Farb- und Textur-Extraktion und -Deskription nach dem MPEG-7-Standard

Wolfgang Spiessl
spiessl@cip.ifi.lmu.de

Universität München
Amalienstrasse 17, 80333 Munich, Germany

Zusammenfassung Diese Arbeit gibt einen Überblick über die Farb- und Texturdeskriptoren, die im MPEG-7-Standard festgelegt wurden. Jeder Deskriptor wird detailliert im Hinblick auf Extraktion und Repräsentation der gewonnenen Metadaten betrachtet, und es werden sowohl Beispiele als auch Anwendungsmöglichkeiten aufgezeigt. Es wird zudem eine Bewertung der Deskriptoren anhand einer statistischen Untersuchung von Horst Eidenberger gegeben. Dabei zeigt sich, dass die betrachteten Deskriptoren hohe Redundanz aufweisen und sich einige schlecht für monochrome Bilder eignen. Der Einsatz dieser Deskriptoren ist in der Realität sehr begrenzt, Privatunternehmen entwickeln meist eigene Lösungen.

1 Einleitung

Der im Jahr 2002 verabschiedete MPEG-7-Standard, auch **Multimedia Content Description Interface** genannt, dient zur Ausstattung von Multimedia Content mit Metainformationen. Der wichtigste Teil von MPEG-7 ist die Festlegung einer Beschreibung, die die Struktur und die Semantik der Metainformation vorgibt. Dazu werden so genannte Deskriptoren verwendet. Es sind Deskriptoren definiert für Farbe (Color), Texturen (Texture), Formen (Shape), Bewegung (Motion) und Audio, als Spezialfall zwischen Texturen und Formen einzuordnen ist die Beschreibung von menschlichen Gesichtern (Face Description). Diese Arbeit beschäftigt sich ausschließlich mit Farb- und Texturdeskription. In Abschnitt 2 und 3 werden die Farb- und Texturdeskriptoren, die MPEG-7 bietet, detailliert betrachtet. Es werden jeweils kurz die Grundlagen der Extraktion von Metadaten sowie die Darstellung der extrahierten Daten vorgestellt, daran anschließend werden in knapper Form Beispiele, Veranschaulichungen und Anwendungsmöglichkeiten aufgezeigt. In Abschnitt 4 findet sich eine Untersuchung, die von Horst Eidenberger an der TU Wien durchgeführt wurde, um die Qualität der MPEG-7 Deskriptoren zu bewerten.
Der Zweck dieser Arbeit besteht nicht darin, eine vollständige Erläuterung aller Aspekte von MPEG-7 Farb- und Texturdeskriptoren zu geben, sondern einen Überblick über deren Funktionsweise und Anwendungsmöglichkeiten zu bieten.

2 Farbdeskriptoren

Deskriptoren in MPEG-7 definieren die Syntax und die Semantik eines spezifischen Charakteristikums von Daten (Feature). Farbdeskriptoren beschreiben

folglich nur Merkmale, Attribute oder Gruppen von Attributen, die die unterschiedlichen Eigenschaften und Darstellungsweisen von Farben betreffen. Farben sind allgemein unabhängig von der Größe und Orientierung eines Bildes und robust gegenüber Änderungen im Hintergrund. Zudem können Farben zum einen von Maschinen sehr gut verarbeitet und von Menschen sehr gut wahrgenommen und unterschieden werden, was die Bedeutung der Farbdeskriptoren verdeutlicht. In MPEG-7 sind folgende Farbdeskriptoren standardisiert: der *Dominant Color Descriptor, (DCD)*, der *Scalable Color Descriptor, (SCD)*, der *Group of Frames/Group of Pictures Descriptor (GoF/GoP)*, der *Color Structure Descriptor (CSD)* und der *Color Layout Descriptor (CLD)* [1]. Anfänglich war lediglich ein Farbhistogrammdeskriptor definiert, der die Aufgabe von SCD, CSD und GoF/GoP übernahm. Nach einer intensiven Testphase wurde dieser jedoch aufgespalten, da er zu viele voneinander unabhängige Dimensionen umfasste, wie den Farbraum, die Wahl der Quantisierung im Farbraum und die Quantisierung der Histogrammwerte, als dass man sie in einem einzigen Deskriptor vereinigen hätte können (siehe Abbildung 1). Es ist zudem ein weiterer Deskriptor definiert, der *Color Space Descriptor*, der allerdings nicht als eigenständiger Deskriptor eingestuft wird, sondern als Hilfsdeskriptor für alle anderen Deskriptoren, die auf unterschiedlichen Farbräumen arbeiten. Dieser soll im Folgenden ebenfalls kurz vorgestellt werden.

2.1 Übersicht über die Farbdeskriptoren

Abbildung 1 [2] bietet einen Gesamtüberblick über die MPEG-7 Farbdeskriptoren. Der SCD, der GoF/GoP Descriptor und der CSD basieren auf Farbhistogrammen, was in der Abbildung durch graue Unterlegung gekennzeichnet ist.

