

3D-Color-Structure-Code – A Hierarchical Region Growing Method for Segmentation of 3D-Images

Patrick Sturm and Lutz Prieße

Universität Koblenz-Landau, Institute of Computational Visualistics,
Universitätsstraße 1, 56070 Koblenz, Germany
{sturm,prieße}@uni-koblenz.de

Abstract. The Color Structure Code (CSC) [RE98] is a very efficient segmentation method for color images. It operates with a hierarchical region growing method. Originally, the CSC was constructed to work on two dimensional images. The generalization of the CSC for 3d images, is not a trivial task. Some important modifications of the original (2d) algorithm have to be done as the underlying 2-dimensional hierarchical hexagon topology cannot be generalized in all aspects to three dimensions.

1 Introduction

Image segmentation is an important step in image analysis. It divides an image into possibly large, pairwise disjoint segments [HP76]. Segments are defined as *spatial connected* pixel sets (pixel = location + color) that fulfill some homogeneity criterion. Generally, segments could be considered to be homogeneous in gray value, color or texture. Segmentation is used to get an abstract, symbolic representation of an image. The quality of an image analysis depends strongly on the quality of the segmentation result. Nowadays 3-dimensional images play an important role. They are produced, e.g., by diagnostic methods like Computer Tomography or Magnet Resonance Tomography. Thus, it is a valuable task to generalize the very successful 2d CSC-segmentation technique ([PR93], [RE98]) to 3d images. However, this approach leads to some surprising difficulties.

2 The Hexagonal 2D-CSC

The CSC follows a hierarchical region growing on a special hexagonal topology that was firstly introduced by Hartmann [HA87]. This hierarchical topology (see fig. 1a) is formed by so-called *islands* of different levels. One island of level 0 consists of seven pixels (one center pixel and its 6 neighbors) in the hexagonal topology. The partition of the image is organized in such a way that the islands are overlapping (each second pixel of each second row is a center of an island of level 0). One island of level $n+1$ consists of seven overlapping islands of level n

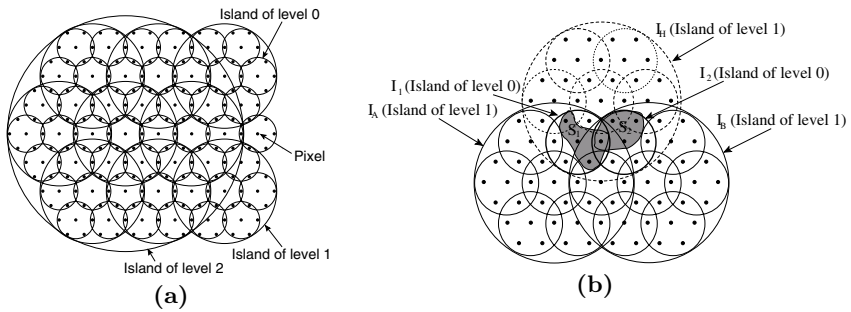


Fig. 1. a) The Hexagonal Island Hierarchy. b) Island I_H covers both islands I_1 and I_2 .

(s. fig. 1a). Repeating this until one island covers the whole image the number of islands decreases from level to level by factor 4. In the following we denote an island I of level n as I^n .

The segmentation algorithm operates essentially in three phases: *Initialization*, *Linking* and *Splitting*. In the *initialization phase* color homogeneous regions (initial segments) in level 0 islands of at most seven pixels are detected independently in each island of level 0. Such an (initial) segment is represented by a *code* c^0 of level 0 which stores the mean color and all pixels ($Sub(c^0)$) of that segment. The pixels of $Sub(c^0)$ are also called sub-codes of c^0 . Two codes c_1^0, c_2^0 are called connected iff they are representing segments possessing a common pixel, i.e. $Sub(c_1^0) \cap Sub(c_2^0) \neq \emptyset$. The result of the initialization phase is a set of codes, each one describing an initial segment.

