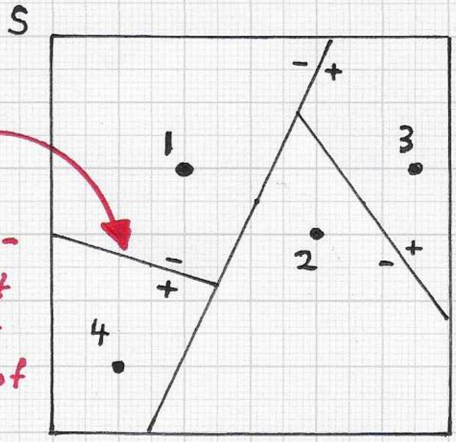
■ OBJECT AND MATERIAL EIGENFUNCTIONS — Cont'd.

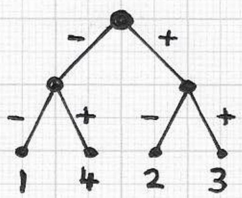
• Laplacian eigenfunctions and neural networks:...

Concept of BSP tree in 2D space for points:

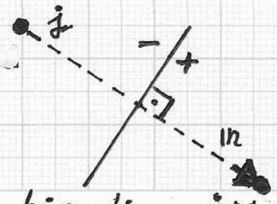


any hyper-plane that separates the tiles of points 1 and 4 — NOT necessarily the perpendicular bisector

Points 1, 2, 3 and 4 are inserted one-by-one in the domain S. Each point insertion causes an update of the spatial subdivision/partition and the binary tree capturing the insertion process. A binary split partitions the tile containing the point to be inserted into a negative (-) and positive (+) halfspace.



Tree resulting from inserting 1, then 2, then 3, then 4



bisecting hyper-plane

Using difference vector n of next point to be inserted, point i+1, and tile owner, point j, to define '-' and '+' halfspaces

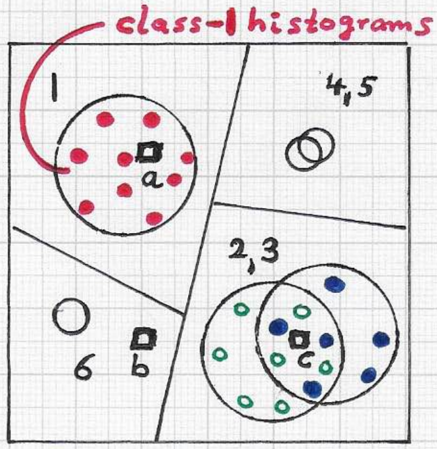
In summary, Voronoi diagrams are too complex to be used for a high-dimensional representation of space, and a simple binary subdivision method using repeated midpoint-value splitting of regions with alignment to a coordinate system's axes is insufficient for our goals. **Henry Fuchs, Z.M. Kedem and B.F. Naylor** introduced the **BINARY SPACE PARTITION (BSP) tree into Computer Graphics** ("On visible surface generation by a priori tree structures," Proc. SIGGRAPH '80). The BSP tree provides the needed representational capabilities for our high-dimensional classification problem — and is computationally acceptable as far as complexity is concerned for required data structures. A simple example is shown in the left figures, where four points are inserted iteratively. We must devise a generalization for disks/hyper-spheres instead of points.

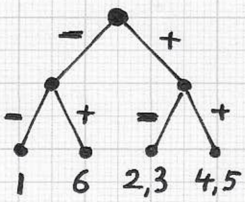
...

■ OBJECT AND MATERIAL EIGENFUNCTIONS — Cont'd.

• Laplacian eigenfunctions and neural networks:...

Generalized BSP tree structure for set of disks, including intersecting disks — 2D scenario:





Disks that intersect produce leaves that are sets of the overlapping disks.

This BSP tree captures only one of H available bounding hyper-sphere configurations for these six classes and their sample sets — with H referring to the number of scales.

The points a, b and c (□) capture three distinct possibilities:

- **a** inside sphere of class-1 samples ⇒ class-1 candidate
- **b** inside tile of class 6, outside sphere of class-6 samples ⇒ class-0 case
- **c** inside spheres of class-2 and class-3 samples ⇒ class-2 and class-3 candidate

The needed generalization of the "basic" BSP tree is illustrated in the left figure, with disks of different radii and intersecting disks (treated like single-disk primitives for tree construction).

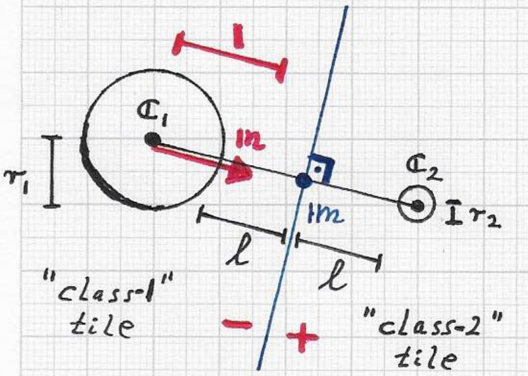
• **Note.** The underlying eigenfunction-based method generates a multitude of coefficient value histograms for each image segment sample, i.e., H histograms for H scales. Due to the availability of these scales, one can assume that two material classes — their coefficient value histogram point sets — can be separated, even when, for some scales, these point sets' bounding hyperspheres intersect.

