

# Computational topology: Lecture 16

Alberto Paoluzzi

May 10, 2019

- 1 Surface extraction from medical imaging
- 2 Terminology
- 3 Computational pipeline

# Surface extraction from medical imaging

# Topologically exact geometric models

Progressive extraction of neural models from high-resolution 3D images of brain, CAD'16,  
June 27-29, 2016, Vancouver, BC, Canada

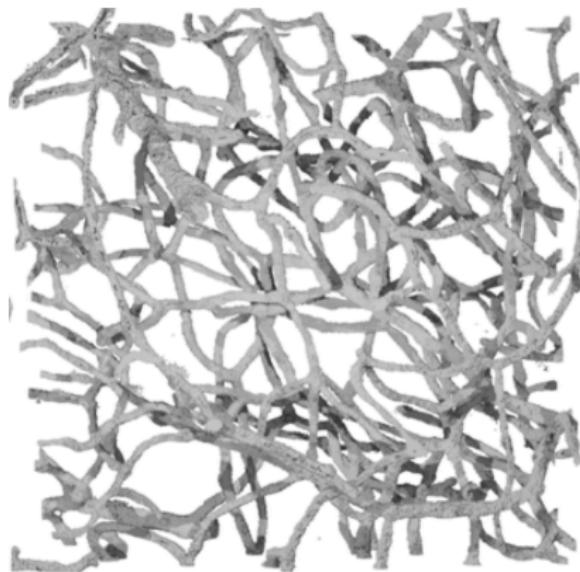


Figure 1: Model of microvessels between neurons of a Rhesus monkey

# Progressive extraction of neural models from high-resolution 3D images of brain

Data source: From [Visus framework](#) (Pascucci et al.  
LLNL, Livermore and SCI, Salt Lake City, 2002-2019)

The ViSUS software framework was [designed](#) to allow the [interactive exploration](#) of [massive scientific models](#) on a variety of hardware, even [geographically distributed](#)

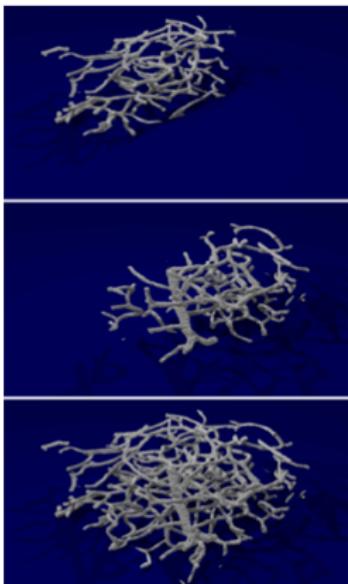


Figure 2: [Portal veins](#)

