

Iterative Fitting After Elastic Registration: An Efficient Strategy for Accurate Estimation of Parametric Deformations

Xinxin Zhang, Christopher Gilliam and Thierry Blu

The Chinese University of Hong Kong



20th September, 2017

Outline

1 Introduction

- Problem Statement
- Overview of Current Methods

2 Method

3 Experimental results

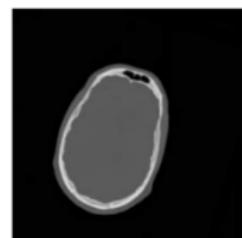
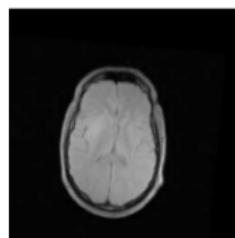
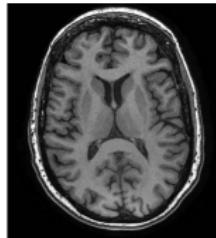
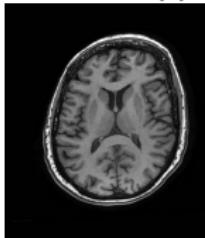
- Synthetic images
- Real images
- Applications

4 Conclusion

Introduction

Problem Statement

- Medical applications



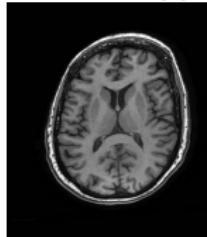
- Remote sensing



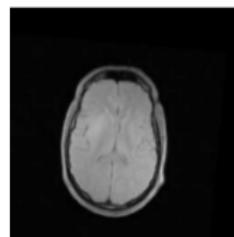
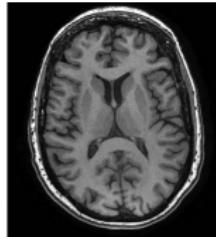
- Others

Problem Statement

- Medical applications



monomodal



multimodal

- Remote sensing



- Others

Problem Statement

■ $I_{\text{target}}(z) = I_{\text{source}}(z + u(z))$
 $z = x + iy, u(z) = u_x(z) + iu_y(z),$

technical details are too complex to cover in the book itself.

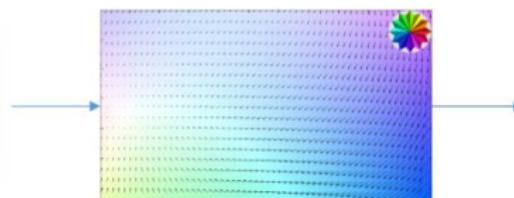
In teaching our courses, we have found it useful for the students to attempt a number of small implementation projects, which often build on one another, in order to get them used to working with real-world images and to understand the challenges that these present. The students are then asked to submit an individual report for each of these projects, first privately. (Sometimes these projects even turn into conference papers!). The exercises at the end of each chapter contain numerous suggestions for smaller student projects, as well as more open-ended problems whose solutions are still active research topics. Whenever possible, I encourage students to try to implement their ideas using photographs, since this better motivates them, and leads to greater variation in the problems, and better acquaints them with the variety and complexity of real-world imagery.

In formulating and solving computer vision problems, I have often found it useful to draw inspiration from these high-level approaches:

$I_{\text{source}}(Z)$

$u(z)$

$I_{\text{target}}(Z)$



technical details are too complex to cover in the book itself.

In teaching our courses, we have found it useful for the students to attempt a number of small implementation projects, which often build on one another, in order to get them used to working with real-world images and the challenges that these present. The students are then asked to submit an individual report for each of these projects, first privately. (Sometimes these projects even turn into conference papers!). The exercises at the end of each chapter contain numerous suggestions for smaller student projects, as well as more open-ended problems whose solutions are still active research topics. Whenever possible, I encourage students to try to implement their ideas using photographs, since this better motivates them, and leads to greater variation in the problems, and better acquaints them with the variety and complexity of real-world imagery.