For example, the unclassified histogram point c included in the space partition shown in the left figure lies inside the intersection region of spheres around class-2 and class-3 samples. By considering several scales, the more likely class can emerge. ...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:...

Half-space construction for two hyper-spheres with different radius values:



• Given: 2 hyper-spheres with center points $\mathbb{c}_1$ and $\mathbb{c}_2$, and radii r_1 and r_2

• Wanted: Separating hyper-plane defined by point $\mathbb{m}$ lying in it and unit normal vector $\mathbb{n}$

• Algorithm: One must use the following vectors and points in B-dimensional space:

$\mathbb{c}_1 = (x^1_1,\dots,x^1_B)^T$, $\mathbb{c}_2 = (x^2_1,\dots,x^2_B)^T$

$\mathbb{n} = (n_1,\dots,n_B)^T = (\mathbb{c}_2-\mathbb{c}_1) / \|(\mathbb{c}_2-\mathbb{c}_1)\|$

$\mathbb{m} = (m_1,\dots,m_B)^T = \mathbb{c}_1 + (r_1 + l)\,\mathbb{n}$,

where $\|\mathbb{c}_2-\mathbb{c}_1\| = r_1 + 2l + r_2$

$\Rightarrow l = (\|\mathbb{c}_2-\mathbb{c}_1\| - r_1 - r_2)$

...

The construction of a BSP tree involves several geometrical and algebraic concepts and computations. We discuss some of them. First, the definition of a separating hyper-plane that partitions space / a tile into a positive (+) and negative (−) half-space is essential. A possible construction of such a hyper-plane is illustrated in the left figure, to partition space when two hyper-spheres are considered. This construction reminds one of the tile boundary construction of the generalized Voronoi diagram for hyper-spheres — but the separators we use here are only hyper-PLANES.

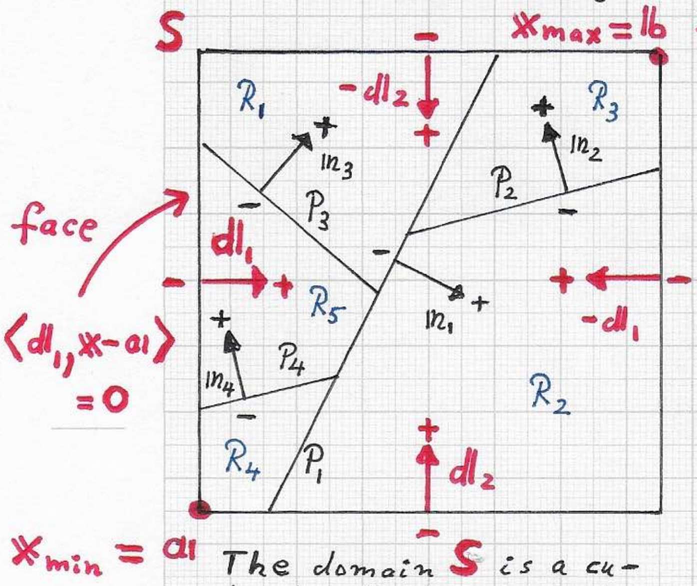
One must define a meaningful midpoint $\mathbb{m}$, e.g., a point on the line passing through the centers of two hyper-spheres, as done here. Further, one must establish a unit normal for the hyper-plane that passes through $\mathbb{m}$; this directed normal $\mathbb{n}$ defines the positive (+) and negative (−) half-spaces.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• *Laplacian eigenfunctions and neural networks:...*

Partition of space into set of tiles resulting from repeated splitting:



The domain S is a cuboid in B-dimensional space with extremal corner points designated as

$$\mathbf{a} = \mathbf{x}_{min} = (x_1^{min}, \ldots, x_B^{min})^T,$$
$$\mathbf{b} = \mathbf{x}_{max} = (x_1^{max}, \ldots, x_B^{max})^T.$$

The $2B$ unit normal vectors of the $2B$ (hyper-) faces of the domain cuboid S are the vectors

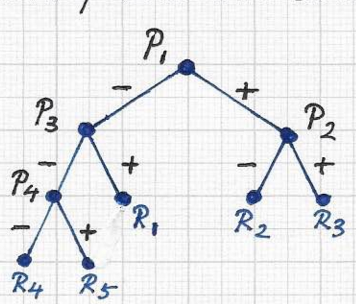
$$\mathbf{d}_1 = (1, 0, \ldots, 0)^T, -\mathbf{d}_1,$$
$$\ldots \mathbf{d}_B = (0, \ldots, 0, 1)^T, -\mathbf{d}_B.$$

The oriented separating hyper-planes P_i have associated (unit) normal vectors $\mathbf{n}_i$. The CONVEX tiles R_j result when repeatedly splitting in the order P_1-, P_2-, P_3-, P_4-split.

