Lectures on Segmentation

by

Lutz Priese

Institute of Computational Visualistics
University Koblenz-Landau
Koblenz, Germany

Contents

1. Notations

2. Filters

3. Recursive Histogram Splitting (RHS)

4. Split and Merge (SaM)

5. Region Growing (RG)

6. Color Structure Code (CSC)

A. Literatur

1 Notations

location (of size $N \times M$)

$\text{Loc} = [0, N[\times[0, M[\quad (\cap \mathbb{N}^2)$

$p \in \text{Loc} \Rightarrow \exists p_1 \in [0, N[\, \exists p_2 \in [0, M[: p = (p_1, p_2)$

values

usual values:
$\text{Val} = [0, 255]$ (greyvalues)
$\text{Val} = [0, 255]^3$ (true colors in RGB)
etc.

image

$X : \text{Loc} \rightarrow \text{Val}$

pixel

$\text{P} = (p, X(p)) \in \text{Loc} \times \text{Val}$
$p = \text{loc}(\text{P})$, location of P
$X(p) = \text{val}(P)$, value of P
sometimes only $p = \text{loc}(\text{P})$ is called a pixel

channel

a projection of Val onto one dimension, if Val is multidimensional, i.e.:

- $\text{Val} = R \times G \times B = [0, 255]^3$
  R is a channel, red-channel
- $\text{Val} = H \times S \times I$
  H is a channel, hue-channel

For $g \in \text{Val}$ let g^c denote the value of g in channel c

Example: $X : \text{Loc} \rightarrow R \times G \times B$ a color image in the (R,G,B) color space, then $X(p)^R$ denotes the value of X at location p in the R-channel. Val^c is the projection of Val onto its channel c.

■

neighbourhood a reflexive and symmetric relation
$\lambda \subseteq \text{Loc} \times \text{Loc}$
$\lambda(p) := \{q \in \text{Loc}; p\lambda q\}$, the neighbourhood of p

similarity a reflexive and symmetric relation $\sigma \subseteq \text{Val} \times \text{Val}$

a set M *of values* is called (σ-)similar iff
$\forall a, b \in M : a\sigma b$.

a set M *of pixels* is called (σ-)similar iff
$\text{val}(M)$ is (σ-)similar.

direct neighbours p and $q \in \text{Loc}$ are called direct neighbours iff
$|p_1 - q_1| \leq 1$ and $|p_2 - q_2| \leq 1$
(sometimes: $|p_1 - q_1| + |p_2 - q_2| \leq 1$)

connected a set M *of locations* is called connected iff
$\forall p, q \in M : \exists n \in \mathbb{N} : \exists p_0, \ldots, p_n \in M :$
$p = p_0 \wedge q = p_n \wedge p_{i+1}$ is a neighbour of
$p_i \, \forall i : 0 \leq i < n$.

a set M *of pixels* is called connected iff
$\text{loc}(M)$ is connected.

region a region is a connected subset of an image

Example: an image is a region itself. ■

segment a segment is a (σ-)similar region

segmentation
i) a *method* to subdivide an image into possibly large pairwise disjoint segments, or
ii) the *result* of such a method.

class	i)	a subset $C \subseteq \mathrm{Val}$ of *values*
	ii)	a subset $M_C \subseteq X$ of *pixels* defined by $M_C := \{P \subseteq X; \mathrm{val}(P) \in C\}$
		All maximal connected regions in M_C are the *regions in class C*.
classification		Given pairwise disjoint classes $C_1, \ldots, C_n \subseteq \mathrm{Val}$, a classification (according to $C_1, \ldots, C_n$) is the set of all regions in class C_i, for $1 \leq i \leq n$.

Examples of similarity:

by threshold: $p\sigma p'$ iff $||\mathrm{Val}(p) - \mathrm{Val}(p')|| \leq s$

by classes: $p\sigma p'$ iff $\exists$ class C s.t. $p \in C \wedge p' \in C$

by homogenity: $p\sigma p'$ iff p and p' belong to a region that is homogenous enough

There are many different concepts for homogenity, i.e.

- a small variance σ_R^2 of a region R
  (σ_R^2 is a statistical feature of dimension 1)
- same texture within a region
  (texture is usually measured by several statistical features of dimension 2)
- by similarity: a region R is homogenous iff all points p' and p'' in R are pairwise similar
  (Thus, similarity and homogenity are equivalent concepts.)

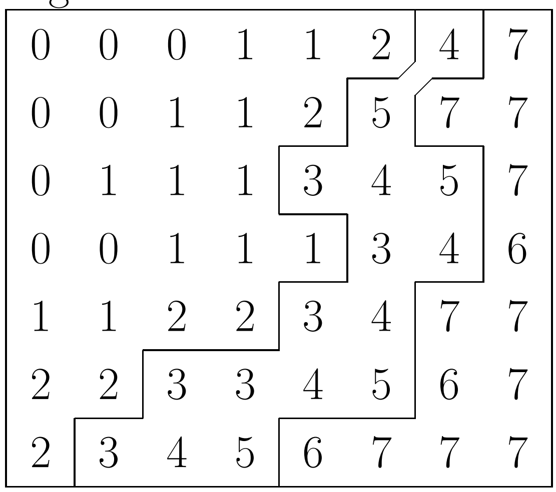
Attention: Even for a fixed similarity σ a segmentation of an image into σ-segments may lead to very different results.