# Marching-cubes (Siggraph, 1987)

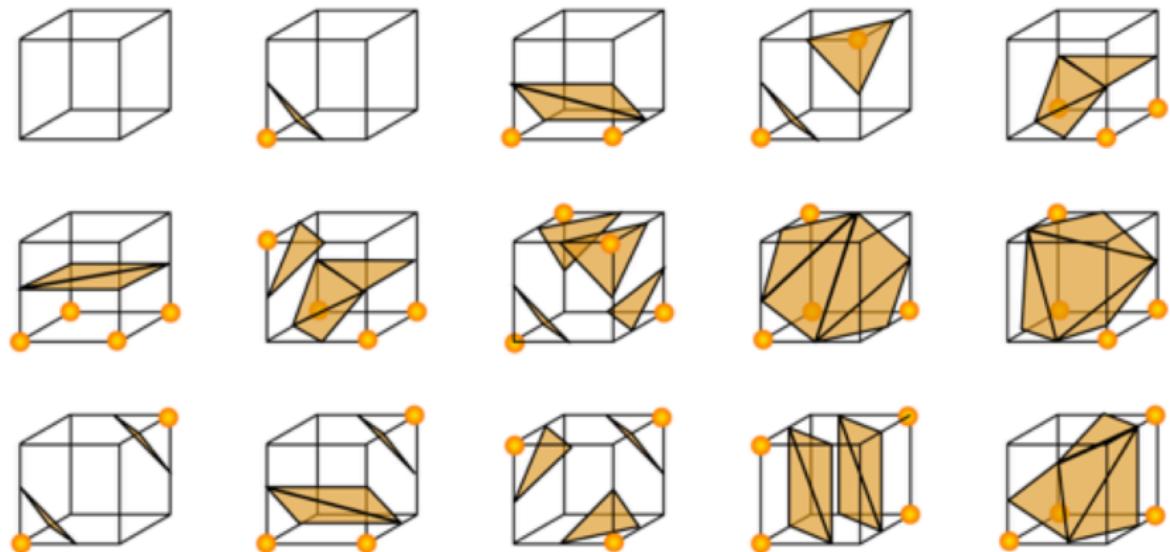


Figure 3: Find the 0-surface (or any iso-surface) of a discrete 3D field

# Progress of high-resolution imaging technology (1987)

William E. Lorensen & Harvey E. Cline, "Marching Cubes: a High Resolution 3D Surface Construction Algorithm", Siggraph, 1987

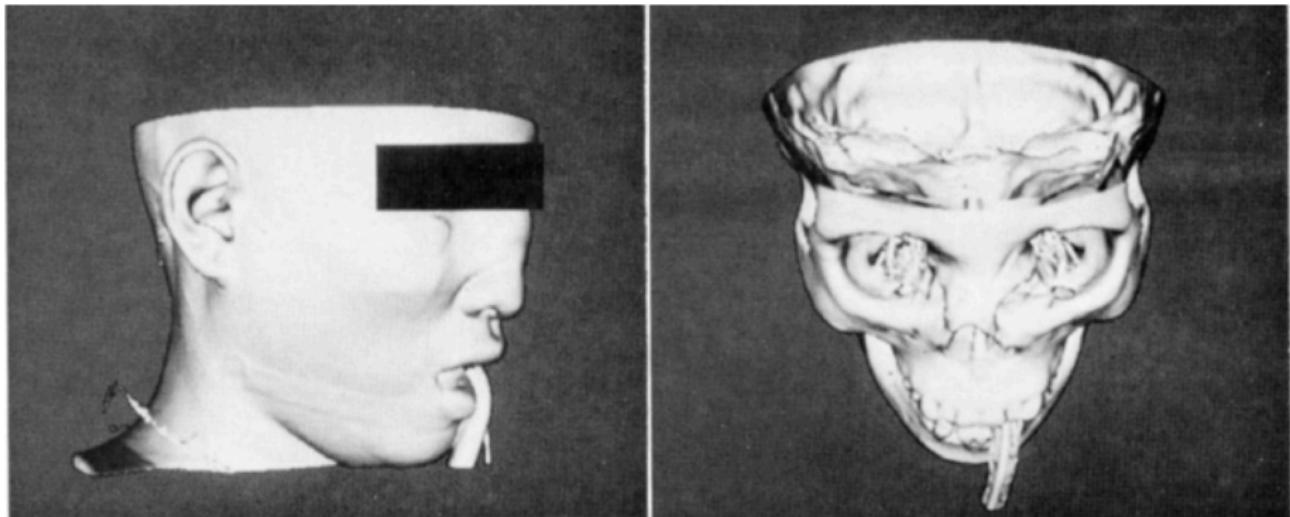


Figure 4:  $260 \times 260 \times 93$  CT images.

