Image Registration: An Overview with an Emphasis on Geometry, Foundations, and Computational Complexity

Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
500 W. University, El Paso, TX 79968, USA
vladik@utep.edu

Joint work with Chandrajit Bajaj and Roberto Araiza

Title Page
◀◀ ▶▶
◀ ▶
Page 1 of 13
Go Back
Full Screen
Close
Quit

1. Image Registration: A Practical Problem

- In many areas of science and engineering, we have two images $I_1(\vec{x})$ and $I_2(\vec{x})$ of the same 2-D or 3-D object.
- Since I_1 and I_2 represent the same object, $I_2(\vec{x}) \approx I_1(\lambda \cdot R\vec{x} + \vec{a})$ for some scaling λ, rotation R, and shift $\vec{a}$.
- Often, we do not know the relative orientation of $I_1(\vec{x})$ and $I_2(\vec{x})$.
- In such situations, we must *register* images, i.e., find λ, R, and $\vec{a}$ after which the images match.
- Similar problem: images of different objects that should match – e.g., protein docking.

2. Existing Image Registration Techniques Are Mainly Heuristic

- *Good news:* there exist many different image registration techniques.
- *Problem:* most existing methods are heuristic.
- *Specifically:*
  - There is often no precise formulation of the problem.
  - Even when there is such a formulation, there is no clear relation between the formulation and the method.
- *Resulting practical problems:*
  - sometimes methods do not work; it is not clear when they are applicable;
  - it is not clear which method is better – or whether a yet better method is possible.
- *Our main objectives:*
  - provide a general formalization of the problem;
  - use this formalization to explain existing techniques – thus providing explanations of when they are applicable;
  - use this formalization to compare existing techniques;
  - if possible, design new, optimal techniques.

3. Towards Formulating the Image Registration Problem in Precise Terms

- *Ideal no-noise case:*
  - we know images $I_1(\vec x)$ and $I_2(\vec x)$;
  - we want to find $\vec a$, R, and λ after which the images match exactly: $I_2(\vec x) = I_1(\lambda \cdot R\vec x + \vec a)$.
- *In practice:* perfect macth is not possible:
  - there is noise,
  - there are measurement errors,
  - change in imaging conditions (or in the image itself) between the time when these two images were taken.
- *Objective:* select transformations for which, according to the user's preferences, the difference between $I_2(\vec x)$ and $I_T(\vec x) \stackrel{\text{def}}{=} I_1(\lambda \cdot R\vec x + \vec a)$ is the "most acceptable" (or, equivalently, the "least unacceptable".
- *Problem:* often, we do not have a clear description of user preferences.
- *Solution:* we must provide the exact description of what "most acceptable" means.

4. Utility theory: a known way to describe users' preferences

- The need to describe user preferences is important in decision making in general.

- To describe these preferences, a special *utility theory* has been developed.

- *Main idea:* sometimes, instead of choosing one of the alternatives $A_1, \ldots, A_n$, a user may choose A_i with probability p_i (e.g., flip a coin).

- We thus consider preference relation $\succ$ between such "lotteries" L_i.

- *Reasonable conditions:* e.g., if for a user, A is preferable to B and B is preferable to C, then for this user A should be preferable to C.

- *Main result:* under reasonable conditions, there exists a function u from the set $\mathcal{L}$ of all possible lotteries into the set $I\!R$ of real numbers for which:

  - $L_1 \succ L_2$ if and only if $u(L_1) > u(L_2)$, and
  - for every lottery L, in which each alternative A_i appears with probability p_i, we have $u(L) = p_1 \cdot u(A_1) + \ldots + p_n \cdot u(A_n)$.

  This function u is called a *utility function.*

- *Uniqueness:* if $u_1(L)$ and $u_2(L)$ describe the same $\succ$, then there exist $a > 0$ and b for which $u_2(L) = a \cdot u_1(L) + b$ for all lotteries L.

5. Utility for image registration: reasonable requirements

- *Objective:* describe a utility function $v(I_1, I_T)$ that describe the quality of matching.
- *Signal-to-noise ratio:*
  - if high, we can match images exactly;
  - is low, i.e., if signals are weak, then we can expand the scoring function $v(I_2, I_T)$ in Taylor series in $I_2(\vec{x})$ and $I_T(\vec{x})$, and keep only the lowest non-zero terms.
- *Comment:* we want to find T for which $I_T(\vec{x}) \approx I_2(\vec{x})$, i.e., $\Delta I(\vec{x}) \stackrel{\text{def}}{=} I_T(\vec{x}) - I_2(\vec{x})$ is small.
- To make this smallness explicit, we describe v in terms of $I_2(\vec{x})$ and $\Delta I(\vec{x})$.
- For perfect match $\Delta I = 0$ we must have $v \to \min$, i.e., $\partial v / \partial \Delta I = 0$ – hence v cannot have linear terms in ΔI; hence, $v(I_2, \Delta I)$ is quadratic.
- In general, a quadratic function can have terms independent on ΔI, terms linear in ΔI, and terms which are quadratic in ΔI.
- Terms that are independent on ΔI do not depend on the choice of $\vec{a}$, R, and λ and thus, do not affect the choice of registration.
- *Conclusion:* v does not depend on I_2, i.e.,

$$v(\Delta I) = \int a_1(\vec{x}) \cdot \Delta I(\vec{x})^2 \, d\vec{x} + \int a_2(\vec{x}, \vec{x}\,') \cdot \Delta I(\vec{x}) \cdot \Delta I(\vec{x}\,') \, d\vec{x} d\vec{x}\,'.$$