Example: $X : [0,8[\times[0,7[\longrightarrow [0,7]$
$a\sigma b :\Leftrightarrow |\text{val}(a) - \text{val}(b)| \leq 2$

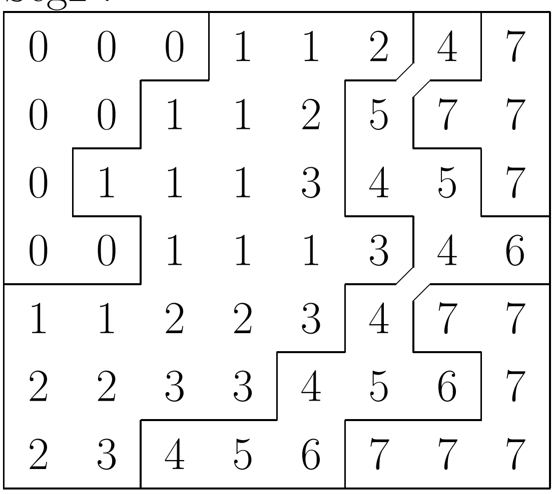
Im1 : X :

0	0	0	1	1	2	4	7
0	0	1	1	2	5	7	7
0	1	1	1	3	4	5	7
0	0	1	1	1	3	4	6
1	1	2	2	3	4	7	7
2	2	3	3	4	5	6	7
2	3	4	5	6	7	7	7

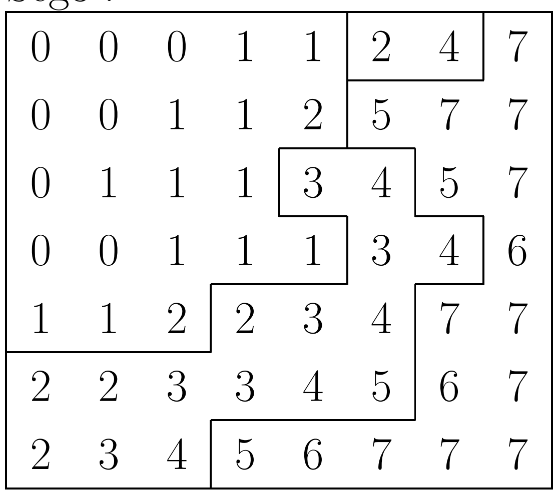
Seg1 :



Seg2 :



Seg3 :



Seg1 is also a classification according to the classes $\{0, 1, 2\}$, $\{3, 4, 5\}$, $\{7\}$.

Seg2 is also a classification according to the classes $\{0\}$, $\{1, 2, 3\}$, $\{4, 5, 6\}$, $\{7\}$.

Seg3 is <u>not</u> a classification: 2 sometimes belongs to a segment with values $\{0, 1, 2\}$, sometimes to a segment with valus $\{2, 3, 4\}$, and sometimes to a segment with valus $\{2, 4\}$.

■

Comparison: Classification $\leftrightarrow$ Segmentation

Let disjoint classes $C_1, \ldots, C_n \subseteq \mathrm{Val}$ and a similarity $\sigma \subseteq \mathrm{Val}^2$ be fixed. We regard a classification $\mathcal{C}$ according to $C_1, \ldots, C_n$ and a segmentation $\mathcal{S}$ according to σ.

- Both, $\mathcal{C}$ and $\mathcal{S}$, are sets of regions.
- $\mathcal{C}$ is unique and easily computed.
- $\mathcal{S}$ is not unique, finding a 'good' $\mathcal{S}$ requires envolved techniques.
- If the light conditions are changing, as happens in practise very often, $\mathcal{C}$ may become instable, $\mathcal{S}$ may stay very stable.

Example: Robocup, detect a red football
Classification: Define a class $C \subseteq \mathrm{Val}$ for 'red'. If a lamp breaks, sun starts shining, a shadow appears, etc, the values of the football may leave C.
Segmentation: Pixels similar in color stay similar if the absolut values change due the above effects. ■

Example: Traffic Signs Detection and Identification
In the HSI color space the hue-values are very stable even when the light conditions change dramatically. Segmentations lead to much better results than classification.
Some examples are found in
http://www.uni-koblenz.de/~lb/lb_research/research.tsr.html ■

- If the light conditions remain the same, classification is preferable, as it is fast and simple.

edge-point — a pixel with 'extrem' non-similar pixels in its neighbourhood
(this is no defnition, only a concept!)

edge — a set of connected edge-points that form a line (i.e. width of 1 or 2 pixels)

Various, very elaborated techniques exist to compute edges.

There are two different methodologies to get a segmentation:

I) Find good edges. Use the edges to define the borders of different regions. Conclude from the borders to the regions itself.

II) Look to the regions in an image. Try to find maximal similar region.

Example: Edges in Im1

0	0	0	1	1	2	4	7
0	0	1	1	2	5	7	7
0	1	1	1	3	4	5	7
0	0	1	1	1	3	4	6
1	1	2	2	3	4	7	7
2	2	3	3	4	5	6	7
2	3	4	5	6	7	7	7

■

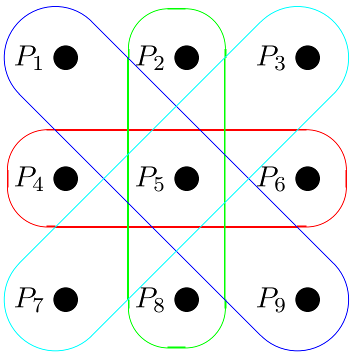
My <u>personal</u> belief: Methodology II leads in most realistic examples to better results.

2 Filters

A preprocessing step befor any segmentation is to filter an image. Here, filters are adeguate that simultaneously sharpen the contrast at the borders of segments and smoothen inside segments. One may use non-linear filters for this purpose. We introduce two such sharpening-smoothening filters here, that usually are applied several times before a segmentation step.

2.1 The Symmetric-nearest-Neighbourhood-Filter (SNN)

The Symmetric-nearest-Neighbourhood-Filter is Harwood [HSHD87].



