

Transformações Geométricas e Interpolação Linear

Alexandre Xavier Falcão

Instituto de Computação - UNICAMP

afalcao@ic.unicamp.br

- Nesta aula vamos considerar **transformações geométricas** ϕ em um **espaço afim** sobre os spels $p \in D_I$ de uma imagem $\hat{I} = (D_I, \vec{I})$.

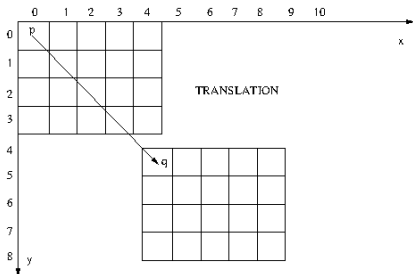
- Nesta aula vamos considerar **transformações geométricas** ϕ em um **espaço afim** sobre os pixels $p \in D_I$ de uma imagem $\hat{I} = (D_I, \vec{I})$.
- O espaço afim é uma generalização do espaço Euclidiano, que inclui pontos, vetores, e certas operações entre eles, tais como adição e multiplicação por um escalar.

- Nesta aula vamos considerar **transformações geométricas** ϕ em um **espaço afim** sobre os pixels $p \in D_I$ de uma imagem $\hat{I} = (D_I, \vec{I})$.
- O espaço afim é uma generalização do espaço Euclidiano, que inclui pontos, vetores, e certas operações entre eles, tais como adição e multiplicação por um escalar.
- Uma transformação geométrica mapeia um ponto (ou vetor) em um outro ponto (ou vetor) do espaço afim (e.g., translação, rotação, escalamento, e projeção).

- Uma transformação ϕ (e.g., translação) sobre um spel $p \in D_I \subset Z^n \subset \mathfrak{R}^n$ gera um novo ponto $q = \phi(p) \in \mathfrak{R}^n$.

Introdução

- Uma transformação ϕ (e.g., translação) sobre um spel $p \in D_I \subset Z^n \subset \mathbb{R}^n$ gera um novo ponto $q = \phi(p) \in \mathbb{R}^n$.
- Se aplicarmos ϕ para todo spel em D_I , a imagem $\hat{I} = (D_I, \vec{I})$ será mapeada em um **domínio real** $D_J \subset \mathbb{R}^n$, gerando $\hat{J} = (D_J, \vec{J})$ tal que para todo $q \in D_J$ existe um $p = \phi^{-1}(q) \in D_I \subset Z^n$ cujo $\vec{I}(p) = \vec{J}(q)$.

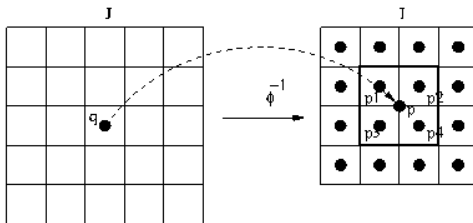
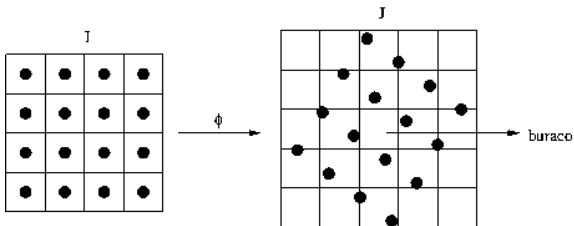


- Para obtermos uma imagem $\hat{J} = (D_J, \vec{J})$ com **domínio inteiro** $D_J \subset Z^n$, devemos aplicar a inversa $\phi^{-1}(q)$ em todo $q \in D_J$, obtendo $p = \phi^{-1}(q) \in \mathfrak{R}^n$.

