Co-registro de imagens

Prof. Luiz Otavio Murta Jr.

Depto. de Computação e Matemática (FFCLRP/USP)

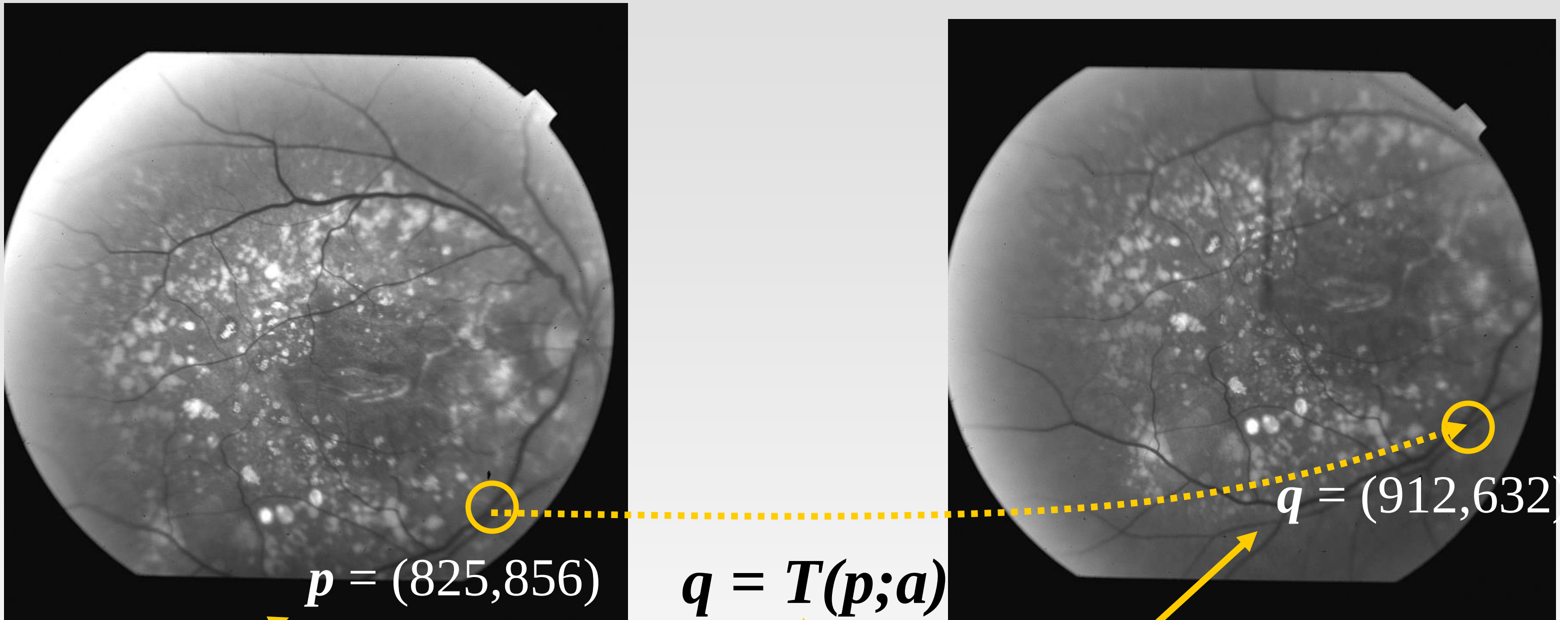
Tópicos

- O problema do co-registro
- Aplicações de co-registro
- Componentes de uma solução

Co-registro de imagens

- **Objetivo:**
  - Determinação do mapeamento entre duas imagens do mesmo objeto, objetos similares, a mesma região ou regiões similares
- **Aspectos abordados:**
  - Matemática básica
  - Imagens
  - Algoritmos
  - Implementações
  - Aplicações

Definição do Problema de Co-registro

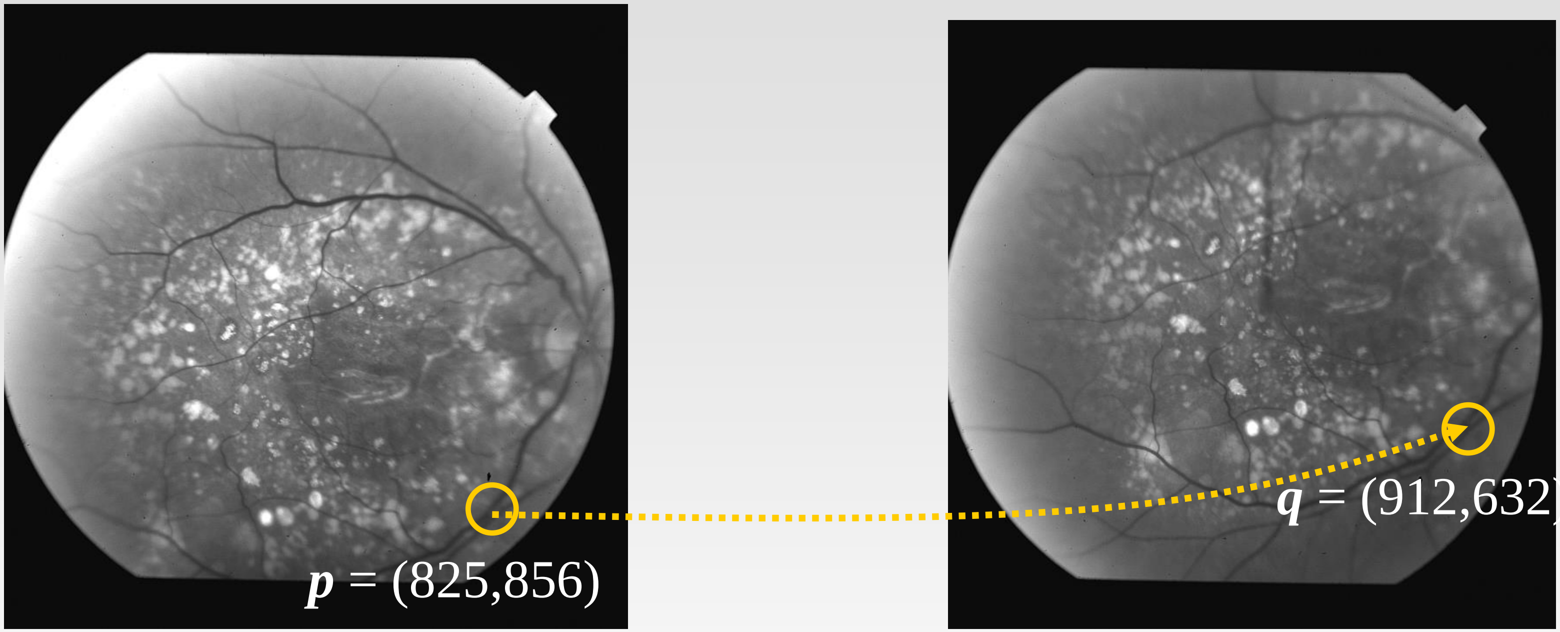


Localização do pixel na primeira imagem

Localização do pixel homólogo na segunda imagem

Função de mapeamento de localização de pixels

Exemplo de Função de Mapeamento

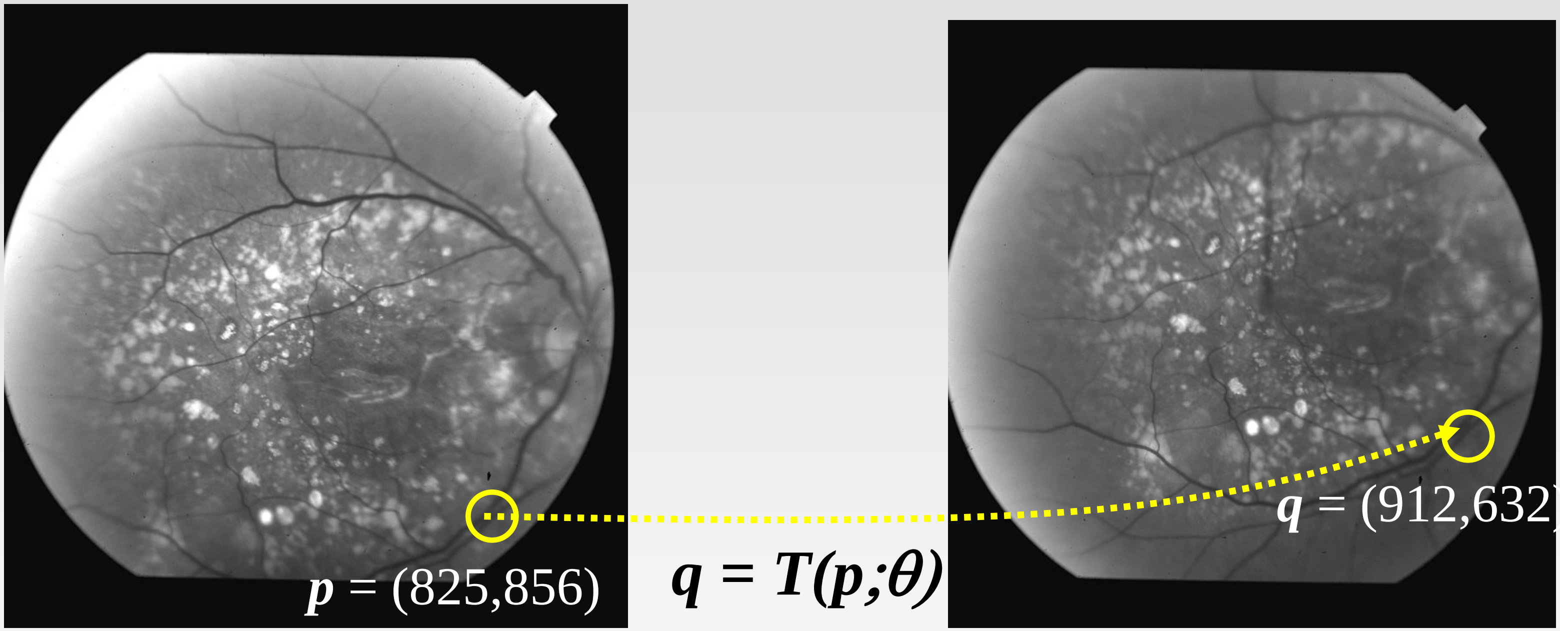


$$p = (x, y)^T$$

$$\Theta = (s, t_x, t_y)^T$$

$$T(p, \Theta) = \begin{pmatrix} s.x + t_x \\ s.y + t_y \end{pmatrix}$$

Definição do Problema de Co-Registro



Problema:

- Formular a função mapeamento $\boldsymbol{T}$
- Parâmetro desconhecido $\boldsymbol{\theta}$
- Correspondências desconhecidas, $\boldsymbol{p,q}$

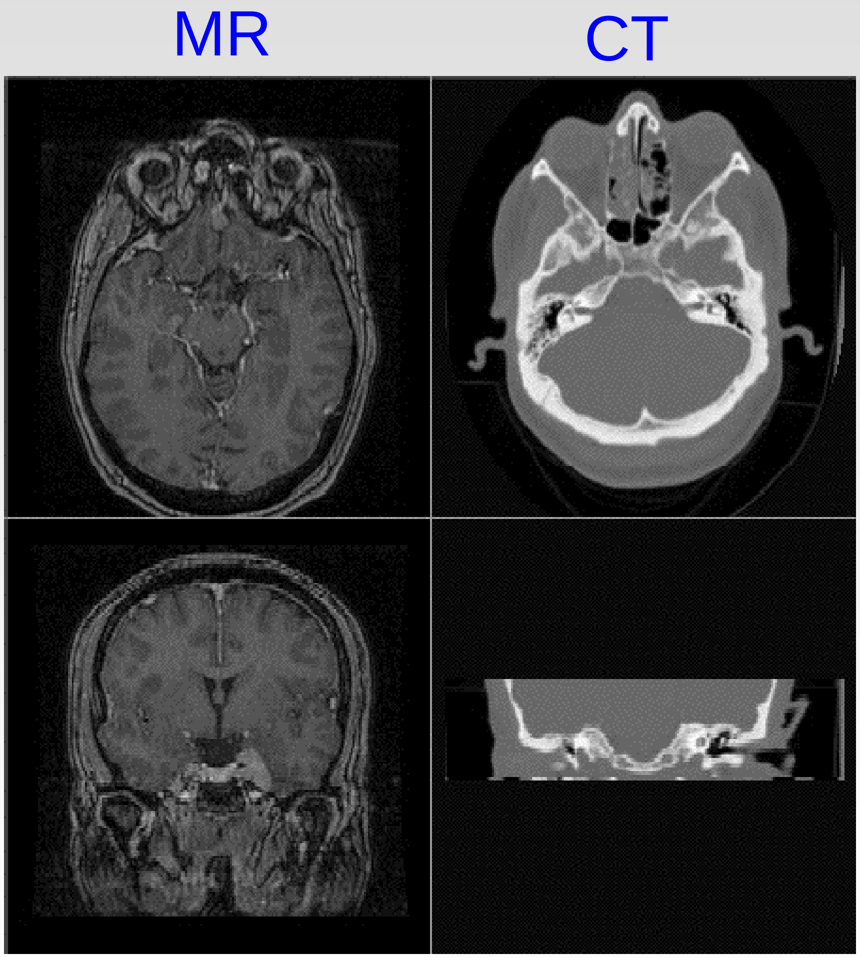
Problema "galinha-ovo"

