VOTs: VOlume doTS as a Point-based Primitive for Volume Data

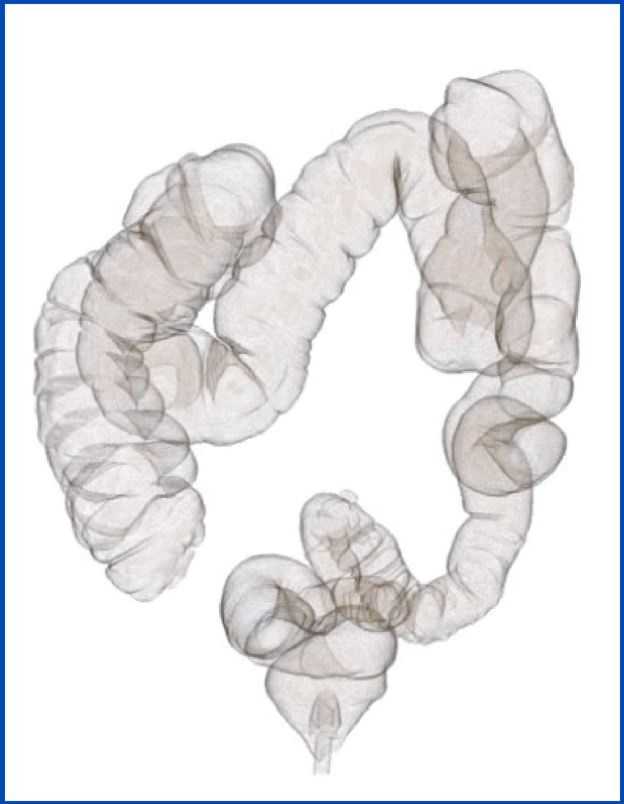
- **Sören Grimm** – dissertation:
  - Real-Time Mono- and Multi-Volume Rendering of Large Medical Datasets on Standard PC Hardware
  - **Supervisor:** Meister Eduard Gröller
  - **Duration:** October 2002 - April 2005

Motivation to VOTs

- Limitation of grid-based volume graphics:
  - Regular sampling, limited resolution – limited position precision, problem with high frequency details
  - Discrete locations – reconstruction inaccuracies
  - Mixing different grid types – very difficult
  - The size of volumetric data sets – huge, also in cases where only small portion is of interest