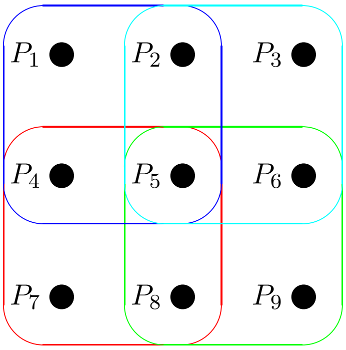
Pixel to be replaced: P_5

Regard 4 opposite sets of pixels:
$S_1 = \{P_1, P_9\}$
$S_2 = \{P_2, P_8\}$
$S_3 = \{P_3, P_7\}$
$S_4 = \{P_4, P_6\}$

Let $cl(S_i)$ denote the pixel in S_i whose value is closest to $\text{val}(P_5)$, i.e. $P = cl(S_i)$ iff $|\text{val}(P) - \text{val}(P_5)|$ is minimal for all $P \in S_i$.
Replace P_5 by $\mu(\{cl(S_1), cl(S_2), cl(S_3), cl(S_4)\})$, the mean of the four closert pixels of each set.

2.2 The Kuwahara-Nagao-Filter (KuNa)

Kuwahara, Hachimura, Eiko, Kimoshita [KHEK76]
Nagao, Matsuyama [NaMa79]



Pixel to be replaced: P_5

Regard 4 overlapping regions:
$R_1 = \{P_1, P_2, P_4, P_5\}$
$R_2 = \{P_2, P_3, P_5, P_6\}$
$R_3 = \{P_4, P_5, P_7, P_8\}$
$R_4 = \{P_5, P_6, P_8, P_9\}$

Compute $\sigma^2_{R_i} \quad , \quad 1 \leq i \leq 4$

Find i_0 s.t. $\sigma^2_{R_{i_0}}$ is minimal

Replace val(P_5) by $\mu_{R_{i_0}}$.

Modifications:

- Replace val(P_5) by that value in val(R_{i_0}) which is closest to $\mu_{R_{i_0}}$.
- Replace val(P_5) as above only if the distance of val(P) to $\mu_{R_{i_0}}$ for $P \in R_{i_0}$ is maximal for P_5.

Obviously:
Both filters have the following properties:

- non-linear
- repair single pixel faults in homogenous regions
- sharpen edges

Implementations are found under:

http://www.uni-koblenz.de/~lb/lb_downloads/

3 Recursive Histogram Splitting (RHS)

Ohlander [Oh75]; Ohta, Kanade, Sakai [OKS80]

Let $X : Loc \to \mathrm{Val}$ be a color image, say $\mathrm{Val} = R \times G \times B$.

We may compute further color representations for this image X,

i.e. $\mathrm{Val}_1 = R \times G \times B$, $\mathrm{Val}_2 = H \times S \times I$, $\mathrm{Val}_3 = I \times Y \times Q$, etc.

and regard X as a multidimensional image
$X : Loc \to \mathrm{Val} = \mathrm{Val}_1 \times \mathrm{Val}_2 \times \mathrm{Val}_3$
with many channels:
$R, G, B, H, S, I, \ldots$, or similar.

For each channel c of X we denote by H^c_X the histogram of X in channel c.

Similar, for any region R of X we denote H^c_R the histogram of region R in channel c, a statistics of dimension 1, telling how many pixels in R have a certain value in channel c:

$H^c_R : \mathrm{Val}^c \to \mathbb{N}$ with $H^c_R(g) = m$ iff exactly m pixels in R have the value g in channel c.

Schema of the algorithm RHS:

Input: image $X : Loc \rightarrow \mathrm{Val}$ with a highdimensional space Val with many channels $c \in \mathrm{Cha}$.

create an (empty) stack S for storing regions
put X on S
repeat
  get R from S
  for all channels c in Cha
    compute H_R^c
  find a channel c_0 and a value g_0 in $\mathtt{Val}^{c_0}$
    s.t. g_0 is a maximal peak in $\{H_R^c | c \in \mathtt{Cha}\}$,
    (i.e., $H_R^{c_0}(g_0)$ is maximal)
  if g_0 is too small
    then define R as a final region
  if g_0 is big enough
    then define a 'natural' intervall C_0 in $\mathtt{Val}^{c_0}$
      around g_0
    compute the classification $\mathfrak{C}_R^{c_0}$ of R
      according to the two classes C_0 and
      $(\mathtt{Val}^{c_0} - C_0)$ of R in channel c_0
    for all regions R' in $\mathfrak{C}_R^{c_0}$
      if R' is large enough
        then put R' on S
        otherwise forget R'
until S is empty

An implementation of a RHS algorithm can be found under:

http://bit.uni-koblenz.de/vibi/segmentation.html

You may use the algorithm on the www to test your own images, or get the source code for free if you fulfill all requirements stated there under:

http://www.uni-koblenz.de/~lb/lb_downloads/

This algorithm uses nine channels, namely $R, G, B, H, S, T, Y, I, Q$ (I has two different meanings in the (H, S, I)- and (Y, I, Q)- color-model). The peaks g_0 and their natural intervalls C_0 are computed with the techniques of graphtheoretical clustering, following [Ma96] and [KNF76].

The idea of graphtheoretical clustering:

choose `some step-width` s
for all `channels` c
for all `values` g `in` $\mathtt{Val}^c$
  find g_m `with` $|g - g_m| \leq s$ `and` $\mathtt{val}^c(g_m)$ `maximal`
  set $\mathit{connnected}(g, g_m)$`:=` **true**
for all g `in` $\mathtt{Val}_c$

$$A_g := \sum_{\substack{g' \in \mathtt{Val}_c \\ \mathit{connnected}(g', g)}} H_R^c(g)$$

  `is the area in the histogram` H_R^c `that 'belongs to` g`'`
g_0^c `is maximal in channel` c `iff` $A_{g_0^c}^c$ `is maximal`
g_0 `is a maximal peak iff` g_0 `is maximal in` $\{g_0^c | c \in \mathtt{Cha}\}$
$C_0 := \{g' | \mathit{connnected}(g', g_0)\}$ `is the natural intervall of` g_0`.`

4 Split and Merge (SaM)

Horowitz, Pavlidis [HoPa76]; Dubuisson, Jain [DuJa93]

Schema of the algorithm SaM:

Input: image $X : Loc \rightarrow \text{Val}$ with $\text{Loc} = [0, 2^n[^2$ for some n