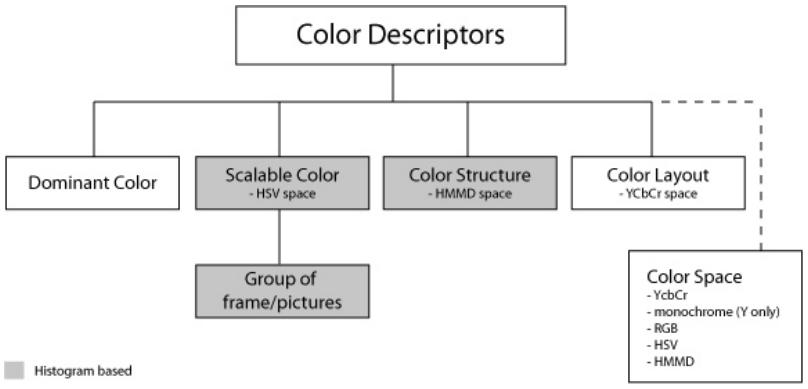

Abbildung 1. Übersicht über die Farbdeskriptoren in MPEG-7

2.2 Deskriptoren im Detail

Color Space Descriptor. Wie bereits eingangs erwähnt, handelt es sich beim Color Space Descriptor um einen Hilfsdeskriptor für Farbdeskriptoren, die auf

einen Farbraum festgelegt sind. Die Farbräume, die in MPEG-7 verwendet werden, sind RGB, YCbCr, HSV und HMMD. Der RGB-Farbraum bildet dabei die Grundlage, alle anderen können aus ihm berechnet werden.
Der **YCbCr**-Farbraum lässt sich durch eine einfache Lineartransformation angeben [1]:

$$\begin{aligned} \mathrm{Y} &= 0.299 * \mathrm{R} + 0.587 * \mathrm{G} + 0.114 * \mathrm{B} \\ \mathrm{Cb} &= -0.169 * \mathrm{R} - 0.331 * \mathrm{G} + 0.500 * \mathrm{B} \\ \mathrm{Cr} &= 0.500 * \mathrm{R} - 0.419 * \mathrm{G} - 0.081 * \mathrm{B} \end{aligned}$$

Für Graustufenbilder wird nur die Helligkeitskomponente (Y) benutzt.

Der **HSV**-Farbraum setzt sich zusammen aus Hue (Farbton, 0-360°), Saturation (Sättigung, 0-1, horizontale Achse) und Value (Helligkeit, 0-1, vertikale Achse) und wird als Zylinder dargestellt (siehe Abbildung 2 [3]). Die Komponenten errechnen sich aus einer nichtlinearen Transformation aus dem RGB-Farbraum. Der HSV-Farbraum wird vom Scalable Color Descriptor und vom Group-of-Frames / Group-of-Picture Descriptor verwendet.

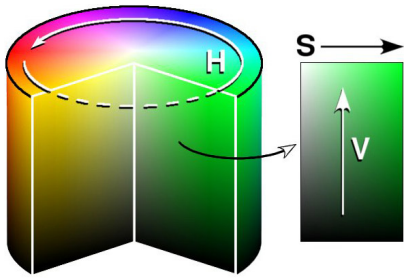

Abbildung 2. HSV Farbraum

Einen wahrnehmungsorientierten Farbraum stellt der **HMMD** (Hue-Max-Min-Diff) dar. Dieser ist als Doppelkegel (siehe Abbildung 3) definiert und setzt sich wie folgt aus dem RGB-Farbraum zusammen [2]:

$$\begin{aligned} Max &= max(R, G, B); \\ Min &= min(R, G, B); \\ Diff &= Max - Min; \\ Sum &= \frac{Max + Min}{2}; \end{aligned}$$

Dieser Farbraum besteht also aus eigentlich fünf Komponenten, dabei ist jeweils eine Menge von drei Komponenten ausreichend, um einen Punkt in Raum eindeutig zu identifizieren: {Hue, Max, Min} oder {Hue, Diff, Sum}. Er wird ausschließlich vom Color Structure Descriptor verwendet.

Dominant Color Descriptor. Mit dem DCD ist es möglich, lokal wie auch global eine kleine Menge von repräsentativen Farben in einem Bild kompakt zu

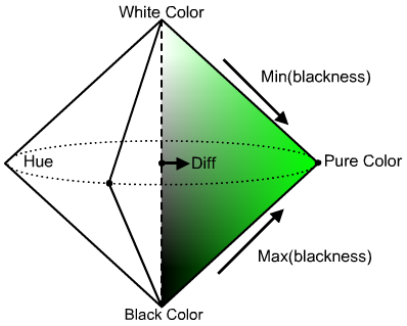

Abbildung 3. HMMD-Farbraum

beschreiben. Damit können Bilddatenbanken effizient nach ähnlichen Bildern durchsucht werden. Wie in Abbildung 1 zu sehen, ist der DCD nicht auf einen Farbraum festgelegt, sondern die repräsentativen Farben werden für jedes zu vergleichende Bild neu berechnet. Der Deskriptor besteht aus den repräsentativen Farben $(1, 2, ...N)$, ihren prozentualen Anteilen p im betrachteten Bildausschnitt, einem optionalen Varianzfaktor v und einem räumlichem Kohärenz-/Zusammenhangsfaktor s:

$$F = \{(c_i, p_i, v_i), s\}, \qquad (i = 1, 2, ..., N)$$

Um diese Daten zu extrahieren, sind verschiedene Schritte notwendig. Zur Clusterung der Pixelfarbwerte wird der Generalized LLoyd Algorithmus [1] verwendet. Dabei wird empfohlen, das Bild vorher in den CIE-LUV-Farbraum zu überführen. Zur Feststellung des räumlichen Zusammenhangs von Pixeln mit einer dominanten Farbe wird mit Hilfe einer 3×3 Pixel großen Maske die Vier-Nachbarschaft (Four Connectivity) dieser Pixel überprüft und daraus die Kohärenz s berechnet.
Um nun eine Ähnlichkeitsabfrage (Similarity Matching) durchzuführen, wird die Euklidische Distanz und die räumliche Kohärenz der Deskriptionen der beiden zu vergleichenden Bilder verglichen. Die Verschiedenheit zwischen F_1 und F_2 berechnet sich aus

$$D^2(F_1, F_2) = \sum_{i=1}^{N_1} p_{1i}^2 + \sum_{j=1}^{N_2} p_{2i}^2 - \sum_{i=1}^{N_1} \sum_{i=1}^{N_1} 2a_{1i,2j} p_{1i} p_{2j}$$

