Diagnostic Medical Image Processing

Rigid Image Registration

Joachim Hornegger, Dietrich Paulus, Markus Kowarschik

Lehrstuhl für Mustererkennung (Informatik 5)

Friedrich-Alexander-Universität Erlangen-Nürnberg

Diagnostic Medical Image Processing

Mathematics for Rigid Transforms

Image Registration:
Image registration is the process of transforming the different images into one common coordinate system. The registration of volumes is also subsumed by the term image registration.

Mathematics for Rigid Transforms

Dependent on the properties of the transform we have two major classes for image registration:

- The term **rigid registration** subsumes the process of computing a rigid transform for registration.
- The term **non-rigid registration** includes all the methods of deforming the different images such that they can be represented in one common coordinate system.

Fiducial Markers for Image Registration

Especially in therapeutic radiology the precise mapping of all available image information is required for the therapy of tumors. Rigid registration methods are mostly applied to images of the skull; in this particular application physicians make quite often use of fiducial markers that are fixed to the patient.

Figure: Fiducial markers used for Gamma Knife treatment

Navigation System for Surgery

Further applications of markers in medical imaging:

Figure: BrainLab system for brain surgery

Rotations and Translations in 2-D

Assume a set of corresponding 2-D points in two images:

$$C = \{(\boldsymbol{p}_k, \boldsymbol{q}_k); k = 1, 2, \ldots, N\}, \qquad (1)$$

where $\boldsymbol{p}_k, \boldsymbol{q}_k \in \mathbb{R}^2$ is the k-th pair of corresponding image points.
Problem: Compute the transform that maps the $\boldsymbol{q}_k$'s to the $\boldsymbol{p}_k$'s.

Point Based 2-D/2-D Registration

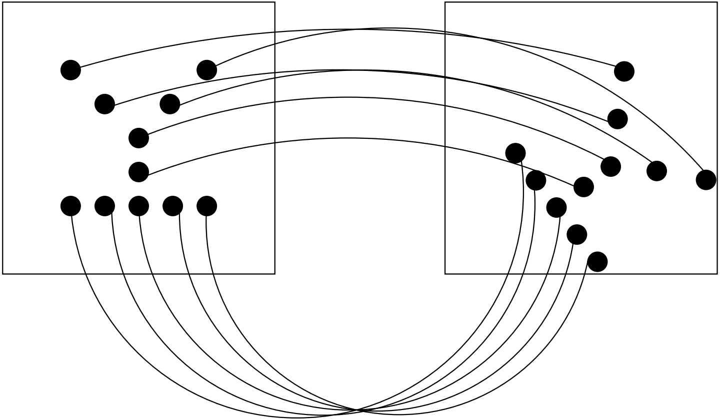


Figure: Corresponding 2-D point features

Optimization Problem

Using a rigid transform defined by rotation

$$\boldsymbol{R} = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} \tag{2}$$

and translation $\boldsymbol{t} = (t_1, t_2)^{\mathrm{T}}$, we have:

$$\boldsymbol{p}_k = \boldsymbol{R}\boldsymbol{q}_k + \boldsymbol{t} \tag{3}$$

Optimization problem for 2D rigid registration:

$$\arg\min_{\varphi, t_1, t_2} \sum_{k=1}^{N} ||\boldsymbol{p}_k - \boldsymbol{R}\boldsymbol{q}_k - \boldsymbol{t}||^2 \tag{4}$$

Conclusion: Using (4) image registration turns out to be a nonlinear optimization problem.

Properties of Rotations

Some important properties of rotation matrices $\boldsymbol{R} \in \mathbb{R}^{n \times n}$.

- columns of the rotation matrix are images of the base vectors of the original coordinate system (valid for all linear mappings!)
- orthogonality:
$$\boldsymbol{R}^{\mathsf{T}} = \boldsymbol{R}^{-1} \tag{5}$$
  and thus we have: $\boldsymbol{R}^{\mathsf{T}}\boldsymbol{R} = \boldsymbol{R}\boldsymbol{R}^{\mathsf{T}} = \boldsymbol{I}_{\mathrm{n}}$
- preserves orientation (left/right-handedness) of the coordinate system
- det ($\boldsymbol{R}$) = 1
- in 3-D: eigenvector corresponding to the eigenvalue $\lambda_1 = 1$ defines the rotation axis

Complex Numbers and Rotations

Complex numbers define a point in 2-D:

$$z = a + i\,b \tag{6}$$

where a is the real part and b the imaginary part of the complex number z.

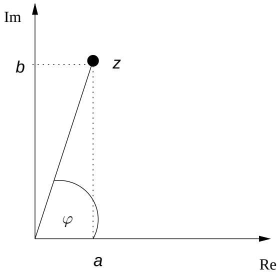


Figure: Geometric representation of complex numbers

Complex Numbers and Rotations

Proposition: Multiplication of complex numbers defines a 2-D scaling and rotation.
Multiplication of complex numbers is defined by:

$$\begin{aligned} z &= z_1 \cdot z_2 = (a_1 + i\, b_1)(a_2 + i\, b_2) \\ &= (a_1 a_2 - b_1 b_2) + i(a_1 b_2 + a_2 b_1) \end{aligned} \tag{7}$$

Complex numbers can be represented using Euler's notation:

$$z = |z| e^{i\varphi} \tag{8}$$

where

- $|z| = \sqrt{a^2 + b^2}$ is the length of the complex number and
- $\varphi = \text{atan2}\,(b, a)$ the angle

Complex Numbers and Rotations

Multiplication of complex numbers using Euler notation:

$$\begin{aligned} z_1 \cdot z_2 &= |z_1|e^{i\varphi_1} \cdot |z_2|e^{i\varphi_2} \\ &= |z_1||z_2|e^{i(\varphi_1+\varphi_2)} \end{aligned} \tag{9}$$

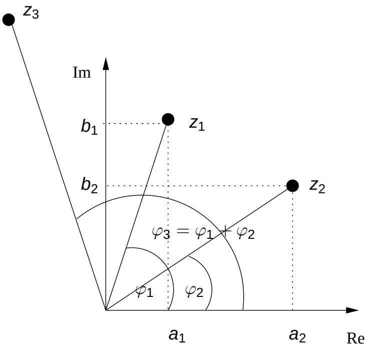


Figure: Geometric representation of complex numbers

Complex Numbers and Rotations

Conclusion: In terms of complex numbers we get for $k = 1, 2, \ldots, N$ the equations:

$$p_{k,1} + i\, p_{k,2} \quad = \quad (r_1 + i\, r_2)(q_{k,1} + i\, q_{k,2}) + t_1 + i\, t_2 \quad . \tag{10}$$

Exercise: Compute the complex number $r = r_1 + i\, r_2$ (where $r_1^2 + r_2^2 = 1$!) that corresponds to the rotation matrix $\boldsymbol{R}$.
The above equation (3) can be rewritten in two equations linear in the components of the complex numbers corresponding to $\boldsymbol{R}$ and $\boldsymbol{t}$: For the real part we get the equation:

$$p_{k,1} \quad = \quad r_1 q_{k,1} - r_2 q_{k,2} + t_1 \quad = \quad (q_{k,1}, -q_{k,2}, 1, 0) \begin{pmatrix} r_1 \\ r_2 \\ t_1 \\ t_2 \end{pmatrix} \tag{11}$$