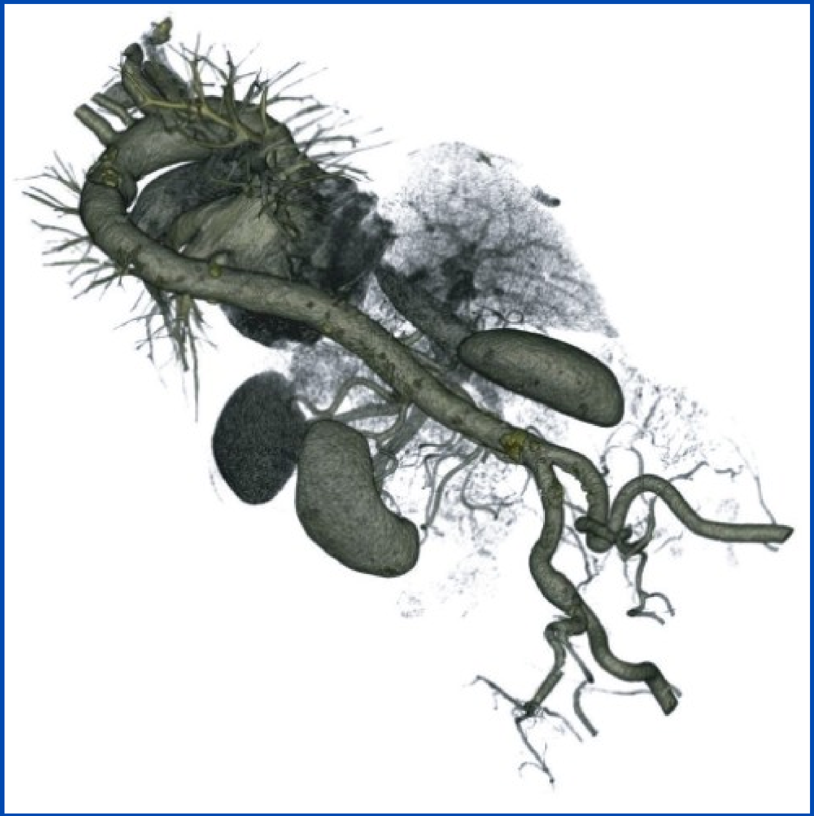
Aim of VOTs Representation

- Volumetric data in which only parts of the volume are of great importance

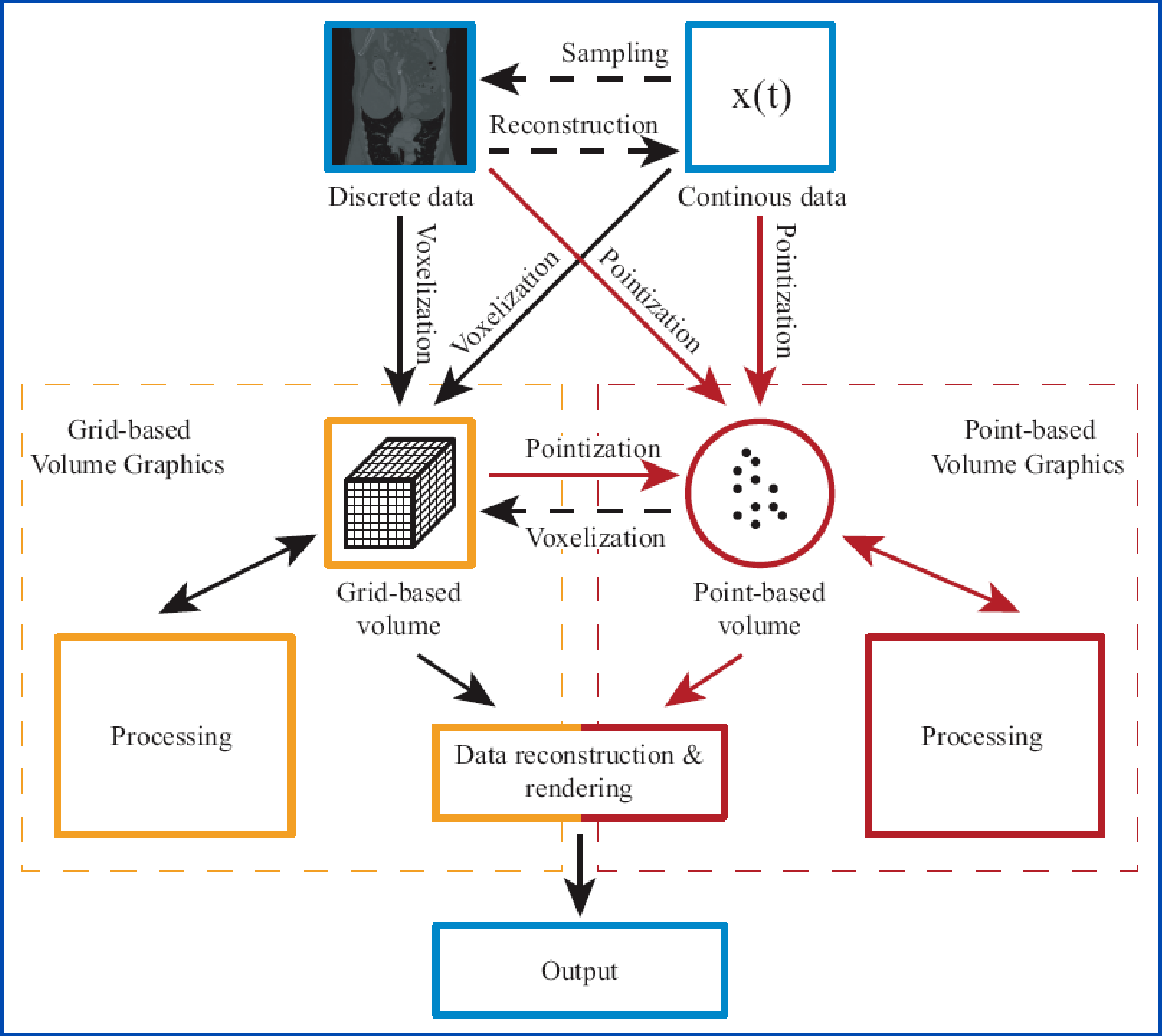


colon

aorta



Grid-Based VG versus Point-Based VG



VOT Data-Structure

- Volumetric data – scalar function: $f: \quad U \subseteq R^3 \rightarrow R$
- Vots – piece-wise representation: $\tilde{f}_i: \quad U_i \subseteq U \rightarrow R$
- Information for local function reconstruction – Vot V_i:
  - Point P_i
  - Function value in P_i and higher order derivatives
  - Validity area U_i