$a_{k,l}$ ist hier der Ähnlichkeitskoeffizient zwischen den beiden Farben c_1 und c_2, dieser berechnet sich aus

$$a_{k,l} = \begin{cases} 1 - d_{k,l}/d_{max} & d_{k,l} \leq T_d \\ 0 & d_{k,l} > T_d \end{cases}$$

wobei $d_{k,l} = ||c_k - c_l||$ die Euklidische Distanz zwischen den Farben c_k und c_l bezeichnet. Als Wert der Maximaldistanz T_d wird zwischen 10 und 20 empfohlen, wenn vorher die Umrechnung nach CIE-LUV gemacht wurde [1].

Anwendungen und Beispiele. Wie bereits erwähnt, eignet sich der DCD gut für Zwecke, wo eine kleine Zahl von repräsentativen Farben ausreichend ist, um ein Bild zu beschreiben. Dies ist oft der Fall bei Firmenlogos oder bei Länderflaggen. In dem Beispiel in Abbildung 4 wird der DCD auf jede Flagge angewendet

Abbildung 4. Beispiel für eine Anwendung des DCDs

und die resultierenden Deskriptionen miteinander verglichen. Dieser Vergleich ergibt ein hohe Ähnlichkeit zwischen der deutschen und der belgischen Flagge, während die niederländische nur eine geringe Ähnlichkeit (aufgrund der roten Farbe) gegenüber den anderen aufweist.

Scalable Color Descriptor. Der SCD beschreibt im Grunde nichts anderes als ein Farbhistogramm, was der Farbverteilung in einem Bild entspricht. Das Histogramm wird mit Hilfe einer Haar-Transformation codiert, damit Skalierbarkeit im Sinne von Anzahl der Histogramm-Bins und Genauigkeit gewährleistet werden kann. Um eine effiziente und kompakte Darstellung zu ermöglichen, durchlaufen die Histogrammwerte verschiedene Schritte (siehe Abbildung 5, [2]):

1. Die Werte werden normalisiert und nicht-linear von einer 11-bit- auf eine 4-bit-Repräsentation abgebildet, wobei kleineren Werten größere Signifikanz zugemessen wird als größeren.
2. Da bei vier Bits für 256 Bins 1024 Bits pro Histogramm erforderlich wären, werden die 256 Bins mit einer Haar-Transformation auf 128 Koeffizienten reduziert. Das geschieht durch primitive Hoch- und Tiefpassfilter, die durch einfache Summen und Differenzen von jeweils zwei benachbarten Werten realisiert werden. Dieser Schritt wird solange wiederholt und dabei jeweils die Anzahl der Koeffizienten auf die Hälfte reduziert, bis noch 16 Koeffizienten übrig sind.
3. Es erfolgt eine lineare Quantisierung, um den Speicherbedarf zu optimieren.

Je nach gewünschter Auflösung kann nun die Anzahl der Bins {256, 128, 64, 32, 16} nach unten skaliert werden. Digitale Fotos weisen für gewöhnlich eine hohe Redundanz zwischen benachbarten Farbwerten auf, was aus leichten Variationen in der Farbwahrnehmung resultiert. Aus diesem Grund können benachbarte Histogrammbins oft ohne große Verluste zusammengefasst werden, um eine möglichste kompakte Darstellung zu erreichen.
Eine andere Art von Skalierbarkeit kann durch eine Skalierung der quantisierten Repräsentation der Koeffizienten auf unterschiedliche Anzahl von Bits erreicht werden. Beispielsweise seien fünf Koeffizienten durch 8,4,7,3,7 Bits repräsentiert.

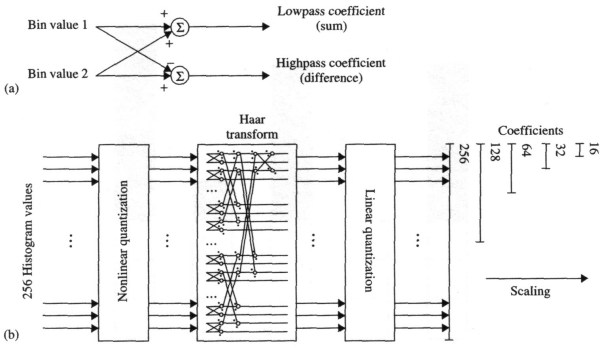

Abbildung 5. Haar-Transformation

Durch Weglassen der drei niederwertigsten Bits (Bitplanes discarding) bleiben für diese Werte 5,1,4,0,4 übrig, wodurch eine noch kompaktere Darstellung möglich ist.
Als Metainformationen gespeichert werden nun die Anzahl der Koeffizienten (*NumberofCoefficients*), die Anzahl der weggelassenen Bits (*NumberofBitplanesDiscarded*), die Vorzeichen der Koeffizienten (*CoefficientsSign*) und der Wert der Koeffizienten (*BitPlane*). Zum Vergleich von Bildern wird die L1-Norm verwendet, also die Summe über die absoluten Differenzen der Histogrammwerte:

$$D = \sum_{i}^{N} |c_{1i} - c_{2i}|$$

Anwendungen und Beispiele. Der SCD wird im Allgemeinen für Fotos verwendet. Diese weisen eine große Zahl von Farben auf, aus denen ein Histogramm mit 256 Einträgen erstellt wird. Darauf den SCD angewendet ergibt ein skalierbare Deskription des Histogramms, welche dann mit anderen SC Deskriptionen verglichen werden kann. In Abbildung 6 [4] sind drei Bilder zusammen mit ihren Farbhistogrammen dargestellt. Das linke und das mittlere Bild weisen in der vereinfachten Darstellung ihrer Histogramme eine deutlich höhere Ähnlichkeit auf als das Bild auf der rechten Seite.