Aplicações: Integração Multimodal

- **Dois ou mais diferentes sensores visualizando a mesma região ou volume**
- **Diferentes pontos de vista**
  - (alguns sensores especializados tem duas ou mais modalidades coincidentes, então o co-registro não é necessário.)
- **Informações diferentes são proeminentes em cada imagem**
- **As imagens podem até possuir dimensões diferentes!**
  - Imagens Range vs. Imagens de intensidade
  - CT volumes vs. imagens de fluoro

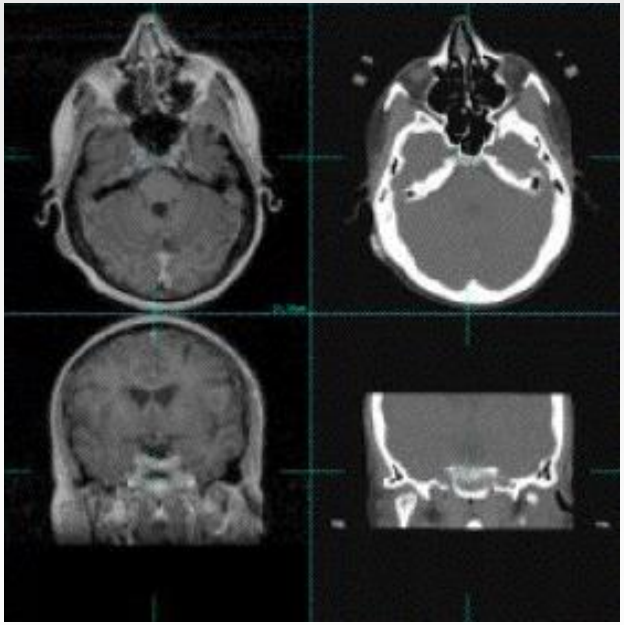
Exemplo: Co-registro cerebral MR-CT

- MR (ressonância magnética) mede conteúdo de água
- CT mede absorção de raios-x
- O osso é mais claro no CT e mais escuro MR
- Ambas as imagens são volumes 3d

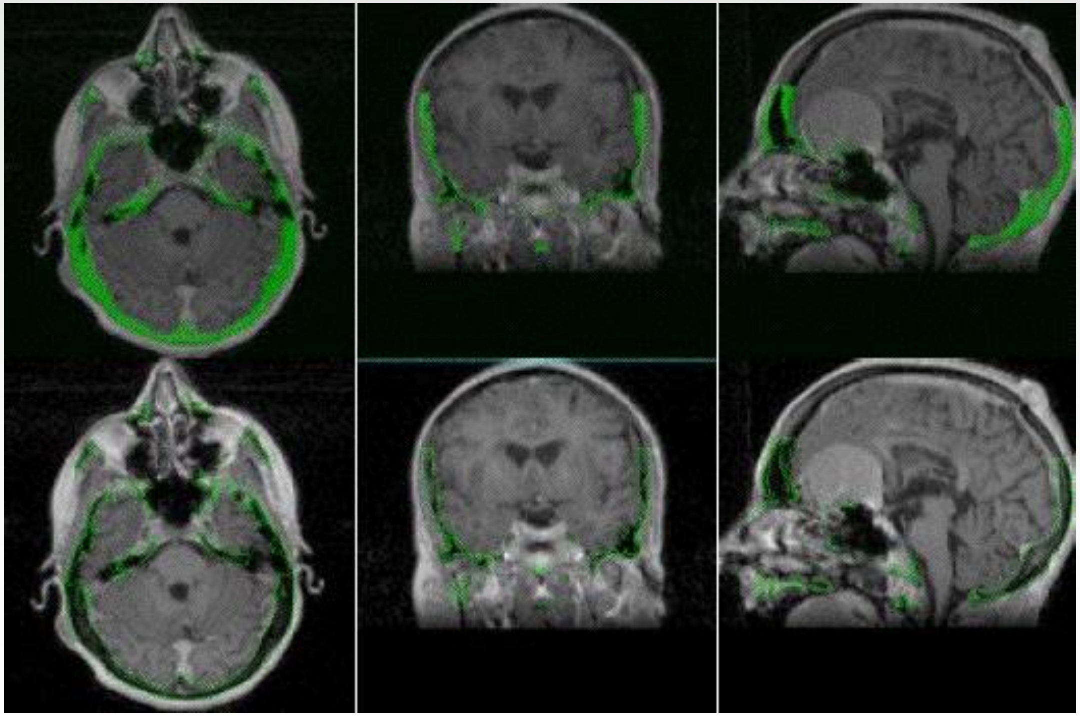


Resultados do co-registro MR-CT

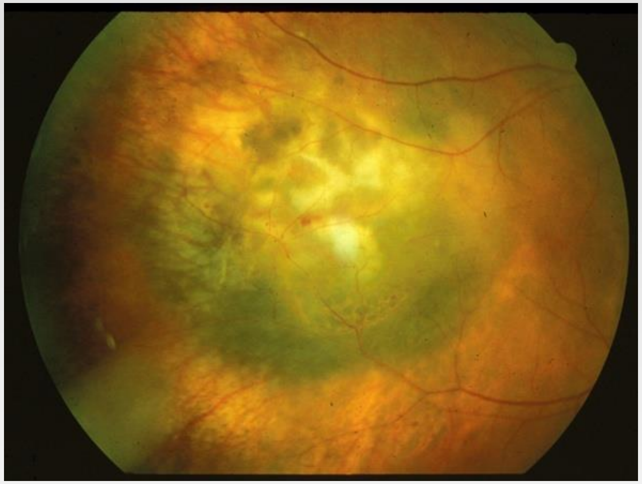
imagens alinhadas

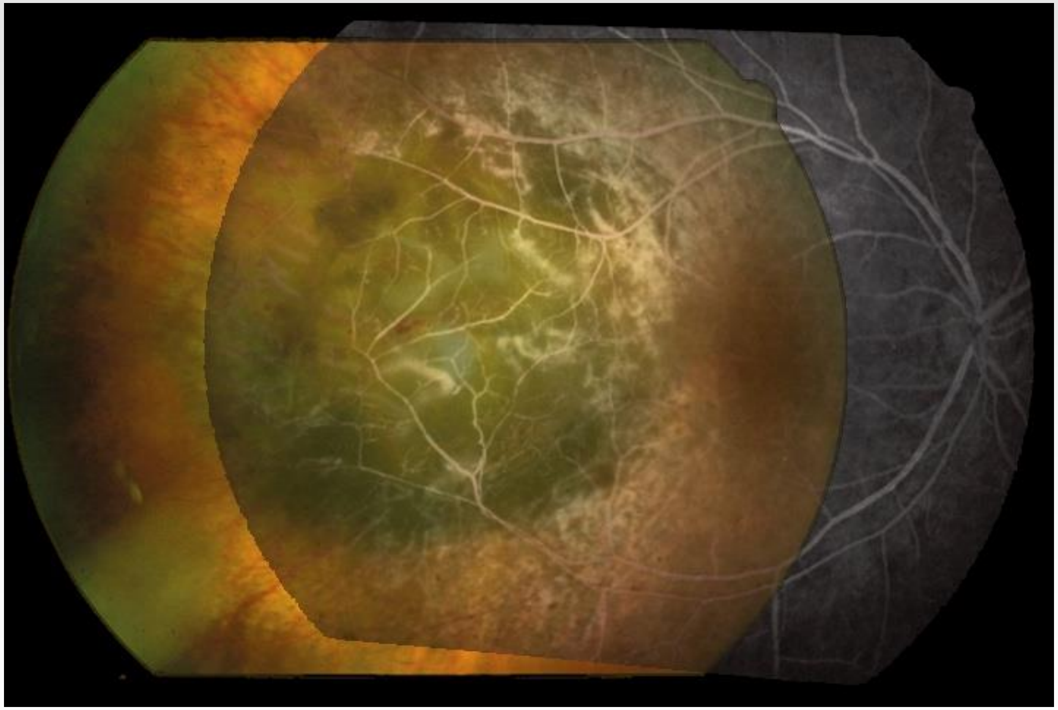


imagens superpostas, com estruturas do crânio provenientes do CT em verde



Angiograma da retina e imagem colorida

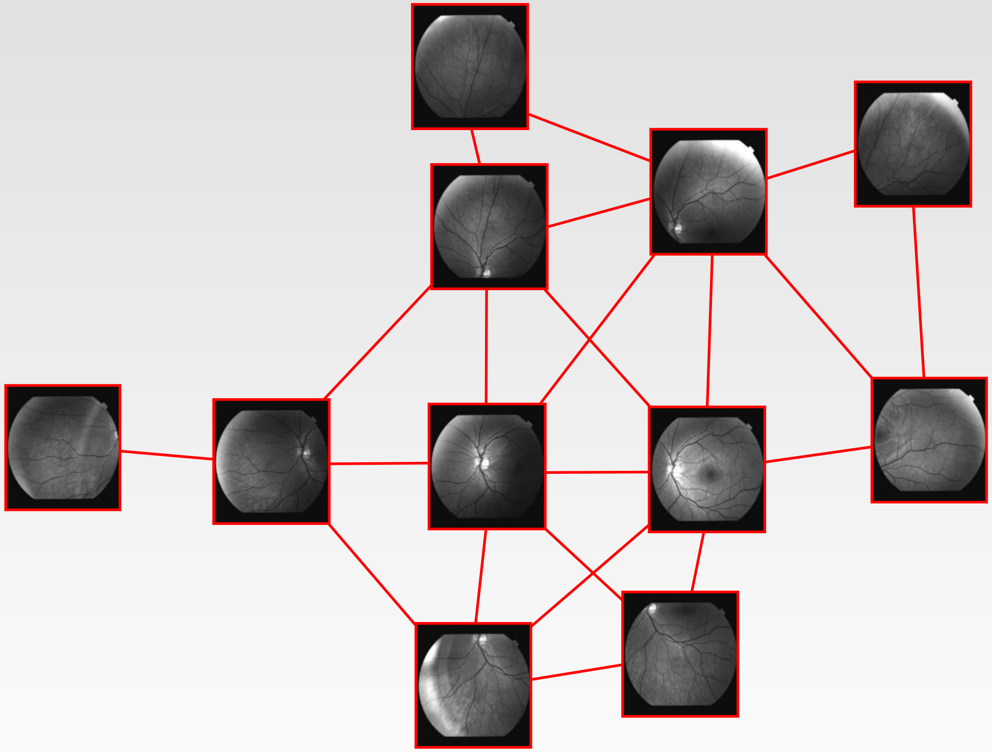


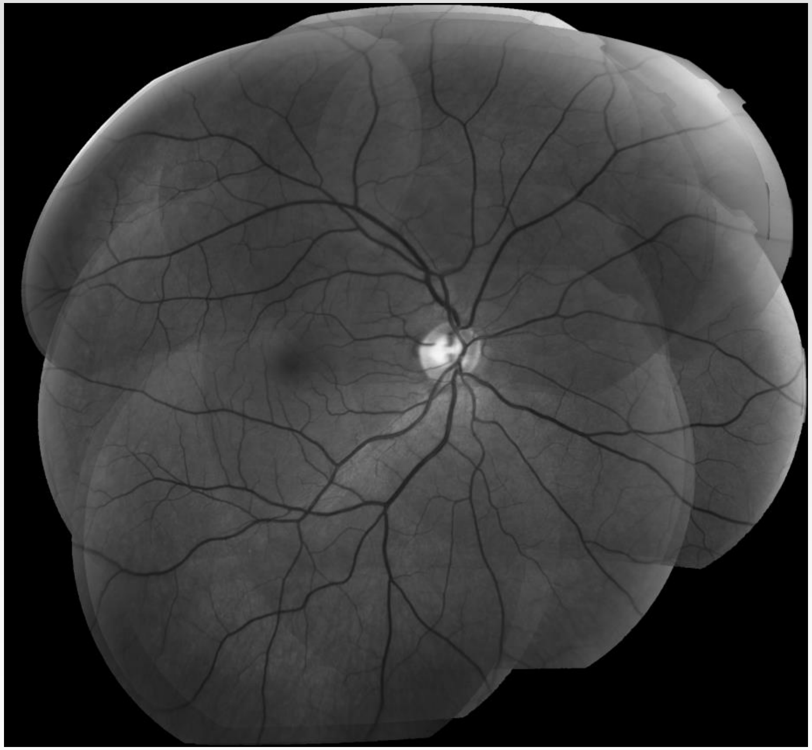


Aplicações: Mosaicos de imagens

- **Em alguns casos, dispomos de várias imagens parciais com sobreposição**
- **Nenhuma delas nos fornece uma visão completa**
- **Neste caso o objetivo do co-registro é “encaixar” estas imagens juntas de modo a obter uma visão mais ampla**
- **Isto requer:**
  - Uma câmara com o campo de visão limitado como é o caso da rotação em torno de centro ótico
  - Uma superfície de geometria simples, tal como plana ou quadrática