In the linking phase these initial segments grow hierarchically to complete color segments. For that purpose in each island I^{n+1} codes of level n in the sub-islands I_1^n, \dots, I_7^n of I^{n+1} are linked to new codes of level $n+1$. Codes will be linked if the segments represented by them are connected and similar in color. The connectivity of codes can easily be determined within the hexagonal island structure: two codes c_1^n, c_2^n are connected if they share a common sub-segment in their common sub-island of level $n-1$. On level 1 this simply means that they possess a common pixel. The linking operations are repeated for all islands on each level. They start from level 1 and stop on the topmost level of the island hierarchy where only one island exists that covers the whole image. The codes c_1^n, \dots, c_m^n of level n that are linked to a new code c^{n+1} of level $n+1$ are so called sub-codes of c^{n+1} ($Sub(c^{n+1})$). Note, codes may have more than seven sub-codes ($m > 7$). Both the mean color of the sub-codes and pointers to them are stored together with the new code c^{n+1} . The linking of color similar and connected codes builds up a code tree. A code forms a root of such a code tree if no link partner could be found. Each code tree represents a complete color segment whose pixels can be computed by descending the code tree starting with its root. We denote $S(c^n)$ for the pixels of a segment that is represented by a code tree with root c^n : $S(c^0) := Sub(c^0)$ and $S(c^{n+1}) := \bigcup_{s^n \in Sub(c^{n+1})} S(s^n)$. The code tree with s^n as root represents the common sub-segment of both codes.

Local region growing methods often link differently colored regions due to a chain of connected pixels with smoothly changing colors (chaining mismatches). In the CSC this problem is solved by additional color checks between connected codes on every linking level. If the color of two connected codes c_1^n , c_2^n are not similar the two codes won't be linked although they are connected by a common sub-code. This means that $S(c_1^n)$ and $S(c_2^n)$ are two different segments that are connected via a common sub-segment. Therefore, their common sub-segment has to be partitioned optimally between $S(c_1^n)$ and $S(c_2^n)$ dependent on the mean color represented in c_1^n and c_2^n . This is done in the splitting phase. (for detailed information see [RE98]).

3 Properties of the Hexagonal Island Hierarchy

The hexagonal island hierarchy has the following properties (see [ST02]):

1. Homogeneity: All islands of level $n+1$ comprise the same number of islands of level n (*sub-islands*).
2. Plainness: Two islands of level $n+1$ overlap each other in at most one island of level n .
3. Saturation: All sub-islands (except the center island) of an island of level $n+1$ are sub-islands of two different islands of level $n+1$.
4. Coverability: Each island of level n (except the topmost island) is a sub-island of at least one island of level $n+1$.
5. Density: Two neighboring islands I_1^n and I_2^n are always sub-islands of a common island I^{n+1} , i.e. $I_1^n \in I^{n+1} \wedge I_2^n \in I^{n+1}$.

The first three properties are important for a simple design of the segmentation algorithm. The CSC would not be as simple as it is if we would allow different island types or complex overlapping structures. The coverability and density properties ensure that segments can grow in all directions without limitations. Suppose the coverability property would be violated. Then there are some pixels that are not covered by any island of level 0. These pixels could never be detected as part of any segment. The density property ensures that two color similar overlapping segments can be linked to a new segment of the next level. Consider the example of fig. 1b. The figure shows two color similar segments S_1 and S_2 that share a common pixel. S_1 was detected in level 0 island I_1 and S_2 in level 0 island I_2 . Because I_1 and I_2 are both sub-islands of level 1 island I_H the segments S_1 and S_2 can only be linked within I_H . S_1 and S_2 would not be linked together if I_H would be missing (violation of the density property). A CSC that works with an island hierarchy that does not fulfill the density property may produce different but overlapping segments. Overlapping segments are represented by overlapping code trees. Two code trees with root nodes $r_1^{m_1}$ and $r_2^{m_2}$ are called overlapping iff there is a code c^n that is reachable via a path from both root codes $r_1^{m_1}$ and $r_2^{m_2}$.