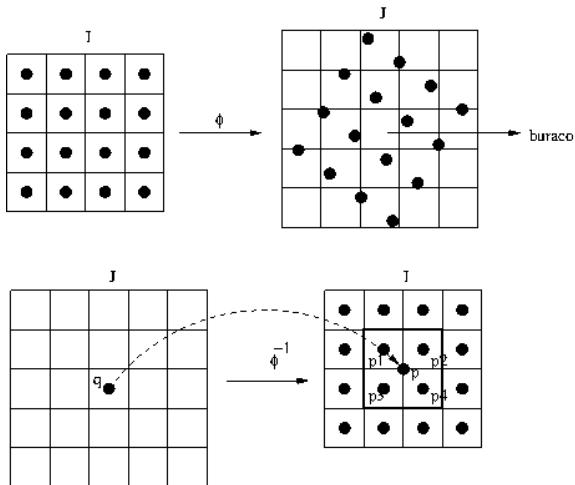
- Para obtermos uma imagem $\hat{J} = (D_J, \vec{J})$ com **domínio inteiro** $D_J \subset Z^n$, devemos aplicar a inversa $\phi^{-1}(q)$ em todo $q \in D_J$, obtendo $p = \phi^{-1}(q) \in \mathfrak{R}^n$.
- Depois o valor $\vec{J}(q) = \vec{I}(p)$ é obtido por **interpolação** dos valores conhecidos $\vec{I}(p_i)$ para $p_i \in D_I \subset Z^n$ em uma dada adjacência $\mathcal{A}(p) \subset D_I$.

- Para obtermos uma imagem $\hat{J} = (D_J, \vec{J})$ com **domínio inteiro** $D_J \subset Z^n$, devemos aplicar a inversa $\phi^{-1}(q)$ em todo $q \in D_J$, obtendo $p = \phi^{-1}(q) \in \mathbb{R}^n$.
- Depois o valor $\vec{J}(q) = \vec{I}(p)$ é obtido por **interpolação** dos valores conhecidos $\vec{I}(p_i)$ para $p_i \in D_I \subset Z^n$ em uma dada adjacência $\mathcal{A}(p) \subset D_I$.
- Esta estratégia também evita a formação de “buracos” na imagem transformada, já que seu domínio deve ser inteiro.

Introdução



Introdução



A adjacência \mathcal{A} é definida pelo piso e teto das coordenadas reais de $p \in D_I$.

Nesta aula iremos estudar as seguintes transformações geométricas:

Nesta aula iremos estudar as seguintes transformações geométricas:

- Translação e escalamento.

Nesta aula iremos estudar as seguintes transformações geométricas:

- Translação e escalamento.
- Rotação em torno da origem e eixo principal (x , y , ou z).

Nesta aula iremos estudar as seguintes transformações geométricas:

- Translação e escalamento.
- Rotação em torno da origem e eixo principal (x , y , ou z).
- Rotação em torno de ponto e eixo arbitrários.

Nesta aula iremos estudar as seguintes transformações geométricas:

- Translação e escalamento.
- Rotação em torno da origem e eixo principal (x , y , ou z).
- Rotação em torno de ponto e eixo arbitrários.

Ao aplicar essas transformações em uma imagem $\hat{I} = (D_I, \vec{I})$, vamos adotar a interpolação linear.

Seja $q = \phi(p) = (x_q, y_q, z_q)$ o ponto $p = (x_p, y_p, z_p)$ transladado do vetor $\vec{t} = (t_x, t_y, t_z)$. Temos que:

$$\begin{bmatrix} x_q \\ y_q \\ z_q \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix}$$

Se representarmos os pontos p e q em **coordenadas homogêneas**, $p = (x_p, y_p, z_p, 1)$ e $q = (x_q, y_q, z_q, 1)$, a translação passa a ser multiplicativa e pode ser facilmente combinada com as demais transformações geométricas, que são multiplicativas.

Se representarmos os pontos p e q em **coordenadas homogêneas**, $p = (x_p, y_p, z_p, 1)$ e $q = (x_q, y_q, z_q, 1)$, a translação passa a ser multiplicativa e pode ser facilmente combinada com as demais transformações geométricas, que são multiplicativas.

$$\begin{bmatrix} x_q \\ y_q \\ z_q \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}$$

Para aplicar a inversa, basta transladar de $(-t_x, -t_y, -t_z, 1)$.