Imagem mosaico da retina





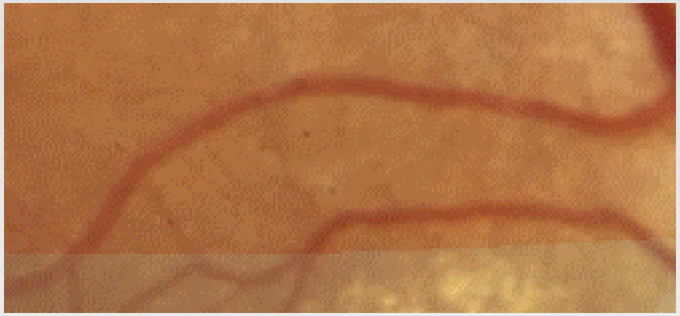
Mosaicos esféricos

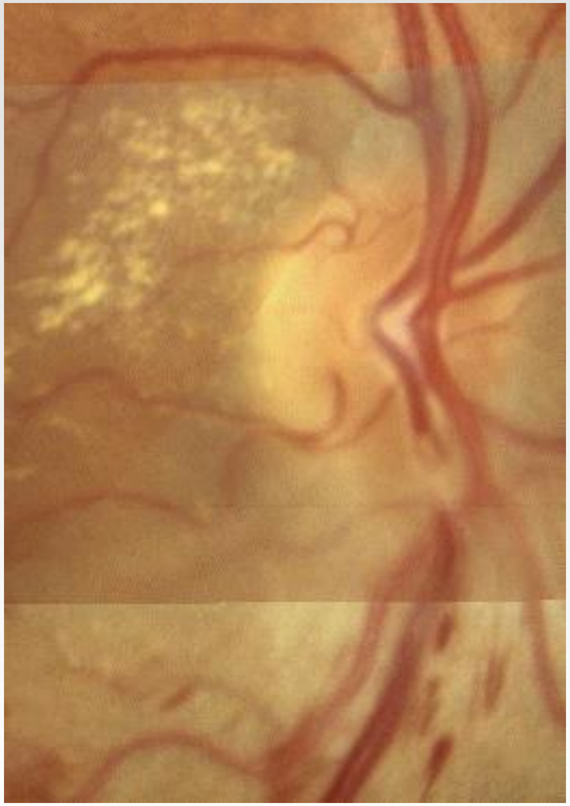
Aplicações: Detecção de mudanças

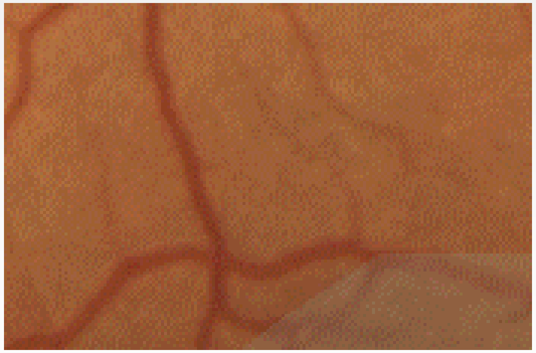
- **Imagens tomadas em tempos diferentes**
- **Com o conseqüente co-registro das imagens as mesmas passam a ter a mesma geometria, assim as diferenças nas imagens podem ser indicativas de mudanças ou variações**
- **Decidir se a mudança existe realmente pode ser difícil**
- **Exemplos:**
  - Imagens de MR com contraste
  - Imagens de perfusão miocárdica

Exemplo da Retina

Regiões mostrando mudanças







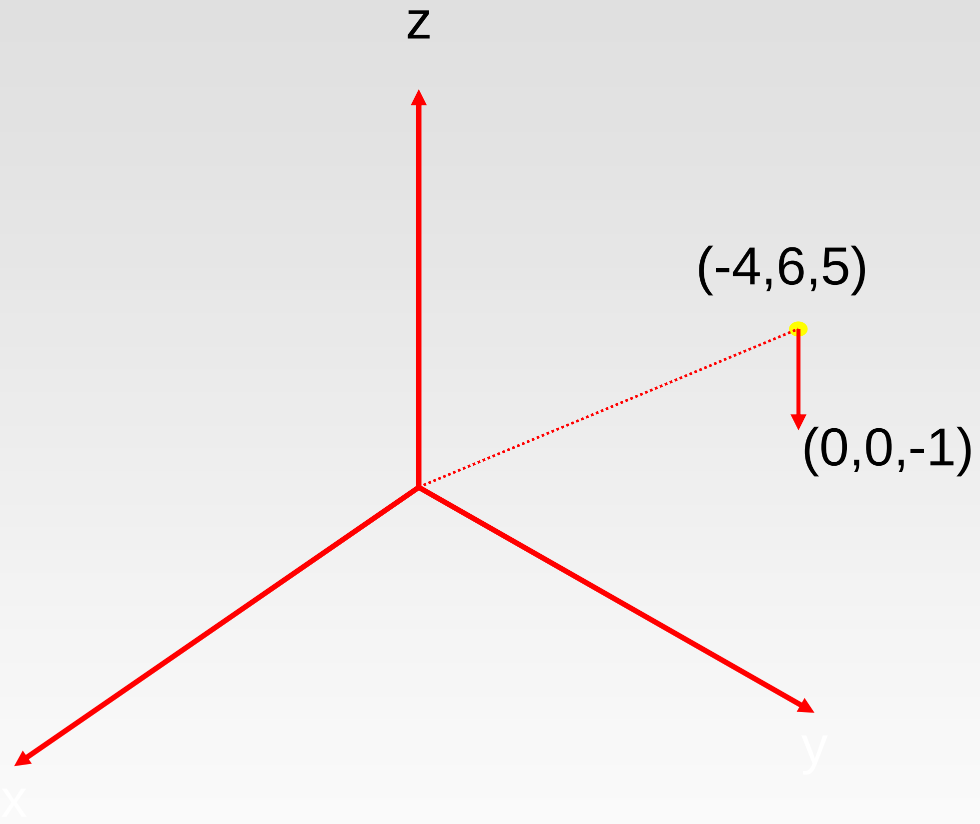
Outras Aplicações

- **Co-registro multi-sujeito para desenvolver atlas de variações para órgãos.**
  - Usado como base para detecção de variações anormais
- **Reconhecimento de objetos - alinhamento de instancia de modelos de objetos e imagem de objeto desconhecido**
- **Inspeção industrial**
  - Compare uma instancia modelo CAD com parte para determinar erros num processo de fabricação

Passos para uma solução

- Análise as imagens
- Determine as primitivas de imagens adequadas
- Determine o modelo de transformação
  - Geométrica e de intensidade
- Especificar uma técnica de inicialização
- Estabelecer limites e uma métrica de erro para a estimativa da transformação
- Desenvolver um algoritmo de minimização
- Desenvolver um critério de convergência

Vetores: Exemplo



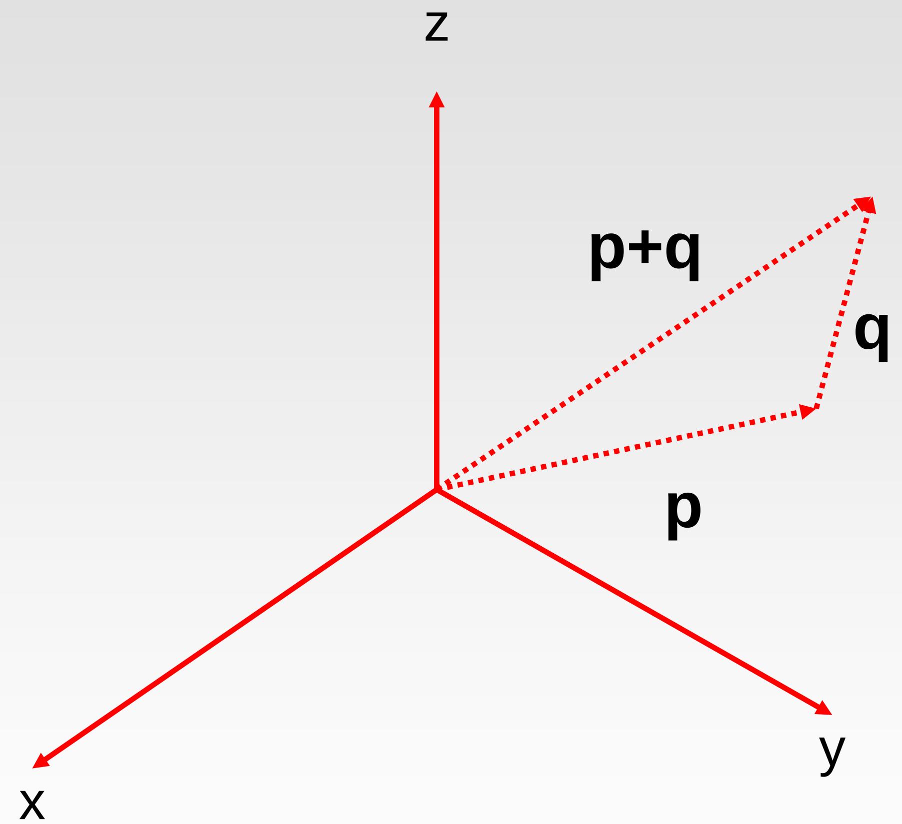
Vetores: Adição

- **Adicionado componente a componente**

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_n + y_n \end{pmatrix}$$

- **Exemplo:**

$$\begin{pmatrix} 2 \\ 3 \\ -1 \end{pmatrix} + \begin{pmatrix} 4 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 6 \\ 2 \\ 0 \end{pmatrix}$$



Interpretação geométrica

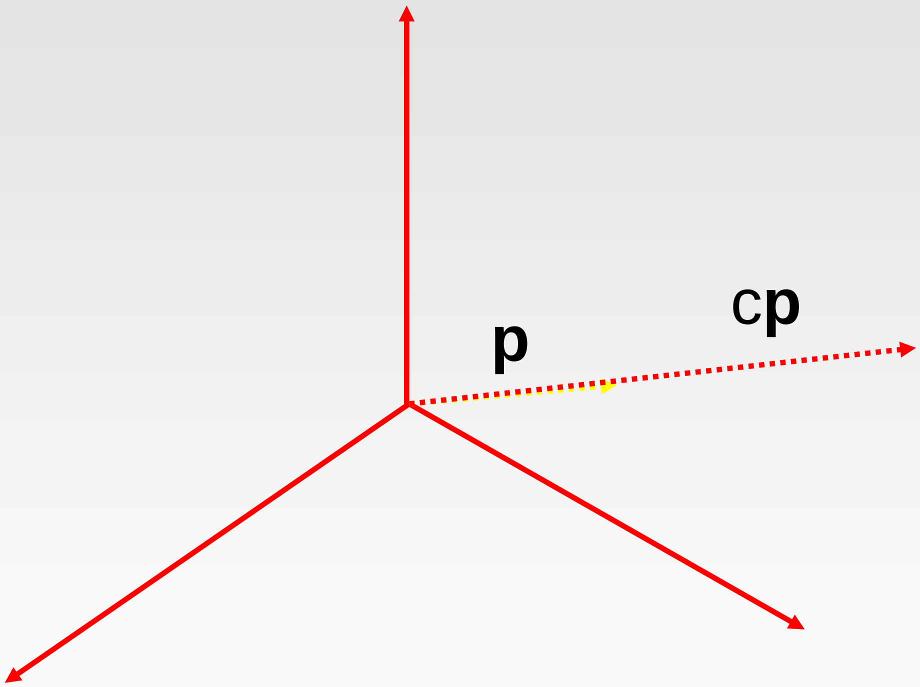
Vetores: Multiplicação escalar

- Forma mais simples de multiplicação envolvendo vetores
- Em particular:

$$c.x = c.\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} cx_1 \\ cx_2 \\ \vdots \\ cx_n \end{pmatrix}$$

- Exemplo:

$$3.\begin{pmatrix} 0,5 \\ -3 \\ 8 \end{pmatrix} = \begin{pmatrix} 1,5 \\ -9 \\ 24 \end{pmatrix}$$



Vetores: Magnitudes, Distancias

- **O comprimento ou magnitude de um vetor é**

$$\|x\| = \left[\sum_{i=1}^{n} x_i^2\right]^{1/2} = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$$

- **A distancia entre dois vetores é**

$$\|x - y\| = \left[\sum_{i=1}^{n} (x_i - y_i)^2\right]^{1/2} = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2}$$