Complex Numbers and Rotations

The imaginary parts results in the equation:

$$p_{k,2} = r_1 q_{k,2} + r_2 q_{k,1} + t_2 = (q_{k,2}, q_{k,1}, 0, 1) \begin{pmatrix} r_1 \\ r_2 \\ t_1 \\ t_2 \end{pmatrix} \quad (12)$$

The final system of linear equations is:

$$\boldsymbol{Ax} = \begin{pmatrix} q_{1,1} & -q_{1,2} & 1 & 0 \\ q_{2,1} & -q_{2,2} & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ q_{N,1} & -q_{N,2} & 1 & 0 \\ q_{1,2} & q_{1,1} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ q_{N,2} & q_{N,1} & 0 & 1 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} p_{1,1} \\ p_{2,1} \\ \vdots \\ p_{N,1} \\ p_{1,2} \\ \vdots \\ p_{N,2} \end{pmatrix} = \boldsymbol{b} \quad (13)$$

Complex Numbers and Rotations

Remarks:

- In this democratic algorithm we compute rotation as well as translation simultaneously.
- Using SVD we compute the pseudo inverse of $\boldsymbol{A}$ and thus get both rotation and translation.
- 2–D/2–D image registration using point correspondences results in a linear problem.
- Rotation matrices imply the constraint that $r_1^2 + r_2^2 = 1$. This can be enforced by a proper scaling of the solution of (13).

Question:

Can we *lift* the complex numbers to characterize 3–D rotations?

Rotations in 3-D

Various representations for rotations:

- Euler angles
- Axis / angle
- Quaternions

Euler Angle Representation

- 3–D rotation can be expressed by 3 $\times$ 3 rotation matrix
- An arbitrary rotation can be composed of 3 rotations around the axes of the coordinate system using the angles φ_{x} (roll), φ_{y} (pitch), φ_{z} (yaw):

$$\boldsymbol{R} = \boldsymbol{R}_{\mathrm{x}}\boldsymbol{R}_{\mathrm{y}}\boldsymbol{R}_{\mathrm{z}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi_{\mathrm{x}} & -\sin\varphi_{\mathrm{x}} \\ 0 & \sin\varphi_{\mathrm{x}} & \cos\varphi_{\mathrm{x}} \end{pmatrix} \begin{pmatrix} \cos\varphi_{\mathrm{y}} & 0 & \sin\varphi_{\mathrm{y}} \\ 0 & 1 & 0 \\ -\sin\varphi_{\mathrm{y}} & 0 & \cos\varphi_{\mathrm{y}} \end{pmatrix} \begin{pmatrix} \cos\varphi_{\mathrm{z}} & -\sin\varphi_{\mathrm{z}} & 0 \\ \sin\varphi_{\mathrm{z}} & \cos\varphi_{\mathrm{z}} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and thus we have

$$\boldsymbol{R} = \begin{pmatrix} \cos\varphi_{\mathrm{y}}\cos\varphi_{\mathrm{z}} & -\cos\varphi_{\mathrm{y}}\sin\varphi_{\mathrm{z}} & \sin\varphi_{\mathrm{y}} \\ \sin\varphi_{\mathrm{x}}\sin\varphi_{\mathrm{y}}\cos\varphi_{\mathrm{z}}+\cos\varphi_{\mathrm{x}}\sin\varphi_{\mathrm{z}} & -\sin\varphi_{\mathrm{x}}\sin\varphi_{\mathrm{y}}\sin\varphi_{\mathrm{z}}+\cos\varphi_{\mathrm{x}}\cos\varphi_{\mathrm{z}} & -\sin\varphi_{\mathrm{x}}\cos\varphi_{\mathrm{y}} \\ -\cos\varphi_{\mathrm{x}}\sin\varphi_{\mathrm{y}}\cos\varphi_{\mathrm{z}}+\sin\varphi_{\mathrm{x}}\sin\varphi_{\mathrm{z}} & \cos\varphi_{\mathrm{x}}\sin\varphi_{\mathrm{y}}\sin\varphi_{\mathrm{z}}+\sin\varphi_{\mathrm{x}}\cos\varphi_{\mathrm{z}} & \cos\varphi_{\mathrm{x}}\cos\varphi_{\mathrm{y}} \end{pmatrix}$$

Euler Angle Representation

This is the most popular parameterization of the rotation matrix $\boldsymbol{R}$

$\boxed{\textbf{NB}\ \text{order is essential for the resulting rotation matrix.}}$

- matrix multiplication is not commutative:

$$\boldsymbol{R}_x\boldsymbol{R}_y\boldsymbol{R}_z \neq \boldsymbol{R}_y\boldsymbol{R}_x\boldsymbol{R}_z \qquad , \tag{14}$$

- for small rotation angles commutativity is approximately true
- Gimbal Lock (Shoemaker):

  When object points are first rotated around the x-axis by $-\frac{\pi}{2}$, then, the y- and the z-axis are aligned and the rotations around the y- and z-axis, respectively, can no longer be distinguished.

- conversion between angles and matrices is computationally not very robust
- representation is not unique and there exist singularities

Axis-Angle Representation

Before we introduce the commonly used axis-angle representation of rotations, we briefly consider the linearity of the cross-product.
For 3-D vectors we have:

$$\boldsymbol{u} \times \boldsymbol{v} = [\boldsymbol{u}]_\times \boldsymbol{v}$$

The skew matrix is defined by:

$$[\boldsymbol{u}]_\times = \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix} . \qquad (15)$$

The matrix $[\boldsymbol{u}]_\times$ is called the *skew matrix* of $\boldsymbol{u}$.

Axis-Angle Representation

Alternatively to the Euler representation, a rotation $\boldsymbol{R}$ can be represented as a rotation with respect to a single unit norm axis $\boldsymbol{u}$ by the angle Θ

Given: rotation axis $\boldsymbol{u} = (u_1, u_2, u_3)^{\mathrm{T}}$ and angle Θ

Compute: rotation matrix $\boldsymbol{R}$

Solution:

$$\boldsymbol{R} = f(\boldsymbol{u}, \Theta) = \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} + \left(\boldsymbol{I}_3 - \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}\right) \cdot \cos\Theta + [\boldsymbol{u}]_{\times} \sin\Theta \tag{16}$$

or in components:

$$\boldsymbol{R} = \begin{pmatrix} u_1^2 + (1 - u_1^2)\cos\Theta & u_1u_2(1-\cos\Theta) - u_3\sin\Theta & u_1u_3(1-\cos\Theta) + u_2\sin\Theta \\ u_1u_2(1-\cos\Theta) + u_3\sin\Theta & u_2^2 + (1-u_2^2)\cos\Theta & u_2u_3(1-\cos\Theta) - u_1\sin\Theta \\ u_1u_3(1-\cos\Theta) - u_2\sin\Theta & u_2u_3(1-\cos\Theta) + u_1\sin\Theta & u_3^2 + (1-u_3^2)\cos\Theta \end{pmatrix}$$