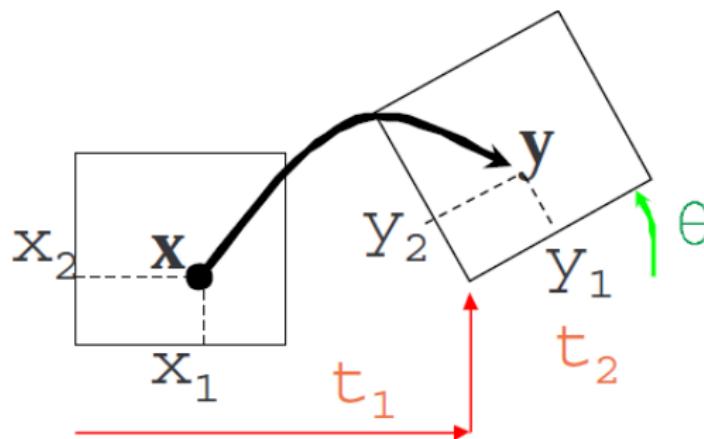
In formulating and solving computer vision problems, I have often found it useful to draw inspiration from these high-level approaches:

Overview of Current Methods

■ Global parametric registration

Calculate the parameters of the model.

Woods, 1992; Woods, 1998; Evangelidis, 2008

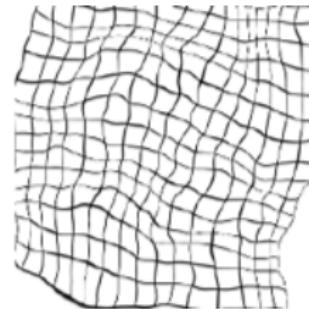
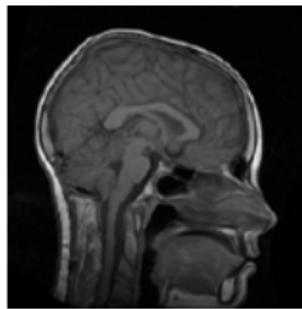
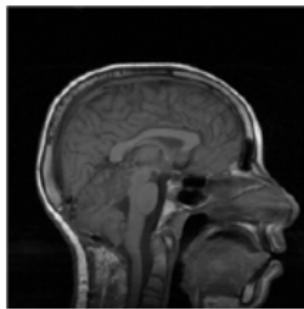


Overview of Current Methods

■ Elastic registration

Estimate a displacement vector per pixel.

Arganda-Carreras, 2006; Bajcsy, 1989; Periaswamy, 2003; Klein, 2010; Periaswamy, 2006; Goshtasby, 1988; Kybic, 2003; Bruhn, 2005