Vetores: Produto escalar

- Segundo nodo de multiplicação envolvendo vetores
- Em particular,

$$x \cdot y = \sum_{i=1}^{n} x_i y_i = (x_1 y_1 + x_2 y_2 + \cdots + x_n y_n)$$

- Veremos uma notação diferente para o produto escalar usando multiplicação de matrizes
- Note que

$$x \cdot x = \|x\|^2$$

Vetor unitário

- **Um vetor unitário (direção) é um vetor de magnitude 1:**

$$\|x\| = x \cdot x = 1$$

- **Convencionalmente, usamos um “chapéu” para denotas vetor unitário, e.g.:**

$$\hat{e} \quad \hat{\mathrm{n}}$$

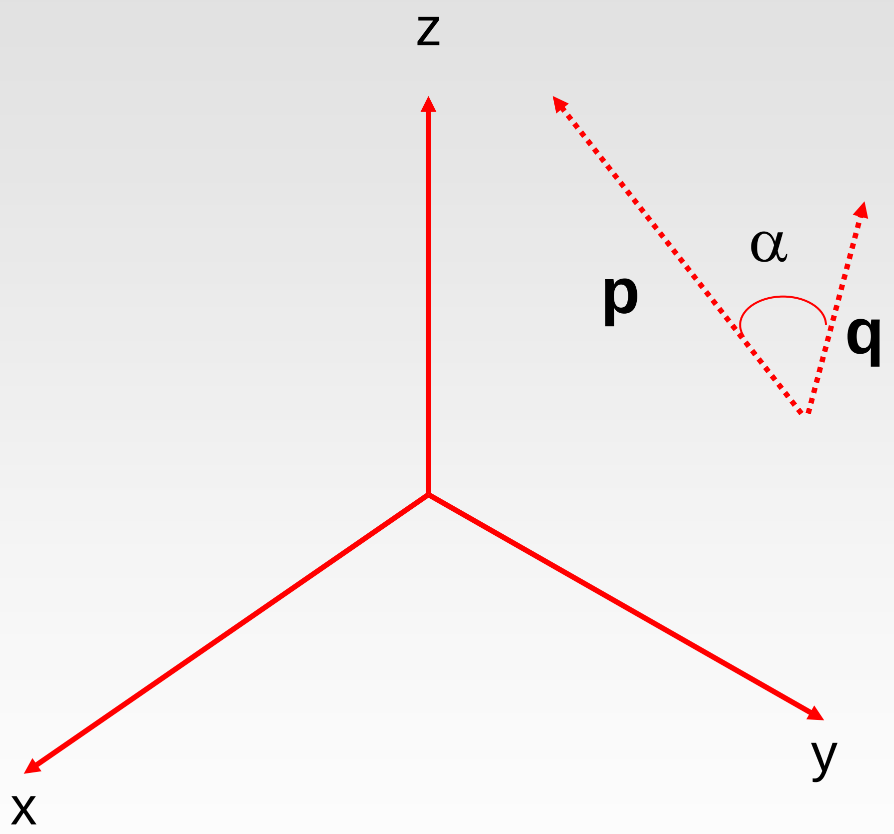
Angulo entre vetores

- **Podemos computar o ângulo entre dois vetores usando o produto escalar:**

$$\cos\alpha = \frac{p \cdot q}{\|p\| \cdot \|q\|}$$

- **Dois vetores não nulos são ortogonais se, e apenas se:**

$$p \cdot q = 0$$



Produto vetorial

- **Dados três vetores em 3 dimensões, p e q, o produto vetorial dos dois é perpendicular a ambos**

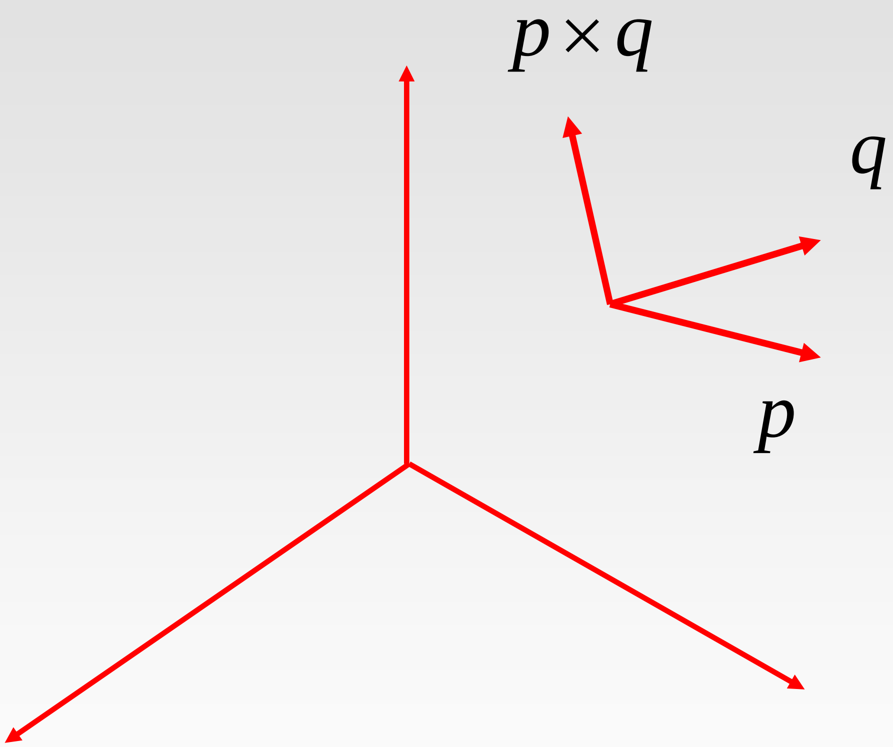
$$r = p \times q$$

- **Na forma componente,**

$$r = \begin{pmatrix} p_2 q_3 - p_3 q_2 \\ p_3 q_1 - p_1 q_3 \\ p_1 q_2 - p_2 q_3 \end{pmatrix}$$

- **Finalmente,**

$$\|p \times q\| = \sin\alpha \|p\| \cdot \|q\|$$



Matrizes - Definição

- **Matrizes são arranjos numéricos retangular, no qual cada número é subscrito por dois índices:**

$$A = \underbrace{\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}}_{n \text{ colunas}} \left.\vphantom{\begin{pmatrix} a \\ a \\ \cdot \\ \cdot \\ a \end{pmatrix}}\right\} m \text{ linhas}$$

- **Uma notação curta para isto é** $A = (a_{ij})$

Matrizes especiais: Identidade

- **A matriz identidade, denotada I, I_n ou I_{nxn}, é uma matriz quadrada, com *n* linhas e colunas tendo 1's na diagonal principal e 0's em qualquer outro lugar:**

$$I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad I_{2\times 2} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Matrizes Diagonal

- A *matriz diagonal* é uma matriz quadrada que tem 0's em toda a matriz exceto na diagonal principal.
- Por exemplo:

$$A = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \underbrace{diag}(2,-1,3,0)$$

Notação curta

Matriz Transposta e Simetria

- **A transposta de uma matriz é aquela que as inscrições de linhas e colunas são invertidas:**

$$A^T = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{1m} \\ a_{12} & a_{22} & \cdots & a_{2m} \\ . & . & \cdots & . \\ . & . & \cdots & . \\ a_{1n} & a_{2n} & \cdots & a_{nm} \end{pmatrix}$$

- **Se** A = A^T **então a matriz é simétrica.**
  - Somente matrizes quadradas (*m*=*n*) são simétricas

Adição de Matrizes

- **Duas matrizes são somadas se, e somente se elas tem o mesmo número de linhas e colunas.**
- **Matrizes são somadas componente a componente:**

$$A + B = (a_{ij} + b_{ij})$$

- **Exemplo:**

$$\begin{pmatrix} 3 & 4 & 5 \\ 4 & -1 & 6 \end{pmatrix} + \begin{pmatrix} 5 & -2 & 0 \\ -1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 8 & 2 & 5 \\ 3 & -1 & 7 \end{pmatrix}$$

Multiplicação escalar de matrizes

- **Qualquer matriz pode ser multiplicada por um escalar**

$$cA = Ac = (ca_{ij})$$

Multiplicação de matrizes

- **O produto de uma matriz *mxn* e uma matriz *nxp* é uma matriz *mxp* :**

$$A \times B = \left(\sum_{k=1}^{n} a_{ik} b_{ki} \right)$$

- **As inscrições *i,j* da matriz resultante é o produto escalar da linha *i* de** A **e coluna *j* de** B
- **Exemplo**

$$\begin{pmatrix} 2 & 2 & 5 \\ 4 & -1 & 0 \end{pmatrix} \times \begin{pmatrix} 5 & 2 \\ -1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 8 & 1 \\ 21 & -8 \end{pmatrix}$$

Vetores como matrizes

- **Vetores, os quais usualmente escritos como vetor coluna, podem ser pensados como matrizes *n*x*1***
- **A transposta de um vetor é uma matriz *1*x*n* – um vetor linha.**
- **Isto permite escrever o produto escalar como multiplicação de matrizes:**

$$a \cdot b = a^T b = b^T a$$

- **Por exemplo,**

$$\begin{pmatrix} -1 \\ 3 \\ 2 \end{pmatrix} \cdot \begin{pmatrix} -2 \\ -1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 & 3 & 2 \end{pmatrix} \begin{pmatrix} -2 \\ -1 \\ 0 \end{pmatrix} = -5$$

Matrizes quadradas

- **Muito da nossa discussão aqui concentrará em matrizes quadradas :**
  - Traço
  - Determinante
  - Inversa
  - Autovalores
  - Matriz ortogonal / ortonormal
- **Quando discutimos decomposições de valores singulares voltaremos em matrizes não quadradas**

Traço de uma Matriz

- Somados termos da diagonal principal de uma matriz:

$$tr(A) = \sum_{i=1}^{n} a_{ii}$$

- O traço é igual a soma dos autovalores da matriz.

Determinante

- **Notação:**

$$\det(A) = |A|$$

- **recursiva:**
  - quando *n=1*,

$$\det(A) = a_{11}$$

  - quando *n=2*

$$\det(A) = a_{11}a_{22} - a_{12}a_{21}$$

Determinante (continuando)

- Para $n>2$, escolha qualquer linha i de **A,** e defina $\mathbf{M}_{i,j}$ sendo $(n-1)\text{x}(n-1)$ matriz formada apagando a linha i e coluna j de **A**, então

$$\det(A) = \sum_{i=1}^{n} (-1)^{i+j} a_{ij} \det(M_{ij})$$

- Temos a mesma formula escolhendo uma coluna *j qualquer de* **A** e somando sobre as linhas.

Algumas propriedades do determinante

- **Se duas linhas ou duas colunas quaisquer são iguais, o determinante é 0**
- **Trocando duas linhas ou trocando duas colunas inverte o sinal do determinante**
- **O determinante de** A **é igual ao produto dos autovalores de** A
- **Para matrizes quadradas**

$$\det(AB) = \det(A)\det(B)$$

$$\det(A) = \det(A^T)$$

Matriz inversa

- **A matriz inversa de uma matriz quadrada** A **é a única matriz** A^{-1} **tal que**

$$A^{-1}A = AA^{-1} = I$$

- **Matrizes que não tem inversa são ditas não inversíveis ou *singular***
- **Uma matriz é inversível se, e somente se seu determinante é não nulo**
- **Não preocuparemos com os mecanismo de cálculos de matrizes inversas, a não ser usando decomposição de valor singular**

Autovalores e autovetores

- **Um escalar λ e um vetor** v **são, respectivamente, um autovalor e um autovetor unitário associado de uma matriz quadrada** A **se**

$$Av = \lambda v$$

- **Por exemplo, se pensarmos em** A **como uma transformação e se** $\lambda=1$, **então** Av=v **implica que** v **é um "ponto-fixo" da transformação.**
- **Auto valores são encontrados resolvendo a equação**

$$\det(Av - \lambda v) = 0$$

- **Uma vez que os auto valores são conhecidos, autovetores são encontrados achando o espaço nulo de**

$$A - \lambda I$$

Autovolores de matrizes simétricas

- São reais (não imaginários), o que pode ser visto estudando a seguinte equação

$$\|Av\|^2 = v^T A^T Av = v^T A^T(\lambda v) = \lambda v^T A^T v = \lambda^2 v^T v$$

- Podemos também mostrar que autovetores associados com autovalores distintos de uma matriz simétrica são ortogonais
- Podemos, portanto, escrever uma matriz simétrica como:

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^T$$

Matrizes ortonormais

- **Uma matriz quadrada é ortonormal (também chamada ortogonal) se**

$$AA^T = I$$

  - Em outras palavras $\mathbf{A}^T$ é a matriz inversa.

- **Baseado nas propriedades de inversas, isto implica**

$$A^T A = I$$

- **Isto significa que para vetores formados por duas linhas quaisquer ou duas colunas quaisquer**

$$a_i^T a_j = \delta_{i,j}$$

delta de Kronecker, o qual é 1 se $i=j$ e 0 caso contrário

Matrizes ortonormais - Propriedades

- O determinante de uma matriz ortonormal é 1 ou -1, porquê:

$$\det(QQ^T) = \det(Q)\det(Q^T) = \det(I) = 1$$

- Multiplicando um vetor por uma matriz ortonormal não muda o comprimento do vetor:

$$\|Qv\|^2 = v^T Q^T Q v = v^T I v = v^T v = \|v\|^2$$

- Uma matriz ortonormal a qual o determinante é 1 (-1) é chamada uma rotação (reflexão).
- É claro que, como já discutimos

$$Q^{-1} = Q^T$$

Decomposição de valor singular (SVD)

- **Considere uma matriz *mxn*,** A**, e assuma *m*≥*n*.**
- A **pode ser "decomposta" no produto de 3 matrizes:**

$$A = UWV^T$$

- **Onde:**
  - **U** é *mxn* com colunas ortonormais
  - **W** é uma matriz diagonal *nxn* de "valores singulares", e
  - **V** é matriz ortonormal *nxn*
- **Se *m*=*n* então** U **é uma matriz ortonormal**

Propriedades dos valores singulares

$$W = diag(w_1, w_2, \ldots, w_n)$$

- **com**

$$w_1 > w_2 > \cdots > w_n > 0$$

- **e**
  - o número valores singulares não zeros é igual ao grau de **A**

SVD e inversão de matriz

- **Para uma matriz não-singular, quadrada, com**

$$A = UWV^T$$

- **A inversa de** A **é**

$$A^{-1} = V diag(1/w_1, 1/w_2, \cdots, 1/w_n) U^T$$

  - Você pode confirmar isto!