Axis-Angle Representation

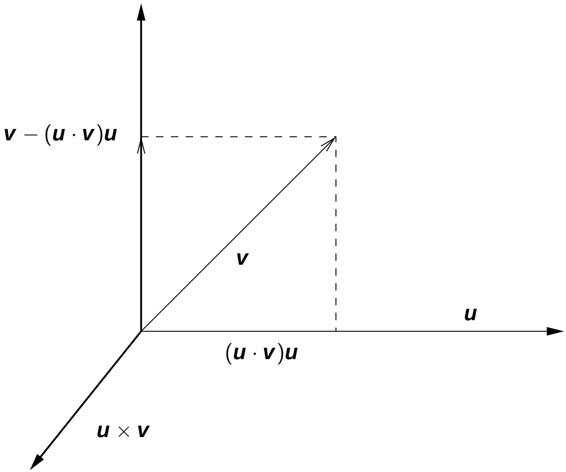


Figure: Orthogonal base vectors of the coordinate system

Axis-Angle Representation

The rotated vector $\boldsymbol{Rv}$ can be written as a linear combination of $\boldsymbol{u} \times \boldsymbol{v}$, $(\boldsymbol{u} \cdot \boldsymbol{v})\boldsymbol{u}$ and $\boldsymbol{v} - (\boldsymbol{u} \cdot \boldsymbol{v})\boldsymbol{u}$.

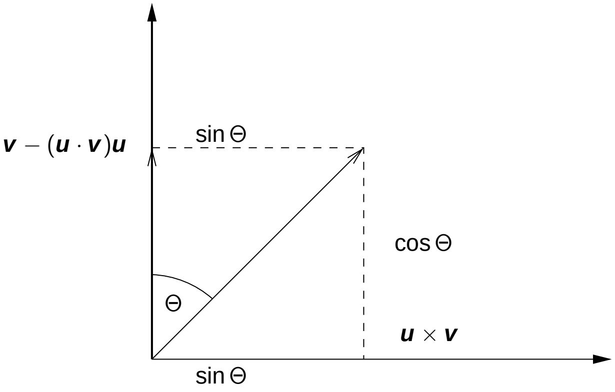


Figure: Coordinate transform implied by rotation around the $\boldsymbol{u}$-axis

Axis-Angle Representation

Formula of Rodrigues:

$$\boldsymbol{R} = f(\boldsymbol{u}, \Theta) = \boldsymbol{u}\boldsymbol{u}^{\mathsf{T}} + \left(\boldsymbol{I}_3 + \boldsymbol{u}\boldsymbol{u}^{\mathsf{T}}\right) \cdot \cos\Theta + [\boldsymbol{u}]_{\times} \sin\Theta \tag{17}$$

i.e. if axis and angle are known $\Rightarrow$ computation of rotation matrix is possible. We require that

$$\boldsymbol{u} = (u_1\ u_2\ u_3)^{\mathsf{T}} \quad \text{with} \quad \|\boldsymbol{u}\|_2 = 1 \tag{18}$$

NB: this description still has three degrees of freedom, two for the direction of the rotation axis and one for the angle

Axis-Angle Representation

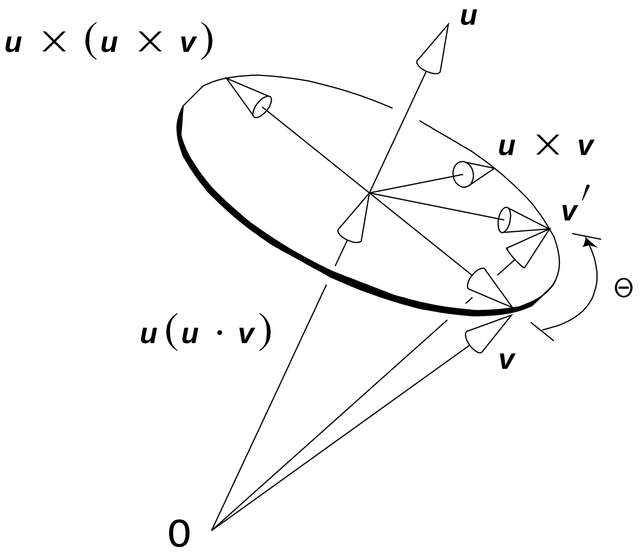


Figure: Formula of Rodrigues

Axis-Angle Representation

Conversely: θ and $\boldsymbol{u}$ from $\boldsymbol{R}$ by eigenvalues and eigenvectors of $\boldsymbol{R}$

- eigenvalues of $\boldsymbol{R}$: $1, \cos\theta + i\sin\theta, \cos\theta - i\sin\theta$
- eigenvector for eigenvalue 1 of $\boldsymbol{R}$: collinear with $\boldsymbol{u}$

Diagnostic Medical Image Processing

Quaternion representation

Rotations in $\mathbb{R}^3$ can be elegantly described by so-called quaternions. Quaternions can be understood as an extension of complex numbers: Three different numbers that are all square roots of -1:

$$\mathrm{i} * \mathrm{i} = -1 \qquad \mathrm{j} * \mathrm{j} = -1 \qquad \mathrm{k} * \mathrm{k} = -1 \tag{19}$$

The products of these numbers are defined as

$$\mathrm{i} * \mathrm{j} = -\mathrm{j} * \mathrm{i} = \mathrm{k} \qquad \mathrm{j} * \mathrm{k} = -\mathrm{k} * \mathrm{j} = \mathrm{i} \qquad \mathrm{k} * \mathrm{i} = -\mathrm{i} * \mathrm{k} = \mathrm{j} \quad . \tag{20}$$

NB: compare this to cross products of unit vectors

Quaternions

Definition

A *quaternion* is a linear combination $\mathbf{r} = w + x\mathrm{i} + y\mathrm{j} + z\mathrm{k}$ where $w, x, y, z \in \mathbb{R}$.

Similar to complex numbers we define:

Definition

The conjugate $\bar{\mathbf{r}}$ and the magnitude $|\mathbf{r}|$ of a quaternion $\mathbf{r} = w + x\mathrm{i} + y\mathrm{j} + z\mathrm{k}$ are

$$\bar{\mathbf{r}} = w - x\mathrm{i} - y\mathrm{j} - z\mathrm{k} \quad (21)$$

$$|\mathbf{r}| = \sqrt{\mathbf{r} * \bar{\mathbf{r}}} = \sqrt{w^2 + x^2 + y^2 + z^2} \quad (22)$$

Quaternions

Definition

A quaternion $\mathbf{r}$ which has length 1 is called a *unit quaternion*.

A few important properties of quaternions:

1. multiplication and summation are associative,
2. multiplication is *not* commutative: $\mathbf{r}_1\mathbf{r}_2 \neq \mathbf{r}_2\mathbf{r}_1$, $\Rightarrow$ quaternions are no algebraic field[1]
3. Conjugate: $\bar{\mathbf{r}} = w - xi - yj - zk$,
4. Norm: $|\mathbf{r}| = \sqrt{\mathbf{r} \cdot \bar{\mathbf{r}}} = \sqrt{w^2 + x^2 + y^2 + z^2}$,
5. Unit quaternions: $|\mathbf{r}| = 1 \Rightarrow \mathbf{r}^{-1} = \bar{\mathbf{r}}$.

[1] Actually, they form a division ring.

Multiplication of Quaternions

Definition

We represent quaternions by a row vector $(w, x, y, z) = (w, \mathbf{v})$ where $\mathbf{v}^T = (x, y, z)$.