Split-Phase:

create `an (empty) tree` T `with regions in its nodes`
create `a node for` T `(which is root and the only leave)`
put X `in this node`
for all `leaves` L `of` T
  take `region` R_L `from` L
  if R_L `is sufficiently homogenous`
    then `return` R_L `in` L
  if R_L `is not sufficiently homogenous`
    then `separate` R_L `into four sub-regions` $R_L^1, R_L^2, R_L^3, R_L^4$
      `of equal size`
    create `four new leaves` L_1, L_2, L_3, L_4 `as sons of` L
    put R_L^i `in` L_1, $1 \leq i \leq 4$

Merge-Phase:

create `an (empty) set` S `of (empty) trees`
for all `leaves` L `of` T
  add L `as a new tree to` S `(with` L `as root and`
    `only leave)`
for any `root` L `in` S
  for any `root` $L' \neq L$ `in` S
    if `the segments` R_L `in` L `and` $R_{L'}$ `in` L' `are`
      `neighboured and similar`
      then create `a new root` $\hat{L}$ `in` S `with sons` L `and` L'
      put `the segment` $R_L \cup R_{L'}$ `into` $\hat{L}$.

Example: $X : [0,8[\times[0,8[\to [0,7]$

$a\sigma b :\Leftrightarrow |\mathrm{Val}(a) - \mathrm{Val}(b)| \leq 1$

X :

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Split 1:

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Split 2:

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Split 3:

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Merge:

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

■

Under

http://bit.uni-koblenz.de/vibi/segmentation.html

an implementation of SaM is found.

Here homogenity in the Split-Phase is measured by

$$\sigma_R^2 + \sigma_G^2 + \sigma_B^2 \qquad (\leq \texttt{Thresh}_S)$$

where σ_c^2, $c \in \{R, G, B\}$ is

$$\sigma_c^2 = \frac{1}{\text{anz}} \sum_{x \in [N_1, N_2[} \sum_{y \in [M_1, M_2[} (\text{Val}^c(x, y) - \mu)^2, \quad c \in \{R, G, B\}$$

with

$$\text{anz} = (N_2 - N_1) \cdot (M_2 - M_1)$$

and

$$\mu = \frac{1}{\text{anz}} \sum_{x \in [N_1, N_2[} \sum_{y \in [M_1, M_2[} \text{Val}^c(x, y), \quad c \in \{R, G, B\}$$

Similarity of two sub-images I_1 and I_2 in the Merge-Phase is measured by

$$|\mu_1^R - \mu_2^R| + |\mu_1^G - \mu_2^G| + |\mu_1^B - \mu_2^B| \qquad (\leq \texttt{Thresh}_M)$$

Excursion:

$$\begin{aligned}
\sigma_c^2 &= \frac{1}{\text{anz}} \sum \sum \big(\text{Val}(x,y) - \mu\big)^2 \\
&= \frac{1}{\text{anz}} \sum \sum \big(\text{Val}^2(x,y) - 2 \cdot \text{Val}(x,y) \cdot \mu + \mu^2\big) \\
&= \frac{1}{\text{anz}} \sum \sum \big(\text{Val}^2(x,y)\big) - 2 \cdot \mu \frac{1}{\text{anz}} \sum \sum \big(\text{Val}(x,y)\big) + \mu^2 \\
&= \frac{1}{\text{anz}} \sum \sum \big(\text{Val}^2(x,y)\big) - \mu^2
\end{aligned}$$

So it is possible to calculate σ and μ in <u>one</u> step, and not first calculate μ and in a second step calculate σ. ■

5 Region Growing (RG)

Schemata:

I) Input: image X : Loc $\rightarrow$ Val

```
create an (empty) set S of segments
repeat
  remove a pixel P from X
  create a new segment R_P consisting of P
  repeat
    remove some pixel P' from X s.t.
      P' is neighboured to some P'' in R_P and
      P' and P'' are similar                         (*)
    add P' to R_P
  until no such P' can be found
  add R_P to S
until X is empty
```

Disadvantage: Chaining Mismatch. One way run by chain of similar pixels from one pixel P_0 to a completely non-similar pixel P_n.

II) replace line `(*)` by

$R_P \cup \{P'\}$ `is homogeneous enough`

Disadvantage: Result depends on the order one chooses pixels from X.

III) Input: image $X : \text{Loc} \to \text{Val}$

```
create an (empty) set S of segments
stage 0:  i:=0
for all pixel P in X
   create a new segment R_P of level 0
      (consisting only of P)
   put R_P in S
repeat
   stage i:
   for all segments R_i of level i in S
      repeat
         find a segment R̂^j of level j ≤ i in S s.t.
            R_i and R̂^j are neighboured and
            R_i ∪ R̂^j is homogenous enough
         remove R_i and R̂^j from S
         redefine R_i := R_i ∪ R̂^j of level i+1
      until no such R̂^j can be found
      add R_i to S
   i:=i+1
until stage i-1 has created no new segment
```

6 Color Structure Code (CSC)

The CSC may be descriped as a Merge-and-Split technique.