The result of the computations summarized on the previous page define the needed algebraic, implicit representation of the separating oriented hyper-plane. Using the (mid)point $\mathbf{m}$ and the "outward" normal $\mathbf{n}$ of the hyper-plane pointing in the '+' half-space, the hyper-plane is defined as

$$n_1 (x_1 - m_1) + \ldots + n_B (x_B - m_B) = 0$$
$$\Leftrightarrow \quad \langle \mathbf{n}_1, \mathbf{x} - \mathbf{m}_1 \rangle = 0 \Leftrightarrow P_1(\mathbf{x}) = 0.$$

where $\mathbf{x} = (x_1, \ldots, x_B)^T$ is a point in B-dimensional space. The partition shown in the left figure has the following underlying tree reflecting the partition's iterative generation:



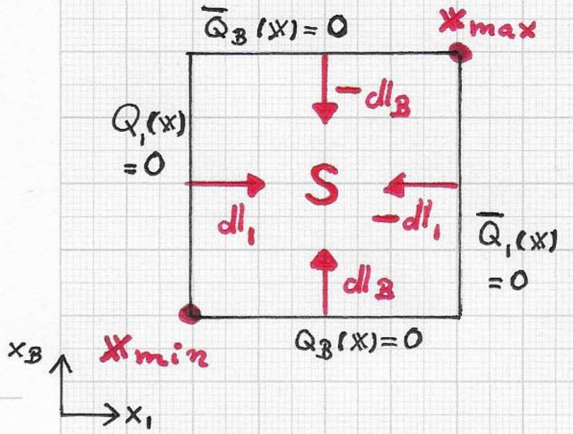
• **Note.** A half-space is convex, and the intersection of convex regions is convex. Thus, all tiles resulting from repeated space/tile partition via hyper-planes generates a BSP having only convex tiles.

...

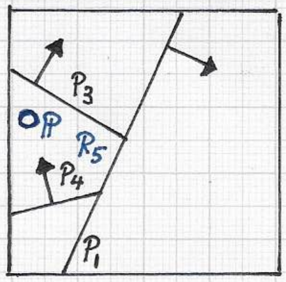
■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:...

Bounding hyper-faces of the cuboid domain in B-dimsional space:



The interior of the rectangle, a hyper-cuboid in B-dimensional space, is bounded by $2B$ hyper-planes that have unit outward normals pointing into the domain S. The figure concerns $B=2$.



Using BSP and tree from the previous page to determine tile for point $\mathbb{p}$:

i) $P_1(\mathbb{p}) < 0 \Rightarrow$ '–'
ii) $P_3(\mathbb{p}) < 0 \Rightarrow$ '–'
iii) $P_4(\mathbb{p}) > 0 \Rightarrow$ '+'
DONE.

Thus, the tile containing $\mathbb{p}$ is tile R_5.

The figure (left) merely stresses that the domain S can itself be defined via $2B$ implicitly represented hyper-planes $Q_1, \ldots, Q_B$ and $\overline{Q}_1, \ldots, \overline{Q}_B$. More specifically, these hyper-planes are

$$Q_1(\mathbb{x}) = \langle dl_1, \mathbb{x} - \mathbb{x}_{min} \rangle = 0, \ldots,$$
$$Q_B(\mathbb{x}) = \langle dl_B, \mathbb{x} - \mathbb{x}_{min} \rangle = 0 \text{ and}$$
$$\overline{Q}_1(\mathbb{x}) = \langle -dl_1, \mathbb{x} - \mathbb{x}_{max} \rangle = 0, \ldots,$$
$$\overline{Q}_B(\mathbb{x}) = \langle -dl_B, \mathbb{x} - \mathbb{x}_{max} \rangle = 0.$$

Of course, by default it is assumed and expected that every classified sample histogram point, and an unclassified histogram point, lies inside the hyper-cuboid S. The hyper-planes $P_i(\mathbb{x}) = 0$ have $P_i(\mathbb{x})$ as their defining linear function that one can use to determine whether a point lies in the negative (–) or positive (+) half-space of the hyper-plane, or inside the plane:

$$P_i(\mathbb{x}) \begin{cases} < 0 \Rightarrow \mathbb{x} \text{ lies in '–' half-space} \\ = 0 \Rightarrow \mathbb{x} \text{ lies inside hyper-plane} \\ > 0 \Rightarrow \mathbb{x} \text{ lies in '+' half-space} \end{cases}$$

Given a specific point $\mathbb{x}$ and a BSP tree for a partition of S, one can determine the region R_j containing $\mathbb{x}$ in logarithmic time.

...