In this notation the product of two quaternions $\mathbf{r}_1 = (w_1, \mathbf{v}_1)$ and $\mathbf{r}_2 = (w_2, \mathbf{v}_2)$ is given by

$$\mathbf{r}_1 * \mathbf{r}_2 \quad = \quad \left(w_1 w_2 - {\mathbf{v}_1}^{\mathrm{T}} \mathbf{v}_2 \; , \; w_1 \mathbf{v}_2 + w_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2 \right) \tag{23}$$

Multiplication of Quaternions

Using the other notation we get:

$$\begin{aligned} \mathbf{r}_1 &= w_1 + x_1\mathrm{i} + y_1\mathrm{j} + z_1\mathrm{k} \\ \mathbf{r}_2 &= w_2 + x_2\mathrm{i} + y_2\mathrm{j} + z_2\mathrm{k} \\ \mathbf{r}_1 * \mathbf{r}_2 &= (w_1w_2 - x_1x_2 - y_1y_2 - z_1z_2) \\ &+ (w_1x_2 + x_1w_2 + y_1z_2 - z_1y_2)\mathrm{i} \\ &+ (w_1y_2 - x_1z_2 + y_1w_2 + z_1x_2)\mathrm{j} \\ &+ (w_1z_2 + x_1y_2 - y_1x_2 + z_1w_2)\mathrm{k} \quad . \end{aligned} \tag{24}$$

Multiplication of Quaternions

This quaternion product can be rewritten in matrix notation. For that purpose define the matrix ${}_*[\mathbf{r}_2]$ such that $\mathbf{r}_1 \ {}_*[\mathbf{r}_2] = \mathbf{r}_1 * \mathbf{r}_2$:
where the matrix ${}_*[\mathbf{r}_2]$ is given by

$$
{}_*[\mathbf{r}_2] = \begin{pmatrix} w_2 & x_2 & y_2 & z_2 \\ -x_2 & w_2 & -z_2 & y_2 \\ -y_2 & z_2 & w_2 & -x_2 \\ -z_2 & -y_2 & x_2 & w_2 \end{pmatrix} \quad . \tag{25}
$$

Note: This matrix shows similarities to the skew matrix that is used to express the cross product of vectors by matrix multiplication where $\boldsymbol{x} \times \boldsymbol{y} = [\boldsymbol{x}]_\times \boldsymbol{y}$

Rotation Quaternion

Let

- $\boldsymbol{p} \in \mathbb{R}^3$ a 3-D point to be rotated,
- $\boldsymbol{u} \in \mathbb{R}^3$ the axis of rotation, $|\boldsymbol{u}| = 1$,
- $\Theta \in \mathbb{R}$ the angle of rotation.

Definition

- The rotation quaternion according to a rotation given in axis/angle representation is defined by:

$$\mathbf{r} = \left(\cos \frac{\Theta}{2}, \sin \frac{\Theta}{2} \cdot \boldsymbol{u}^T \right) \tag{26}$$

- The quaternion associated with a 3-D point $\boldsymbol{p}$ is defined by $\mathbf{p}' = (0, \boldsymbol{p})$.

Rotation Quaternion

Then the rotation of $\boldsymbol{p}$ can be computed by:

$$\mathbf{p}'_{\mathrm{rot}} = \mathbf{r} * \mathbf{p}' * \bar{\mathbf{r}} \tag{27}$$

Note:

- The quaternion $\boldsymbol{p}'_{\mathrm{rot}}$ should be $(0, \boldsymbol{p}_{\mathrm{rotated}})$. Actually, we could put any value into the scalar part of $\boldsymbol{P}$, i.e. $\boldsymbol{P} = (c, \boldsymbol{p})$ and after performing the quaternion multiplication, we should get back $\boldsymbol{P}_{\mathrm{rotated}} = (c, \boldsymbol{p}_{\mathrm{rotated}})$.
- You may want to confirm that $\boldsymbol{r}$ is a *unit quaternion*, since that will allow us to use the fact that the inverse of $\boldsymbol{r}$ is $\bar{\boldsymbol{r}}$ if $\boldsymbol{r}$ is a unit quaternion, i.e. $\|\mathbf{r}\| = 1$, $\mathbf{r}^{-1} = \bar{\mathbf{r}}$

Marker Based Image Registration

Let us summarize the results so far:

- Complex numbers allowed us to treat rotations and translations simultaneously.
- Quaternions are limited to 3-D rotation. Translation must be dealt with separately.
- Next section: Dual quaternions provide a framework to incorporate both rotations and translations.

Estimation of 3-D Rotation

The optimization problem to estimate the 3-D rotation is:

$$\widehat{\boldsymbol{R}} = \operatorname{argmin}_{\boldsymbol{R}} \sum_{i=1}^{N} \|\boldsymbol{p}_{\mathrm{rot},i} - \boldsymbol{R} \cdot \boldsymbol{p}_i\|^2$$

Using quaternions for representing rotations we get the following relationship between the original and the rotated points:

$$\begin{aligned} (0, \boldsymbol{p}'_i) &= \boldsymbol{q}(0, \boldsymbol{p}_i)\bar{\boldsymbol{q}} && (28) \\ (0, \boldsymbol{p}'_i)\boldsymbol{q} &= \boldsymbol{q}(0, \boldsymbol{p}_i) && (29) \end{aligned}$$

and thus we get the optimization problem:

$$\operatorname{argmin}_{\boldsymbol{q}} \sum_{i=1}^{N} ||(0, \boldsymbol{p}'_i)\boldsymbol{q} - \boldsymbol{q}(0, \boldsymbol{p}_i)||^2$$

Epipolar Geometry Revisited

The decomposition of the essential matrix $\boldsymbol{E}$ can be done by using quaternions.

- essential matrix is defined by

$$\boldsymbol{E} \quad = \quad \boldsymbol{R}[\boldsymbol{t}]_\times \quad . \tag{30}$$

- translation vector spans the kernel of $\boldsymbol{E}$
- least square estimator:

$$||\boldsymbol{E} - \boldsymbol{R}[\boldsymbol{t}]_x||^2 \quad \rightarrow \quad \text{min} \quad , \tag{31}$$

where $\boldsymbol{R}\boldsymbol{R}^\mathrm{T} = \boldsymbol{1}$ and $\det(R) = 1$.

Epipolar Geometry Revisited

Closed form Solution:
If the SVD of the essential matrix is $\boldsymbol{E} = \boldsymbol{U}^{\circ}\boldsymbol{V}^{\mathrm{T}}$ then the rotation is:

$$\boldsymbol{R} = \boldsymbol{U} \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & \det(\boldsymbol{U}) & \det(\boldsymbol{V}) \end{pmatrix} \boldsymbol{V}^{T} \tag{32}$$

Epipolar Geometry Revisited

The objective function for the estimation of the essential matrix can be rewritten in the following form:

$$\mathrm{argmin}_{\boldsymbol{R}} ||\boldsymbol{E} - \boldsymbol{R}[\boldsymbol{t}]_x||^2 \;=\; \mathrm{argmin}_{\boldsymbol{R}} \sum_{i=1}^{3} ||\boldsymbol{p}_i' - \boldsymbol{R} \cdot \boldsymbol{p}_i||^2$$

Using quaternions for representing rotations we get:

$$\mathrm{argmin}_{\boldsymbol{q}} \sum_{i=1}^{3} ||(0, \boldsymbol{p}_i') - \boldsymbol{q}(0, \boldsymbol{p}_i)\bar{\boldsymbol{q}}||^2 \;=\; \mathrm{argmin}_{\boldsymbol{q}} \sum_{i=1}^{3} ||(0, \boldsymbol{p}_i')\boldsymbol{q} - \boldsymbol{q}(0, \boldsymbol{p}_i)||^2$$

Conclusion:
The objective function is linear in the rotation quaternion. The rotation can be estimated by solving a system of linear equations.