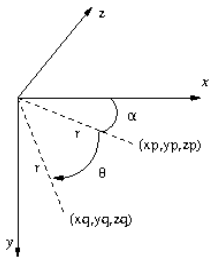
Fatores s_x , s_y e s_z podem ser aplicados às coordenadas dos pontos para aumentar/reduzir o tamanho de um objeto (imagem), ou refletí-lo em relação a um dos planos de coordenadas.

$$\begin{bmatrix} x_q \\ y_q \\ z_q \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}$$

Fatores maiores que 0 e menores que 1 ocasionam redução de tamanho, fatores maiores que 1 ocasionam aumento, e fatores menores que 0 ocasionam reflexão. A inversa $\mathbf{S}^{-1}(s_x, s_y, s_z)$ é $\mathbf{S}(1/s_x, 1/s_y, 1/s_z)$.

Rotação em torno da origem e eixo z

Seja $\vec{V} = (0, 0, 1, 1)$ o vetor que representa o eixo z com origem em $(0, 0, 0, 1)$, a rotação em torno da origem e eixo z



modifica apenas as coordenadas x e y dos pontos, seguindo a regra da mão direita (polegar direito na direção e sentido de \vec{V} , e os demais dedos girando para dentro da mão.).

Rotação em torno da origem e eixo z

Esta rotação é representada por uma matriz $\mathbf{R}_z(\theta)$ obtida das equações abaixo.

$$x_p = r \cos(\alpha)$$

$$y_p = r \sin(\alpha)$$

$$x_q = r \cos(\theta + \alpha)$$

$$x_q = r \cos(\alpha) \cos(\theta) - r \sin(\alpha) \sin(\theta)$$

$$x_q = x_p \cos(\theta) - y_p \sin(\theta)$$

$$y_q = r \sin(\theta + \alpha)$$

$$y_q = r \cos(\alpha) \sin(\theta) + r \sin(\alpha) \cos(\theta)$$

$$y_q = x_p \sin(\theta) + y_p \cos(\theta)$$

$$z_q = z_p$$

Rotação em torno da origem e eixo z

$$\begin{bmatrix} x_q \\ y_q \\ z_q \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}$$

Rotações em torno da origem e eixos x e y

As rotações em torno da origem e eixos x e y são obtidas de forma similar.

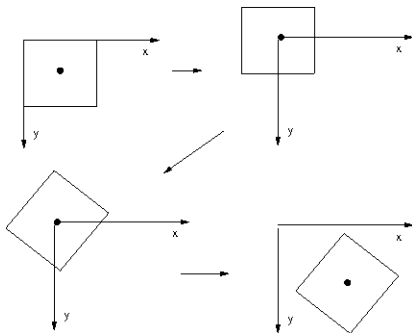
$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note apenas que o sinal dos senos está trocado em $\mathbf{R}_y(\theta)$, porque a rotação é no sentido inverso da regra da mão direita.

Rotação em torno de um ponto arbitrário e eixos x , y , ou z

Desejamos normalmente rotacionar o objeto (imagem) em torno do seu centro geométrico. Neste caso, o objeto (imagem) deve ser transladado para que seu centro geométrico fique na origem do sistema de coordenadas. Após aplicar a rotação, transladamos de volta o objeto (imagem) evitando cortes de cena.



Rotação em torno de um ponto arbitrário e eixos x , y , ou z

Por exemplo, a rotação $\mathbf{R}_x(\theta)$ de um ângulo θ em torno do vetor $\vec{V} = (1, 0, 0, 1)$ e do centro geométrico $(x_c, y_c, z_c, 1)$ de um objeto (imagem) é dada por

$$\begin{bmatrix} 1 & 0 & 0 & d/2 \\ 0 & 1 & 0 & d/2 \\ 0 & 0 & 1 & d/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_c \\ 0 & 1 & 0 & -y_c \\ 0 & 0 & 1 & -z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

onde d é a diagonal do objeto (imagem).

Rotação em torno de ponto e eixos arbitrários

- Suponha agora uma rotação de ângulo θ em torno de um ponto $p_0 = (x_0, y_0, z_0, 1)$ e vetor $\vec{V} = (V_x, V_y, V_z, 1)$, com $|\vec{V}| = 1$.

