Volumetric Image Visualization

Alexandre Xavier Falcão

LIDS - Institute of Computing - UNICAMP

afalcao@ic.unicamp.br

- In the previous lecture, we learned that for a set $\mathcal{P}$ of points p_k, $k \in [1, n]$, in the scene region, the intensities $I(p_k)$ must be obtained by interpolation.

- In the previous lecture, we learned that for a set $\mathcal{P}$ of points p_k, $k \in [1, n]$, in the scene region, the intensities $I(p_k)$ must be obtained by interpolation.

- Today, we will learn how to create sequences of cuts (new slices) in the scene for a given viewing direction and orientation $\mathbf{n}'$, which also requires interpolation.

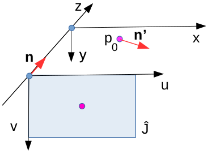
Planar cuts, reslicing, and interpolation

- In the previous lecture, we learned that for a set $\mathcal{P}$ of points p_k, $k \in [1, n]$, in the scene region, the intensities $I(p_k)$ must be obtained by interpolation.
- Today, we will learn how to create sequences of cuts (new slices) in the scene for a given viewing direction and orientation $\mathbf{n}'$, which also requires interpolation.
- We will then learn how to estimate the intensity at any point inside the scene region from the intensities of the nearby spels — i.e., the interpolation operation.

Planar cuts and reslicing

In order to obtain a planar cut (slice) at a point p_0 inside the scene, such that $\mathbf{n}'$ is the ortogonal vector to the cut, one needs to apply the following transformations:

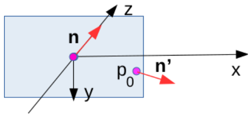
- Translate the viewing (cut) plane (i.e., center of $\hat{J}$) to the origin of the (x, y, z) coordinate system.
- Align the $\mathbf{n}$ vector to the desired viewing vector $\mathbf{n}'$ — this is the inverse of the transformation that aligns an arbitrary vector to the z-axis.
- Translate the rotated viewing plane to the desired location p_0.



Planar cuts and reslicing

In order to obtain a planar cut (slice) at a point p_0 inside the scene, such that $\mathbf{n}'$ is the ortogonal vector to the cut, one needs to apply the following transformations:

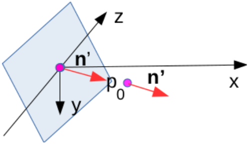
- Translate the viewing (cut) plane (i.e., center of $\hat{J}$) to the origin of the (x, y, z) coordinate system.
- Align the $\mathbf{n}$ vector to the desired viewing vector $\mathbf{n}'$ — this is the inverse of the transformation that aligns an arbitrary vector to the z-axis.
- Translate the rotated viewing plane to the desired location p_0.



Planar cuts and reslicing

In order to obtain a planar cut (slice) at a point p_0 inside the scene, such that $\mathbf{n}'$ is the ortogonal vector to the cut, one needs to apply the following transformations:

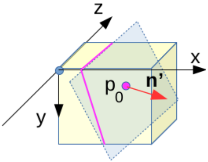
- Translate the viewing (cut) plane (i.e., center of $\hat{J}$) to the origin of the (x, y, z) coordinate system.
- Align the $\mathbf{n}$ vector to the desired viewing vector $\mathbf{n}'$ — this is the inverse of the transformation that aligns an arbitrary vector to the z-axis.
- Translate the rotated viewing plane to the desired location p_0.



Planar cuts and reslicing

In order to obtain a planar cut (slice) at a point p_0 inside the scene, such that $\mathbf{n}'$ is the ortogonal vector to the cut, one needs to apply the following transformations:

- Translate the viewing (cut) plane (i.e., center of $\hat{J}$) to the origin of the (x, y, z) coordinate system.
- Align the $\mathbf{n}$ vector to the desired viewing vector $\mathbf{n}'$ — this is the inverse of the transformation that aligns an arbitrary vector to the z-axis.
- Translate the rotated viewing plane to the desired location p_0.