4 The 3D-CSC

For a generalization of the hexagonal CSC that operates on 3d images a three-dimensional island hierarchy is necessary that has (nearly) the same properties as the hexagonal island hierarchy (see section 3). A 3d lattice that is very similar to the hexagonal lattice is the most dense sphere packing (see fig. 2). We define that two spheres of the packing are called neighbors if they touches each other. By slightly increasing each sphere we will get the desired overlapping property. Each sphere has exactly 12 neighbors – 6 neighbors in the same layer, 3 neighbors in the layer above and 3 neighbors in the layer below. Each neighbor should overlap. We define in a first step a 3d island of level 0 as a set of 13 spheres – one center sphere and its 12 neighbors. Each second sphere of each second row of each second layer has to be a center sphere to ensure that two 3d islands of level 0 overlaps in at most one common sphere (Plainness) and that each sphere except the center sphere is covered by exactly two different islands (Saturation). As one may see in fig. 2 there are some (black) spheres that are not a neighbor of any center sphere. Thus, such spheres (holes) are not covered by any 3d island of level 0 (violation of the coverability property). To ensure the coverability property a 3d island of level 0 must consist of 15 spheres – a center sphere, its 12 neighbors and two additional holes that lie close to the island center (s. fig. 2d). Now each 3d island of level 0 overlaps with exactly 14 other islands of level 0. Generally a 3d island of level $n+1$ consists of one center island of level n and its 14 neighbors. Also a 3d island of level $n+1$ overlaps with exactly 14 other islands of level $n+1$. From level to level the number of islands decreases by factor 8.

The 3d island hierarchy fulfills all properties of the hexagonal island hierarchy except the density property (for a proof see [ST02]). Thus, the 3D-CSC may produce overlapping code trees that represents overlapping segments. Although we tried a lot of different 3d island hierarchies we could not find any that fulfills all 5 properties of the hexagonal island hierarchy. Thus, we decide to use the above 3d island hierarchy, that is defined upon the most dense sphere packing and to deal with overlapping segments. Two overlapping segment have to be

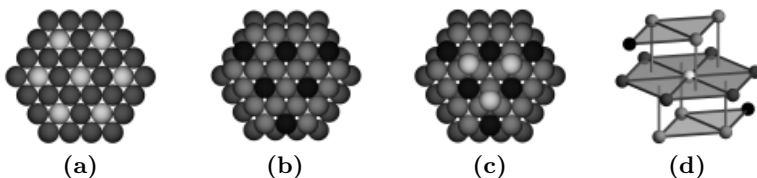


Fig. 2. a) First layer of the most dense sphere packing. White spheres are island centers of level 0. Gray Spheres are direct neighbors of island centers. b) Second layer of the sphere packing. Black spheres represent holes. Holes are spheres with no center sphere as neighbor. c) Third layer of the sphere packing. c) Schematical representation of a 3d island of level 0.

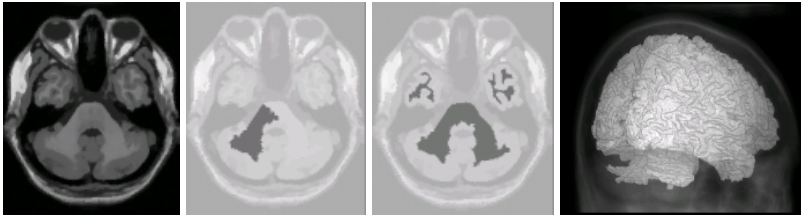


Fig. 3. The first image shows a slice taken from a 3d MRT-image of a human head. The second (without merge) and the third image (with merge) shows each a slice taken of one detected 3d segment. The last image shows a projection of the 3d segment of the third image.

merged if they are color similar. If not, they have to be split optimally. This is done in the *deferred splitting-and-linking phase*.