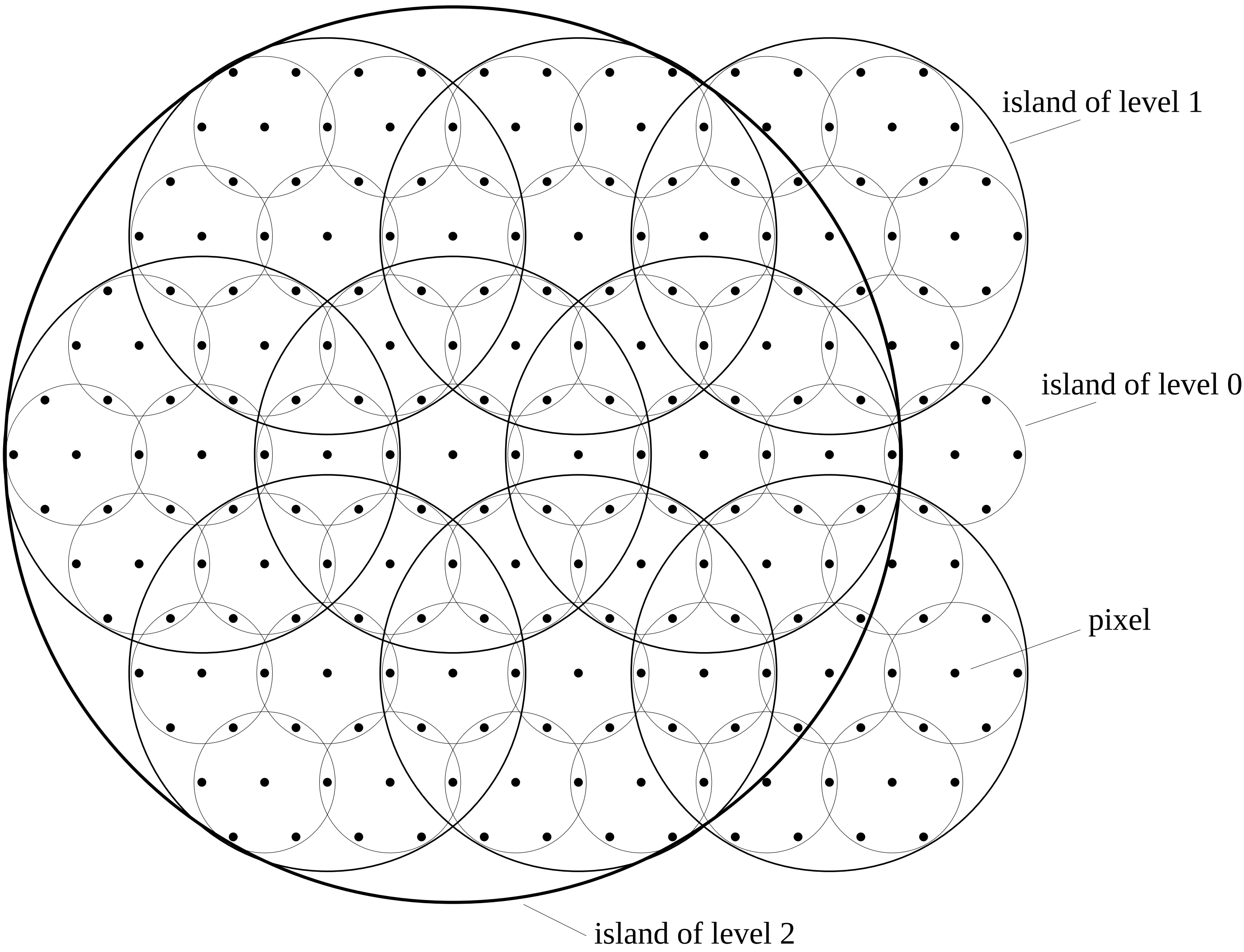
Segments are created following the schema III of region growing methods.

During the creation (Merge-Phase) different segments may overlap in common sub-segment. If it turns out that overlapping segments R_1 and R_2 must not be merged their common sub-segment R' is splitted between both into R'_1 and R'_2 s.t.
$R' = R'_1 \cup R'_2$ and R'_i fitts best to R_i, $1 \leq i \leq 2$,
using knowledge of $R_1 \cup R_2$.

The CSC is from Rehrmann [Re94], using an inherently parallel data-structure from Priese, see e.g. [PrRe92], based on a hierarchical hexagonal pixelstructure from Hartmann [Ha87].

A hierarchical hexagonal pixel structure

For the logic of the following algoritm of CSC we image an image being represented by a hexagonal topology of the pixels in Loc, see the following figure:



Here, the location of the pixels is shown by a dot. Seven pixels, arranged by one pixel in the middle and its six directly neighboured pixels (in this hexagonal topology), form a so-called *island of level 0*.

Seven islands of level n, arranged by one island in the middle and its six directly neighboured islands of level n, form an island of level $n + 1$.

In such a hexagonal topology of pixels we canonically also may identify *lines* of pixels. An image of size $N \times M$ thus consists of N lines of M pixels each. However, each line $i+1$ is slightly shifted against line i. If the distance of pixels on one line is d, the distance of lines becomes $\sqrt{2} \cdot d$. Now the distance of each pixels to its 6 direct neighbours (two on adjacent lines, two on the same line) is always d.

Every second pixel of every second line forms the centre of an island of level 0. Thus, an image of size $N \times M$ possesses $\lfloor \frac{N}{2} \rfloor \times \lfloor \frac{M}{2} \rfloor$ islands of level 0 and $\lfloor \frac{N}{2^n} \rfloor \times \lfloor \frac{M}{2^n} \rfloor$ islands of level n.

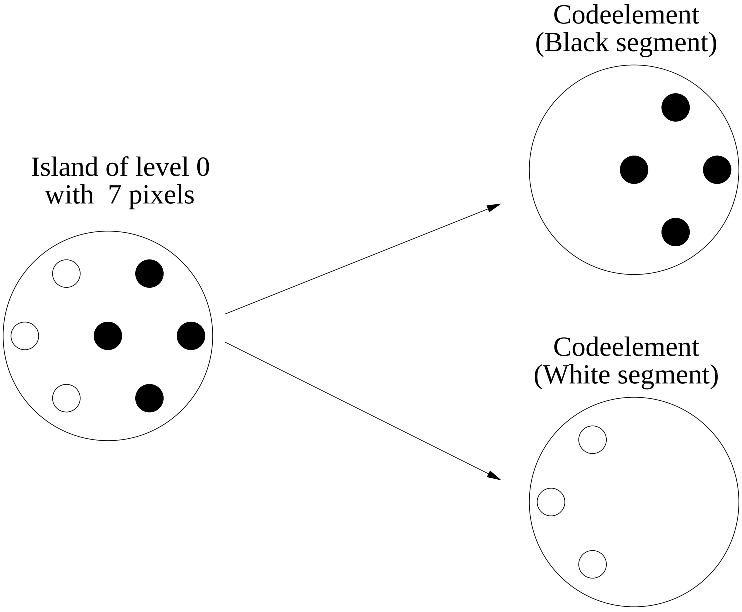
Thus, all six border pixels (i.e., non-centre pixels) of an island of level 0 belong also to an adjacent island of level 0. Any two adjacent islands of level 0 overlap with a common pixel.