Therefore, for every $p \in D_J$, we must apply the transformation Ψ below, being $\Psi_r = \mathbf{R}_x(-\alpha)\mathbf{R}_y(\beta)$ the alignment of $\mathbf{n}$ with $\mathbf{n}'$:

$$\begin{bmatrix} x_{p'} \\ y_{p'} \\ z_{p'} \\ 1 \end{bmatrix} = \mathbf{T}(+x_{p_0}, +y_{p_0}, +z_{p_0})\Psi_r\mathbf{T}(\frac{-d}{2}, \frac{-d}{2}, \frac{d}{2}) \begin{bmatrix} u_p \\ v_p \\ \frac{-d}{2} \\ 1 \end{bmatrix}$$

- The coordinates $p' = (x_{p'}, y_{p'}, z_{p'})$ may not be integers, which requires interpolation to assign $J(p) \leftarrow I(p')$.
- For reslicing a scene from p_0 with spacement $\lambda > 0$ between slices, one can move the cut plane to $p_k = p_{k-1} + \lambda\mathbf{n}'$, $k = 1, 2, \ldots, n-1$.

Algorithm to get a slice given p_0 and $\mathbf{n}'$

Input: Scene $\hat{I} = (D_I, I)$, p_0, and $\mathbf{n}'$.
Output: Image $\hat{J} = (D_J, J)$ of the slice.

1 Compute Ψ and Ψ_r from p_0, $\mathbf{n} = (0, 0, 1, 0)$, and $\mathbf{n}'$.
2 For each $p \in D_J$ do
3 $\quad$ $p' \leftarrow \Psi(p)$.
4 $\quad$ If $(\lceil x_{p'} \rceil, \lceil y_{p'} \rceil, \lceil z_{p'} \rceil) \in D_I$ then
5 $\quad\quad$ $J(p) \leftarrow I(p')$ using interpolation.

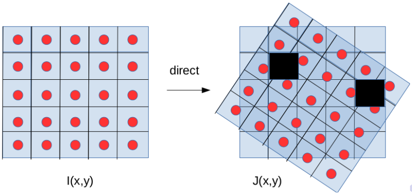
Reslicing algorithm

Input: Scene $\hat{I} = (D_I, I)$, p_0, $\mathbf{n}'$, λ, and number n of slices.
Output: Scene $\hat{J} = (D_J, J)$, with axial slices $J_0, J_1, \ldots, J_{n-1}$.

1 $J_0 \leftarrow GetSlice(\hat{I}, p_0, \mathbf{n}')$.
2 For $k = 1$ to $n-1$ do
3 $\quad p_k \leftarrow p_{k-1} + \lambda\mathbf{n}'$.
4 $\quad J_k \leftarrow GetSlice(\hat{I}, p_k, \mathbf{n}')$

Interpolation

- Interpolation estimates the intensity $I(p')$ at a point $p' = (x_{p'}, y_{p'}, z_{p'})$ with real coordinates inside a scene by using the spel values $I(q_k)$, $k = 1, 2, \ldots, |\mathcal{A}(p')|$, where $q_k \in \mathcal{A}(p') \subset D_I$ and $\mathcal{A}(p')$ is a set of nearby spels of p'.
- Interpolation is required when geometric transformations Φ on spels of an image $\hat{I} = (D_I, I)$ create an image $\hat{J} = (D_J, J)$.
- In order to avoid holes (i.e., unfilled spels in $\hat{J}$), one must apply Φ^{-1} on each spel $p \in D_J$ to obtain $p' = \Phi^{-1}(p)$, with real coordinates for interpolation $J(p) \leftarrow I(p')$.

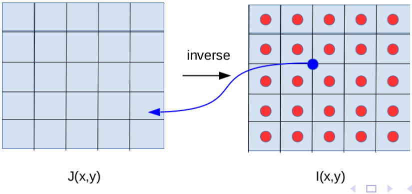


Interpolation

- Interpolation estimates the intensity $I(p')$ at a point $p' = (x_{p'}, y_{p'}, z_{p'})$ with real coordinates inside a scene by using the spel values $I(q_k)$, $k = 1, 2, \ldots, |\mathcal{A}(p')|$, where $q_k \in \mathcal{A}(p') \subset D_I$ and $\mathcal{A}(p')$ is a set of nearby spels of p'.