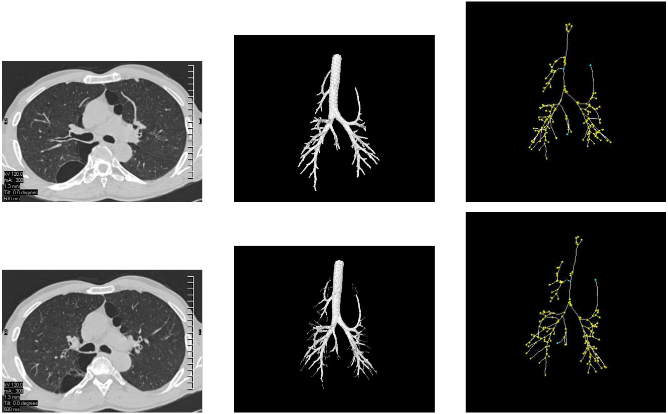
Example of 3-D/3-D Registration

Rigid Registration of the Airways:

Segmentation | Skeletonization | Registration

Patient A

Patient B



From Quaternions to Dual Quaternions

What do we have so far?

- Complex numbers allowed us to treat rotations and translations simultaneously.
- Rotation matrices lead to a non-linear optimization problem.
- Quaternions are limited to 3-D rotation. Translation must be dealt with separately.

Dual Numbers (1)

Dual numbers were introduced 1873 by William Kingdon Clifford

- Dual numbers are defined as follows:

$$z = a + \epsilon\, b \quad \text{with} \quad \epsilon^2 = 0. \tag{33}$$

- $a \in \mathbb{R}$ is called the real part
- $b \in \mathbb{R}$ is called the dual part
- conjugate of z is defined by:

$$\bar{z} = a - \epsilon b \tag{34}$$

Dual Numbers (2)

Properties of dual numbers:

- addition:

$$z_1 + z_2 = (a_1 + \epsilon b_1) + (a_2 + \epsilon b_2) = a_1 + a_2 + \epsilon(b_1 + b_2); \quad (35)$$

- multiplication:

$$z_1 z_2 = (a_1 + \epsilon b_1)(a_2 + \epsilon b_2) = a_1 a_2 + \epsilon(a_1 b_2 + b_1 a_2) \quad (36)$$

- Taylor expansion of functions with dual arguments:

$$f(z) = f(a + \epsilon\, b) = f(a) + \epsilon\, b f'(a), \quad (37)$$

  where $f'(z)$ denotes the first derivative of $f(z)$.
- dual numbers form an Abelian Ring Δ according to addition and multiplication

Dual Vectors

Dual vectors in Δ^3, where the real and dual parts are orthogonal vectors represent lines in $\mathbb{R}^3$ with direction $\boldsymbol{l}$ through a point $\boldsymbol{p}$ and:

- real part $\boldsymbol{l}$: direction of line
- dual part $\boldsymbol{m} = \boldsymbol{l} \times \boldsymbol{p}$: line moment

These vectors are called *Plücker coordinates*.

Constraints:

- $\|\boldsymbol{l}\| = 1$ and
- $\boldsymbol{l}^T \boldsymbol{m} = 0$.

Thus the dof of an arbitrary line equals four.

Dual Quaternions (1)

A dual quaternion $\widehat{\boldsymbol{q}}$ is defined by:

$$\widehat{\boldsymbol{q}} = (s, \boldsymbol{q}), \tag{38}$$

where

- s is a dual number, and
- $\boldsymbol{q} \in \Delta^3$ is a dual vector .

or alternatively:
A dual quaternion is defined by

$$\widehat{\boldsymbol{q}} = \boldsymbol{a} + \epsilon \boldsymbol{b} \tag{39}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are each quaternions.

Dual Quaternions (2)

A few **operations** on dual quaternions:

- Squared magnitude

$$||\widehat{\boldsymbol{q}}||^2 = \widehat{\boldsymbol{q}}\widehat{\boldsymbol{q}}^{\mathsf{T}} = \boldsymbol{a}\bar{\boldsymbol{a}} + \epsilon(\boldsymbol{a}\bar{\boldsymbol{b}}\boldsymbol{b}\bar{\boldsymbol{a}}) \quad (40)$$

- Addition:

$$\widehat{\boldsymbol{q}}_1 + \widehat{\boldsymbol{q}}_2 = (s_1 + s_2, \boldsymbol{q}_1 + \boldsymbol{q}_2). \quad (41)$$

- Scaling:

$$\lambda \cdot (s, \boldsymbol{q}) = (\lambda s, \lambda \boldsymbol{q}) \quad (42)$$

- Multiplication:

$$\widehat{\boldsymbol{q}}_1 \cdot \widehat{\boldsymbol{q}}_2 = (s_1 s_2 - \widehat{\boldsymbol{q}}_1{}^{\mathsf{T}}\widehat{\boldsymbol{q}}_2, s_1\boldsymbol{q}_1 + s_2\boldsymbol{q}_1 + \boldsymbol{q}_1 \times \boldsymbol{q}_2) \quad (43)$$

Dual Quaternions (3)

Further Properties:

- Conversion of a dual vector (line) $\boldsymbol{q}$ to a dual quaternion $\widehat{\boldsymbol{q}}$:
  $\widehat{\boldsymbol{q}} = (0, \boldsymbol{q})$.
- unit dual quaternion:

$$||\widehat{\boldsymbol{q}}||^2 = 1 + \epsilon\, 0. \tag{44}$$

- Unity Condition $||\widehat{\boldsymbol{q}}|| = \widehat{\boldsymbol{q}}\bar{\widehat{\boldsymbol{q}}}$ =>

$$\boldsymbol{a}\bar{\boldsymbol{a}} = 1 \quad \text{and} \quad \bar{\boldsymbol{a}}\boldsymbol{b} + \bar{\boldsymbol{b}}\boldsymbol{a} = 0. \tag{45}$$

Transformation of Lines

Given:
A line given by its dual quaternion $\widehat{\boldsymbol{l}}_a = \boldsymbol{l}_a + \epsilon \boldsymbol{m}_a$ that is transformed to a line $\widehat{\boldsymbol{l}}_b = \boldsymbol{l}_b + \epsilon \boldsymbol{m}_b$. The transformation is defined by a rotation matrix $\boldsymbol{R}$ and a translation vector $\boldsymbol{t}$.

Problem:
Show that there exists a dual quaternion $\widehat{\boldsymbol{q}}$ such that $\widehat{\boldsymbol{l}}_a = \widehat{\boldsymbol{q}} \widehat{\boldsymbol{l}}_b \bar{\widehat{\boldsymbol{q}}}$.