The same holds hierarchically for all levels:
All six border (i.e., non-centre) sub-islands level $n-1$ of one island I of level n are also a border sub-island of some adjacent island I' (of level n) to I. Any two adjacent islands of level $n \geq 1$ overlap in a common sub-island of level $n-1$.

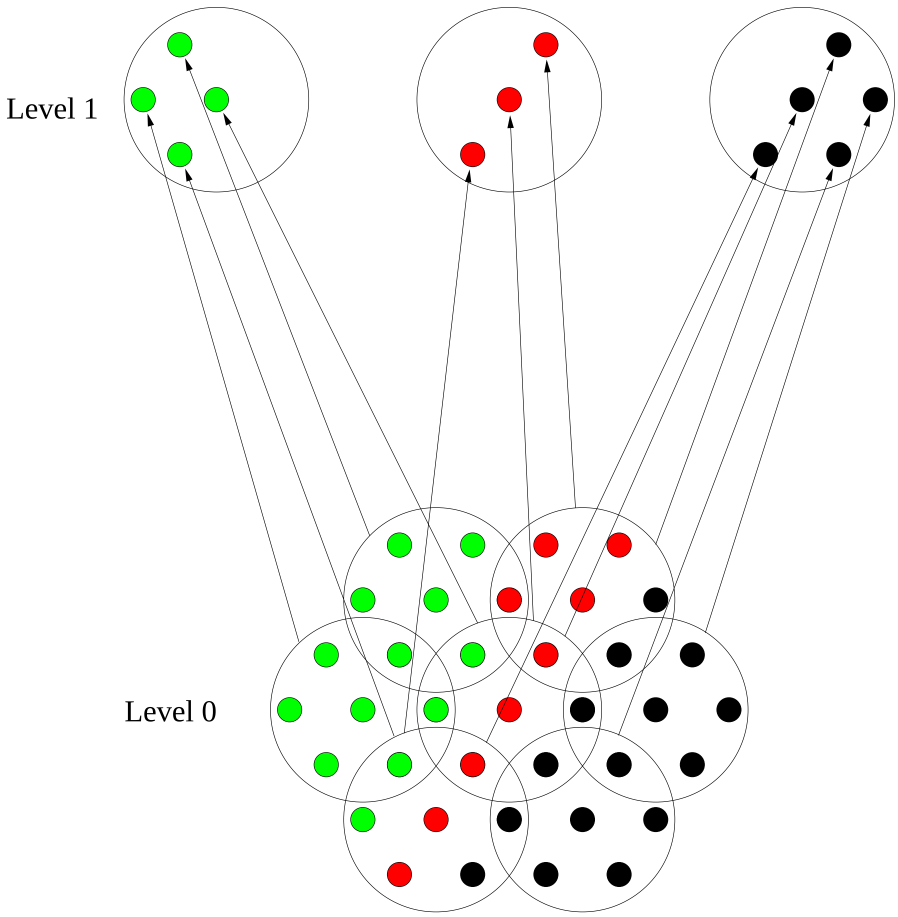
Properties of the Hexagonal Island Hierarchy

The following list contains the essential properties of the hexagonal islands of level $n+1$:

- All islands of level $n+1$ posses the same number of sub-islands. (Homogenity)
- Two islands of level $n+1$ intersect each other in at most one island of level n. (Plainness)
- All sub-islands (except the center island) of an island of level $n+1$ are sub-islands of two different islands of level $n+1$. (Saturation)
- Each island of level n is a sub-island of at least one island of level $n+1$. (Cover)
- Two neighbouring islands I_1 and I_2 of level n are always sub-islands of a common island I of level $n+1$, i.e. $I_1 \in I \wedge I_2 \in I$. (Cover of neighbourship)



Getting a segmentation on level 0.



Getting 3 segments on level 1 from 12 segments in 7 islands of level 0.

Schema for CSC

Input: image $X : \text{Loc} \rightarrow \text{Val}$
regard Loc as if it was equipped with a hexagonal topology

Phase `Initialize`:

```
for all island I of level 0
compute a segmentation 𝒮_I for X|_I
```

Phase `Merge`:

for all `levels` $n > 0$
  for all `islands` I `of level` $n+1$
    create `an (empty) set` $\mathcal{S}_I$ `of segments`
    for all `islands` I^I `of` I `(of level` n`)`
      for all `segments` $S^I \in \mathcal{S}_{I^I}$
        for all `sub-islands` I^{II} `of` I `that overlap with` I^I `in a common island` $I^{I,II}$ `of level` $n-1$
          if S^I `possesses a sub-segment` $S^{I,II}$ `in` $\mathcal{S}_{I^{I,II}}$
          then find `the segment` S^{II} `in` $\mathcal{S}_{I^{II}}$ `that also possesses` $S^{I,II}$ `as a sub-segment`
          if $S^I \cup S^{II}$ `is not homogenous enough`
            then
              compute $\mathtt{SPLIT}^I(S^I, S^{II}, S^{I,II})$
              compute $\mathtt{SPLIT}^{II}(S^I, S^{II}, S^{I,II})$
              $S^I := (S^I - S^{I,II}) \cup \mathtt{SPLIT}^I(S^I, S^{II}, S^{I,II})$
              $S^{II} := (S^{II} - S^{I,II}) \cup \mathtt{SPLIT}^{II}(S^I, S^{II}, S^{I,II})$
          if $S^I \cup S^{II}$ `is homogenous enough`
            then remove S^I `from` $\mathcal{S}_{I^I}$ `and` S^{II} `from` $\mathcal{S}_{I^{II}}$
            add $S := S^I \cup S^{II}$ `to` $\mathcal{S}_I$

$$\mathcal{S} := \bigcup_{\substack{I \text{ island} \\ \text{of any level}}} \mathcal{S}_I$$ `is the found segmentation`