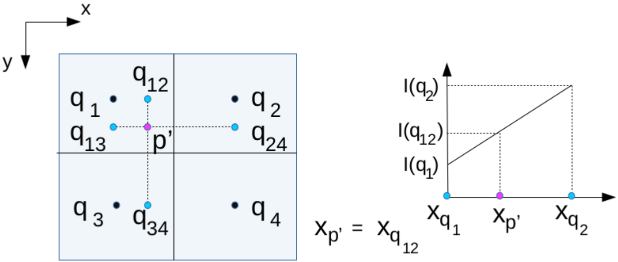
- Interpolation is required when geometric transformations Φ on spels of an image $\hat{I} = (D_I, I)$ create an image $\hat{J} = (D_J, J)$.

- In order to avoid holes (i.e., unfilled spels in $\hat{J}$), one must apply Φ^{-1} on each spel $p \in D_J$ to obtain $p' = \Phi^{-1}(p)$, with real coordinates for interpolation $J(p) \leftarrow I(p')$.



Bilinear interpolation

In 2D, the nearby spels $q_k \in \mathcal{A}(p') \subset D_I$, $k = 1, 2, 3, 4$, may be obtained as $q_1 = (\lfloor x_{p'} \rfloor, \lfloor y_{p'} \rfloor)$, $q_2 = (\lfloor x_{p'} \rfloor + 1, \lfloor y_{p'} \rfloor)$, $q_3 = (\lfloor x_{p'} \rfloor, \lfloor y_{p'} \rfloor + 1)$, and $q_4 = (\lfloor x_{p'} \rfloor + 1, \lfloor y_{p'} \rfloor + 1)$.



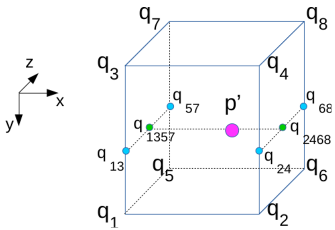
$$
\begin{aligned}
I(p') &= (y_{p'} - y_{q_{12}})I(q_{34}) + (y_{q_{34}} - y_{p'})I(q_{12}) \\
I(q_{12}) &= (x_{p'} - x_{q_1})I(q_2) + (x_{q_2} - x_{p'})I(q_1) \\
I(q_{34}) &= (x_{p'} - x_{q_3})I(q_4) + (x_{q_4} - x_{p'})I(q_3)
\end{aligned}
$$

Bilinear interpolation

Scaling and rotation around the center of the image and axis z.

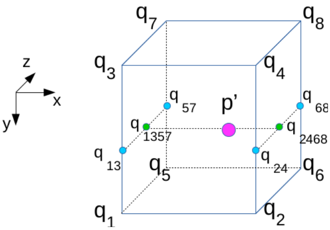
Trilinear interpolation

In 3D, the nearby spels $q_k \in \mathcal{A}(p') \subset D_I$, $k = 1, 2, \ldots, 8$, are obtained similarly, based on the floor operations on the $x_{p'}$, $y_{p'}$, and $z_{p'}$ real coordinates of p'.



$$
\begin{aligned}
I(p') &= (x_{p'} - x_{q_{1357}})I(q_{2468}) + (x_{q_{2468}} - x_{p'})I(q_{1357}) \\
I(q_{2468}) &= (z_{p'} - z_{q_{24}})I(q_{68}) + (z_{q_{68}} - z_{p'})I(q_{24}) \\
I(q_{1357}) &= (z_{p'} - z_{q_{13}})I(q_{57}) + (z_{q_{57}} - z_{p'})I(q_{13})
\end{aligned}
$$

Trilinear interpolation



$$
\begin{aligned}
I(q_{24}) &= (y_{p'} - y_{q_4})I(q_2) + (y_{q_2} - y_{p'})I(q_4) \\
I(q_{68}) &= (y_{p'} - y_{q_8})I(q_6) + (y_{q_6} - y_{p'})I(q_8) \\
I(q_{13}) &= (y_{p'} - y_{q_3})I(q_1) + (y_{q_1} - y_{p'})I(q_3) \\
I(q_{57}) &= (y_{p'} - y_{q_7})I(q_5) + (y_{q_5} - y_{p'})I(q_7)
\end{aligned}
$$

Reformatting a scene

Therefore, a scene $\hat{I} = (D_I, I)$ with spels sizes (d_x, d_y, d_z) can be reformatted into a scene $\hat{J} = (D_J, J)$ with spels sizes (d'_x, d'_y, d'_z) by scaling D_J, with factors $s_x = \frac{d_x}{d'_x}$, $s_y = \frac{d_y}{d'_y}$, and $s_z = \frac{d_z}{d'_z}$, and determining the intensities $J(p)$ via 3D interpolation in $\hat{I} = (D_I, I)$.