Transformation of lines (2)

Proof:

1. Apply transformation parameters:

$$
\begin{aligned}
\boldsymbol{l}_a &= \boldsymbol{R}\boldsymbol{l}_b, && (46)\\
\boldsymbol{m}_a &= \boldsymbol{p}_a \times \boldsymbol{l}_a = (\boldsymbol{R}\boldsymbol{p}_b + \boldsymbol{t}) \times \boldsymbol{R}\boldsymbol{l}_b && (47)\\
&= \boldsymbol{R}(\boldsymbol{p}_b \times \boldsymbol{l}_b) + \boldsymbol{t} \times \boldsymbol{R}\boldsymbol{l}_b && (48)\\
&= \boldsymbol{R}\boldsymbol{m}_b + \boldsymbol{t} \times \boldsymbol{R}\boldsymbol{l}_b && (49)
\end{aligned}
$$

2. Substitute quaternions:

$$
\begin{aligned}
\boldsymbol{l}_a &= \boldsymbol{q}\boldsymbol{l}_b\bar{\boldsymbol{q}}, && (50)\\
\boldsymbol{m}_a &= \boldsymbol{q}\boldsymbol{m}_b\bar{\boldsymbol{q}} + \frac{1}{2}(\boldsymbol{q}\boldsymbol{l}_b\bar{\boldsymbol{q}}\bar{\boldsymbol{t}} + \boldsymbol{t}\boldsymbol{q}\boldsymbol{l}_b\bar{\boldsymbol{q}}) && (51)
\end{aligned}
$$

Estimation of the Transformation Parameters

Theorem (Walker & Daniilidis)
Rotation and translation of a 3-D line $\widehat{\boldsymbol{l}}_a$ (represented as dual quaternions) is defined by

$$\widehat{\boldsymbol{l}}_a = \widehat{\boldsymbol{q}}\widehat{\boldsymbol{l}}_b\bar{\widehat{\boldsymbol{q}}}, \tag{52}$$

where the dual quaternion $\widehat{\boldsymbol{q}} = \boldsymbol{a} + \epsilon \boldsymbol{b}$ is defined by

$$\widehat{\boldsymbol{q}} = \boldsymbol{r} + \frac{1}{2}\epsilon \boldsymbol{t}\boldsymbol{r} \tag{53}$$

with the unit quaternion $\boldsymbol{r}$ representing the rotation and the quaternion $\boldsymbol{t}$ representing the translation.

Estimation using SVD

1. Split the fundamental equation (52) into non-dual and dual parts:

$$\boldsymbol{l}_a = \boldsymbol{a}\boldsymbol{l}_b\bar{\boldsymbol{a}}, \tag{54}$$
$$\boldsymbol{m}_a = \boldsymbol{a}\boldsymbol{l}_b\bar{\boldsymbol{b}} + \boldsymbol{a}\boldsymbol{m}_b\bar{\boldsymbol{a}} + \boldsymbol{b}\boldsymbol{l}_b\bar{\boldsymbol{a}} \tag{55}$$

2. Multiply on the right with $\boldsymbol{a}$ and use $\bar{\boldsymbol{a}}\boldsymbol{b} + \bar{\boldsymbol{b}}\boldsymbol{a} = 0$

$$\boldsymbol{l}_a\boldsymbol{a} = \boldsymbol{a}\boldsymbol{l}_b, \tag{56}$$
$$\boldsymbol{m}_a\boldsymbol{a} = -\boldsymbol{l}_a\boldsymbol{b} + \boldsymbol{a}\boldsymbol{m}_b + \boldsymbol{b}\boldsymbol{l}_b \tag{57}$$

3. Rearrange to

$$\boldsymbol{l}_a\boldsymbol{a} - \boldsymbol{a}\boldsymbol{l}_b = 0, \tag{58}$$
$$(\boldsymbol{m}_a\boldsymbol{a} - \boldsymbol{a}\boldsymbol{m}_b) + (\boldsymbol{l}_a\boldsymbol{b} - \boldsymbol{b}\boldsymbol{l}_b) = 0 \tag{59}$$

Estimation using SVD (2)

These equations can be written as a matrix vector equation:

$$\begin{pmatrix} \boldsymbol{l}_a - \boldsymbol{l}_b & [\boldsymbol{l}_a + \boldsymbol{l}_b]_\times & \boldsymbol{0}_{3\times 1} & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{m}_a - \boldsymbol{m}_b & [\boldsymbol{m}_a + \boldsymbol{m}_b]_\times & \boldsymbol{l}_a - \boldsymbol{l}_b & [\boldsymbol{l}_a + \boldsymbol{l}_b]_\times \end{pmatrix} \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{pmatrix} = \boldsymbol{0} \tag{60}$$

Properties:

- Dimension of the matrix $\boldsymbol{S}$ is 6×8.
- Vector of unknowns $(\boldsymbol{a}^\mathrm{T}, \boldsymbol{b}^\mathrm{T})$ has a dimension of 8.
- DoF is 6 (constraints on unit quaternion).
- Two redundant equations (Plcker Coordinates).

At least two lines are required for the estimation.

Projections from 3-D to 2-D

Motivation

2-D/3-D image fusion is important for applications where volume data and X-ray projections have to be registered.

Figure: 2-D/3-D image fusion

Projection Models

In computer vision different projection models are used:

1. orthographic projection
2. perspective projection
3. weak perspective projection
4. paraperspective projection

Illustration of Different Projection Models

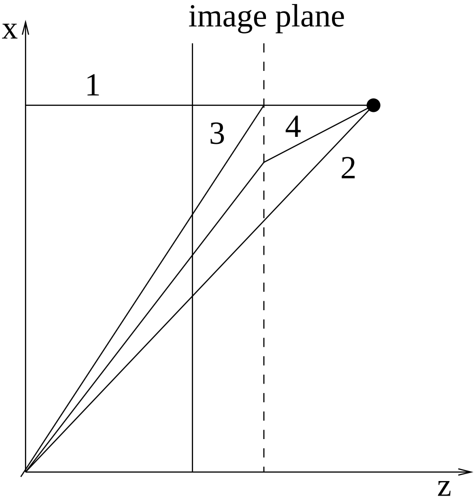


Figure: Projection models: 1 orthographic, 2 perspective, 3 weak perspective, 4 para perspective

Intramodal Registration

Examples for the requirement of intramodal registration:

- digital subtraction angiography
- dual energy X-ray and CT
- visualization of perfusion
- visualization of differences (therapy control)
- motion estimation (for instance, in cardiac reconstruction)

Digital Subtraction Angiography

Figure: Mask image (left), fill image (middle), angiogram (right)

Digital Subtraction Angiography

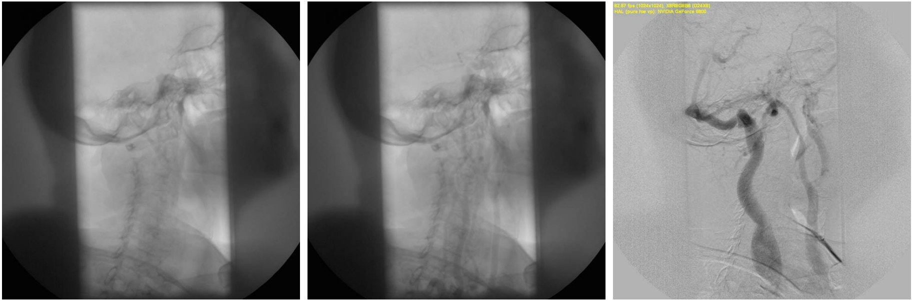


Figure: Mask image (left), fill image (middle), angiogram (right)