Procedure `SPLIT`

Input: Three segments S^I, S^{II}, and $S^{I,II}$
with $S^{I,II} \subseteq S^I \cap S^{II}$

compute `two subsets` $\texttt{SPLIT}^I$ `and` $\texttt{SPLIT}^{II}$ `s.t.`

- $S^{I,II} = \texttt{SPLIT}^I \cup \texttt{SPLIT}^{II}$
- $\texttt{SPLIT}^I \cap \texttt{SPLIT}^{II} = \emptyset$
- $S^I \cup \texttt{SPLIT}^I$ `is connected`
- $S^{II} \cup \texttt{SPLIT}^{II}$ `is connected`
- $S^I \cup \texttt{SPLIT}^I$ `and` $S^{II} \cup \texttt{SPLIT}^{II}$
  `are (almost) optimal homogenous`

An implementation of the CSC can be found under:

http://www.uni-koblenz.de/~lb/lb_downloads/

Obviously, all islands I of level 0 in Phase 0 (Initialize) or at level $n+1$ in the Merge-Phase may be treated independently. Thus, the CSC structure offers an inherent parallel implementation.

Any segment S on level 0 is represented by $\hat{S}$ with

$$\hat{S}=<\texttt{head, fatherpointer}_1\texttt{, fatherpointer}_2\texttt{, } P_1, \ldots, P_7 >,$$

where $P_i, 1 \leq i \leq 7$, are the pixels belonging to S.
head denotes some ID for S and some homogenity measure if S and the mean color of S, or similar.
Both fatherpointers of $\hat{S}$ are empty when S is created.

Any segment S of level $n > 0$ is represented by $\hat{S}$ with

$$\hat{S}=<\texttt{head, fatherpointer}_1\texttt{, fatherpointer}_2\texttt{, } P_1, \ldots, P_n >,$$

where $P_1, \ldots, P_n$ ($n \geq 7$ may occur!) are pointers to all sub-segments of level $n-1$ that were used to form S and that are sub-segments of S.

When S is created both fatherpointers of $\hat{S}$ are empty. Whenever a sub-segment S^I of level $n-1$ is added to S one fatherpointer of S^I is set to point to S.

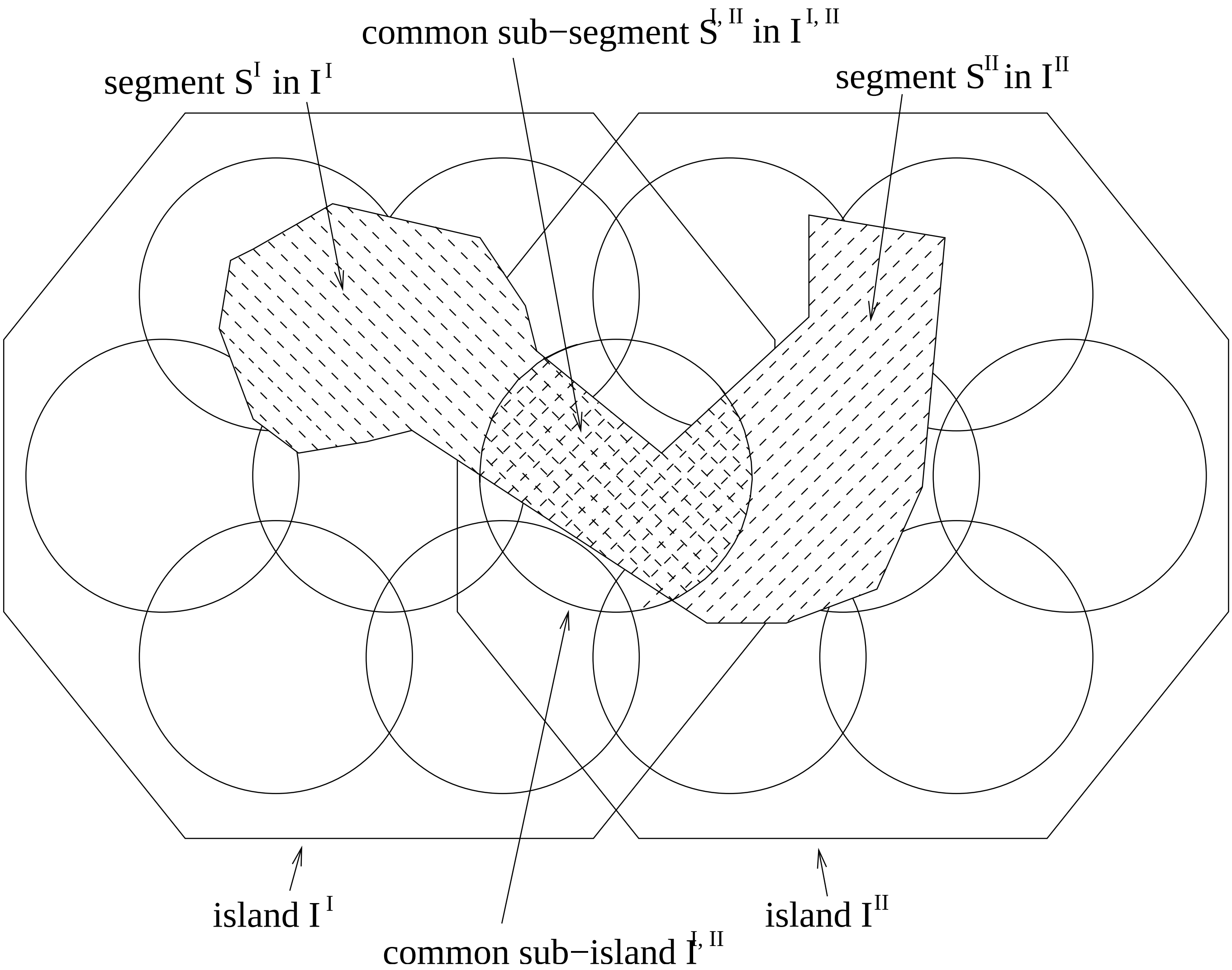
As any segment S^I may belong to at most most two-islands I_1, I_2 of level $n-1$, S^I may be a sub-segment of two different 'father'-segments S_1 in I_1 and S_2 in I_2. Thus, two fatherpointers are sufficient.