Reformatting a scene

Therefore, a scene $\hat{I} = (D_I, I)$ with spels sizes (d_x, d_y, d_z) can be reformatted into a scene $\hat{J} = (D_J, J)$ with spels sizes (d'_x, d'_y, d'_z) by scaling D_J, with factors $s_x = \frac{d_x}{d'_x}$, $s_y = \frac{d_y}{d'_y}$, and $s_z = \frac{d_z}{d'_z}$, and determining the intensities $J(p)$ via 3D interpolation in $\hat{I} = (D_I, I)$. However, it is more efficient to

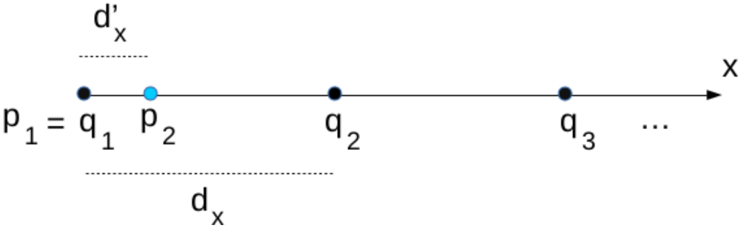
- interpolate $\hat{I} = (D_I, I)$ along x, generating $\hat{I}' = (D_{I'}, I')$ with spels sizes (d'_x, d_y, d_z),
- then interpolate $\hat{I}' = (D_{I'}, I')$ along y, generating $\hat{I}'' = (D_{I''}, I'')$ with spels sizes (d'_x, d'_y, d_z), and
- finally interpolate $\hat{I}'' = (D_{I''}, I'')$ along z, generating $\hat{J} = (D_J, J)$ with spels sizes (d'_x, d'_y, d'_z).

Reformatting a scene

- For instance, let $q_1, q_2, \ldots, q_{n_x}$ be the voxels along the x direction of $\hat{I} = (D_I, I)$ for a fixed row y and a fixed slice z.

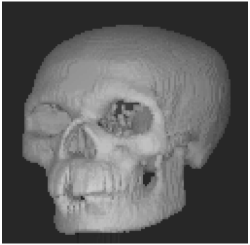
- By starting at $p_1 = q_1 = (x_{q_1}, y, z) \in D_I$ and adding d'_x to generate the subsequent voxels $p_{k+1} = (x_{p_1} + kd'_x, y, z) \in D_{I'}$, $k = 1, 2, \ldots, n'_x - 1$, the intensities $I'(p_{k+1})$ of the subsequent voxels $p_{k+1} \in D_{I'}$ can be found by linear interpolation.

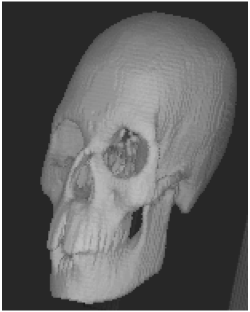
$$I'(p_{k+1}) = (x_{p_{k+1}} - x_{q_k})I(q_{k+1}) + (x_{q_{k+1}} - x_{p_{k+1}})I(q_k).$$

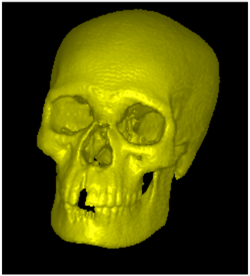


Effects of reformatting

A scene should be reformatted to become isotropic — i.e., with spels of equal sizes in all axes. This is paramount for rendering and segmentation.







Axial slices from bottom to top: $d'_x = d'_y < d'_z$ (left), $d'_x = d'_y > d'_z$ (center), and $d'_x = d'_y = d'_z$ (right, isotropic).