Image Registration: A . . .
Existing Image . . .
Towards Formulating . . .
Utility theory: a . . .
Utility for image . . .
Utility for image . . .
Why Fourier Methods
Why Fourier Methods . . .
Possibility of Further . . .
Case of Rotation Only
General Case: Shift . . .
Acknowledgments

Title Page
◀◀ ▶▶
◀ ▶

Go Back
Full Screen
Close
Quit

6. Utility for image registration: reasonable requirements (cont-d)

- A quadratic scoring function must be *non-degenerate*: for a function $\Delta I(\vec{x})$ which is equal to a finite number inside the bounded region and to 0 everywhere else, we get a finite value of $v(\Delta I)$.
- *Invariance:*
  - We want to find the shift, rotation, and scaling after which the images match as much as possible.
  - It is therefore reasonable to assume that the relative quality of two possible matches does not change if we simply shift, rotate, and/or scale both images.
  - Two utility functions $v_1(A)$ and $v_2(A)$ lead to the same preference relation if and only if they can be obtained from each by using a linear transformation: $v_2(A) = a \cdot v_1(A) + b$.
  - *Conclusion:* for every $\vec{a}$, R, and λ there exist real numbers $a(\vec{a}, R, \lambda)$ and $b(\vec{a}, R, \lambda)$ for which, for every function $\Delta I(\vec{x})$, we have $v(\Delta I_T) = a \cdot v(\Delta I) + b$.
- *Result:* Every non-degenerate invariant scoring function has the L^2-form $v(\Delta I) = c \cdot \int \Delta I(\vec{x})^2 \, d\vec{x}$ for some real number c.

7. Why Fourier Methods

- *Problem:* given two images $I_1(\vec{x})$ and $I_2(\vec{x})$, find a vector $\vec{a}$ for which

$$\int (I_2(\vec{x}) - I_1(\vec{x}+\vec{a}))^2\,d\vec{x} \to \min_{\vec{a}}.$$

- *Straightforward approach and its computational complexity:* for n^d images, we need $O(n^d)$ steps for each of $O(n^d)$ vectors $\vec{a}$, to the total of $O(n^{2d})$ – too long.
- *Solution:* instead of representing $I_i(\vec{x})$ pixel-by-pixel – i.e., a delta-function basis, use a different basis $e_1(\vec{x})$, $e_2(\vec{x})$, etc.
- *Auxiliary question:* which basis is optimal?
- *Comment:* for each n, selecting $e_1, \ldots, e_n$ is equivalent to selecting their linear combination.
- *Example:* polynomials can be represented as 1, x, x^2, $\ldots$, or by orthonormal basis.
- *Natural requirement:* modulo this non-uniqueness, the basis should be shift-invariant: $e_i(\vec{x}+\vec{a}) = \sum\limits_j c_{ij}(\vec{a}) \cdot e_j(\vec{x})$.
- *Result:* $e_i(\vec{x}) = \exp(\vec{\omega}\cdot\vec{x}) \cdot P_0(\vec{x})$.

8. Why Fourier Methods (cont-d)

- *Important case:* $e_i(\vec{x}) = \exp(\vec{\omega} \cdot \vec{x})$.
- *Why Fourier coefficients:* the coefficients at $e_i(\vec{x})$ are $\int I(\vec{x}) \cdot e_i(\vec{x})\, d\vec{x}$ – i.e., Fourier coefficients.
- *Alternative approach:* we have a preference relation on the set of all bases which is
  - shift-invariant and
  - there exists exactly one optimal basis.
- *Result:* $e_i(\vec{x}) = \exp(\vec{\omega} \cdot \vec{x}) \cdot P_0(\vec{x})$.
- *Fourier coefficients really help:*
  - the problem
    $$\int (I_2(\vec{x}) - I_1(\vec{x} + \vec{a}))^2\, d\vec{x} \to \min_{\vec{a}}$$
    is equivalent to
    $$\int I_2(\vec{x}) \cdot I_1(\vec{x} + \vec{a})\, d\vec{x} \to \max;$$
  - FT of convolution is a product of FTs;
  - by using $O(n^d \cdot \log(n))$ FFT algorithm, we can thus compute convolution – and find $\vec{a}$ – in time $O(n^d \cdot \log(n)) \ll O(n^{2d})$.