Rotação em torno de ponto e eixos arbitrários

- Suponha agora uma rotação de ângulo θ em torno de um ponto $p_0 = (x_0, y_0, z_0, 1)$ e vetor $\vec{V} = (V_x, V_y, V_z, 1)$, com $|\vec{V}| = 1$.
- Primeiro aplicamos uma translação $\mathbf{T}(-p_0)$ para colocar a origem do sistema no ponto p_0 .

Rotação em torno de ponto e eixos arbitrários

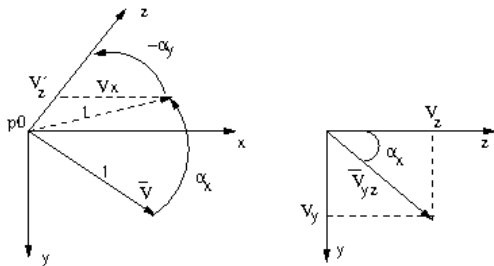
- Suponha agora uma rotação de ângulo θ em torno de um ponto $p_0 = (x_0, y_0, z_0, 1)$ e vetor $\vec{V} = (V_x, V_y, V_z, 1)$, com $|\vec{V}| = 1$.
- Primeiro aplicamos uma translação $\mathbf{T}(-p_0)$ para colocar a origem do sistema no ponto p_0 .
- Depois alinhamos o vetor \vec{V} com um dos eixos (e.g., o eixo z) e rotacionamos de θ em torno deste eixo e origem p_0 .

Rotação em torno de ponto e eixos arbitrários

- Suponha agora uma rotação de ângulo θ em torno de um ponto $p_0 = (x_0, y_0, z_0, 1)$ e vetor $\vec{V} = (V_x, V_y, V_z, 1)$, com $|\vec{V}| = 1$.
- Primeiro aplicamos uma translação $\mathbf{T}(-p_0)$ para colocar a origem do sistema no ponto p_0 .
- Depois alinhamos o vetor \vec{V} com um dos eixos (e.g., o eixo z) e rotacionamos de θ em torno deste eixo e origem p_0 .
- Por fim invertemos o alinhamento e aplicamos a translação $\mathbf{T}(p_0)$ de volta ao sistema original.

Rotação em torno de ponto e eixos arbitrários

Seja z o eixo escolhido, o alinhamento de \vec{V} com o vetor $(0, 0, 1, 1)$ requer rotação em torno do eixo x e depois em torno do eixo y .



A rotação α_x em torno de x até o plano xz , seguida da rotação $-\alpha_y$ em torno de y , alinha o vetor \vec{V} com o eixo z (à esquerda). A projeção \vec{V}_{yz} de \vec{V} no plano yz forma o mesmo ângulo α_x com o eixo z (à direita)

Rotação em torno de ponto e eixos arbitrários

A operação desejada pode ser escrita como:

$$\mathbf{T}(p_0)\mathbf{R}_x^t(\alpha_x)\mathbf{R}_y^t(-\alpha_y)\mathbf{R}_z(\theta)\mathbf{R}_y(-\alpha_y)\mathbf{R}_x(\alpha_x)\mathbf{T}(-p_0)$$

ou ainda, como:

$$\mathbf{T}(p_0)\mathbf{R}_x(-\alpha_x)\mathbf{R}_y(\alpha_y)\mathbf{R}_z(\theta)\mathbf{R}_y(-\alpha_y)\mathbf{R}_x(\alpha_x)\mathbf{T}(-p_0)$$

Lembrando que a inversa $\mathbf{R}^{-1}(\theta)$ de uma matriz rotação $\mathbf{R}(\theta)$ é a **matriz transposta** $\mathbf{R}^t(\theta) = \mathbf{R}(-\theta)$ e a inversa $(\mathbf{R}_1(\theta_1)\mathbf{R}_2(\theta_2))^{-1}$ da multiplicação entre duas matrizes é $\mathbf{R}_2^{-1}(\theta_2)\mathbf{R}_1^{-1}(\theta_1) = \mathbf{R}_2^t(\theta_2)\mathbf{R}_1^t(\theta_1) = \mathbf{R}_2(-\theta_2)\mathbf{R}_1(-\theta_1)$.

