farbräume:
 - RGB geräteabhängig, benötigt Weiß und die drei Primärfarben als Referenz, außerdem entspricht berechnete Distanz oft nicht der wahrgenommenen
 - CIE Luv, CIELab, HSI und HVC besser geeignet
 - wichtig: Gamma-Korrektur, Primärfarben, Weiß müssen mit erfasst werden (z. B. TIFF 6.0)

Formbasierte Suche

- **erfordert Segmentierung**
 - halbautomatische Methoden verfügbar
- **dann Darstellung der Formen und Ähnlichkeitsmessung**
 - jede Form sollte *eindeutige* Darstellung haben, invariant gegenüber Translation, Rotation und Skalierung
 - ähnliche Formen sollten auch ähnliche Darstellungen haben, so dass Suche wieder mit Distanzen arbeiten kann
- **Anfrage**
 - Beispielbild oder Skizze von Formen
- **Begriffe**
 - **Hauptachse**: gerade Linie zwischen den Randpunkten mit der größten Entfernung
 - **Nebenachse**: gerade Linie senkrecht zur Hauptachse und so lang, dass ein Rechteck parallel zur Haupt- und Nebenachse, das die Form gerade umschließt, aus den Längen der Haupt- und Nebenachse gebildet werden kann

Formbasierte Suche (2)

- **Begriffe (Forts.)**
 - **Basisrechteck**: das eben eingeführte Rechteck mit den Längen von Haupt- und Nebenachse als Seiten
 - **Exzentrizität**: das Verhältnis von Haupt- zu Nebenachse (≥ 1)
- **einfache Formendarstellung**
 - auf der Basis dieser vier Maße
 - im Prinzip für Beschreibung und Suche verwendbar
 - kennzeichnen eine Form aber nur sehr grob
 - daher meist zusammen mit weiteren Beschreibungen verwendet
 - z. B. QBIC (von IBM): Fläche, Zirkularität, Orientierung der Hauptachse und invariante Momente

Invariante Momente

□ Definition

- für Bild $f(x, y)$ ist Moment der Ordnung $(p + q)$ definiert als

$$m_{p,q} = \sum_x \sum_y x^p y^q f(x, y)$$

mit x, y Pixelposition im Bild und $f(x, y)$ Pixelintensität

- daraus berechenbar: zentrale Momente, normalisierte zentrale Momente usw.

□ Invarianz

- einige davon invariant gegenüber Translation, Rotation und Skalierung

□ Nutzen

- allerdings bei gleichen Momenten keine Garantie für gleiche Form, und bei sichtbar ungleichen Formen einige Momente ungleich, andere ähnlich
- Leistung in Vergleichsuntersuchungen nicht sehr hoch

Fourier-Deskriptoren

□ Form-Signatur

- Darstellung einer Form mittels einer Merkmalsfunktion

□ diskrete Fourier-Transformation

- auf die Signatur anwenden
- liefert Fourier-Deskriptoren der Form
- als Index genutzt

□ Beispiele für Form-Signaturen:

- Verwendung von Kurven, Radien oder Randkoordinaten
- Leistung der drei Varianten sehr ähnlich
- Radien am einfachsten zu implementieren

□ Radien

- geordnete Distanzen einer Reihe von Punkten auf dem Rand der Form vom Zentroiden, gleichmäßig verteilt (z. B. 64)

Histogramm signifikanter Kanten

- ❑ **Kantenerkennung**
- ❑ **Histogramm**
 - Kanten nach Richtungen gruppieren und Gruppengröße zählen:
- ❑ **Distanz**
 - Metrik L-1
- ❑ **Normalisierung**
 - zunächst nicht invariant gegen Rotation und Skalierung
 - Skalierung: Normalisierung nach Anzahl der Kanten
 - Zahl der Kanten allerdings nicht direkt proportional zur Größe der Form
 - Rotation: gemeinsame Verschiebung zwischen den Bins, also alle Verschiebungen probieren
 - langsam

Geordnete Liste interessanter Punkte

- ❑ **Randpunkte**
- ❑ **Basisvektor**
 - Liste von n Punkten, ein Paar herausnehmen
 - normalisieren auf Einheitsvektor entlang der x -Achse
 - alle anderen interessanten Punkte in dieses Koordinatensystem transformieren
 - ergibt Darstellung der Form
- ❑ **Skalierung**
 - invariant wegen der Transformation auf Einheitsvektor
- ❑ **Rotation**
 - alle Paare nacheinander zum Basisvektor machen
 - $n - 1$ Punktlisten
- ❑ **Bewertung**
 - Punktlisten groß \rightarrow langsam; schwierig: Listen nicht gleich lang

Anpassung elastischer Muster

- „elastic template matching“ (ETM)
- **Anfrage-Form verbiegen**
 - so nah wie möglich an Formen in der Datenbank bringen
- **Parameter zur Ermittlung der Ähnlichkeit**
 - Übereinstimmung der deformierten Anfrage mit Datenbank-Form
 - Energie, die zum Verbiegen benötigt wurde
 - Komplexität der Anfrage-Form
 - Korrelation der ursprünglichen und der deformierten Anfrage-Form
- **neuronales Netz**
 - berechnet einzelnen Ähnlichkeitswert auf der Basis dieser Parameter