Motion Artifacts in DSA

Figure: Motion Artifacts in DSA (images: LME, Yu Deuerling-Zheng)

Motion Artifacts in DSA

Figure: Motion Artifacts in DSA (images: LME, Yu Deuerling-Zheng)

Similarity measures

Notation:

- $[f_{i,j}]$: reference image
- $[g_{i,j}]$: image to be registrated
- T: transform
- $\bar{f}$: mean intensity value of reference image
- $\bar{g}$: mean intensity value of second image

Similarity Measures

Sum of squared differences (SSD):

$$\widehat{T} = \operatorname{argmin}_T \sum_{i,j} ||f_{i,j} - T\{g_{i,j}\}||^2 \tag{61}$$

Correlation coefficient:

$$\widehat{T} = \operatorname{argmax}_T \frac{\sum_{i,j}(f_{i,j} - \bar{f})(T\{g_{i,j}\} - \bar{g})}{\sqrt{\sum_{i,j}(f_{i,j} - \bar{f})^2 \sum_{i,j}(T\{g_{i,j}\} - \bar{g})^2}} \tag{62}$$

Difference Imaging in CT

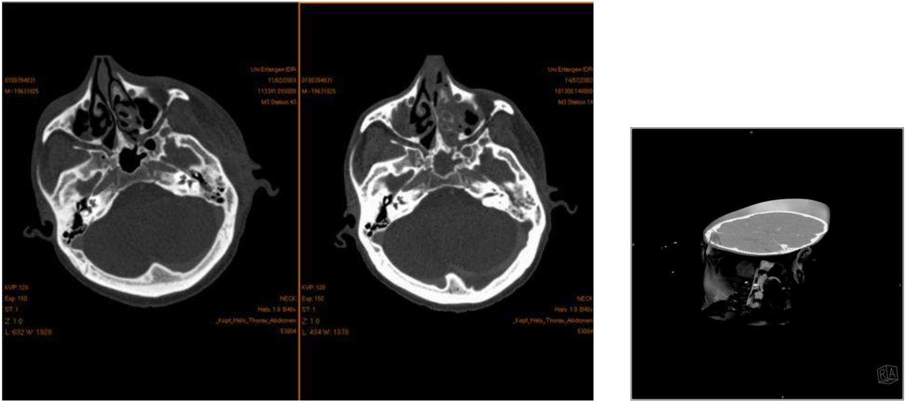


Figure: Difference imaging in CT (images: LME, Dieter Hahn)

Registration Combined with Segmentation

Problem:
Differences in images lead to a bias if all voxels are used for registration.

Solution:
Apply a weighting scheme to voxels. Voxels that belong to bones are rigid and allow for a reliable estimate for the transform (high weights). Soft tissue deforms, for instance, with tumor growth and thus implies a bias (low weights).

Registration in CT using Transfer Functions

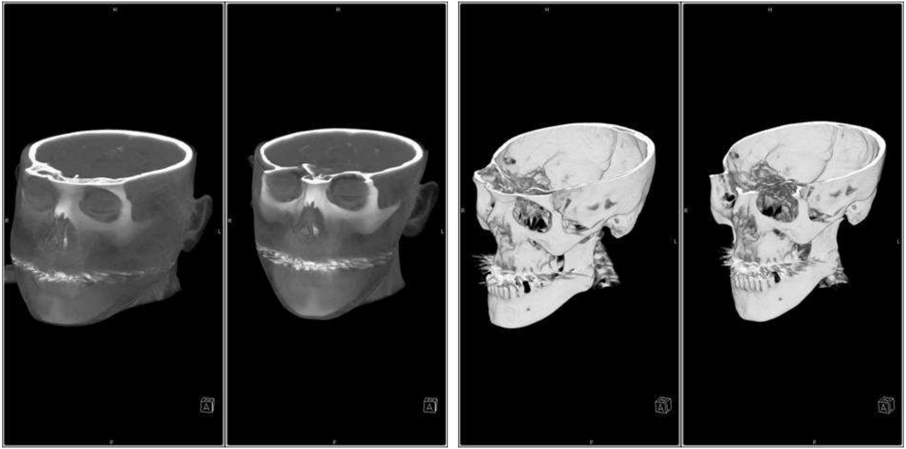


Figure: Difference imaging in CT: Segmentation of bones (images: LME, Dieter Hahn)

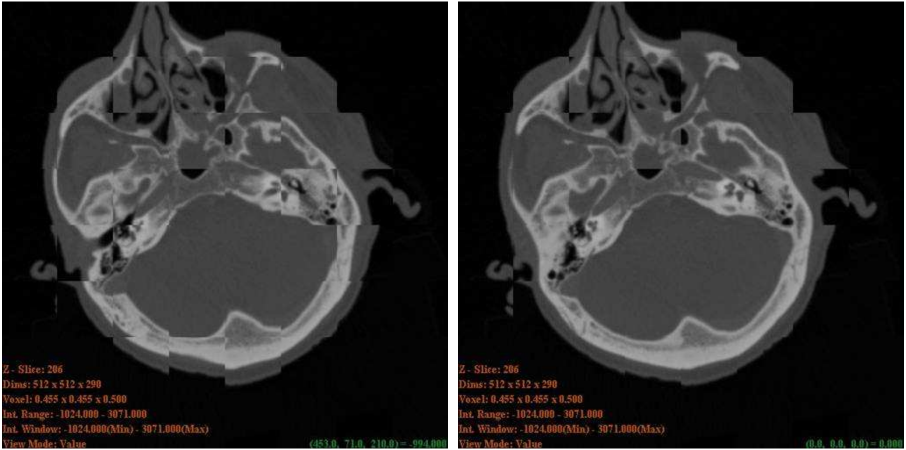


Figure: Checker board representation of results: no bone segmentation (left), bone segmentation (right) (images: LME, Dieter Hahn)

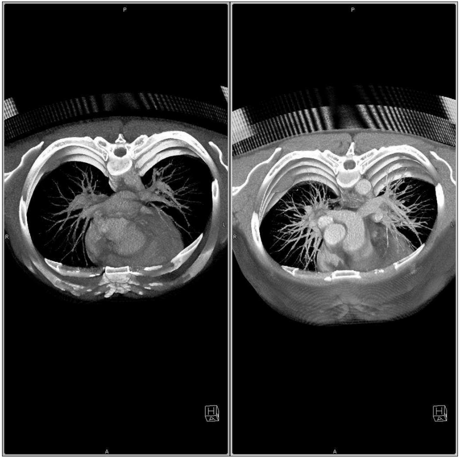


Figure: Thorax: tumor at two different therapy stages (images: LME, Dieter Hahn)

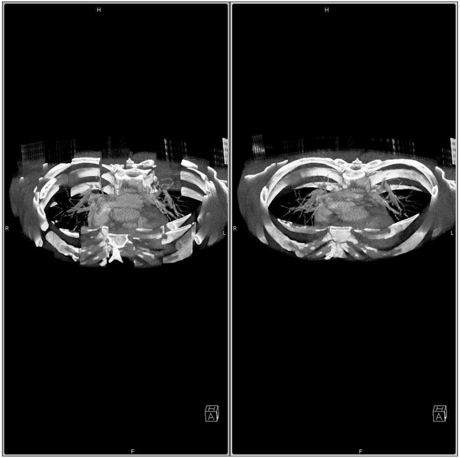


Figure: Checker board representation of results: no bone segmentation (left), bone segmentation (right) (images: LME, Dieter Hahn)

Figure: Difference image (images: LME, Dieter Hahn)

Multimodal Registration

Historical remarks on the application of mutual information in computer vision and image processing

- maximization of mutual information is applied to solve parameter estimation estimation problems, e.g. speech recognition (˜ 1980)
- maximization of mutual information first applied to image registration by P. Viola, W. Wells and by Collignon (1995)
- maximization of mutual information first applied to active object recognition by B. Schiele and J. Crowley (1997)
- ... and many other applications.