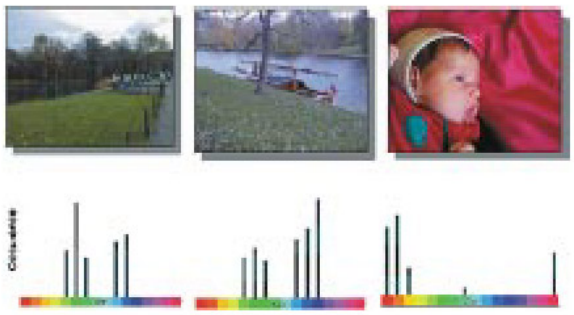

Abbildung 6. Beispiel einer Anwendung des SCD

Group-of-Frame / Group-of-Picture Descriptor. Der GoF/GoP-Descriptor wird zur Repräsentation von Farbmerkmalen in mehreren Einzelbildern oder in mehreren Frames eines Videosegments verwendet. Das geschieht durch Aggregation der einzelnen Histogramme aller Einzelbilder auf drei unterschiedliche Arten: Durchschnitts-, Median- oder Intersektionsaggregation [5]. Das Durchschnittshistogramm wird durch Aufsummierung und anschließende Normalisierung der Bin-Werte durch die N Einzelbilder gebildet. Beim Medianhistogramm wird eine Liste der einzelnen Bin-Werte für alle Einzelbilder erstellt und daraus jeweils der Median genommen. So bleiben einzelne Extremwerte ohne Einfluss auf das Aggregationshistogramm. Das Intersektionshistogramm ist eine Art "kleinster gemeinsamer Nenner" aller Einzelbilder, da jeweils der Minimalwert eines Bins in allen Bildern gewählt wird. So wird erreicht, dass das Intersektionshistogramm die Farbwerte beschreibt, die in allen Einzelbildern vorkommen - in denen sie sich also überschneiden.
Wie in Abbildung 1 zu sehen, stellt dieser Deskriptor eine Erweiterung des SCD dar. Der Vergleich zwischen unterschiedlichen GoF/GoPs wird exakt so durchgeführt, wie im oberen Abschnitt unter SCD beschrieben wurde. Die Darstellung der gewonnenen Metadaten erfolgt schlichtweg so, dass die Darstellung des SCD verwendet wird mit der zusätzlichen Angabe, welche Art von Aggregation durchgeführt wurde [1].

Anwendungen und Beispiele. Wie der Name bereits besagt, wird der GoF/GoP Descriptor verwendet, um Bildfolgen zu vergleichen. Eine praktische Anwendung hierfür ist das sog. Video-Segment-to-Segment Matching. Videosegmente werden also auf ihre Ähnlichkeit zu anderen Videosegmenten verglichen, jedoch nur im Hinblick auf Ähnlichkeit in der Farbdomäne. Bewegungen oder Formen lassen sich mit dem GoF/GoP Descriptor nicht feststellen.

Color Layout Descriptor. Der CLD dient in erster Linie dazu, räumliche Farbverteilung in beliebigen Bildbereichen zu beschreiben. Damit ist eine effiziente Möglichkeit gegeben, skizzenbasiert nach relevanten Bildern zu suchen. Die selbe Funktionalität kann durch eine gerasterte Strukturanalyse in Kombination mit dominanten Farben erreicht werden, jedoch mit wesentlich mehr Speicheraufwand und deutlich langsamer als mit dem CLD.
Die Extraktion erfolgt in vier Schritten (siehe Abbildung 7, [1]):

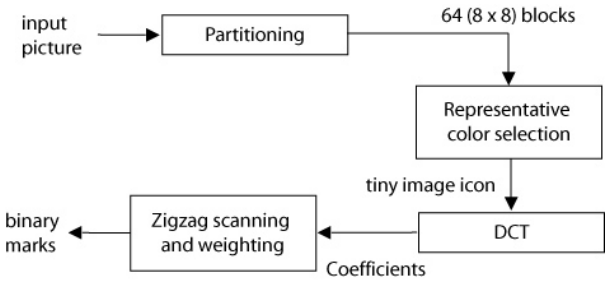

Abbildung 7. Der Extraktionsprozess des CLD

1. Das Bild wird in 64 Blöcke (8×8 Raster) eingeteilt. Damit wird eine Unabhängigkeit von Auflösung oder Skalierung erreicht.
2. Aus jedem Block wird eine repräsentative Farbe aus dem YCbCr Farbraum berechnet. Es wird ausdrücklich empfohlen, den Durchschnittswert zu bilden, da dies effizient zu bewerkstelligen ist und gute Resultate liefert.
3. Über die Durchschnittswerte wird eine 8×8 DCT (Diskrete Cosinus Transformation) berechnet mit 64 Koeffizienten als Resultat.
4. Niederfrequente Koeffizienten werden benutzt, um im Zick-Zack das Bild zu scannen und zu quantisieren, um den Speicherverbrauch zu optimieren.

Zum Vergleich von CLDs von zwei Bildern DY, DCr, DCb und DY', DCr', DCb' wird wieder die Distanz berechnet:

$$D = \sqrt{\sum_i w_{yi}(DY_i - DY_i')^2} + \sqrt{\sum_i w_{bi}(DCb_i - DCb_i')^2} + \sqrt{\sum_i w_{ri}(DCr_i - DCr_i')^2}$$

w bezeichnet hier die Gewichtung und i die Reihenfolge, in der das Bild im Zickzack quantisiert wird, wobei die Distanzen nicht-linear gewichtet werden, indem niederfrequenteren Komponenten ein größeres Gewicht beigemessen wird.