Rotação em torno de ponto e eixos arbitrários

A questão é quais são os parâmetros α_x e α_y das matrizes $\mathbf{R}_y(\cdot)$ e $\mathbf{R}_x(\cdot)$? Pela figura temos que:

$$\alpha_x = \tan^{-1} \left(\frac{V_y}{V_z} \right)$$
$$\alpha_y = \tan^{-1} \left(\frac{V_x}{V'_z} \right),$$

onde $V'_z = \frac{V_z}{\cos(\alpha_x)}$.

A questão é quais são os parâmetros α_x e α_y das matrizes $\mathbf{R}_y(\cdot)$ e $\mathbf{R}_x(\cdot)$? Pela figura temos que:

$$\alpha_x = \tan^{-1} \left(\frac{V_y}{V_z} \right)$$
$$\alpha_y = \tan^{-1} \left(\frac{V_x}{V'_z} \right),$$

onde $V'_z = \frac{V_z}{\cos(\alpha_x)}$. Note que, quando $V_z = 0$, as equações acima necessitam de tratamento especial. Isto é, se $V_x = 0$, então $\alpha_y = 0$; se $V_x \neq 0$, então $\alpha_y = 90$; se $V_y = 0$, então $\alpha_x = 0$; e se $V_y \neq 0$, então $\alpha_x = 90$.

Alinhamento com os eixos principais

Uma aplicação bastante útil é o alinhamento de um objeto com seus eixos principais.

- Sejam $\mathcal{X} = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n)\}$ o conjunto das coordenadas dos voxels de um objeto em relação ao seu centro geométrico e Σ a matriz de covariância \mathcal{X} .

Alinhamento com os eixos principais

Uma aplicação bastante útil é o alinhamento de um objeto com seus eixos principais.

- Sejam $\mathcal{X} = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n)\}$ o conjunto das coordenadas dos voxels de um objeto em relação ao seu centro geométrico e Σ a matriz de covariância \mathcal{X} .
- A decomposição $\Sigma = \mathbf{U}\mathbf{S}\mathbf{V}^t$ em valores singulares calcula na primeira coluna de \mathbf{U} o eixo principal do objeto.

Alinhamento com os eixos principais

Uma aplicação bastante útil é o alinhamento de um objeto com seus eixos principais.

- Sejam $\mathcal{X} = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n)\}$ o conjunto das coordenadas dos voxels de um objeto em relação ao seu centro geométrico e Σ a matriz de covariância \mathcal{X} .
- A decomposição $\Sigma = \mathbf{U}\mathbf{S}\mathbf{V}^t$ em valores singulares calcula na primeira coluna de \mathbf{U} o eixo principal do objeto.
- Se aplicarmos a rotação de \mathcal{X} por \mathbf{U} teremos o alinhamento do objeto com os eixos principais.

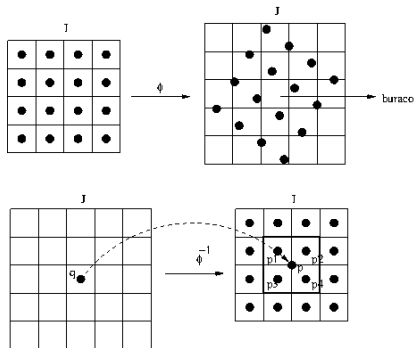
Alinhamento com os eixos principais

Uma aplicação bastante útil é o alinhamento de um objeto com seus eixos principais.

- Sejam $\mathcal{X} = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n)\}$ o conjunto das coordenadas dos voxels de um objeto em relação ao seu centro geométrico e Σ a matriz de covariância \mathcal{X} .
- A decomposição $\Sigma = \mathbf{USV}^t$ em valores singulares calcula na primeira coluna de \mathbf{U} o eixo principal do objeto.
- Se aplicarmos a rotação de \mathcal{X} por \mathbf{U} teremos o alinhamento do objeto com os eixos principais.
- Se aplicarmos o escalamento de \mathcal{X} por \mathbf{S}^{-1} teremos a normalização das dimensões do objeto ao longo dos eixos principais (uma elipse vira um círculo).