The 93 axial slices are 1.5 mm thick, with pixel dimensions of 0.8 mm

# Progress of high-resolution imaging technology (2015)

Nobel Prize in Chemistry 2014 awarded to Betzig, Moerner & Hell for "development of super-resolved fluorescence microscopy," which brings "optical microscopy into the nanodimension"

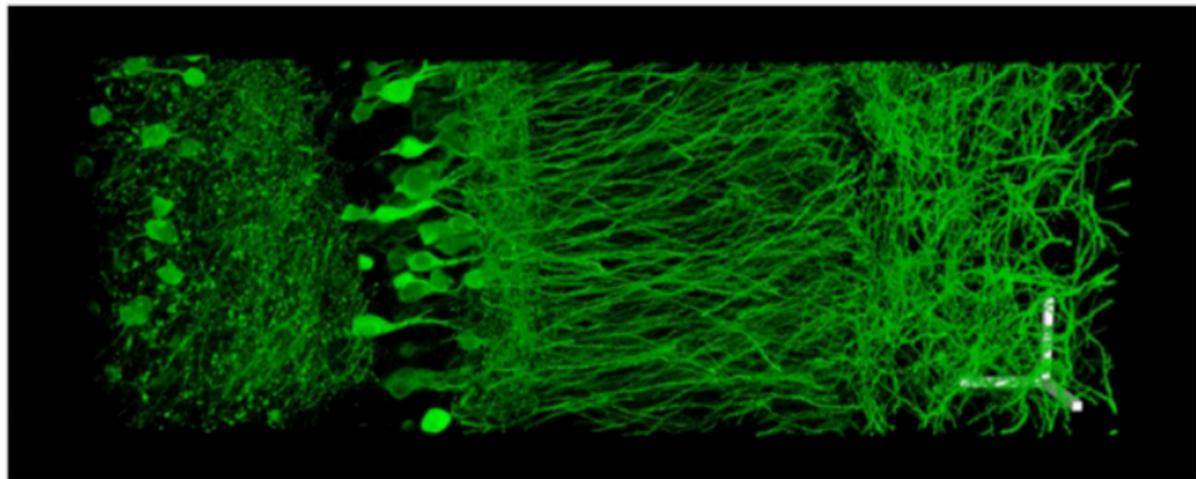


Figure 3: Volume rendering (i.e. dynamic images) of a portion of hippocampus showing neurons and synapses

Source: Fei Chen, Paul W. Tillberg, Edward S. Boyden. Expansion microscopy. Science, January 15 2015; DOI: 10.1126/science.1260088

# Progress of high-resolution imaging technology (2015)

Nobel Prize in Chemistry 2014 awarded to Betzig, Moerner & Hell for "development of super-resolved fluorescence microscopy," which brings "optical microscopy into the nanodimension"

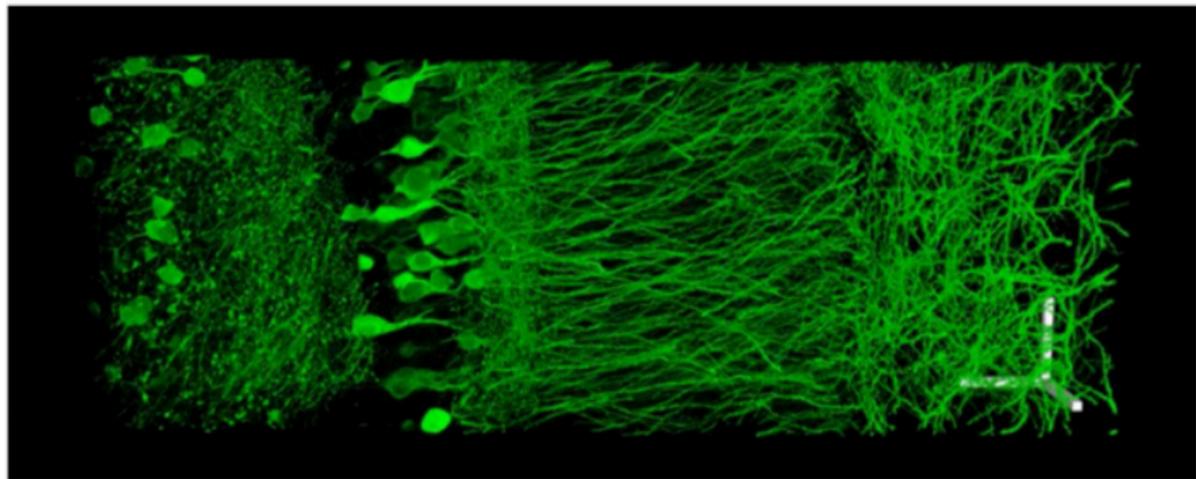


Figure 3: Volume rendering (i.e. dynamic images) of a portion of hippocampus showing neurons and synapses