Anwendungen und Beispiele. Wie bereits angesprochen, eignet sich der CLD besonders dafür, anhand von Skizzen ähnliche Bilder aus einer Datenbank zu finden. Außerdem kann der CLD zur Content Filterung durch Bildindizierung und Visualisierung genutzt werden.

Color Structure Descriptor. Der Color Structure Descriptor, auch Color Structure Histogram genannt, repräsentiert ein Bild sowohl durch seine Farbverteilung (Histogramm) als auch durch lokale räumliche Struktur. So können Bilder unterschieden werden, selbst wenn sie das exakt gleiche Farbhistogramm aufweisen. Dies geschieht durch ein Strukturierungselement, das über das Bild läuft. Der CSD ist allgemein definiert als

$$CSD = \overline{h}_s(m), \qquad m \in \{1, ..., M\}$$

wobei M aus $\{184, 120, 32, 16\}$ stammt, was die Menge der repräsentierten Farben (Bins) darstellt, und s die Kantenlänge des Strukturierungselementes.
Zur Extraktion des CS Histogramms muss das Bild im HMMD Farbraum repräsentiert sein, falls es das nicht ist, muss es dorthin konvertiert werden. Ein Strukturierungselement scannt das Bild einmal, wobei es sich immer gänzlich innerhalb der Grenzen des Bildes befinden muss. An jeder Position wird geprüft, welche Farben vom Strukturierungselement eingeschlossen werden und für diese Farben wird der Zähler um eins erhöht (siehe Abbildung 8, [6]).

Da nicht die absolute Häufigkeit der vorkommenden Farben aufsummiert wird, sondern nur, ob sie auftreten oder nicht, werden seltener vorkommende Farben häufiger gezählt, als es ihrem Bildanteil entspricht. Die Größe des Strukturierungselements ist nicht beliebig, sie errechnet sich folgendermaßen:

$$p = max\{0, \lfloor log_2\sqrt{WH} - 7.5 \rfloor\}, \qquad K = 2^p, E = 8K$$

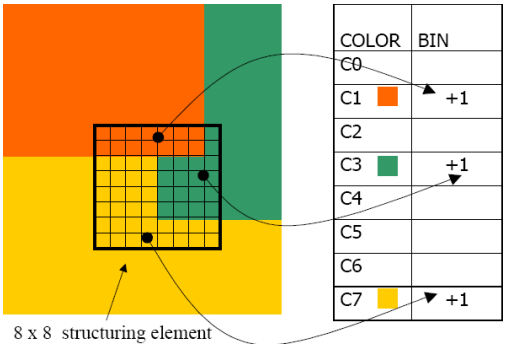

Abbildung 8. Erstellung des CS Histogramms

W entspricht hierbei der Anzahl der Pixel in der Breite und H in der Höhe des Bildes, K ist der Subsamplingfaktor und die Größe des Strukturierungselements entspricht dann $E \times E$. Beispielsweise ist also bei einer Bildgröße von 640×480 Pixel $p = 1$, $K = 2$ und somit $E = 16$, also ist die Größe des Strukturierungselements 16×16 [6]. Alternativ können auch beliebige geformte Regionen des Bildes verwendet werden.
Da das Ergebnis der Extraktion des Farbstrukturhistogramms 256 Bins aufweist, müssen in einem nächsten Schritt die Bins auf die gewünschte Anzahl (184, 120, 32 oder 16) herunterskaliert werden. Anschließend werden die Werte auf $[0, 1[$ normalisiert, um Bilder mit unterschiedlich vielen Bildpunkten vergleichbar zu machen.

Anwendungen und Beispiele. Der CSD wird nur für Einzelbilder verwendet und meistens für Naturaufnahmen. Abbildung 9 [7] zeigt das visualisierte Ergebnis nach Anwendung des CSD, mit dem bereits eine rudimentäre Art der Formerkennung möglich ist:

Abbildung 9. Beispiel für den Color Structure Descriptor

2.3 Weitere Beispiele

Abbildung 10 [8] zeigt Zusammenstellung von unterschiedlichen Flaggen, deren Merkmale durch unterschiedliche Deskriptoren, nicht nur Farbdeskriptoren, beschrieben werden können. Dabei wird deutlich, dass sich kein Deskriptor finden lässt, anhand dessen alle Flaggen zureichend beschrieben werden könnten.
Listing 1 [8] zeigt eine XML-Beschreibung als Ergebnis des CLD.