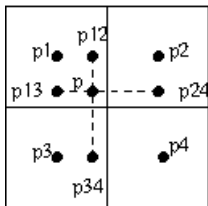
Interpolação

Ao aplicar a transformação inversa para obter a transformação direta, a interpolação dos valores da imagem original é adotada em uma adjacência \mathcal{A} .



A adjacência \mathcal{A} é definida pelas coordenadas inteiras mais próximas, abaixo e acima, das coordenadas reais (x_p, y_p, z_p) de $p \in D_I$.

A interpolação linear assume que os valores dos spels variam linearmente em todas as direções.

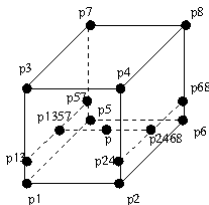


$$\vec{I}(p_{12}) = (x_p - x_{p_1})\vec{I}(p_2) + (x_{p_2} - x_p)\vec{I}(p_1)$$

$$\vec{I}(p_{34}) = (x_p - x_{p_3})\vec{I}(p_4) + (x_{p_4} - x_p)\vec{I}(p_3)$$

$$\vec{I}(p) = (y_p - y_{p_{12}})\vec{I}(p_{34}) + (y_{p_{34}} - y_p)\vec{I}(p_{12})$$

No caso 3D, temos



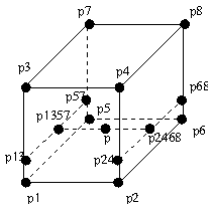
$$\vec{I}(p_{24}) = (y_p - y_{p_4})\vec{I}(p_2) + (y_{p_2} - y_p)\vec{I}(p_4)$$

$$\vec{I}(p_{68}) = (y_p - y_{p_8})\vec{I}(p_6) + (y_{p_6} - y_p)\vec{I}(p_8)$$

$$\vec{I}(p_{13}) = (y_p - y_{p_3})\vec{I}(p_1) + (y_{p_1} - y_p)\vec{I}(p_3)$$

$$\vec{I}(p_{57}) = (y_p - y_{p_7})\vec{I}(p_5) + (y_{p_5} - y_p)\vec{I}(p_7)$$

No caso 3D, temos



$$\vec{I}(p_{2468}) = (z_p - z_{p_{24}})\vec{I}(p_{68}) + (z_{p_{68}} - z_p)\vec{I}(p_{24})$$

$$\vec{I}(p_{1357}) = (z_p - z_{p_{13}})\vec{I}(p_{57}) + (z_{p_{57}} - z_p)\vec{I}(p_{13})$$

$$\vec{I}(p) = (x_p - x_{p_{1357}})\vec{I}(p_{2468}) + (x_{p_{2468}} - x_p)\vec{I}(p_{1357})$$

Se uma imagem $\hat{I} = (D_I, \vec{I})$ tem voxels de dimensões $(d_{x_1}, d_{y_1}, d_{z_1})$ e desejamos gerar por interpolação uma imagem $\hat{J} = (D_J, \vec{J})$ com voxels $(d_{x_2}, d_{y_2}, d_{z_2})$, então é mais rápido

- interpolar primeiro $\hat{I} = (D_I, \vec{I})$ ao longo de x , gerando $\hat{I}_1 = (D_I, \vec{I}_1)$ com tamanho de voxel $(d_{x_2}, d_{y_1}, d_{z_1})$,

Se uma imagem $\hat{I} = (D_I, \vec{I})$ tem voxels de dimensões $(d_{x_1}, d_{y_1}, d_{z_1})$ e desejamos gerar por interpolação uma imagem $\hat{J} = (D_I, \vec{J})$ com voxels $(d_{x_2}, d_{y_2}, d_{z_2})$, então é mais rápido

- interpolar primeiro $\hat{I} = (D_I, \vec{I})$ ao longo de x , gerando $\hat{I}_1 = (D_I, \vec{I}_1)$ com tamanho de voxel $(d_{x_2}, d_{y_1}, d_{z_1})$,
- depois interpolar $\hat{I}_1 = (D_I, \vec{I}_1)$ ao longo de y , gerando $\hat{I}_2 = (D_I, \vec{I}_2)$ com tamanho de voxel $(d_{x_2}, d_{y_2}, d_{z_1})$, e