- **Note, contudo, esta nem sempre é a melhor maneira de computar a inversa**

SVD e solução de sistemas lineares

- **Muitas vezes o problema se reduz a achar o vetor** x **que minimiza**

$$\left\| Ax - b \right\|^2$$

- **Tomando a derivada em relação a** x**, e igualando o resultado a** 0 **e resolvendo implica**

$$A^T A x = A^T b$$

- **Computando o SVD de** A **(assumindo grau máximo) resulta em**

$$x = V W^{-1} U^T b$$

Tópicos

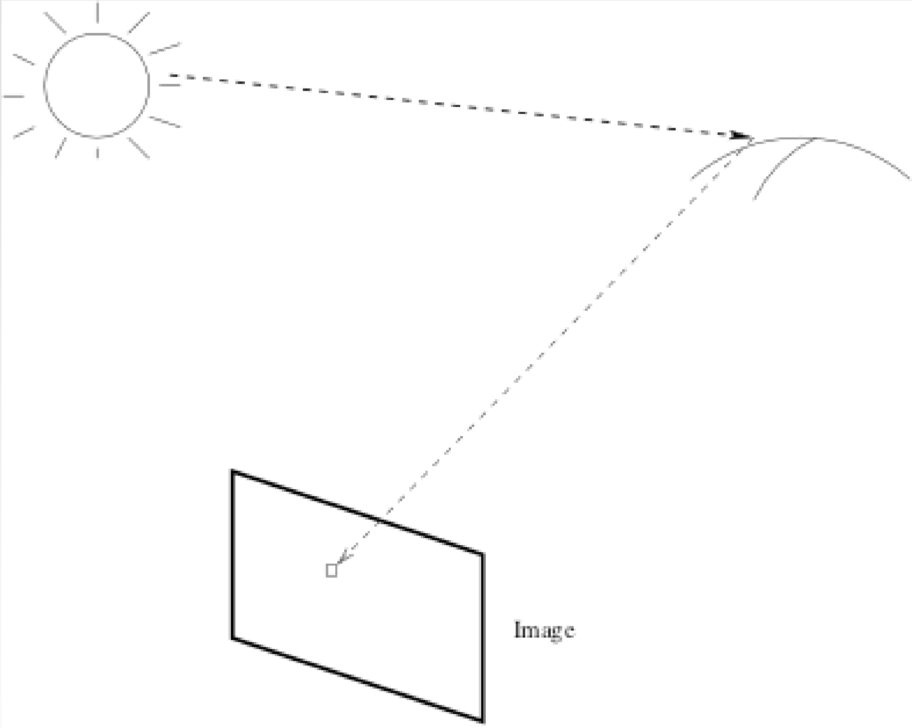
- **Imagens:**
  - Tipos de imagens
  - Sistemas de coordenadas
- **Transformações**
  - Transformações de similaridade em 2d e 3d
  - Transformações affine em 2d e 3d
  - Transformações projetivas
  - Transformações não-lineares e deformáveis

Imagens

- **Imagens tomográficas:**
  - CT e MRI
- **Imagens de intensidade / vídeo**
- **Imagens de intervalo**

Imagens de vídeo

- **Intensidades de pixel dependem de:**
  - Iluminação
  - Orientação de superfícies
  - Refletância
  - Direção da projeção
  - Digitalização
- **Por si só:**
  - Dimensões de pixels não tem significados físicos
  - Intensidades de pixels não tem significados físicos

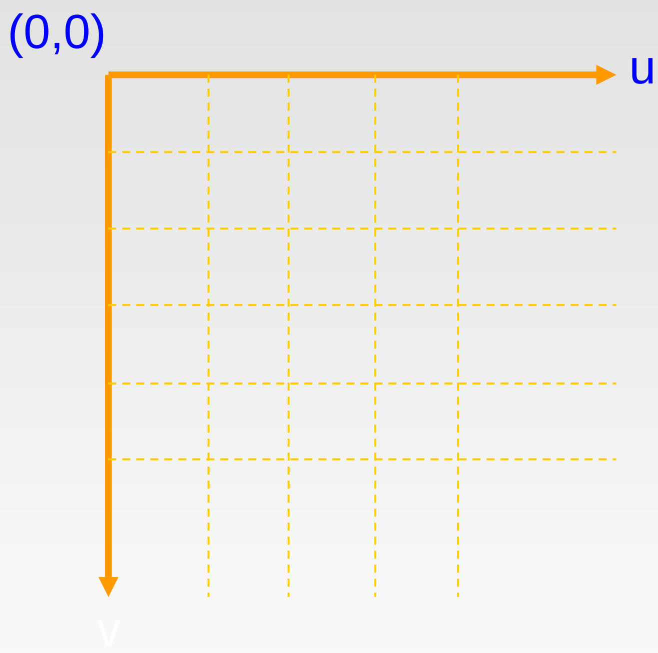


Sistemas de coordenadas

- **Topicos:**
  - Coordenadas da imagem
  - Coordenadas físicas
  - Mapeamento entre elas
- **Deve-se ficar atento a isto, especialmente na impelementação de detalhes de algorítrmo do co-registro**

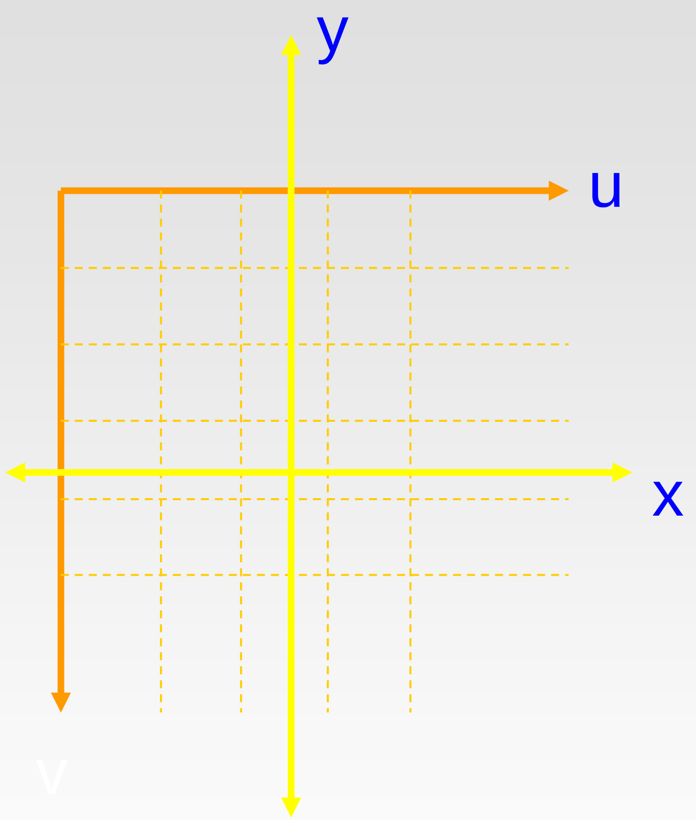
Sistema de coordenadas da imagem

- **A origem é geralmente no canto da imagem, freqüentemente esquerdo superior**
- **Eixos:**
  - Eixo *x* é para direita
  - Eixo *y* é para baixo
- **Difere das coordenadas da matriz e das coordenadas Cartesianas**
- **As unidades são incrementos inteiros nos índices da grade**



Coordemadas físicas

- **Origem:**
  - Próximo ao centro em imagens de vídeos
  - Definido pelo scanner em imagens medicas
- **Eixos:**
  - Coordenadas frame “mão direita”
- **Unidades são millimetros (imagens medicas)**



Voxels não isotrópicos em imagens médicas

- Dimensão axial (*z*) é freqüentemente maior que as dentro da fatia (dimensões *x* e *y*)
- Na pior das hipóteses, a oredem de grandeza é próxima
- Novos scanners estão se aproximando das dimensões isotrópicas

Mudanças de coordenadas como multiplicação de matrizes

- **Coordenadas físicas par coordenadas de pixels :**

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \boxed{s_x} & 0 \\ 0 & \boxed{s_y} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \boxed{\begin{pmatrix} t_u \\ t_v \end{pmatrix}}$$

Escalonamento das unidades físicas para pixels: parâmetros são pixels por mm

Translação da origem em unidades de pixels

Forma homogênea

- **Usando coordenadas homogêneas --- nos termos mais simples, adicionando 1 no fim de cada vetor --- nos permite escrever usando uma única multiplicação de matriz:**

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & t_u \\ 0 & -s_y & t_v \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Transformações

- Transformações geométricas
- Transformações de intensidades

Transformação geométrica direta

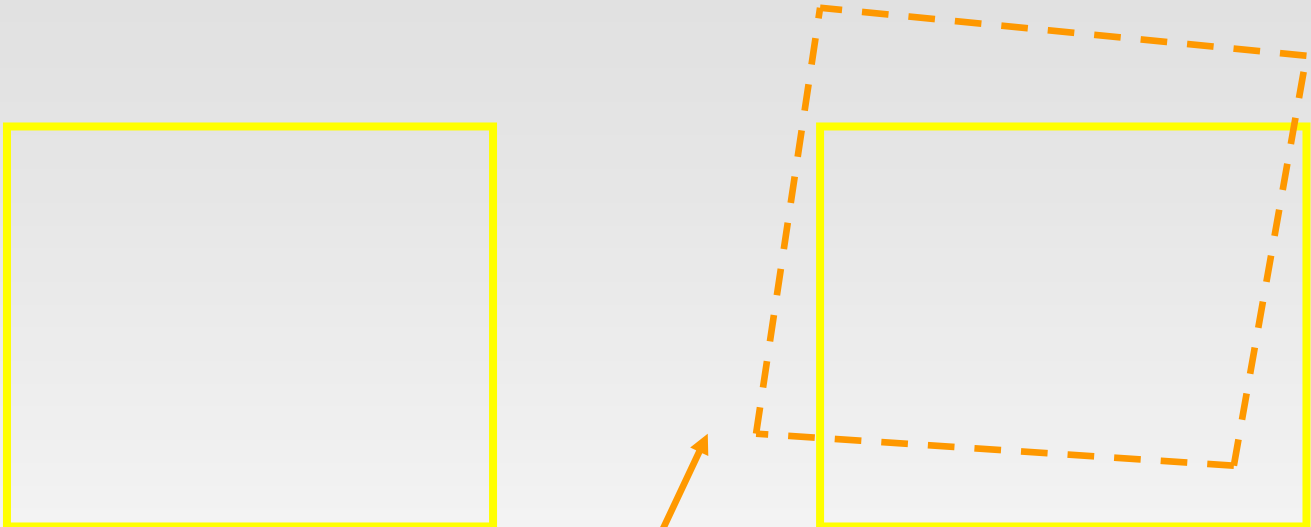


Imagem em "Movimento", $\mathbf{I}_m$

Imagem "Fixa", $\mathbf{I}_f$

Imagem em movimento transformada, $\mathbf{I}_m$'. Ela é mapeada no sistema de coordenadas de $\mathbf{I}_f$

Mapeando valores de pixels

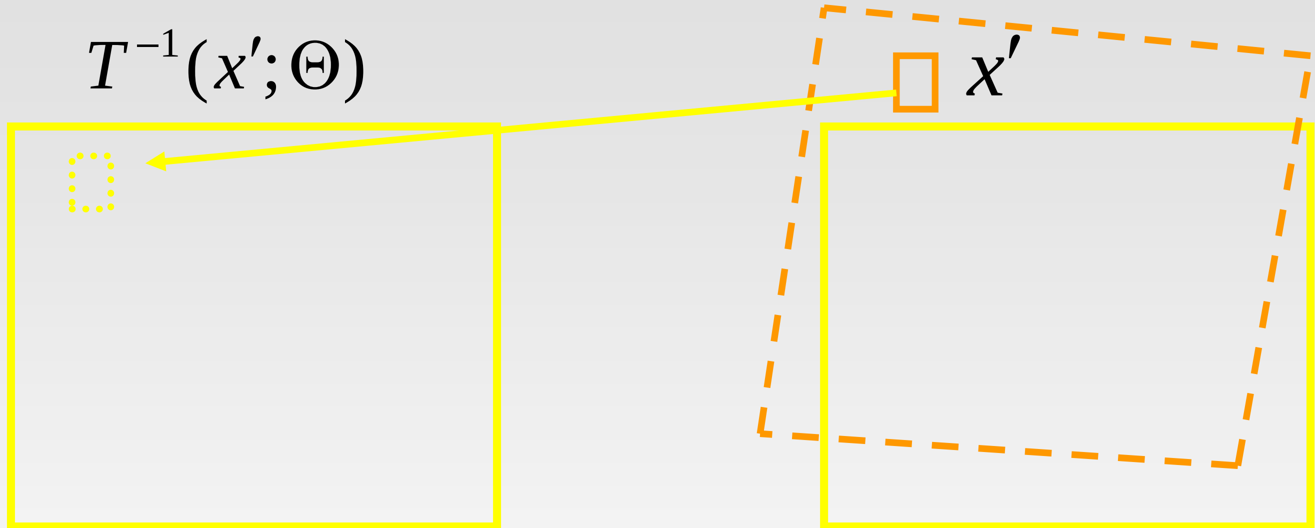


Imagem em “Movimento”, $\mathbf{I}_m$ Imagem “Fixa”, $\mathbf{I}_f$

$$I'_m(x')I_m\left(T^{-1}(x';\Theta)\right)$$

É preciso retornar para obter o valor do pixel. Isto geralmente não mapeia num pixel exato, mas sim entre pixels. Por isso é necessário interpolação.