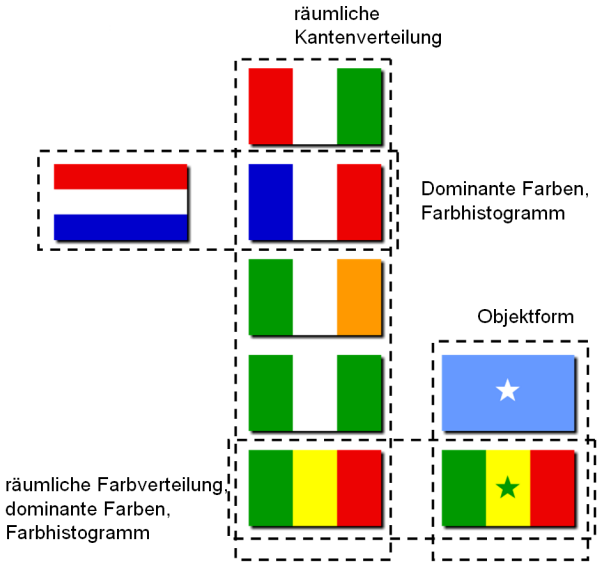

Abbildung 10. Unterschiedliche Deskriptoren im Vergleich

```
<Grid Layout numOfPartX="2" numOfpartY="2" descriptorMask="0110">
        <!--instance at (0 1) -->
        <Descriptor xsi:type="ColorLayoutType">
                <YDCCoeff>50</YDCCoeff>
                <CbDCCoeff>34</CbDCCoeff>
                <CrDCCoeff>30</CrDCCoeff>
                <YACCoeff5>16 12 15 12 17 11</YACCoeff5>
                <CbACCoeff2>12 17</CbACCoeff2>
                <CrACCoeff2>12 14</CrACCoeff2>
        </Descriptor>
        <!--instance at (1 0) -->
        <Descriptor xsi:type="ColorLayoutType">
                <YDCCoeff>48</YDCCoeff>
                <CbDCCoeff>34</CbDCCoeff>
                <CrDCCoeff>32</CrDCCoeff>
                <YACCoeff5>12 10 13 9 10 15</YACCoeff5>
                <CbACCoeff2>14 15</CbACCoeff2>
                <CrACCoeff2>16 12</CrACCoeff2>
        </Descriptor>
</Grid>
```

3 Texturdeskriptoren

3.1 Überblick über die Texturdeskriptoren

Als weiteres wichtiges visuelles Merkmal von Bildern haben sich Texturen erwiesen. Der Begriff Textur (vgl. lat.: *textura* - Gewebe) [9] hat je nach Gebrauchsdomäne durchaus unterschiedliche Bedeutungen. Im Allgemeinen versteht man

darunter strukturelle Beschaffenheit einer zusammenhängenden Oberfläche. Gemeinhin können Texturen visuell durch fünf Charakteristika beschrieben werden: Grobkörnigkeit (*Coarseness*), Kontrast (*Contrast*), Ausrichtung oder Gerichtetheit (*Directionality*), Linienartigkeit (*Line-Likeness*) und Regelmäßigkeit (*Regularity*) [9]. Wichtig zu bemerken ist, dass Texturen *nicht* von der Farbgebung abhängig sind, weswegen auch aus Speicherplatzgründen meistens Graustufenbilder zur Texturanalyse verwendet werden. MPEG-7 stellt drei verschiedene Texturdeskriptoren zur Verfügung: den *Homogeneous Texture Descriptor (HTD)*, den *Texture Browsing Descriptor, (TBD)* und den *Edge Histogram Descriptor (EHD).*

3.2 Deskriptoren im Detail

Homogeneous Texture Descriptor. Der HTD ist gut geeignet zur quantitativen Charakterisierung von Texturen, die homogene Eigenschaften aufweisen. Ein gutes Beispiel für ein solches homogenes Muster wäre ein von oben betrachteter Parkplatz, auf dem in gleichmäßigen Abständen Autos parken. In der Tat wird der HTD in der Praxis hauptsächlich verwendet, um Aufnahmen aus der Luft und Satellitenbilder von Städten und ländlichen Gebieten zu indizieren (siehe Abbildung 12).
Die Analyse der Bilder wird beim HTD im Frequenzraum durchgeführt. Dazu wird das Frequenzspekturm in 30 Kanäle eingeteilt, einheitlich zu Anteilen von je 30°(Abbildung 11 [10], [4]): Der Deskriptor basiert auf Anwendung

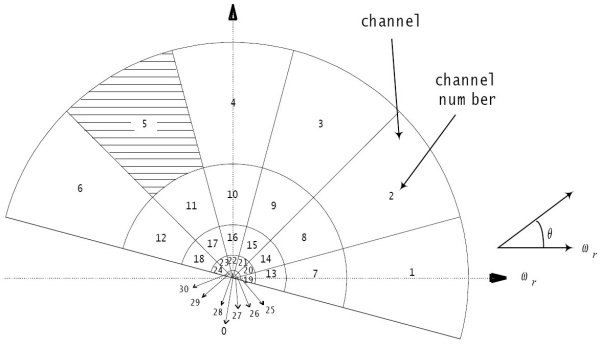

Abbildung 11. Frequenzkanäle im HTD

einer Filterbank, die skalierungs- und orientierungssensitive Filter beinhaltet. Brauchbare Metainformationen über die Texturen lassen sich aus der Berechnung von Mittelintensität und Standardabweichung von Frequenzkoeffizienten aus den oben genannten Frequenzkanälen extrahieren, indem so genannte 2D-Gabor-Funktionen durch eine Fourier-Transformation für die einzelnen Kanäle in Polarkoordinaten dargestellt werden. Die Ähnlichkeit zwischen zwei Bildern lässt sich folgendermaßen berechnen:

$$d(TD_{query}, TD_{database}) = \sum_k \left| \frac{TD_{query}(k) - TD_{Database}(k)}{a(k)} \right|$$

wobei k den Frequenzkanal bezeichnet und $a(k)$ die Standardabweichung $TD_{Database}$ der Datenbank bezeichnet.
Experimentell wurde für den HTD beim Vergleich von Bildern ein Genauigkeitswert von etwa 77% ermittelt [10].

Anwendungen und Beispiele. Auf Abbildung 12 [11] ist eine typische Luftaufnahme zu sehen, welche mit Hilfe des HTD sehr gut indiziert werden kann.