Se uma imagem $\hat{I} = (D_I, \vec{I})$ tem voxels de dimensões $(d_{x_1}, d_{y_1}, d_{z_1})$ e desejamos gerar por interpolação uma imagem $\hat{J} = (D_I, \vec{J})$ com voxels $(d_{x_2}, d_{y_2}, d_{z_2})$, então é mais rápido

- interpolar primeiro $\hat{I} = (D_I, \vec{I})$ ao longo de x , gerando $\hat{I}_1 = (D_I, \vec{I}_1)$ com tamanho de voxel $(d_{x_2}, d_{y_1}, d_{z_1})$,
- depois interpolar $\hat{I}_1 = (D_I, \vec{I}_1)$ ao longo de y , gerando $\hat{I}_2 = (D_I, \vec{I}_2)$ com tamanho de voxel $(d_{x_2}, d_{y_2}, d_{z_1})$, e
- por fim interpolar $\hat{I}_2 = (D_I, \vec{I}_2)$ ao longo de z , gerando $\hat{J} = (D_I, \vec{J})$ com tamanho de voxel $(d_{x_2}, d_{y_2}, d_{z_2})$.

Interpolação Linear

Por exemplo, a interpolação linear ao longo de x em uma dada linha y_p e fatia z_p é realizada para todo p de p_1 até o último voxel da linha, com incrementos d_{x_2} .

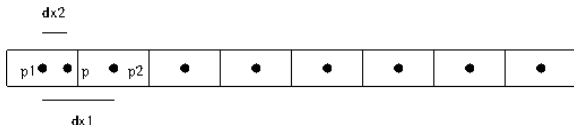
$$y_p = y_{p_1} = y_{p_2}$$

$$z_p = z_{p_1} = z_{p_2}$$

$$x_p = x_{p_1} + d_{x_2}$$

$$\vec{I}(p) = (x_p - x_{p_1})\vec{I}(p_2) + (x_{p_2} - x_p)\vec{I}(p_1)$$

onde p_2 é o próximo voxel na linha após p .



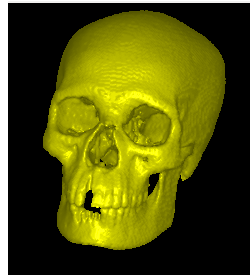
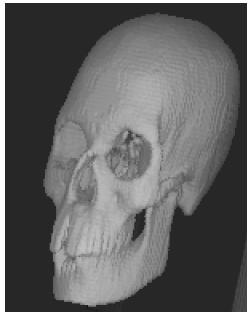
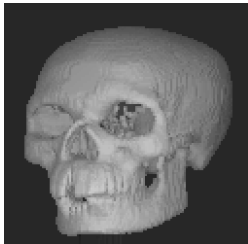
Interpolação Linear

Exemplo de escalamento e rotação em torno do centro da imagem e eixo z.



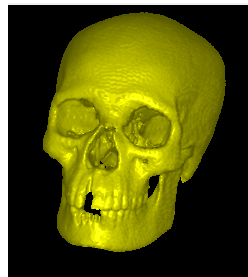
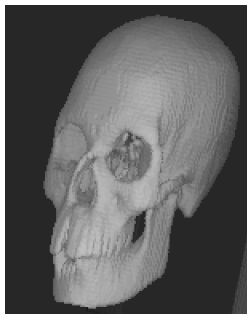
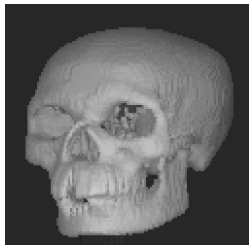
Interpolação Linear

Exemplo de rotação em torno do centro do objeto e eixos x e y , seguida de projeção ortogonal e tonalização (rendering).



Interpolação Linear

Exemplo de rotação em torno do centro do objeto e eixos x e y , seguida de projeção ortogonal e tonalização (rendering).



Distorções ocorrem se a imagem original é amostrada com (d_x, d_y, d_z) tais que $d_x \neq d_y$ ou $d_x \neq d_z$ ou $d_y \neq d_z$. Só a imagem colorida apresenta amostragem isotrópica.