Source: Fei Chen, Paul W. Tillberg, Edward S. Boyden. Expansion microscopy. Science, January 15 2015; DOI: 10.1126/science.1260088

# LAR examples (1/2)

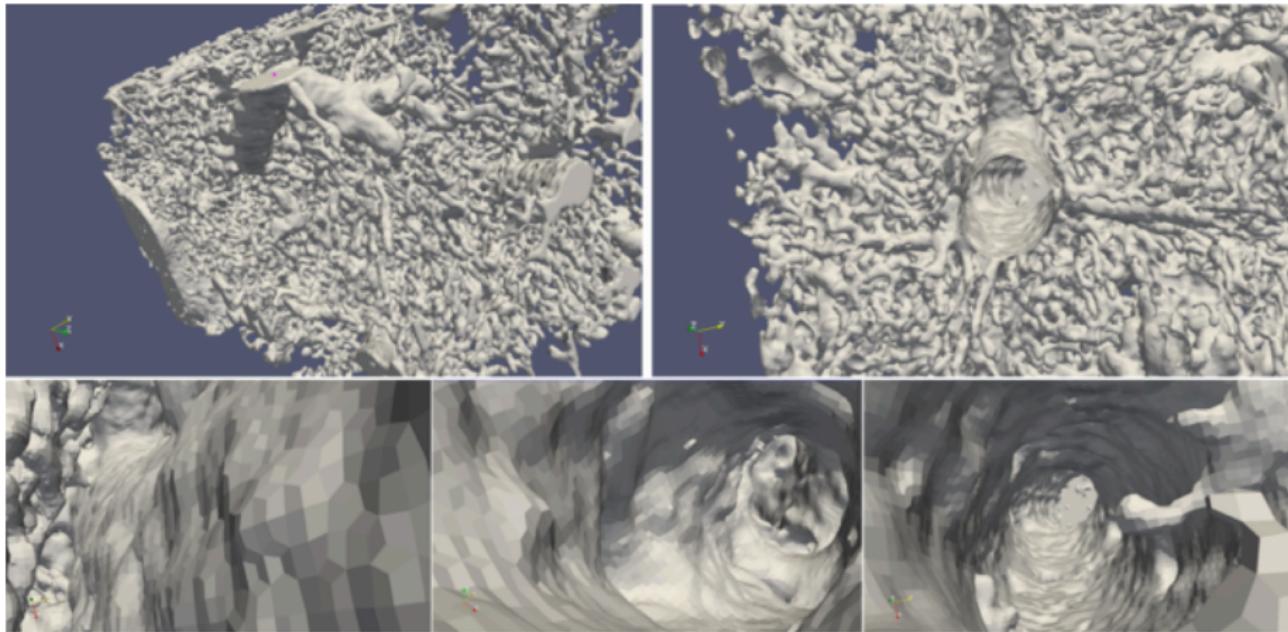


Figure 7: Paoluzzi, DiCarlo, Furiani, Jirik, CAD models from medical images using LAR, Computer-Aided Design and Applications, 2016

