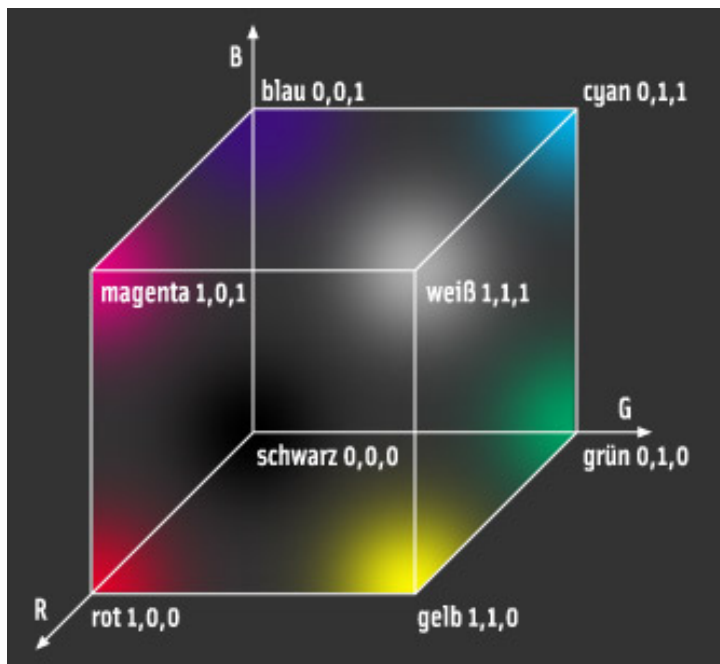


## Projekt kd-Trees

### 1. Erläuterung des Umfeldes

Bei der Bildanalyse sollen Farbpunkte eines Bildes klassifiziert werden. Die Farbe eines Bildpunktes (Farbobjekt) hat im RGB Farbraum drei Farbdimensionen **rot**, **grün** und **blau**,  $rgb \in [0, 1]^3 \subset \mathbb{R}$  bzw.  $rgb \in [0: 255]^3 \subset \mathbb{N}^3$ . Die Farben werden in einem achsenparallelen Farbwürfel angeordnet. Die beiden schwarz-weiß-Ecken verbindet eine Graulinie.



(From: [http://www.beta45.de/farbcodes/theorie/farbmodelle/rgb\\_modell.html](http://www.beta45.de/farbcodes/theorie/farbmodelle/rgb_modell.html))

Anhand eines vorgegebenen Lernbildes können aus den darin enthaltenen Farbpunkten Kategorien gebildet werden, die nicht notwendig disjunkt sein müssen. Angenommen, ein Testbild soll analysiert werden. Folgende Anfragen sind von Interesse:

1. Zu welchen Kategorien gehört Farbobjekt aus dem Testbild.
2. Welches ist der nächste Nachbar eines Farbobjekts

Der Abstand zweier Farbobjekte wird durch den euklidischen Abstand ermittelt. Zwei Punkte  $x, y$  haben im RGB-Farbraum den euklidischen Abstand

$$d(x, y) = \sqrt{\sum_{i=r,g,b} (x_i - y_i)^2}$$

Der euklidische Abstand zwei Punkte ändert sich mit allen drei Farbdimensionen rot, grün und blau.

## 2. Darstellung von RGB-Farbräume durch kd-Bäume

Die Farbbobjekte eines Bildes werden mit dem RGB-Farbwürfel modelliert und in binären kd-Baum mit den drei Dimensionen **rot**, **grün** und **blau** gespeichert. Ein Bildpunkt (x,y) in einem 2-dimensionalen Bild ist im RGB-Farbmodell ein Farbbobjekt mit den Koordinaten (r,g,b):