Transformação reversa

- **Pelo que já foi dito, algumas técnicas estimam os parâmetros da transformação reversa, T^{-1}:**
  - Da imagem fixa para a imagem em movimento
- **Isto pode se tornar um problema se a função de transformação é não reversível**
- **Algoritmos baseados em intensidades geralmente estimam a transformação reversa**

Transformações de intensidades

- **Quando as intensidades devem ser tambem transformadas, as equações se tornam mais complicadas:**

$$I'_m(x') = \boxed{S}\left(I_m\left(T^{-1}\left(x' : \Theta\right)\right) : \boxed{\gamma}\right)$$

Função de mapeamento de intensidade

Parâmetros do mapeamento de intensidade

Felizmente, não estaremos muito preocupados com mapeamentos de intensidades

Modelos de transformação

- Translação e escalonamento em 2d e 3d
- Transformações de similaridade em 2d e 3d
- Transformações affine em 2d e 3d

Translação e escalonamento

- **Nós já vimos em 2d:**
  - Transformações entre coordenadas de pixels e coordenadas físicas
  - Translação e escalonamento na aplicação da retina
- **Na forma matricial:**

$$T(x;\Theta) = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} x + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

- **onde**

s_x

$$\Theta = \begin{pmatrix} s_x & s_y & t_x & t_y \end{pmatrix}^T$$

- **Na forma homogênea:**

$$T(x;\Theta) = \begin{pmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \boxed{x} \\ 1 \end{pmatrix}$$

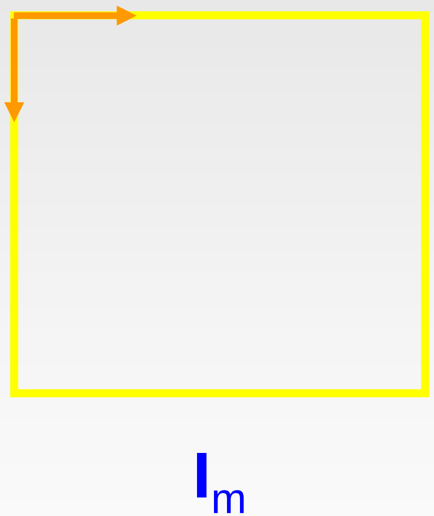
Vetor de duas componentes

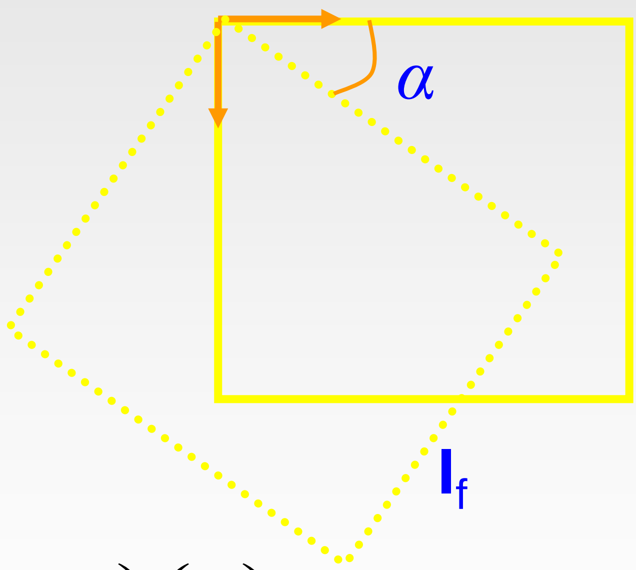
Transformações de Similaridade

- **Componentes:**
  - Rotação (próximo slide)
  - Translação
  - Escalonamento independente
- **“Invariantes”**
  - Ângulos entre vetores permanecem fixos
  - Todos os comprimentos escalonam proporcionalmente, então as razões entre os comprimentos são preservadas
- **Exemplo de aplicação:**
  - Multi-modalidade, co-registro cerebral intra-subjetivo (mesma pessoa)

Rotações em 2d

- **Assumam (por enquanto) que a transformação direta é somente uma rotação**
- **A origem das imagens são os cantos superiores direito; angulos são mensurados no sentido horário no eixo *x***





$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\alpha & -sen\alpha \\ sen\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Transformações de Similaridade em 2d

- **Como resultado:**

$$T(x:\Theta) = s\begin{pmatrix} \cos\alpha & -sen\alpha \\ sen\alpha & \cos\alpha \end{pmatrix} x + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

$$\Theta = \left(s, \alpha, t_x, t_y\right)^T$$

- **Ou:**

$$T(x:\Theta) = \begin{pmatrix} s.\cos\alpha & -s.sen\alpha \\ s.sen\alpha & s.\cos\alpha \end{pmatrix} x + \begin{pmatrix} t_x \\ t_y \end{pmatrix} = \begin{pmatrix} a & -b \\ a & a \end{pmatrix} x + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

$$\Theta = \left(a, b, t_x, t_y\right)^T$$

Matrizes de rotação e rotações 3d

- **A matriz de rotação 2d é ortonormal com determinante 1:**

$$R = \begin{pmatrix} \cos\alpha & -sen\alpha \\ sen\alpha & \cos\alpha \end{pmatrix}$$

- **em dimensões arbitrárias, estas duas propriedades preservam.**
- **Portanto, escrevemos a transformação de similaridade em dimensões arbitrárias como:**

$$T(x:\Theta) = sRx + t \qquad T(x:\Theta) = \begin{pmatrix} sR & t \\ 0^T & 1 \end{pmatrix}\begin{pmatrix} x \\ 1 \end{pmatrix}$$

Representando Rotaçoes em 3d

- R **é uma matriz 3x3, dando 9 parametros, mas tem somente 3 graus de liberdade.**
  - Suas propriedades de ortonormsalidade removem os outros 6.
- **Questão: Como representar** R**?**
- **Algumas respostas (de muitas):**
  - Aproximações para pequenos angulos em torno de cada eixo
  - Quaternions

Transformações Affine

- **Geometria:**
  - Rotação, escalonamento (cada dimensão separadamente), translação e cisalhamento
- **Invariantes:**
  - Linhas paralelas permanecem paralelas
  - Razão de comprimentos de segmentos numa linha permanecem fixas
- **Exemplo de aplicação:**
  - Construção de mosaicos para imagens de superfície da terra (superfície plana e câmaras relativamente não obliquas)

Transformações Affine em 2D

- **Em 2D, multiplique a localização** x **por uma matriz 2x2 arbitrária mas inversível**

$$T(x:\Theta) = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} x + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

- **generalizando,**

$$T(x:\Theta) = Ax + t$$

- **ou, na forma homogenea,**

$$T(x:\Theta) = \begin{pmatrix} A & t \\ 0^T & 1 \end{pmatrix} x + \begin{pmatrix} x \\ 1 \end{pmatrix}$$

Transformações Affine e SVD

- **Considere o SVD da matriz affine** A**:**

$$A = UWV^T$$

  - (se necessário, volte e reveja as propriedades do SVD.)
- **Fazendo**

$$Q = UV^T$$

- **Temos**

$$A = QVWV^T$$

- **A transformação geométrica**

Transformações projetivas

- **Transformação linear geral**
- **Comprimentos, angulos, e linhas paralelas MÃO são preservadas**
- **Coincidencia, tangencia, e razões de comprimentos complicadas são invariantes**
- **Aplicação:**
  - Mosaicos de imagens tiradas de câmaras girando sobre o centro ótico
  - Mosaico de imagens de superfíces planares

Algebra de transformações projetivas

- **Adicione uma linha "não zerada" abaixo na matriz homogênea:**

$$T(x;\Theta) = \begin{pmatrix} A_0 & t \\ v^T & a_{nn} \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix}$$

- **Localização do ponto transformada:**

$$x' = \frac{Ax + t}{v^T x + a_{nn}}$$

- **O calculo do estimador é um pouco complicada que a transformação affine**

Transformações não-lineares

- **Transformação usada em imagens mosaico da retina é uma função não linear das coordenadas da imagem:**

$$T(x;\Theta) = \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \theta_{14} & \theta_{15} & \theta_{16} \\ \theta_{21} & \theta_{22} & \theta_{23} & \theta_{24} & \theta_{25} & \theta_{26} \end{pmatrix} \begin{pmatrix} x^2 \\ xy \\ y^2 \\ x \\ y \\ 1 \end{pmatrix}$$

Modelos deformáveis

- **Funções não-globais**
- **Freqüentemente baseadas em PDE:**
  - Splines placas-finas
  - Mecânica de fluidos
- **Representações**
  - B-splines
  - Elementos finitos
  - Bases de Fourier
  - Bases de funções radiais

Tópicos

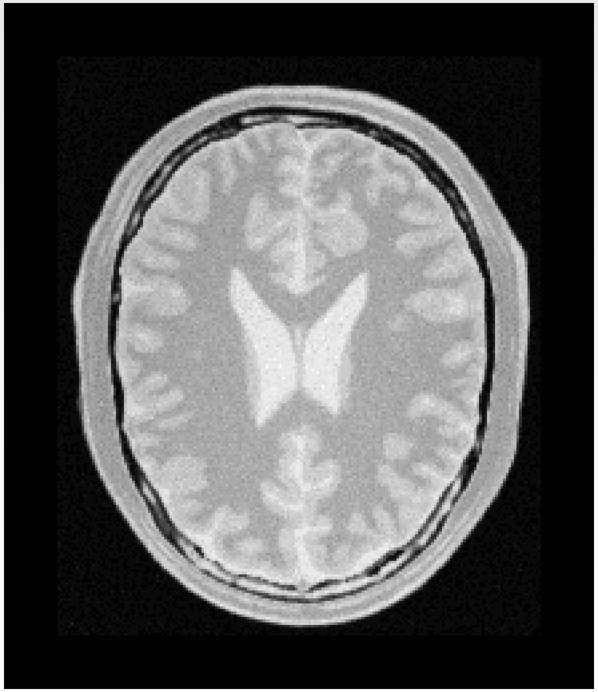
- **Exemplo**
- **Co-registro baseado em intensidade**
- **Função de erro SSD**
- **Mapeamento de imagens**
- **Minimização de funções:**
  - Gradiente descendente
  - Cálculo variacional
- **Algoritmo**
- **Resultados e discussão**

Material para leitura

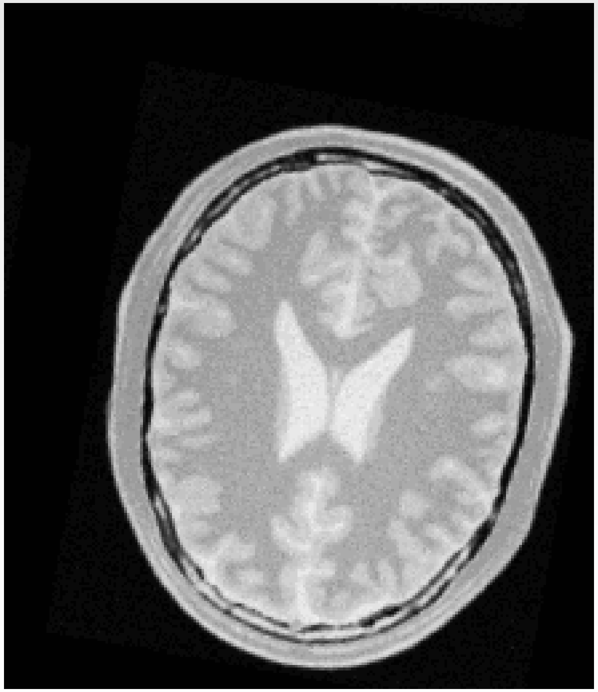
- **Paper:**
  - Hill, Batchelor, Holden and Hawkes, Medical Image Registration, *Physics of Medicine and Biology* **46** (2001) R1-R45.
- **Artigo disponível na internet**
- **Excelente introdução ao problema do co-registro, porem muito focalizado em aplicações médicas utilizando transformações de similaridade**

Exemplo

- **Co-registro de imagens de MRI, transformação de similaridade (rodada por 10 graus, com uma tranlação de 17mm e 13mm)**



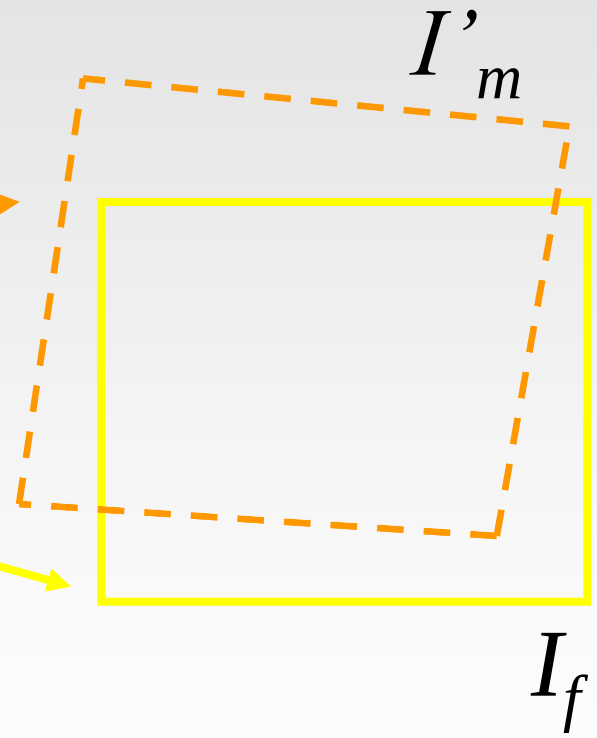
I_m



I_f

Co-registro baseado em intensidade

- **Use a intensidade mais ou menos diretamente**
- **Compare intensidades entre**
  - Versão mapeada (transformada) da imagem movimentada $\mathbf{I}_m$ (baseada na transformação estimada) e
  - Imagem fixa $\mathbf{I}_f$
- **Precisa:**
  - Medida de erro pixel-a-pixel
  - Técnica de mapeamento
  - Técnica de Minimização