## LAR examples (2/2)

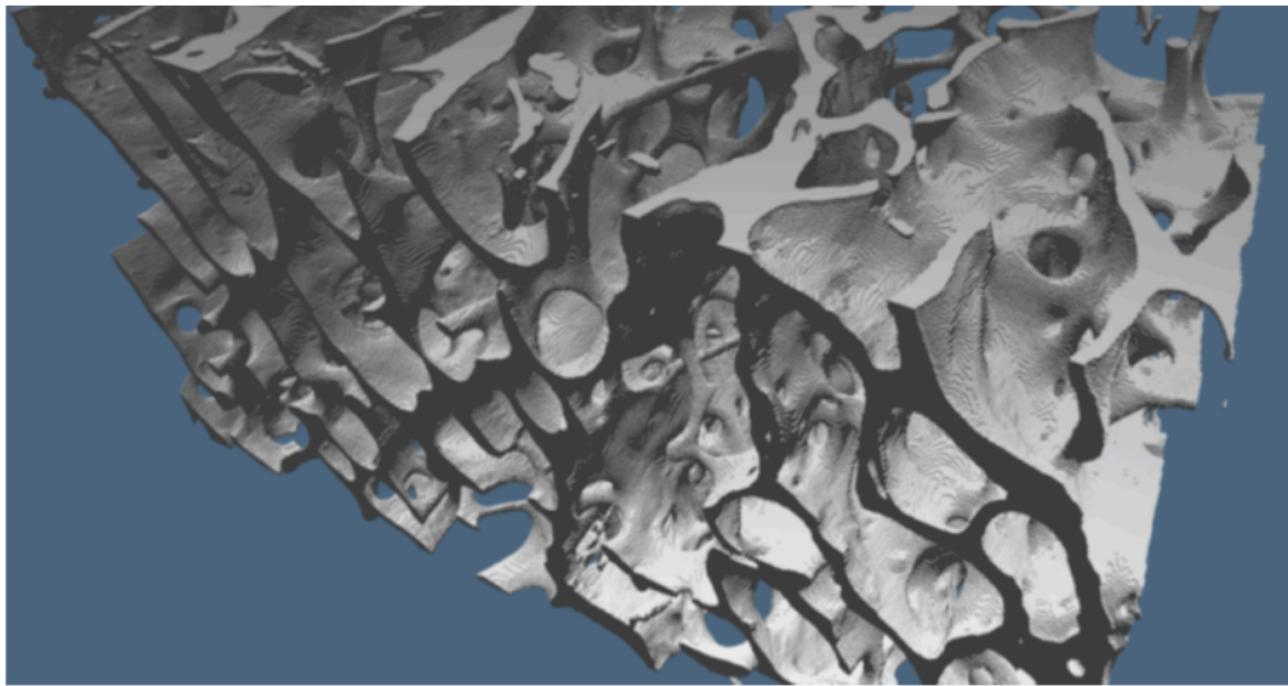


Figure 8: Paoluzzi, DiCarlo, Furiani, Jirik, CAD models from medical images using LAR, Computer-Aided Design and Applications, 2016

# Terminology

# Medical imaging devices

Radiography using X-rays or gamma rays to view the internal form of an object

Magnetic resonance imaging MRI scanner or NMR (nuclear magnetic resonance) scanner uses powerful magnets to polarize and excite hydrogen nuclei (i.e., single protons) of water molecules\* in human tissue

Nuclear medicine Gamma cameras and PET scanners are used (e.g. scintigraphy) to detect regions of biologic activity that may be associated with a disease

Ultrasound uses high frequency broadband sound waves in the megahertz range, that are reflected by tissue to varying degrees to produce (up to 3D) images

Elastography Elastography maps the elastic properties of soft tissue, as elasticity can discern healthy from unhealthy tissue for specific organs/growths.

Tomography X-ray computed tomography (CT), or Computed Axial Tomography (CAT) scans produces a 2D image of the structures in a thin section of the body. Positron emission tomography (PET) also used in conjunction with computed tomography, PET-CT, and magnetic resonance imaging PET-MRI.

Echocardiography When ultrasound is used to image the heart it is referred to as an echocardiogram. Echocardiography allows detailed structures of the heart

# Medical imaging

Digital Imaging and Communications in Medicine (DICOM) is the standard for the communication and management of medical imaging information and related data.