Regionengestützte Formdarstellung

- **Problem**
 - bei allen bisherigen Verfahren entspricht die berechnete Ähnlichkeit nicht immer der wahrgenommenen
- **weiterer Ansatz: Regionen einbeziehen**
 - verspricht bessere Ergebnisse in dieser Hinsicht
 - ausführlicher betrachten
 - sehr leistungsfähig
 - als Beispiel für den vollständigen Prozess der Merkmalsdarstellung, Normalisierung und Distanzberechnung

Regionen – Idee

- ❑ **Raster überlagern**
 - quadratische Zellen gleicher Größe
 - gerade groß genug für die Form
- ❑ **Zellen**
 - vollständig, teilweise oder gar nicht von der Form ausgefüllt
 - erhalten eine 1, wenn *zu mindestens 15 % ausgefüllt*, sonst eine 0
 - Anordnung von links nach rechts und von oben nach unten durch das Raster: **Binärfolge** für die Form
 - kompakt, leicht zu ermitteln, translationsinvariant
- ❑ **Rastergröße**
 - je kleiner die Zellen, desto genauer die Darstellung der Form und desto größer der Aufwand beim Speichern und Berechnen
 - Kompromiss: Zellen zwischen 10x10 und 20x20 Pixeln

Regionen – Rotations- und Skalierungs-Normalisierung

- ❑ **Formen in einheitliche Ausrichtung bringen**
- ❑ **rotieren:**
 - (Hauptachse) parallel zur x-Achse machen
- ❑ **dann aber immer noch zwei mögliche Positionen:**
 - normal und auf dem Kopf
 - ergibt zwei Binärfolgen
 - nicht bei den gespeicherten Objekten
 - doppelter Speicherplatzbedarf
 - sondern bei den Anfragen
- ❑ **skalieren:**
 - Formen proportional vergrößern oder verkleinern, bis Hauptachse die gleiche Länge hat (in Experimenten: 192 Pixel)

Regionen – Eindeutige Formdarstellung

- ❑ **nach Normalisierung und Entscheidung für Zellgröße**
 - falls Hauptachse eindeutig
- ❑ **Länge der Binärfolge**
 - Raster gerade groß genug für normalisierte Form:
Zahl der Zellen in der x-Richtung immer gleich
 - bei Zellgröße 24x24 und Hauptachse von 192 Pixeln: 8 Zellen
 - Zahl der Zellen in y-Richtung von Exzentrizität abhängig,
aber nicht mehr als in x-Richtung
 - zwischen 1 und 8

Regionen – Ähnlichkeitsmaß

- ❑ **grundsätzlich**
 - Distanz = Zahl der ungleichen Zellen
- ❑ **Exzentrizität noch berücksichtigen**
 - bei gleicher Rastergröße bitweiser Vergleich
 - bei sehr großem Unterschied in der Zahl der Zellen in y-Richtung:
Formen müssen ungleich sein
 - Schwellenwert hängt ab von der Zellgröße und der Anwendung
 - typisches Beispiel: 3
 - bei kleinem Unterschied in der Zahl der Zellen in y-Richtung:
mit Nullen auffüllen
und wieder bitweise vergleichen

Regionen – weitere Details

- **Spiegelung**
 - horizontal oder vertikal
 - soll auch als ähnlich gewertet werden
 - zwei weitere Binärfolgen – zu einer Anfrage
- **mehrere Hauptachsen**
 - jede auswerten und Binärfolge speichern
 - Distanz dann paarweise berechnen (jede mit jeder) und Minimum verwenden

Regionen – Indexierungs- und Retrieval-Prozess

- **für jede Form in der Datenbank**
 - Haupt- und Nebenachsen sowie Exzentrizität bestimmen
 - Rotation, so dass Hauptachse parallel zur x-Achse, und Skalierung, so dass Hauptachse einheitliche Länge
 - Raster mit fester Zellgröße überlagern
 - Zuordnung von 0 und 1 zu den Zellen; zeilenweises Lesen liefert Binärfolge
 - Binärfolge und Länge der Nebenachse speichern
- **für Anfragen**
 - Binärfolge analog ermitteln – allerdings je eine für normal und kopfüber sowie die Spiegelungen, also vier
 - suchen nach Binärfolgen in der Datenbank mit (nahezu) gleicher Länge der Nebenachse (gleicher Exzentrizität)
 - Distanzberechnung (Anzahl unterschiedlicher Bits)
 - Ablieferung aufsteigend nach Distanz geordnet

Texturbasierte Suche

- ❑ **schwierig zu beschreiben, Wahrnehmung subjektiv**
- ❑ **ein Ansatz (Tamura et al.): sechs Merkmale**
 - Grobheit
 - wichtigstes Merkmal (manchmal einziges)
 - Größe unterscheidbarer Bildelemente
 - Kontrast
 - Bereich der Grauwerte, Kantenschärfe, Periode der Wiederholungselemente
 - Gerichtetheit
 - Form und Ort der Elemente
 - Linienartigkeit
 - Form der Elemente: Linie oder Fleck
 - Regularität
 - Variation in der Platzierung der Elemente
 - Rauheit

Systeme

- ❑ **müssen Kombination bieten**
 - besonders elementare Merkmale *und* Text
- ❑ **QBIC**
 - Beispielbild (Farbe, Form und Textur) plus Schlagworte
 - einiges davon in DB2 aufgenommen
 - www.qbic.almaden.ibm.com
 - Demonstration
- ❑ **Virage**
 - Merkmale Farbe, Form, Textur, aber auch anwendungsabhängig
 - www.virage.com
- ❑ **WebSEEK**
 - www.ctr.columbia.edu/webseek

Munsell-Farbmodell (vereinfacht)