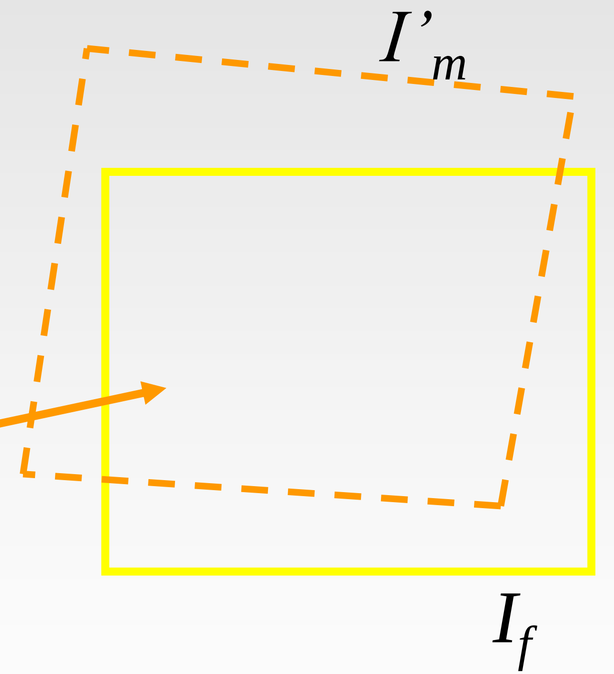


Exemplo de medida de erro: SSD

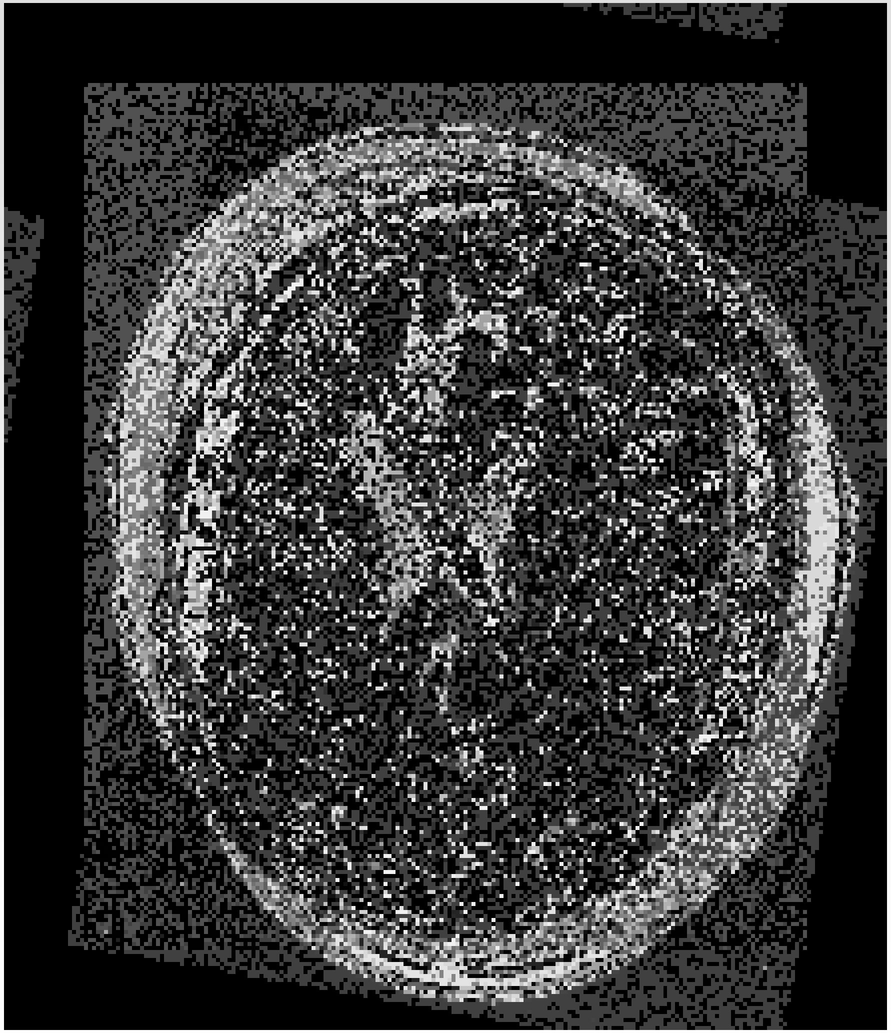
$$\sum_{p \in \Omega} \left[I_f(p) - I'_m(p) \right]^2$$

$\Omega \rightarrow$ Região de intersecção entre imagens

$\mathbf{p} \rightarrow$ Localizações de pixels dentro da região



Exemplo SSD: Alinhamento Inicial



SSD: soma dos quadrados dos erros

- **Vantagens:**
  - Simples de computar
  - Diferenciavel
  - Otimo para distribuição de erro Gaussiana
- **Desvantagens:**
  - Não permite ganho variável entre as imagens, o qual pode ser causado poriluminação diferente ou conjunto de camaras
  - Viesado por grandes erros em intensidades
    - » Ex. Causado por injeção de agente de contraste

Trabalhando nos parametros

- **Lembre:**

$$I_m^{'}(p) = I_m(T^{-1}(p:\Theta))$$

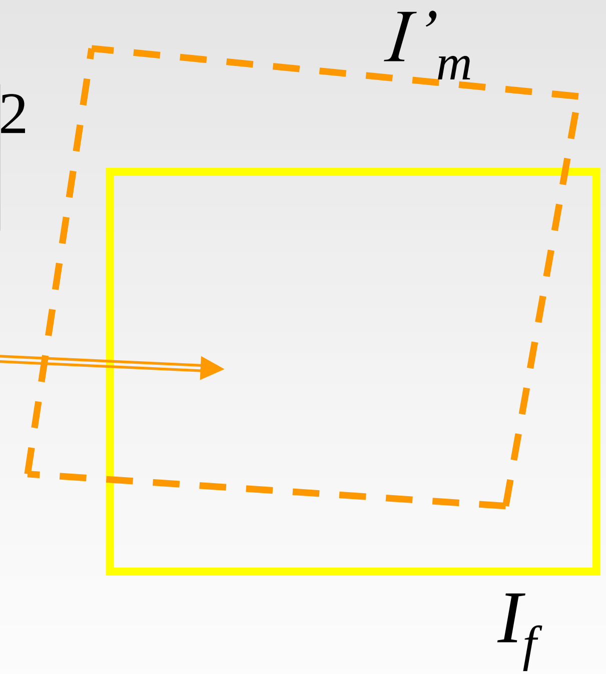
- **Isto significa que para avaliar o efeito da estimativa da transformação, o que devemos avaliar é**

$$\sum_{p\in\Omega}\left[I_f(p) - I_m(T^{-1}(p:\Theta))\right]^2$$

O papel da região

- **Observe: a região sobre a qual a transformação é avaliada depende dos parâmetros:**

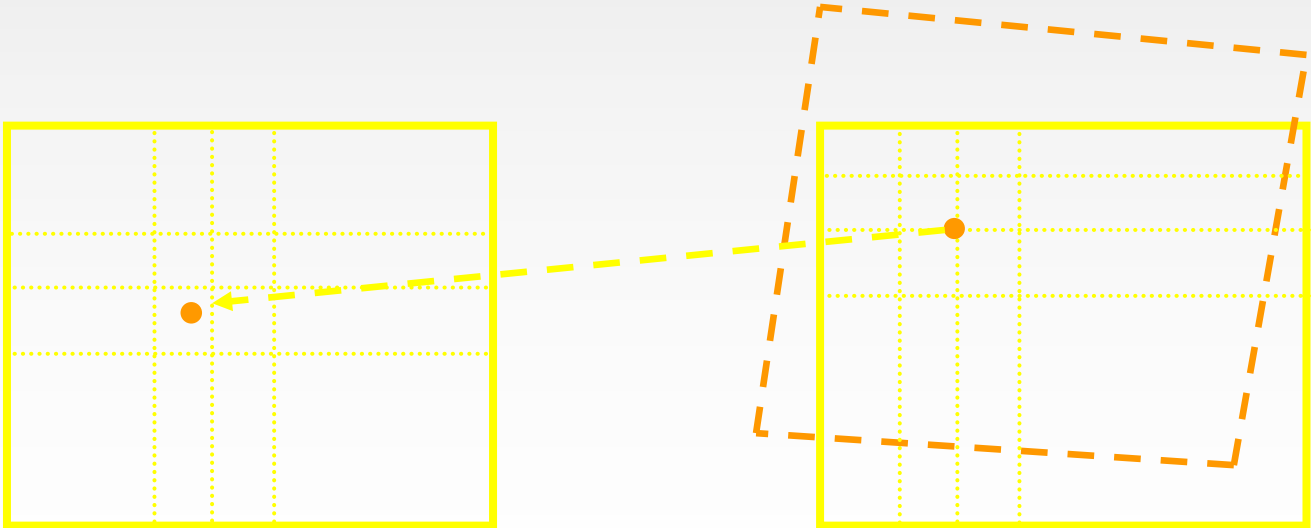
$$\sum_{\boxed{p\in\Omega}} \left[I_f(p) - I_m(T^{-1}(p:\Theta))\right]^2$$



- **Isto pode causar problemas na prática:**
  - Uma transformação resultando em nenhuma sobreposição leva a erro 0!

Avaliando função custo

- Avaliação pixel-a-pixel dentro da região
- Apique o mapeamento inverso mapping em cada pixel
- Problem: mapeamento inverse de pixel não leva a localização discreta de pixels!

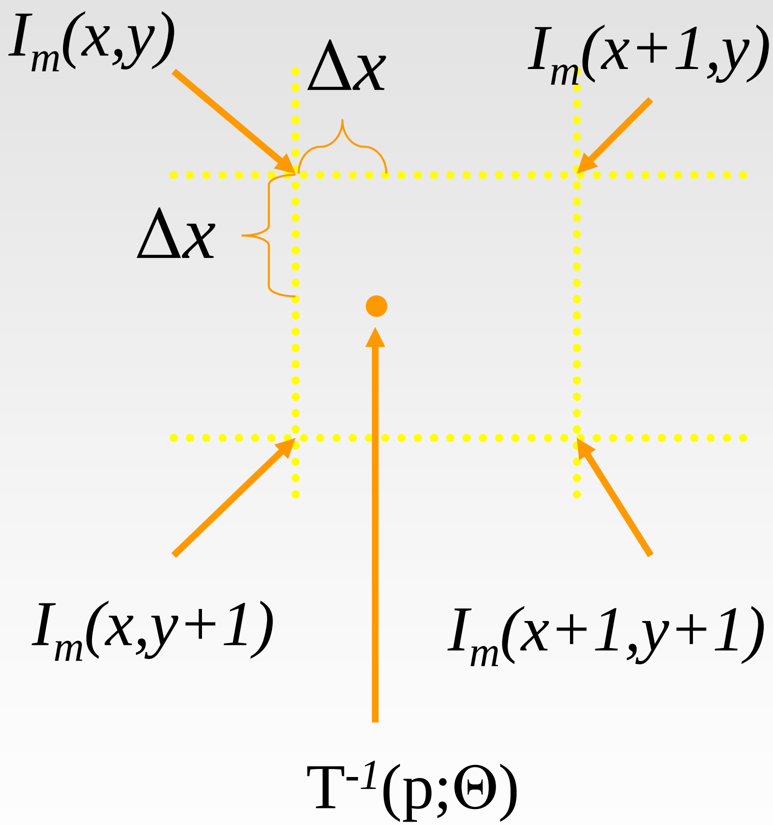


Muitas opções de interpolação

- Nearest neighbor (vizinhos próximos)
- Bilinear (ou trilinear em 3d)
- Spline
- Sinc