# Medical imaging standard

Medical imaging creates **visual representations** of the **interior of a body** for clinical analysis and medical intervention, **as well as visual representation of the function** of some organs or tissues (physiology)



*Digital Imaging and Communications in Medicine*

ABOUT DICOM®

STANDARD

ACTIVITY

USING DICOM®

RESOURCES

CONFERENCES



*The Association of Electrical Equipment  
and Medical Imaging Manufacturers*

Standards  
STORE



Public Policy

Member Products

NEMA/BIS

Standards

Communications

About Us

Join

Standards

All Standards

All Standards by Product

About NEMA Standards

## Medical Imaging Standards

NEMA > Standards

MEDICAL IMAGING

[View all products](#)



Search Standards

Search



Related

# DICOM Data format

DICOM® — Digital Imaging and Communications in Medicine — is the international standard for medical images and related information

- It defines the formats for medical images that can be exchanged with the data and quality necessary for clinical use

DICOM groups information into data sets

- The file of a chest x-ray image, for example, actually contains the patient ID within the file, so that the image can never be separated from this information by mistake
- This is similar to the way that image formats such as JPEG can also have embedded tags to identify and otherwise describe the image

# DICOM Data object

A **DICOM data object** consists of a **number of attributes**, including **items** such as **name**, **ID**, etc., and also one special attribute containing the image pixel data (i.e

- The **main object** has no “header” as such, being merely a **list of attributes**, including the pixel data
- A single **DICOM object** can have only one **attribute** containing **pixel data**
- For many modalities, **this corresponds** to a **single image**
- The attribute **may contain** multiple “frames”, allowing storage of cine loops or other multi-frame data
- Another example is **NM data**, where an NM image, **by definition**, is a **multi-dimensional multi-frame image**
- Three- or four-dimensional data can be encapsulated in a **single DICOM object**
- Pixel data can be **compressed** using a variety of standards, including **JPEG**, **Lossless JPEG**, **JPEG 2000**, and **RLE** (Run-length encoding)
- **LZW (zip) compression** can be used for the **whole data set** (not just the pixel data), but this has rarely been implemented.

# Computational pipeline

# LAR-surf terminology

*dD Image* ( $d \in \{2, 3\}$ )

*d-Block or Brick (cuboidal portion of Image)* 3D image portions as 3-cell complexes: (a) image portion seen exploded; (b) divide et impera paradigm; (c) surface assembling by removal of double 2-cells, via a sort-based MapReduce algorithm

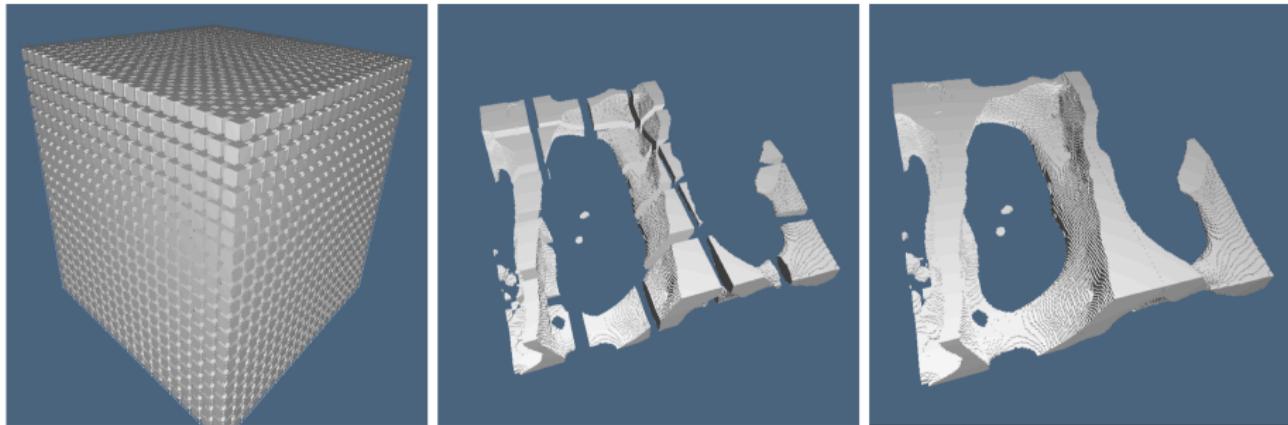


Figure 9: 3-Block or 3-brick of image

# *d*-Block or Brick

$B$  is a function of its element of the lowest block coordinates  $i, j, k \in \mathbb{N}$  and of block dimension  $n$ :  $[B(i,j,k,n) := \text{Image}([in:in+n, jn:jn+n, kn:kn+n])]$