Taylor Series Expansion

- Approximation:
- N = 2 or 3

$$\begin{aligned} \alpha &\in \{x, y, z\}^* \\ \partial^\alpha &= \frac{\partial^{\alpha_x}}{\partial x^{\alpha_x}} \frac{\partial^{\alpha_y}}{\partial y^{\alpha_y}} \frac{\partial^{\alpha_z}}{\partial z^{\alpha_z}} \\ \alpha! &= \alpha_x! \alpha_y! \alpha_z! \\ P^\alpha &= \underbrace{P^x \cdots P^x}_{\alpha_x} \underbrace{P^y \cdots P^y}_{\alpha_y} \underbrace{P^z \cdots P^z}_{\alpha_z} \end{aligned}$$

$$f(P + \Delta P) \approx \tilde{f}(P + \Delta P) = \sum_{|\alpha| \leq N} \frac{1}{\alpha!} \partial^\alpha \tilde{f}(P) \Delta P^\alpha$$

$$\begin{aligned} \nabla \tilde{f}(P) &= \begin{pmatrix} \tilde{f}_x \\ \tilde{f}_y \\ \tilde{f}_z \end{pmatrix} \\ H_{\tilde{f}}(P) &= \begin{pmatrix} \tilde{f}_{xx} & \tilde{f}_{xy} & \tilde{f}_{xz} \\ \tilde{f}_{yx} & \tilde{f}_{yy} & \tilde{f}_{yz} \\ \tilde{f}_{zx} & \tilde{f}_{zy} & \tilde{f}_{zz} \end{pmatrix} \end{aligned}$$

$$T_{\tilde{f}}(P) = \begin{matrix} \begin{pmatrix} \tilde{f}_{xxx} & \tilde{f}_{xyx} & \tilde{f}_{xzx} \\ \tilde{f}_{yxx} & \tilde{f}_{yyx} & \tilde{f}_{yzx} \\ \tilde{f}_{zxx} & \tilde{f}_{zyx} & \tilde{f}_{zzx} \end{pmatrix} \\ \begin{pmatrix} \tilde{f}_{xxy} & \tilde{f}_{xyy} & \tilde{f}_{xzy} \\ \tilde{f}_{yxy} & \tilde{f}_{yyy} & \tilde{f}_{yzy} \\ \tilde{f}_{zxy} & \tilde{f}_{zyy} & \tilde{f}_{zzy} \end{pmatrix} \\ \begin{pmatrix} \tilde{f}_{xxz} & \tilde{f}_{xyz} & \tilde{f}_{xzz} \\ \tilde{f}_{yxz} & \tilde{f}_{yyz} & \tilde{f}_{yzz} \\ \tilde{f}_{zxz} & \tilde{f}_{zyz} & \tilde{f}_{zzz} \end{pmatrix} \end{matrix}$$

Validity Area

- U_i can be of any shape
- Practical – convex shapes: spheres, ellipsoids, boxes,...
- Each validity area is defined in such a way that the function approximation error is smaller than a given tolerance – it can vary according to the importance of the appropriate region

Vot Generation of a Cell (1)

- A cell definition: $\left(P_{ijk}, f_{P_{ijk}}\right)_{i,j,k \in \{0,1\}}$
- Taylor expansion point – center of the cell: $P = \frac{1}{8}\sum P_{ijk}$
- The terms for the Taylor series expansion:

$$
\begin{aligned}
\tilde{f}(P) &= \tfrac{1}{8}\sum f_{P_{ijk}} \\
\nabla \tilde{f}(P) &= \tfrac{1}{4}\begin{pmatrix} \sum_{j,k} f_{P_{1jk}} - \sum_{j,k} f_{P_{0jk}} \\ \sum_{i,k} f_{P_{i1k}} - \sum_{i,k} f_{P_{i0k}} \\ \sum_{i,j} f_{P_{ij1}} - \sum_{i,j} f_{P_{ij0}} \end{pmatrix} \\
H_{\tilde{f}}(P) &= \tfrac{1}{2}\begin{pmatrix} 0 & \tilde{f}_{xy} & \tilde{f}_{xz} \\ \tilde{f}_{xy} & 0 & \tilde{f}_{yz} \\ \tilde{f}_{xz} & \tilde{f}_{yz} & 0 \end{pmatrix} \\
T_{\tilde{f}}(P) &= \tilde{f}_{xyz}
\end{aligned}
$$