$$[x, y, r, g, b]$$

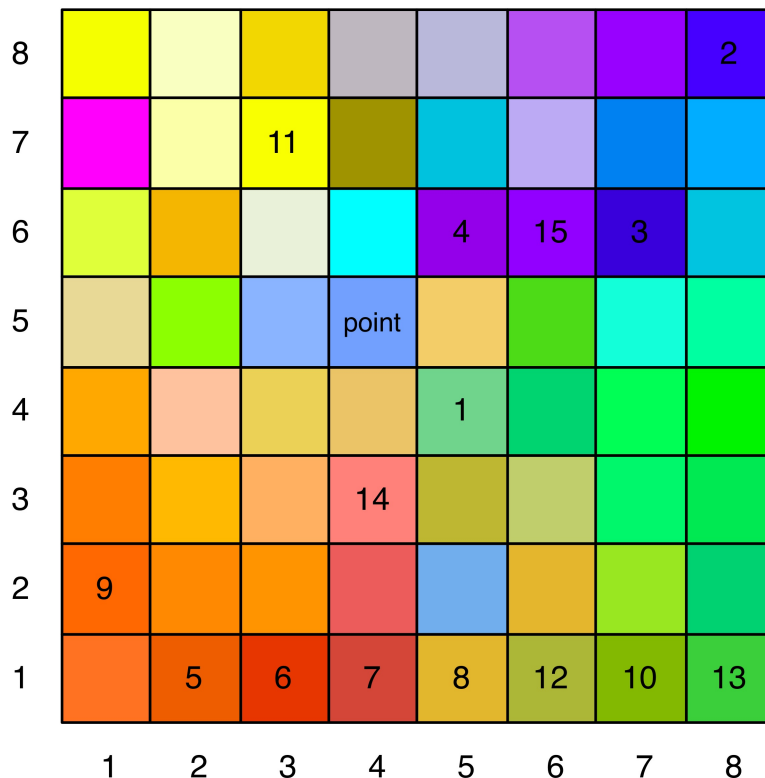
### Aufgaben

1. Überlegen Sie sich **sinnvolle Farbkategorien** und erstellen Sie dazu ein Lernbild, z.B. könnten die 8 Eckpunkte des RGB-Würfels als Farbkategorien verwendet werden. Sie müssen keine realen Bilder verwenden, sondern können 2-dimensionale Arrays mit rgb-Werten verwenden.
2. **Speichern Sie die Farbbobjekte** des Bildes in einem kd-Baum. Als Schlüssel sollen die rgb-Werte und als beschreibende Daten die x,y-Koordinaten verwendet werden.
3. **Suche nach dem nächsten Nachbarn** mit dem Euklidischen Abstand als Ähnlichkeitsmaß. Diese Aufgabe beschleunigt der kd-Baum. Wenn der Abstand vom aktuellen Knoten auf der projizierten Dimension (r, g oder b) schon weiter ist als der bisher als bestmögliche gefundene Gesamtabstand, dann muss die Suche beim zweiten Kind nicht mehr fortgesetzt werden. Denn auf den anderen Dimensionen kommt man nicht näher an das Farbbobjekt heran.
4. **Suche und Auflisten der Farbkategorien**, zu denen ein aktuelles Farbbobjekt gehört.
5. Entwickeln Sie eine **Testumgebung**, mit der Anfragen wie „Nächster Nachbar“ oder „zugehörige Kategorien“ durchgeführt und verständlich dargestellt werden könnten.
6. Überlegen Sie sich **weitere Anfragen**, die bei Bildanalysen interessant sind und realisieren Sie diese.

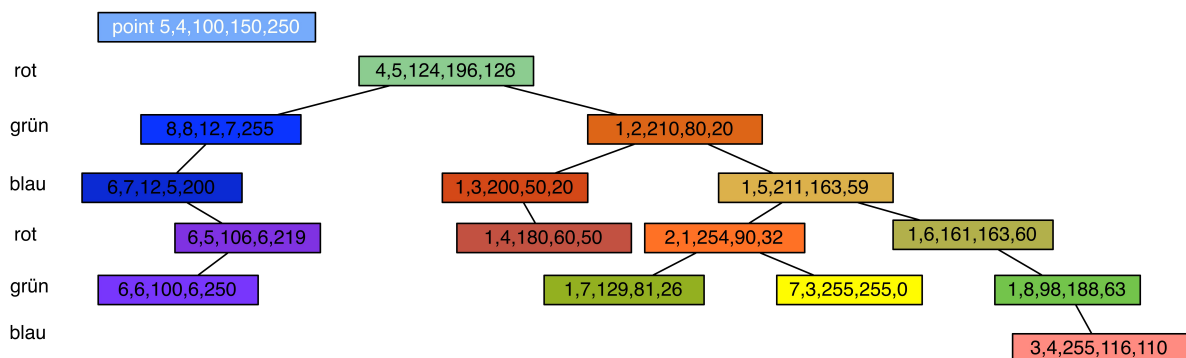
Die Wahl der Programmiersprache ist freigestellt. Auch freigestellt ist, ob Sie reale Bilder verwenden oder Bilder einfach durch 2-dim. Arrays simulieren.

### 3. Beispiel

Gegeben ist folgendes Bild mit  $8 \times 8 = 64$  Farbobjekten (Pixeln).