block sides may not correspond to image edges

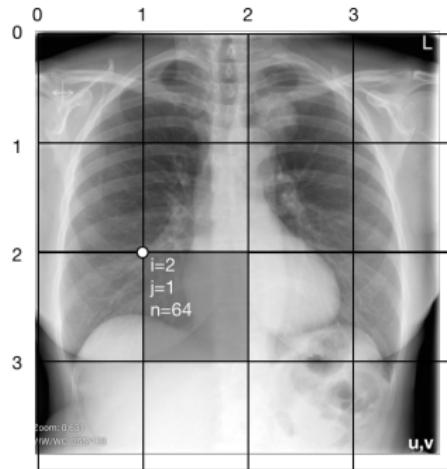


Figure 10: A possible **block partition** of a **radiologic image**. The evidenced 2D block, of size  $n^d = 64^2$ , is sliced by  $B([2, 1, 64]) = \text{Image}([128 : 172], [64 : 128])$

## Block chain ;-)

Map each image subspace  $B(i, j, n)$  to the linear **chain space  $C_2$**  of **dimension  $n \times n$** , using coordinate vectors  $c_{h,k} \in \mathbb{B}^{n \times n} := \{0, 1\}^{n \times n}$ , where the basis element  $c = c_{h,k} \in C_2$  is mapped via **Cartesian-to-linear map** to the Boolean vector

$$\text{Image}(h, k) \mapsto c_{h,k} := [0 \cdots 0 \ 1 \ 0 \cdots 0]^t \in \mathbb{B}^{n \times n}$$

for each  $0 \leq h, k \leq n$ , and where the (single) unit cell in position  $nk + h \leq n \times n$ .

Each **pixel** (or voxel) in a block is seen as a **basis vector** in  $C_2$  (or  $C_3$ ), and each **subset of image elements**, as a **vector  $c \in C_2$** , with many ones as the **cardinality** of the subset

Each  **$B(i, j, n)$  box** maps to the unit vector  $[1 \cdots 1 \cdots 1]^t \in \mathbb{B}^{n^d}$

# Number of non-zeros in boundary matrix $[\partial_d]_{n^d}$

It is easy to see that the operator's matrix  $[\partial_d]$  is very sparse

- $[\partial_d]_{n^d}$  contains  $2d$  non-zero elements (ones) for each column, i.e.~4 ones and 6 ones for 2D and 3D, respectively
- the number of columns is  $n^d$
- the number of rows is the number of 2-cells in  $B(n^d)$

## Question

How to compute the dimension (cardinality of basis) of  $C_{d-1}(B(n^d))$  ?

## Sparsity of boundary matrix $[\partial_d]$

2d non-zero elements for each column, hence their total number is  $2d n^d$ .

The number of matrix element is  $d n(1 + n)^{d-1} \times n^d$ , giving a ratio of:

$$\frac{\text{non-zero elements}}{\text{total elements}} = \frac{2d \times n^d}{d n(1 + n)^{d-1} \times n^d} = \frac{2}{n + n^d}$$

Using sparse matrices in CSC (Compressed Sparse Column) format we get the storage size:

$$mem([\partial_d]_{n^d}) = 2 \times \#\text{ nonzero} + \#\text{columns} = 2 \times 2d n^d + n^d = (4d + 1)n^d$$

# Storage size of boundary matrix $[\partial_d]$

In conclusion:

- for **block size  $n = 64$** ,  $[\partial_d]$  requires for **2D images**  $9 \times 64^2 = 36,864$  memory elements, and for **3D images**  $13 \times 64^3 = 3,407,872$  memory elements
- Counting the bytes for the standard implementation of a **sparse binary matrix** (1 byte for values and 8 bytes for indices) we get:
- $(18d + 8)n^d$  bytes, giving **176,KB** for **2D** and **15.872,MB** for **3D**

## Set the number $n$ of blocks

Let set the size  $n$  of the block, in order to decompose the input  $\text{Image}(u, v, w)$  into a fair number  $M$  of blocks:

$$M = \lceil u/n \rceil \times \lceil v/n \rceil \times \lceil w/n \rceil \approx \frac{uvw}{n^3}$$

Then consider each image portion  $c_{i,j,k} = S \cap B(i, j, k, n)$  and compute its (binary) coordinate representation  $[c]_{i,j,k} \in C_3(n, n, n)$ .