Kullback–Leibler Divergence

A proper similarity measure for denisty functions is the *Kullback–Leibler Divergence*
The *Kullback–Leibler Divergence* (KL divergence) between two bivariate probability density functions $p(f,g)$ and $q(f,g)$ is defined as:

$$KL(p,q) = \int_f \int_g p(f,g) \log \frac{p(f,g)}{q(f,g)} \, df \, dg \qquad (63)$$

Idea for Image Registration:

Images that are correctly registrated imply the highest probabilistic dependency of aligned intensity values. The more the random variables depend on each other, the more the pdf's $p(f,g)$ and $p(f) \cdot p(g)$ are different. This difference can be measured by the KL divergence.

$$KL(p,q) = \int_f \int_g p(f,g) \log \frac{p(f,g)}{p(f)p(g)} \, df \, dg \tag{64}$$

Optimization problem for image registration:

The transform T that maps the gray–level of the model image to the image point of the reference image can be estimated as follows:

$$\widehat{T} \quad = \quad \text{argmax}_T \sum_i p(f_i, g_{T(i)}) \log \frac{p(f_i, g_{T(i)})}{p(f_i)\, p(g_{T(i)})} \qquad (65)$$

In terms of information theory this is the maximization of **Mutual Information**:
We send image $[f_i]_{i=1,\ldots,N}$ and receive the image $[g_i]_{i=1,\ldots,N}$ where the channel applies a transform to image coordinates and intensity values change.

Registration of CT and SPECT images.[2]

Figure: CT image (left), SPECT image (middle), result of image registration (right)

[2]Images in courtesy of Dr. W. Römer, Nuclear Medicine, Univ. Erlangen

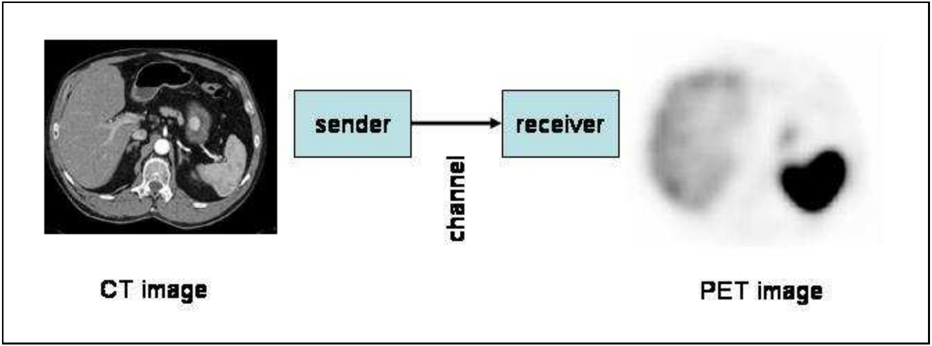


Figure: Channel model that defines the transition of CT to PET images (according to Wells et al., 1995)

Definition of Mutual Information

- Entropy is defined as:

$$H(F) = -\sum_f p(f) \log p(f) \tag{66}$$

- Entropy in the bivariate case:

$$H(F,G) = -\sum_{f,g} p(f,g) \log p(f,g) \tag{67}$$

- Conditional entropy:

$$H(F|g) = -\sum_f p(f|g) \log p(f|g) \tag{68}$$

and

$$H(F|G) = -\sum_g p(g) \sum_f p(f|g) \log p(f|g) \tag{69}$$

- Mutual information is defined as:

$$I(F;G) = H(F) + H(G) - H(F,G) = I(G,F) \tag{70}$$

Stochastic Maximization of Mutual Information

	draw N_A samples for i
	draw N_B samples for i
	Set $\widehat{T} = \widehat{T} + \lambda \frac{dI}{dT}$
UNTIL convergence	

Figure: Stochastic Maximization

Exercises

- Write a function that returns a rotation matrix $\boldsymbol{R}$ for a given rotation axis $\boldsymbol{u}$ and a rotation angle ϕ using the Rodrigues formula (17).
- Implement a datatype `quaternion` in Matlab with the appropriate functions as described in Sect. 2. Hint: as matlab – unlike e.g. C++ – does not provide user defined operator overloading, we need to use several functions for operations on quaternions. Solutions are given on p. **??**
- Implement a function `crossprod` in matlab that has two arguments which need to be vectors of 3 values. The function returns the cross product of the argument vectors.
- Write a function that returns a 3×3 matrix $\boldsymbol{U}$ for a given vector $\boldsymbol{u}$ where the multiplication with this matrix is equivalent to the cross product with $\boldsymbol{u}$ as in (15).

Discussion

- linear method to estimate 3–D rotation
- translation has to be known
- for 2-D/2-D image registration we found a linear estimator for both rotation and translation

Further Readings

- **Survey papers on medical image registration**
  - D.L.G. Hill, P.G. Batchelor, M. Holden, D.J. Hawkes: Medical Image Registration, Phys. Med. Biol. 46, 2001, pp. R1–R45, (pdf).
  - J.B.A. Maintz, M.A. Viergever: A survey of medical image registration, Medical Image Analysis, 2, p. 1-36, (pdf).
  - L.G. Brown: A survey of image registration techniques, ACM Computing Surveys 24(4) (1992), 325-376.
  - J. P. W. Pluim, J. B. A. Maintz, and M. Viergever: *Mutual-Information-Based Registration of Medical Images: A Survey*, IEEE Transactions on Medical Imaging, Vol. 22, No. 8, August 2003, pp. 986–1004, (pdf).
- **Complex numbers, quaternions and dual quaternions**
  - A paper that inspired all the sections on complex numbers, quaternions, and dual quaternions:
    K. Daniliidis, *Hand-Eye Calibration using Dual Quaternions*, The International Journal of Robotics Research, Vol. 18, No. 3, March 1999, pp. 286–298 (pdf).

Further Readings

Non–parametric mappings for image registration

- Non–linear registration methods applied to DSA can be found in Erik Meijering's papers.
- J. Moderzitzki: *Numerical Methods for Image Registration*, Habilitationsschrift, Institute of Mathematics, Medical University of Lübeck, 2002.
  Most of Jan Modersitzki's and Bernd Fischer's papers on image registration can be downloaded from his homepage.
- The group of M. Rumpf is also working on non-parametric image registration. Details on their work can be found on the institutes .