Die Punkte mit der Nummerierung werden in einem kd-Baum als Referenz-Objekte gespeichert. Wird als Einfügereihenfolge die angegebene Nummerierung verwendet, so entsteht folgender kd-Baum (bitte auf Richtigkeit prüfen):



**Aufgabe:** Prüfe, welcher Knoten im kd-Baum der **nächste Nachbar** zu dem Farbobjekt (point 5,4,100,150,250) ist.

**Lösung:** Traversiere den kd-Baum unter Beachtung der Dimensionen rot, grün, blau. Berechne für jeden besuchten Knoten den Euklidischen Abstand zu (point 5,4,100,150,250) und merke den bisher kleinsten gefundenen Abstand. Die Excel-Tabelle zeigt diese Berechnung:

x,y	rot	grün	blau	d(point,(x,y)	d_red	d_green	d_blue	Kriterium
point	100	150	250					
4,5	124	196	126	134,42	24	46	124	red
8,8	12	7	255	167,98	112	143	5	green
6,7	12	5	200	176,83	88	191	74	blue
6,5	106	6	219	147,42	6	144	31	red
1,2	210	80	20	264,39	110	70	230	green
1,3	200	50	20	270,00	100	100	230	blue
1,4	180	60	50	233,45	80	90	200	red
1,5	211	163	59	221,29	111	13	191	blue
2,1	254	90	32	273,57	154	60	218	red
1,7	129	81	26	236,17	29	69	224	green
7,3	255	255	0	312,33	155	105	250	green
1,6	161	163	60	199,97	61	13	190	red
1,8	98	188	63	190,83	2	38	187	green
3,4	255	116	110	211,62	155	34	140	blue
6,6	100	6	250	144,00	0	144	0	blue

Das Pixel (4,5) ist der nächste Nachbar zu (point, 100, 150, 200).

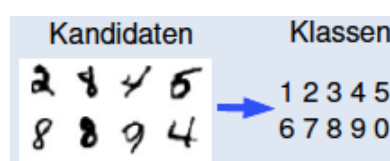
Beobachtungen:

1. Die Grün-Distanz von (8,8) mit 145 ist bereits größer als die bisher gefundene kleinste Distanz von (4,5) mit 134,42. Alle Knoten im kd-Baum, die links unter (8,8) liegen, würden die Gründistanz und damit den euklidischen Abstand nicht kleiner machen als die Gründistanz des Punktes (8,8). Diese Knoten müssen also nicht mehr geprüft werden.
2. Die Rotdistanz des Knotens (2,1) mit 154 ist ebenfalls schon größer als der bisher gefundene kleinste Abstand von (4,5) mit 134,42. Jede weitere Verzweigung nach rechts würde den euklidischen Abstand niemals unter den Wert von 154 sinken lassen. Daher müssen Knoten, die rechts unter (2,1) liegen, nicht mehr betrachtet werden.

## 4. Anwendungsbeispiele

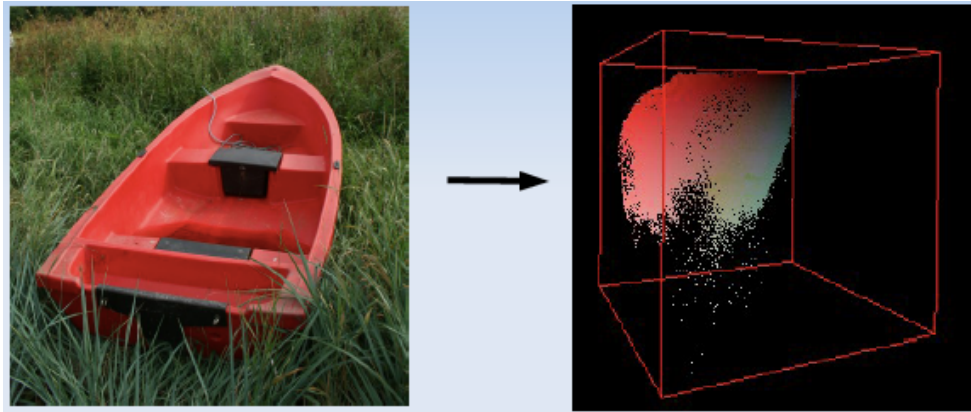
kd-Bäume werden heute in vielen Softwaresystemen verwendet, bei denen die Partitionierung von Objekten nach mehreren Dimensionen erfolgen soll. Beispiele finden sich in Anwendungen für Automatisierung (Industrieroboter), Bildersuche, Verkehrsnavigationssysteme, usw. Ein wichtiger Vorverarbeitungsschritt ist oft die (Farb-) Segmentierung von Objekten, die dadurch klassifiziert werden: Gleichfarbige Flächen zusammenfassen, um Objekte im Bild einfacher zu erkennen und ihre Position zu bestimmen. Durch Klassifizierung werden die Objekte (Kandidaten) nach vorgegebenen oder gelernten Kriterien bestimmten Klassen zugeordnet. Beispiele finden sich in der

- Medizin: Bilder von Zellen (gesund, gefährdet, erkrankt)



- Handschrifterkennung:
- Fernerkundungen anhand von Bildern -> Wälder, Gewässer, Nutzfläche