## 2-chains of surface portions

let assemble the  $M$  vectors of 2-chains  $c_{i,j,k}$  of surface portions, into a sparse binary matrix  $\mathbf{S}$ , of dimension  $n^d \times M$

Then compute a matrix  $\mathbf{B}$  of boundary portions of  $S$ , represented by columns as chain coordinate vectors in  $C_2$ :

$$\mathbf{B} = [\partial_3(n)] \mathbf{S},$$

where the boundary matrix has dimension  $d n (1 + n)^{d-1} \times n^d$ .

A final computational step is needed, in order to embed the 2-chains in Euclidean space  $\mathbb{E}^3$  and to assemble the whole resulting surface

# Embedding

Compute the **embedding function**  $\mu : C_0 \rightarrow \mathbb{E}^3$ , where  $C_0$  is the space of 0-chains, **one-to-one with vertices** of the extracted surface

- The **simplest solution** associates **four 0-cells** to **each 2-cell** of the extracted surface, i.e.~to **each non-zero entry** in every column of **B**
- The  **$\mu$  function** is computed by **mapping**, via **element position** in the column, a **triple of integers**  $0 \leq x \leq u$ ,  $0 \leq y \leq v$ , and  $0 \leq z \leq w$  **for each vertex**
- The **mapping** can be implemented with a **dictionary**, that stores the **inverse coordinate transformation** used at the beginning, i.e.~the one from linear to Cartesian coords, in order of not duplicating the output vertices.

# Surface assembling

$$\text{Lar}(X) := (\text{Geom}(X), \text{Top}(X)) = (V, CV)$$

w.r.t.~the **chain complexes**  $C_3 \rightarrow C_2 \rightarrow C_1 \rightarrow C_0$  induced by the input image  $Im$  and by each segment  $S_{i,j,k}$ , we have

$$\text{Geom} := \mu(C_0(S_{i,j,k})) = V, \quad (1)$$

$$\text{Top} := C_3(S) = S \mapsto FW. \quad (2)$$

$$\text{Lar}(B_{i,j,k}) := (\text{Geom}, \text{Top}) = (W, FW), \quad (3)$$

$$\text{Geom} := \mu(C_0(B_{i,j,k})) = W \subset V, \quad (4)$$

$$\text{Top} := C_2(B_{i,j,k}) = B_{i,j,k} \mapsto FW \subset FW. \quad (5)$$

A **translation transformation** applied to each **vertex subset**  $W_{i,j,k}$  with translation vector  $\mathbf{t} = [i, j, k]$  will therefore move it in the **final space position**:

$$\text{Lar}(B) = \bigoplus_{i,j,k} \text{Lar}(\partial_3 S_{i,j,k}) = \bigoplus_{i,j,k} (W, FW).$$

# Block-level parallelism

Several steps can be performed in parallel at **image-block level**, depending on the **embarassingly data parallel** nature of the problem

- In particular, little effort is needed to **decompose the problem** into a number of **parallel tasks**  $S_{i,j,k}$ , using **multiarray slicing**
- The **granularity of parallelism**, depending on the block size  $n$ , is further enforced by the computation of a **single boundary matrix**  $[\partial_d(n)]$ , in turn depending on  $n$
- The initial cost of **broadcasting the matrix** to nodes can be carefully controlled, and **finely tuned** for the **system architecture**
- The approach is appropriate for **SIMD hybrid architectures** of CPUs and GPUs, since only the **initial block setup** of boundary matrix and image slices, as well the **final collection** of computed surface portions, **require inter-process communication**.