Abbildung 12. Luftaufnahme

Texture Browsing Deskriptor. Der TBD charakterisiert Texturen nach sehr menschlichen Kriterien, nämlich nach Ausrichtung, Gleichmäßigkeit und Grobkörnung eines Musters. Dabei wird folgendes Format vorgegeben:

$$TBD = [v_1 v_2 v_3 v_4 v_5]$$

wobei
$v_1 \in \{1, 2, 3, 4\}$ angibt, in welchem Maße die Textur als gleichmäßig einzuordnen ist, also je größer v_1, desto gleichmäßiger die Textur.
v_2, $v_3 \in \{1, ..., 6\}$ gibt an, in welche Richtung die Textur ausgerichtet ist. Da eine Textur mehr als eine dominante Richtung haben kann, sind hier zwei Werte vorgesehen. Dabei können Winkelwerte von 0° - 150° in 30° Schritten auftreten.
v_4, $v_5 \in \{1, ..., 4\}$ gibt an, wie grobkörning die Textur eingestuft wird - je größer der Wert, desto grobkörniger die Textur. Auch hier sind zwei Werte vorgesehen. Durch diese Einteilung ist eine sehr kompakte Speicherung der Metadaten möglich, für die fünf Koeffizienten sind nur 12 Bits erforderlich. Die Extraktion der einzelnen Komponenten geschieht durch eine Gabor-Filterbank, die eine Reihe von skalierungs- und orientierungssensitiven Filtern enthält und das betrachtete Bild in mehrere gefilterte Bilder aufspaltet. Die Berechnung der einzelnen Merkmale und ihre Ausprägung geschieht sehr ähnlich zum HTD.

Anwendungen und Beispiele. Der Name des Deskriptors sagt bereits viel über die Absicht aus, mit der er entwickelt wurde. Er ist gedacht, um schnell eine

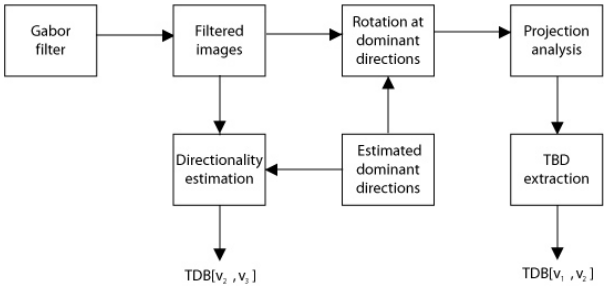

Abbildung 13. Extraktion des TBD

große Menge von Texturen zu durchsuchen und vor allem anhand der Ausrichtung zu filtern. In Abbildung 14 [1] zeigt die erste Reihe Texturen mit zwei dominanten Richtungen, die zweite Reihe mit einer dominanten Richtung.

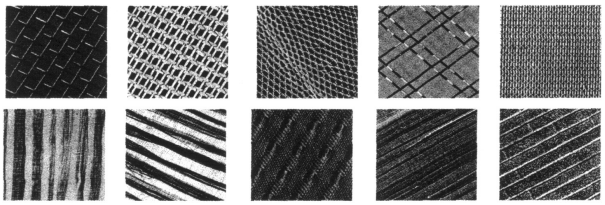

Abbildung 14. Texturen mit zwei und einer dominanten Richtung

Edge Histogram Descriptor. Der EHD, auch Non-Homogeneous Texture Descriptor genannt, beschreibt die räumliche Anordnung von Kanten innerhalb eines Bildes. Dazu wird das Bild in 16 Blöcke (4×4) eingeteilt und ein Histogramm aus den in jedem Teilbild gefundenen Kanten erstellt. Man unterscheidet Kanten anhand ihres Verlaufswinkels: vertikal (90°), horizontal (0°), 45° diagonal, 135° diagonal und non-direktional für Kanten, deren Richtung nicht eindeutig bestimmbar ist [12]. Somit sind $5 \times 16 = 80$ Histogrammbins erforderlich. Um das Kantenhistogramm zu berechnen, wird jedes der 16 Unterbilder

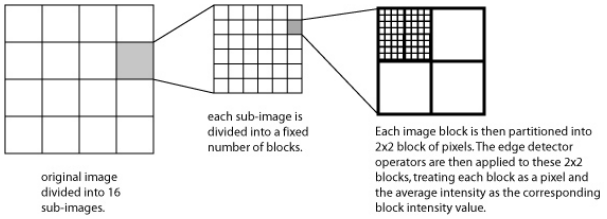

Abbildung 15. Berechnung des EHD

noch einmal in eine feste Anzahl von Blocks unterteilt. Diese werden dann von einer Kantendetektorroutine wie 2×2 Pixel große Bilder behandelt, indem für

jeden Block ein Durchschnittswert der Pixel gebildet wird, aus denen er besteht (vgl. Abbildung 15 [2]). Solche Blocks, die einen gegebenen Minimumintensitätswert überschreiten, die also eine Kante bilden, werden zur Berechnung des Kantenhistogramms verwendet. Zusätzlich zu den 80 Einzelwerten können noch globale und semi-globale Kanteninformationen gespeichert werden, da die lokalen Kanten oftmals nicht ausreichen. Dies geschieht, indem einmal aus dem ganzen Bild, einmal aus den vier horizontalen Unterbildern, aus den vier vertikalen Unterbildern und aus jeweils in den vier Ecken und in der Mitte befindlichen Unterbildern die Kanten berechnet werden. Die Distanz zwischen verschiedenen Bildern lässt sich wieder über die L1-Norm [4] berechnen.

Anwendungen und Beispiele. Da der EHD zur Deskription ausschließlich Kanteninformation verwendet, eignet er sich besonders gut zum Vergleich von Bildern mit klaren Kanten. Dies ist vor allem bei Cliparts, Skizzen und Bildern mit starken Kontrasten der Fall.