9. Possibility of Further Speed-Up

- *Problem:* even with FFT, the method is sometimes too slow.
- *Natural solution:* before comparing, we keep only partial information about the images.
- *Possibilities:*
  - only keep some coefficients of expanding $I_i(\vec{x})$ over a basis;
  - only keep values of $I_i(\vec{x})$ at special points – symmetry leads to critical points $I_{,i} = 0$.
- *Comment:* optimal $e_i(\vec{x})$ are polynomials – where coefficients are moments – or waves – and Fourier transforms.
- *Moments-based techniques:* we match images by their centers of mass.
- *Limitations:* works well for astronomical images (surrounded by empty space) but not for images cut off from larger ones.
- *FFT-based – idea:* $F_2(\vec{\omega}) \approx r(\vec{\omega}) \cdot F_1(\vec{\omega})$.
- *FFT-based – implementation:* least squares leads to

$$r(\vec{\omega}) = \frac{F_1(\omega) \cdot F_2^*(\vec{\omega})}{|F_1(\omega) \cdot F^*(\vec{\omega})|}.$$

- *After that:*

$$\int |F_1(\vec{\omega})|^2 \cdot |r(\vec{\omega}) - \exp(\mathrm{i} \cdot \vec{\omega} \cdot \vec{x})|^2 \, d\vec{\omega} \to \min.$$

Possible approximations: $|F_1(\omega)| = \text{const}$, threshold (Shiek), etc.

Title Page
◀◀ ▶▶
◀ ▶
Page 10 of 13
Go Back
Full Screen
Close
Quit

10. Case of Rotation Only

- *Description:* $\int (I_2(\vec{x}) - I_1(R\vec{x}))^2 \, d\vec{x} \to \min$.
- *Straightforward approach:* $O(n^3)$ in 2-D case, $O(n^6)$ in 3-D case.
- *2-D case:* we can reduce to shift by using log-polar coordinates $(\ln(\rho), \theta)$.
- *2-D and 3-D cases:*
  - only keep some coefficients of expanding $I_i(\vec{x})$ over a basis;
  - only keep values of $I_i(\vec{x})$ at special points – symmetry leads to critical points $I_{,i} = 0$;
  - only keep values at a rays $\lambda \cdot \vec{e}$ where rays match best.
- *Resulting methods:*
  - *Moments:* if center of mass is not 0, $J_i = \int I(\vec{x}) \cdot x_i \, d\vec{x} \neq 0$, we get rotation;
  - if center of mass is at 0, we compare moments of inertia $J_{ij} = \int I(\vec{x}) \cdot x_i \cdot x_j \, d\vec{x} \neq 0$ – and compare then, e.g., by eigenvectors.
  - *Matching critical points:* good if there are few distinct points; bad for road networks and cellular samples.
  - *Matching rays:*

$$\int a(\lambda) \cdot (I_2(\lambda \cdot \vec{e}) - I_1(\lambda \cdot \vec{e}))^2 \to \min_{\vec{e}};$$

this requires $O(n^2) \cdot O(n \cdot \log(n)) = O(n^3 \cdot \log(n))$ time.

Image Registration: A . . .
Existing Image . . .
Towards Formulating . . .
Utility theory: a . . .
Utility for image . . .
Utility for image . . .
Why Fourier Methods
Why Fourier Methods . . .
Possibility of Further . . .
Case of Rotation Only
General Case: Shift . . .
Acknowledgments
Title Page
◀◀ ▶▶
◀ ▶

Go Back
Full Screen
Close
Quit

11. General Case: Shift and Rotation (and Maybe Scaling)

- *Problem:* $\int (I_2(\vec{x}) - I_1(\lambda \cdot R\vec{x} + \vec{a}))^2 \to \min$.
- *Straightforward approach:* $O(n^6)$ in 2-D case; $O(n^{10})$ in 3-D case.
- *With FFT-based convolution:* we need convolution for every R and λ, i.e., $O(n^4 \cdot \log(n))$ in 2-D case and $O(n^7 \cdot \log(n))$ for 3-D case.
- *Main idea:*
  - replace images with coefficients – Fourier transforms $F_i(\vec{\omega})$;
  - then, replace each coefficient with its shift-invariant combination $M_i(\vec{\omega}) = |F_i(\vec{\omega})|$.

  Once we find rotation, we rotate images and find shift.
- *Resulting* $O(n^d \cdot \log(n))$ *methods:*
  - *2-D case:* turn to log-polar coordinates and use FFT-based matching to find rotation and scaling;
  - *3-D case:* get rotation by matching eigenvectors of moments of inertia $J_{ij} = \int M_i(\vec{\omega}) \cdot \omega_i \cdot \omega_j \, d\vec{\omega}$;
  - *3-D case:* get the rotation axis as the direction $\vec{e}$ at which M_i best match.
- *Additional idea* NFFT uses polynomials in addition to FFT.

12. Acknowledgments

This work was supported in part:

- by NASA under cooperative agreement NCC5-209,
- by NSF grant EAR-0225670,
- by NIH grant 3T34GM008048-20S1,
- by Army Research Lab grant DATM-05-02-C-0046,
- by Star Award from the University of Texas System,
- and by Texas Department of Transportation grant No. 0-5453.

Title Page
◀◀ ▶▶
◀ ▶
Page 13 of 13
Go Back
Full Screen
Close
Quit