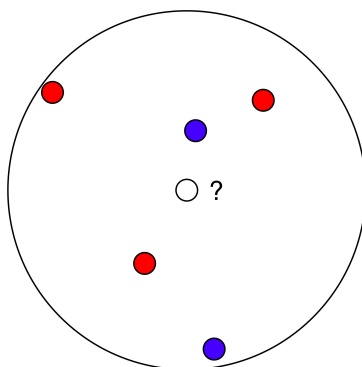
Die Objekte werden durch mehrere quantitative Merkmale in einem Merkmalsraum beschrieben, z.B. Pixel durch RGB-Farbwerte im RGB Farbraum:



Bei der Klassifizierung unterscheidet man zwischen

- **überwachten Verfahren**, bei denen die Klassen anhand von Stichproben vorgegeben werden und
- **unüberwachten Verfahren**, wo sich die Klassen durch die Analyse des Merkmalsraums ergeben.

Bei der **k-nächste-Nachbar Klassifizierung** handelt es sich um ein überwachtes Verfahren, bei dem anhand von Beispiel-Pixeln die typischen Merkmale der Klassen vorgegeben werden. Durch Trainingsbeispiele kann die Klassenzuordnung z.B. unter Berücksichtigung der k-nächsten Nachbarn gelernt werden:



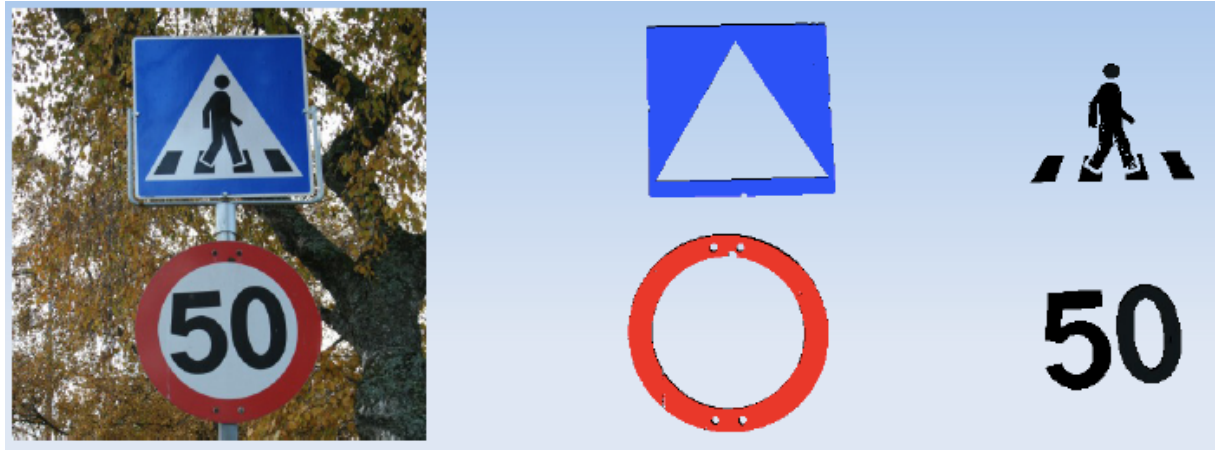
Im Referenzbereich des Kreises um das zu kategorisierende Objekt befinden sich 3 rote und 2 blaue Objekte,  $k=5$ . Damit wird das neue Objekt der Kategorie „rote Kreise“ zugeordnet.

Die Trainingsdaten werden in einem kd-Baum gespeichert. Neue Objekte werden entsprechend einem Distanzkriterium, z.B. euklidischer Abstand im RGB-Farbraum, den Kategorien zugeordnet.

Eine Strategie zum Auffinden des nächsten Nachbarn ist folgende:

1. Blattposition des neuen Kandidaten (hier Farbobjekt) finden
2. Euklidischen Abstand zum Vaterknoten bestimmen
3. Rekursiv Suchregionen ausschließen (vgl. Beispiel unter Punkt 3) und ggf. näheren oder mehrere nähere Nachbarn finden

Das folgende Bildbeispiel zeigt die Extraktion von Merkmalen aus Bildern. Vorgegeben sind hier als Kategorien die drei Farbobjekte „rot“, „weiß“ und „schwarz“. Alle anderen Bildpixel diesen Kategorien zugeordnet.



In der Literatur werden kd-Bäume auch oft im Zusammenhang mit Instanzbasiertem Lernen genannt. Dieses Thema soll hier aber nicht weiter vertieft werden, weil es in diesem Algorithmik-Praktikumsprojekt in erster Linie um die Implementierung und das Verständnis von kd-Bäumen und nicht primär um ihre Anwendung geht.