$$
\begin{aligned}
\tilde{f}_{xy} &= \sum_{ijk \in \{000,001,110,111\}} f_{P_{ijk}} - \sum_{ijk \in \{010,011,100,101\}} f_{P_{ijk}} \\
\tilde{f}_{xz} &= \sum_{ijk \in \{000,010,101,111\}} f_{P_{ijk}} - \sum_{ijk \in \{001,011,100,110\}} f_{P_{ijk}} \\
\tilde{f}_{yz} &= \sum_{ijk \in \{000,011,100,111\}} f_{P_{ijk}} - \sum_{ijk \in \{001,010,101,110\}} f_{P_{ijk}}
\end{aligned}
$$

$$
\tilde{f}_{xyz} = \sum_{ijk \in \{001,010,100,111\}} f_{P_{ijk}} - \sum_{ijk \in \{000,011,101,110\}} f_{P_{ijk}}
$$

Vot Generation of a Cell (2)

- Vots representation of a cell – 8 values (due to the symmetry or zero values):
  - 1 – function value
  - 3 – gradient
  - 3 – Hessian
  - 1 – third partial derivatives
- Only every other cell must be represented by a Vot
- Equal memory demands, but more intuitive representation

General Vot Generation

- Aim: generate a Vot for a given set of *m* scattered data points Q_j with function values f_{Q_j}
- Minimization of the mean square error: $E(\cdots) = \sum_{j=1}^{m} (\tilde{f}(Q_j) - f_{Q_j})^2$
- Expansion point – center of gravity: $P = \frac{1}{m} \sum Q_j$
- Unknown variables (20 for N=3): $\tilde{f}, \tilde{f}_i, \tilde{f}_{ij}, \tilde{f}_{ijk}$
- Partial derivatives of *E* with respect to the unknowns are taken and set to zero – system of 20 linear equations

Vot-Space

- Basic question: Given an arbitrary point *P* find the corresponding Vot V_i so that the function value at position *P* can be determined.
- Efficient indexing structures (range trees, interval trees, octrees,...) – depends on the shape of the validity area and on the application
- Vot space:
  - Set of Vots
  - Set of indexing structures (at least unsorted list)

From Grid to Vots

- Input: rectilinear grid with grid points P_i and values f_{P_i}
- Output: small number of Vots V_j which completely cover the underlying volumetric data
- Algorithm:

  1) Find the largest possible Vot for every grid position with a given accuracy
     - Validity area: grid aligned box

  2) Find a minimal subset of all possible Vots which covers the whole input grid

From Grid to Vots – Step 1

- Input: a position in the given grid
- Output: largest possible Vot covering this position with a given accuracy
- Growing process for the validity area (grid aligned box):
  - Initially: cell-sized
  - Iteratively: increase it in one direction (± x, y, z) and compute the error for the given Vot defined by the current point set *T* (consists of grid points inside a box)
  - Error evaluation: $\mathcal{F}(T) = max_j |\tilde{f}(Q_j) - f_{Q_j}|, Q_j \in T$