Allthough the structure of the Merge-Phase looks rather complicated and complex (in computation time), due to the nested 'for all' structure, the creation of a new segment S of an island I of level $n+1$ becomes now very simple and fast:

If one segment S^I of level n from a sub-island I^I of I already belongs to S, simply move down in $\hat{S^I}$ on a pointer of $\hat{S}$ to a sub-segment S^+ of S^I of level $n-1$ in a sub-island I^+ of I^I that also is a subisland of a second island I^{II} in I.

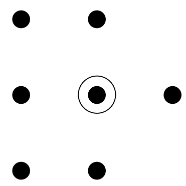
Now, move up the second fatherpointer of $\hat{S^+}$ and one immediately has navigated to the sub-segment S^{II} of I^{II} which overlaps with S^I and is a candidate for merging with S^I to form S.

An implementation of the SPLIT-procedure is not that simple. However, we already have some information on the homogenity of S^I and S^{II} even on level $n+1$ and may use this more global view to decide which pixel of $S^{I,II}$ (of level $n-1$ (!)) shall be added to S^I and which to S^{II}. The difficulty is to not destroy the connectedness, i.e. add only such pixels P to S^I (or S^{II}) s.t. $S^I \cup P$ (or $S^I \cup P$) remains connected.

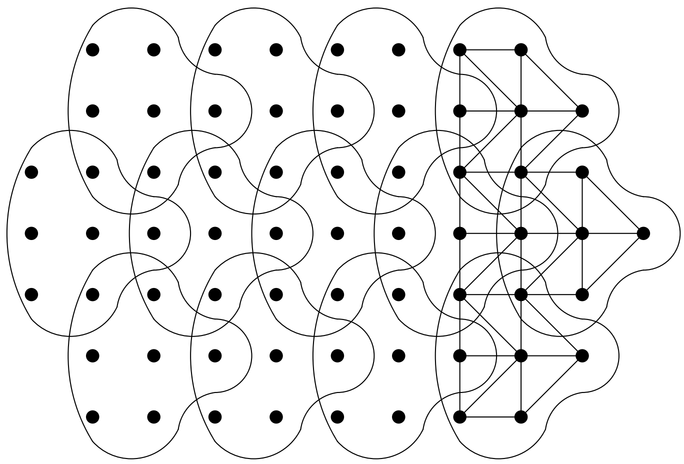


CSC with a standard orthogonal topology

In practice, any standard image $X : \mathrm{Loc} \rightarrow \mathrm{Val}$ is segmented by this technique, where Loc has no hexagonal structure but the standard orthogonal topology of $\mathrm{Loc} = [0, N[\times[0, M[$. We simulate an island of level 0 by seven pixels in a window of the following form



where $\odot$ is the center pixel, all others are border pixels. Analogously for higher levels.
The following figure shows the overlapping structure of those windows.



References

[DuJa93] M. P. Dubuisson, A. K. Jain:
Object contour extraction using color and motion
Proceedings 'Conference on Computer Vision and Pattern Recognition', pp. 471-476, New York, 15-17th June, 1993.

[Ha87] G. Hartmann:
Recognition of 'Hierarchically Encoded Images by Technical and Biological Systems
Biological Cybernetics Vol. 57, pp. 73-84, Springer, 1987.

[HSHD87] D. Harwood, M. Subbarao, H. Hakalahti, and L.S. Davis:
A New Class Of Edge-Preserving Smoothing Filters
PRL Vol. 6, pp. 155-162, 1987.

[HoPa76] S. L. Horowitz, T. Pavlidis:
Picture Segmentation by a Tree Traversal Algorithm
Journal of the ACM Vol. 23, pp. 368-388, 1976.

[KNF76] W.L.G. Koontz, P.M. Narendra, and K. Fukunaga:
A graph-theoretic approach to nonparametric clustering
IEEE Trans. Computers, C-25, pp.936-944, 1976.

[KHEK76] M. Kuwahara, K. Hachimura, S. Eiho, M. Kinoshita:
Processing of RI-angiocardiographic images
Digital Processing of Biomedical Images, New York, pp. 187-202, 1976.

[Ma96] J. Matas:
Colour-based Object recognition
Ph. D. Thesis, Department of Electronic and Electrical Engineering, University of Surrey, 1996

[NaMa79] M. Nagao, T. Matsuyama:
Edge preserving smoothing
Computer Graphics and Image Processing, vol. 9, no. 4, pp. 394-407, 1979.

[Oh75] R. Ohlander:
Analysis of Natural Scenes
Ph. D. Thesis, Department of Computer Science, Carnegie-Mellon University, 1975.

[OKS80] Y. Ohta, T. Kanade, T. Sakai:
Color Information for Region Segmentation
Computer Graphics and Image Processing Vol. 13, pp. 222-241, 1980.

[PrRe92] L. Priese, V. Rehrmann:
A Fast Hybrid Color Segmentation Method
Fachberichte Informatik, Universität Koblenz-Landau, 1992

[PrRe93] L. Priese, V. Rehrmann:
On Hierarchical Color Segmentation and Applications
Proceedings 'Conference on Computer Vision and Pattern Recognition', pp. 633-634, New York, 15-17th June, 1993.

[Re94] V. Rehrmann:
Stabile, echtzeitfähige Farbbildauswertung
Ph. D. Thesis, Department of Computer Science, Universitity of Koblenz, Germany, 1994