4 Beurteilung der MPEG-7 Deskriptoren

Horst Eidenberger von der Universität Wien führte im Jahr 2003 eine Evaluation der visuellen Deskriptoren im Hinblick auf ihr Design, ihre Abhängigkeit vom betrachteten Medientyp und ihre Redundanz durch [13]. Im Folgenden soll kurz Eidenbergers Vorgehensweise und die Ergebnisse erläutert werden.

4.1 Untersuchungsdesign und Zielsetzung

Untersucht wurden alle Farbdeskriptoren, alle Texturdeskriptoren und ein Formdeskriptor, der Region-based Shape Deskriptor. Alle anderen MPEG-7 Deskriptoren wurden nicht untersucht, da aufgrund ihrer Beschaffenheit ein Vergleich zu den obengenannten nicht möglich war. Als Testdatensatz wurden eine Reihe von Texturen, Naturbildern und Wappengrafiken verwendet, insgesamt 798 Bilder, auf die jeweils alle genannten Deskriptoren angewendet wurden.
Ziel der Untersuchung war, grundlegende Richtlinien zum Gebrauch der unterschiedlichen Deskriptoren zu erstellen, wann welcher Deskriptor in Kombination mit welchen anderen sinnvoll eingesetzt werden kann und wie man eventuell den Extraktionsprozess verbessern könnte. Dazu wurden drei Gruppen von Fragestellungen formuliert, die jeweils eine Reihe von Unterfragen zu eben genannten Gesichtspunkten enthalten.

4.2 Ergebnisse

Die Hauptergebnisse der Evaluierung besagen, dass die am besten zur Kombination geeigneten Deskriptoren der Color Layout Descriptor, der Dominant Color Descriptor, der Edge Histogram Descriptor und der Texture Browsing Descriptor sind. Der Color Structure und der Scalable Color Descriptor sind schlecht für Graustufen- oder Binärbilder geeignet, so sollte anstatt des SCDs auch für Gruppen von Frames oder Bildern eher der DCD verwendet werden.
Alle Deskriptoren weisen eine sehr hohe Redundanz gegeneinander auf und

durch entsprechende Kompressionsmaßnahmen können bis zu 80% des Speicherplatzes eingespart werden. Trotzdem wurden einige Bildmerkmale, vor allem bei Graustufenbildern und bei feinen Farbunterunterschieden, von keinem der Deskriptoren hinreichend gut erkannt, so dass man einige Deskriptoren verfeinern bzw. neue Deskriptoren in den Standard einführen müsste.

5 Zusammenfassung und Fazit

Diese Arbeit gibt einen Überblick über die Funktionsweise und Anwendungsmöglichkeiten der Farb- und Texturdeskriptoren, die in MPEG-7 festgeschrieben wurden.
Zusammenfassend ist zu sagen, dass durch die MPEG-7 Farb- und Texturdeskriptoren eine Möglichkeit geschaffen wurde, effizient in großen Bilddatenbeständen zu suchen und diese mit Metadaten zu versehen. In der Realität ist der Einsatz der MPEG-7 Deskriptoren allerdings praktisch nur im universitären Umfeld und in Forschungsprojekten zu finden. Privatunternehmen und private Webseitenbetreiber setzen in den allermeisten Fällen auf proprietäre Lösungen, um ähnliche Dienste anzubieten. Die Gründe dürften hauptsächlich in bereits bekannten Technologien liegen, die eine schnelle und effiziente Bildsuche in einer Datenbank erlauben, wie etwa die Wavelet-Technologie, die im Allgemeinen recht gute Ergebnisse liefert.
Im Hinblick auf stetige Neuerungen und steigende Anforderungen an Metadaten-Anwendungen bleibt in den kommenden Jahren die Entwicklung der Signifikanz der MPEG-7-Technologie abzuwarten.

Literatur

1. Salembier, P., Sikora, T.: Introduction to MPEG-7: Multimedia Content Description Interface. John Wiley & Sons, Inc., New York, NY, USA (2002)
2. Manjunath, B.S., Ohm, J.R., Vasudevan, V.V., Yamada, A.: Color and Texture Descriptors. IEEE Trans. Circuits Syst. Video Techn. **11** (2001) 703–715
3. Wikipedia, the free encyclopedia: HSV Color Space (2006)
4. Sikora, T.: The MPEG-7 Visual Standard for Content Description-An Overview. IEEE Trans. Circuits Syst. Video Techn. **11** (2001) 696–702
5. Ferman, A., Krishnamachari, S., Tekalp, A., AbdelMottaleb, M., Mehrotra, R.: Group-of-frames/pictures color histogram descriptors for multimedia applications. In: International Conference on Image Processing. (2000)
6. Buturovic, A.: MPEG 7 Color Structure Descriptor for visual information retrieval project VizIR (2005)
7. De Natale, F.G.B., Granelli, F.: Structure-based image retrieval using a structured color descriptor (2001)
8. International Organization for Standardization: Text of ISO/IEC 15938-8 PDTR (Extraction and Use of MPEG-7 Descriptions) (2001)
9. Wikipedia, the free encyclopedia: Textur (2006)
10. Ro, Y.M., Kim, M., Kang, H.K., Manjunath, B.S., Kim, J.: MPEG-7 Homogenous Texture Descriptor. ETRI Journal **32** (2001) 41–51
11. Newsam, S., Tesic, J., El-Saban, M., Manjunath, B.S.: MPEG-7 Homogeneous Texture Descriptor Demo (2001)
12. Park, D.K., Jeon, Y.S., Won, C.S.: Efficient use of local edge histogram descriptor. In: MULTIMEDIA '00: Proceedings of the 2000 ACM workshops on Multimedia, New York, NY, USA, ACM Press (2000) 51–54
13. Eidenberger, H.: How good are the visual MPEG-7 features? In: VCIP. (2003) 476–488