From Grid to Vots – Step 2

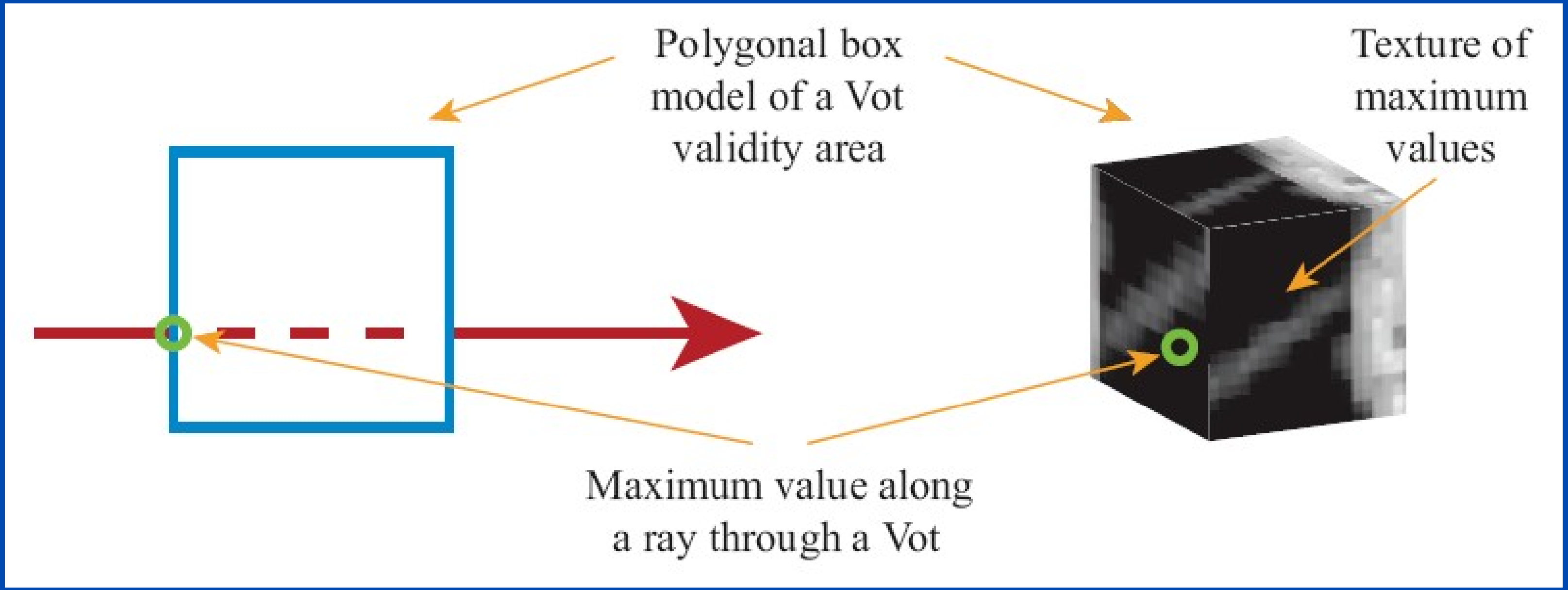
- Input: a set of largest possible Vots for every grid position
- Output: minimal subset of these Vots which covers the whole volume
- Algorithm:
  - Vots are weighted – initially according to the size of val. area
  - Iteratively:
    - Vot with the largest weight is included in the minimal subset
    - Weights of other Vots are adapted (uncovered grid points are counted)
    - ...until all grid points are covered

Maximum Intensity Projection (1)

- Question: Given a Vot *V* and a viewing direction *r*, how to compute the maximum along this ray?
- Taking *V* and *r* an 1-dimensional function *F* is determined – restriction of the Vot function to the ray line
  - Set the first derivative of *F* to zero – find extremes
  - Check the sign of the second derivative of *F* – distinguish between minimum and maximum
  - Maximum is found either within the validity area using derivatives of *F* or it occurs at one of the intersection points (validity area box and the ray)

Maximum Intensity Projection (2)

- Validity area of each Vot – polygonal model
- Intersection point is calculated and maximum along the ray computed and stored in an image (one image for each visible face) – images are textured onto the corresponding faces



Results

- MIP

- Number of Vots for different error bounds (data range: [0, 4095])

Error ε:	0.4096	4.096	40.96	409.6
(a) # Vots:	570 690	538 919	333 650	35 079
(b) # Vots:	114 665	114 665	112 604	60 353

Issues to be Addressed

- High computational complexity of Vots generation – need for more sophisticated algorithms
- Discontinuity at the borders of Vots
- Evaluation of error function only in sample points
- New rendering approach competitive with conventional volume rendering approaches
- Vots construction from other types of data structures (point clouds, unstructured grids, curvilinear grids,...)

Conclusion

- Vots – a novel primitive for volumetric data modeling, processing and rendering
- More intuitive and constructive representation of the data
- Size (and number) of Vots can be adjusted by setting of the appropriate error bound
- User-centric importance sampling – a few Vots for unimportant regions, many Vots for the important ones

Thank you for attention!