The first step is to detect whether two code trees T_1, T_2 with root codes $r_1^{m_1}, r_2^{m_2}$ are overlapping in some sub-code c^n . This is done by a simple top-down labeling of sub-codes with their root-codes. If a sub-code c^n receives for the first time two different root codes, the labeling is interrupted, and a decision is made to which tree either T_1, T_2 or a new tree $T_{1,2}$ the sub-code c^n shall belong to. If the mean color in $r_1^{m_1}$ and $r_2^{m_2}$ are similar T_1 and T_2 are melt into a new code tree $T_{1,2}$ by adding a meta code μ that is a new root code and a parent for $r_1^{m_1}$ and $r_2^{m_2}$. In this case c^n belongs to tree $T_{1,2}$ and is labeled with μ . Some sub-codes may still labeled with $r_1^{m_1}$ resp. $r_2^{m_2}$ although $r_1^{m_1}$ and $r_2^{m_2}$ are no root codes anymore. Therefore these labels are renamed into μ . If the mean colors of $r_1^{m_1}$ and $r_2^{m_2}$ are too different, c^n shall belong solely to the tree T_1 or T_2 which is most color similar (in its root) to c^n . Let us assume that c^n is more color similar to $r_2^{m_2}$. Then c^n has to be deleted from T_1 which is done by simply removing the pointer to c^n from its parent code p_1^{n+1} in T_1 . This does not mean that the sub-segment represented by c^n is completely assigned to the code tree T_2 : Although c^n is deleted from T_1 it is still connected with some sub-codes of p_1^{n+1} due to the overlapping of some sons of p_1^{n+1} . There must be at least one sub-code b^n of p_1^{n+1} that is a brother of c^n overlapping with c^n on level $n-1$. Thus, the trees T_1 and T_2 are still overlapping due to some connection between c^n and its brother b^n . These further connections between T_1 and T_2 are discovered and handled by the labeling algorithm on level $n-1$ in the same way. Thus, c^n may loose some sub-segments on lower levels and is not assigned completely to the code tree T_2 . In this manner a optimal separation between T_1 and T_2 is reached. Please note, that the top-down approach of the labeling algorithm ensures that each sub-code is labeled with just one root code. Therefore the overlapping of only two code trees have to be solved in any step although several trees T_1, \dots, T_n may overlap in a common sub-tree c^k . However, there is a further difficulty: by deleting some codes at different levels from a code tree T it may happen that the voxel set represented by T is not spatial connected anymore, and thus T does

not represent a segment. The above description is just an outline. Avoiding a loss of connectivity and reaching a color optimal separation of c^n between T_1 and T_2 are two contradict aims. Practical solutions to that problem become rather involved.

Code trees generated by the hexagonal CSC (standard code trees) are obviously different from those generated by the 3D-CSC. Therefore evaluation methods that are constructed for standard code trees have to be modified to deal with those virtually merged code trees.

5 Conclusions

The advantages of the 2d-CSC are its good performance and its accuracy in finding color homogeneous regions without any chaining mismatches. We introduced in this paper a method how the 2d-CSC can be generalized to a three-dimensional segmentation. Because of some defects in the three-dimensional island hierarchy, an additional merging and splitting phase is necessary to ensure that each segment is as large as possible and that there are only pairwise disjoint segments in the segmentation result. Fig. 3 shows a segmentation result. The left image shows a slice that is taken from a 3d-MRT-image (size: $181 \times 217 \times 181$ voxel). The next image shows the same slice in which the voxels belonging to a certain 3d-segment are highlighted. The 3d-segmentation is computed by a 3D-CSC without deferred splitting-and-linking phase. The second image shows also a slice of a highlighted 3d-segment. This time it is computed by a 3D-CSC with deferred splitting-and-linking phase. The last image shows the whole 3d-segment of the third image.

References

- [HA87] G. Hartmann. Recognition of Hierarchically Encoded Images by Technical and Biological Systems. In: *Biological Cybernetics*, 57:73–84, 1987.
- [HP76] S. L. Horowitz, T. Pavlidis. Picture Segmentation by a Traversal Algorithm. *Journal of the ACM*, 23:368–388, 1976.
- [PR93] L. Priese, V. Rehrmann. A Fast Hybrid Color Segmentation Method. In: S.J. Pöppel and H. Handels, editors, *Mustererkennung 1993*, pages 297–304. Springer Verlag, 1993. 15. DAGM-Symposium, Lübeck, 27.–29. Sept. 1993.
- [RE98] V. Rehrmann, L. Priese. Fast and Robust Segmentation of Natural Color Scenes. 3rd Asian Conference on Computer Vision, Hongkong, 8–10th January 1998.
- [ST02] P. Sturm, L. Priese. Properties of a Three-Dimensional Island Hierarchy for Segmentation of 3D Images with the Color Structure. In: Luc van Gool, editor, *Pattern Recognition*, pages 274–281. Springer Verlag, 2002. 24th DAGM-Conference, Zürich, 16–18th September 2002.