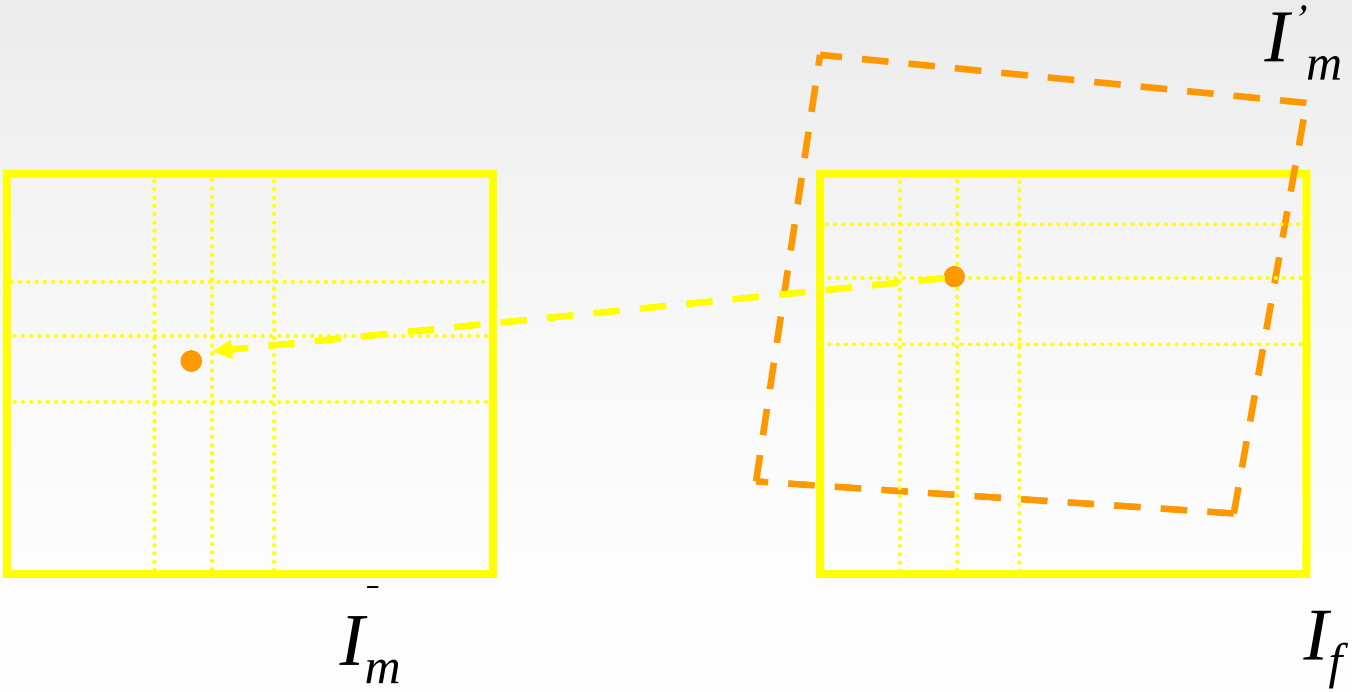
Interpolação Bilinear

- **Média ponderada sobre 4 pixels vizinhos**
  - 8 pixels vizinhos em 3d
- **Pesos proporcional as distancia em x e em y**



Duas opções na prática

- Criar intensidades, pixel-a-pixel, mas não cria uma imagem explícita I'_m
- Criar imagem real I'_m



Estagio de otimização

- **Temos:**
  - Formulado a função peso SSD;
  - Discutido como avalia-la.
- **Próximo passo é como minimizá-la em relação aosparametros de transformação.**

Antes de prosseguir

- Estimaremos os parametros da transformação reversa
- Minimizaremos a equação:

$$\sum_{p \in \Omega} \left[I_f(p) - I_m(T^{-1}(p : \Theta)) \right]^2$$

- Devemos entender (implicitamente) que esta é a tranformação inversa e os valores dos paramretros serão diferentes

Minimização de função

- **Função para minimizar:**

$$E(\Theta) = \sum_{p \in \Omega} \left[I_f(p) - I_m(T^{-1}(p;\Theta)) \right]^2$$

- **Possibilidades**
  - Método simplex - não-diferencial
  - Gradiente descendente
  - Linearização (levando ao "mínimos quadrados")
  - Método de Newton
  - Muitos outros…

Gradiente Descente

- **Compute o gradiente da função peso (com relação aos parâmetros de transformação), avaliado na estimativa atual dos parâmetros**

$$\nabla E(\Theta_t) = \frac{\partial E}{\partial \Theta}(\Theta_t)$$

- **Faça uma pequena mudança nos parâmetros na direção contrária ao gradiente**

$$\Theta_{t+1} = \Theta_t - \eta \nabla E(\Theta_t)$$

  - η é chamado “taxa de aprendizagem”
- **Reavalie a função peso e aceite se for reduzido (caso contrário reduza a taxa de aprendizagem)**
- **Continue até não ser possível mais mudanças**

Computando a derivada

- **problema:**
  - Imagens são discretas
  - Parâmetros são contínuos
- **Dois métodos**
  - Numérico
  - Contínuos (eventualmente numérico também)
- **Definição abstrata de vetor de parâmetros:**

$$\Theta_{t+1} = \left(\theta_1\ \theta_2\ \ldots \theta_k\right)^T$$

Derivadas numéricas

- **Forme cada derivada parcial tomando um pequeno passo em cada parâmetro, *i* = 1,..,*k*:**

$$\frac{\partial E}{\partial \theta_i} \approx \frac{E(\theta_1,\ldots,\theta_i+\Delta\theta_i,\ldots,\theta_k)-E(\theta_1,\ldots,\theta_i,\ldots,\theta_k)}{\Delta\theta_i}$$

- **A escolha do tamanho do passo pode ser difícil**
- **Requer *k*+1 avaliações da função para computar a derivada**
- **As vezes isto é tudo que se pode fazer!**

Computação contínua da derivada

- **Aplique a regra da cadeia:**

$$\frac{\partial E}{\partial \theta_i} = \sum_{p \in \Omega} -2 \underbrace{\left[I_f(\mathrm{p}) - I_m(\mathrm{T}(\mathrm{p};\Theta)) \right]}\,\underbrace{\frac{\partial I_m}{\partial T}}\,\underbrace{\frac{\partial \mathrm{T}}{\partial \Theta}}$$

erro corrente no pixel atual $\Delta I(\mathrm{p})$

gradiente na imagem movimentada

variação na transformação para variação nos parâmetros

Computando derivadas da imagem

- **Muitas maneiras.**
  - A mais simples é diferenças de pixels

$$I_x \equiv \frac{\partial I}{\partial x} \approx \frac{I(x+\Delta x, y) - I(x, y)}{\Delta x}$$

$$I_y \equiv \frac{\partial I}{\partial y} \approx \frac{I(x, y+\Delta y) - I(x, y)}{\Delta y}$$

  - Métodos mais sofisticados levam em conta ruídos da imagem
- **Computado a cada pixel**

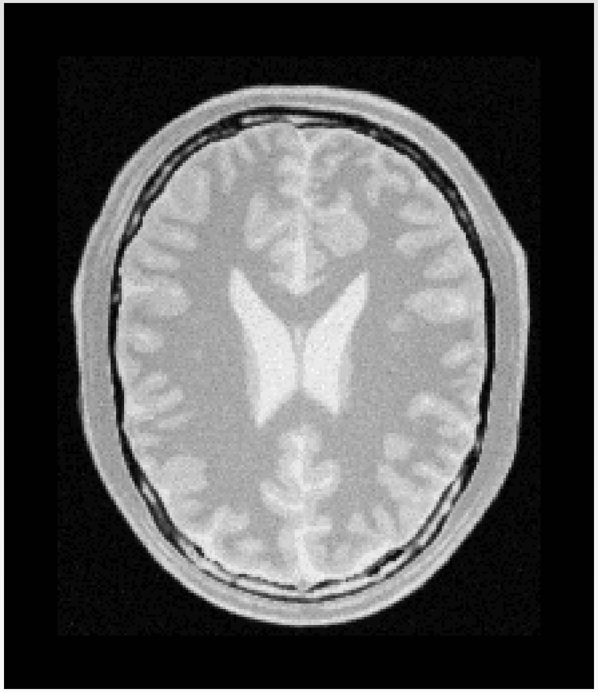
Derivada da imagem em movimeto

- **Equação**

$$\frac{\partial I_m}{\partial \mathrm{T}}(\mathrm{p}) = \left(I_{mx}\left(\mathrm{T}(\mathrm{p};\Theta)\right) \quad I_{my}\left(\mathrm{T}(\mathrm{p};\Theta)\right) \right)$$

- **Em detalhes**
  - Pre-compute derivadas na imagem movimentada $\mathbf{I}_m$
  - Durante minimização, mapeie os pixels de volta para o sistema de coordenadas da imagem movimentada e interpole

Exemplo de derivada da imagem



I_m

I_{mx}

I_{my}

dT/dθ

- **Transformação de similaridade :**

$$\mathrm{T}(\mathrm{p};\Theta)=\begin{pmatrix} ax-by+t_x \\ bx+ay+t_y \end{pmatrix}$$

  - onde

$$\Theta=\begin{pmatrix} a & b & t_x & t_y \end{pmatrix}^T \qquad \mathrm{p}=(x,y)^T$$

- **Então a derivada é a matriz 2x4 (Jacobian):**

$$\frac{\partial \mathrm{T}}{\partial \Theta}=\begin{pmatrix} x & -y & 1 & 0 \\ y & x & 0 & 1 \end{pmatrix}$$

Resumindo

- **Para cada pixel na região de sobreposição:**
  - Calcule diferença de intensidade (escalar)
  - Multiplique pelo vetor gradiente 1x2 computado mapeando a localização do pixel de volta para a imagem movimentada
  - Multiplique pela Jacobiana 2x4 da transformação, avaliada no pixel
  - O resultado é o vetor gradiente 1x4 em cada pixel
- **Some cada componente do vetor sobre todos os pixels**

Algoritmo

- **Inicialize a transformação**
- **Repita**
  - Compute gradiente
  - Dê um passo na direção do gradiente
  - Atualize a equação de mapeamento
  - Remapeie a imagem
- **Até convergência**

Initialização

- **Como é uma técnica de minimização, uma estimativa inicial é preciso,**

$$\Theta_0$$

- **Existem vária maneiras de gerar esta estimativa:**
  - Por ex., transformação identidade

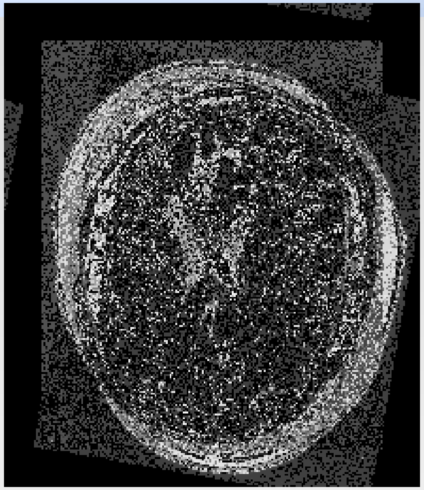
$$a = 1,\ b = 0,\ tx = ty = 0$$

  - Informação a priori
  - Técnica diferente
- **Gradiente descente acha somente um mínimo local da função peso**

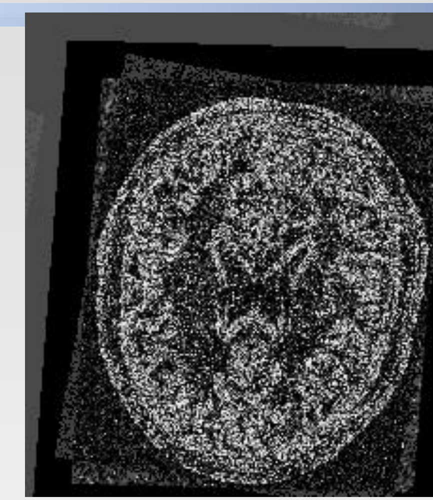
Convergência

- **Ideal é que o gradiente seja 0.**
- **Na pratica, o algoritmo é parado quando:**
  - Tamanho do passo se torna muito pequeno
  - A função peso se torna suficientemente pequena
  - Um número máximo de iterações é alcançado

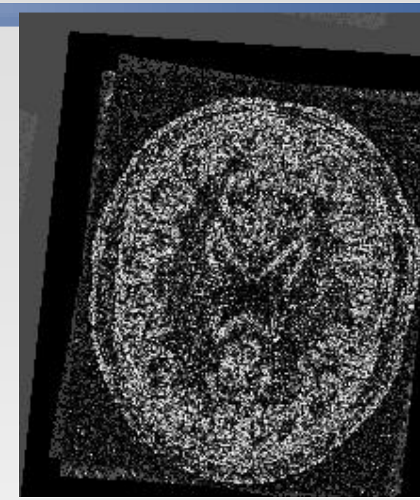
Exemplo



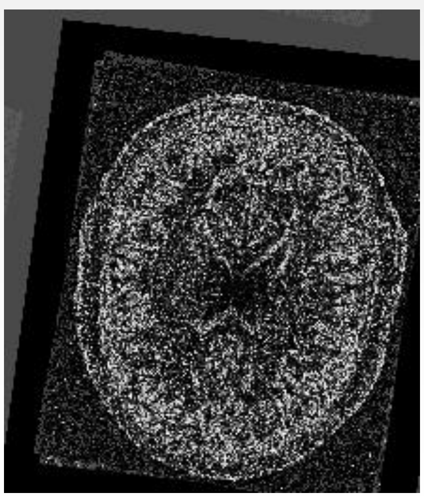
Erros iniciais



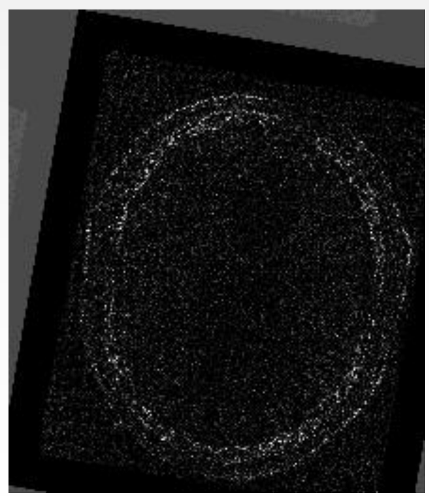
Iteração 100



Iteração 200



Iteração 300



Final: 498 iterações

Discussão

- **Gradiente descendente é simples, mas possui limitações:**
  - Mínimo local
  - Convergência lenta (linear)