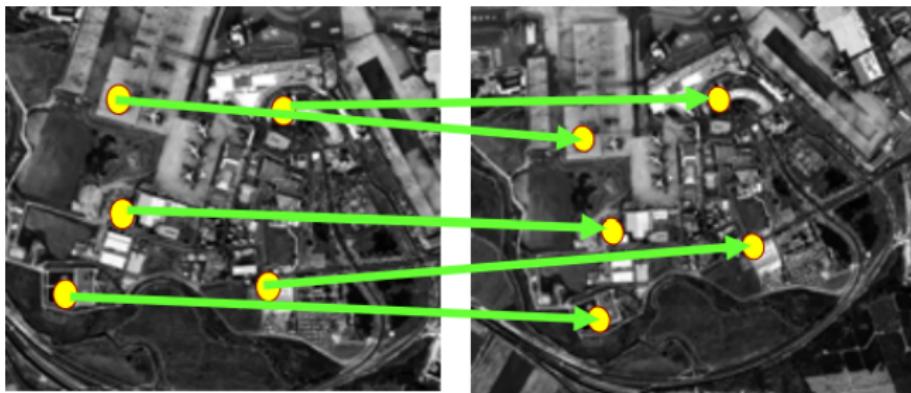


Overview of Current Methods

■ Landmark/Feature-based Registration

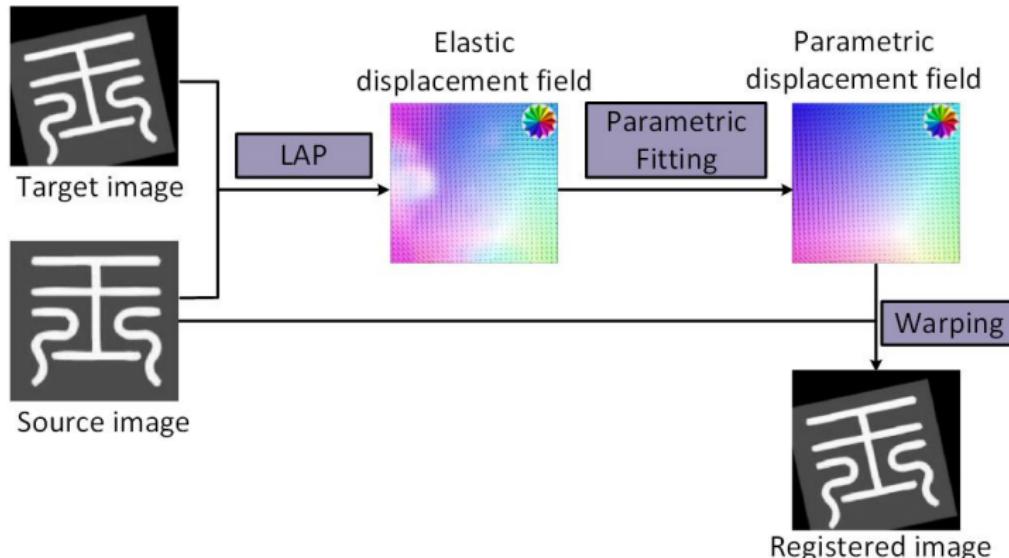
Establish the correspondence between the extracted features or landmarks from the two images.

Alhichri, 2003; Davatzikos, 1996; Li, 2009; Ma, 2015



Method

Parametric fitting of the elastic displacement field



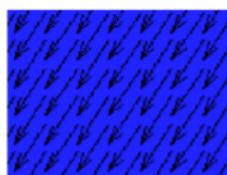
LAP: Local All-Pass Filters

C. Gilliam and T. Blu, "Local all-pass filters for optical flow estimation," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1533-1537, 2015.

Parametric model

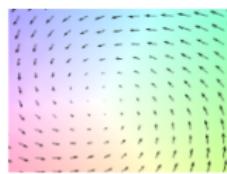
Geometric Transformation

Translation



$$u(z) = c_1$$

Rotation

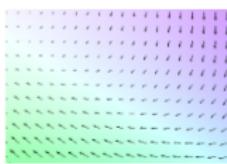


Scaling



$$u(z) = c_2 z$$

This paper



$$u(z) = c_1 + c_2 z + c_3 \bar{z} + c_4 z\bar{z} + c_5 z^2 + c_6 \bar{z}^2$$

Parametric model fitting

Fitting model: $u_{\text{fit}}(z) = c_1 + c_2 z + c_3 \bar{z} + c_4 z \bar{z} + c_5 z^2 + c_6 \bar{z}^2$,

Fitting criterion: $\min_c \sum_{z \in \Omega} |u_{\text{fit}}(z) - u_{\text{LAP}}(z)|^2$,

$c = [c_1, c_2, c_3, c_4, c_5, c_6]^T$ to be calculated.

→ solution of a linear system of equations

★ Fast, efficient and flexible.

Parametric model fitting

Fitting model: $u_{\text{fit}}(z) = c_1 + c_2 z + c_3 \bar{z} + c_4 z \bar{z} + c_5 z^2 + c_6 \bar{z}^2$,

Fitting criterion: $\min_c \sum_{z \in \Omega} |u_{\text{fit}}(z) - u_{\text{LAP}}(z)|^2$,

$c = [c_1, c_2, c_3, c_4, c_5, c_6]^T$ to be calculated.

→ solution of a linear system of equations

★ Fast, efficient and flexible.

Choosing reliable pixels to fit parameters

Source image



Target image



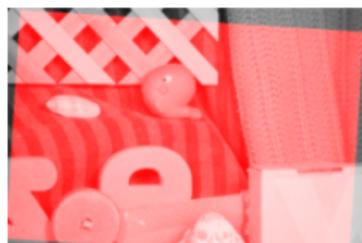
Common region

Choosing reliable pixels to fit parameters

Source image

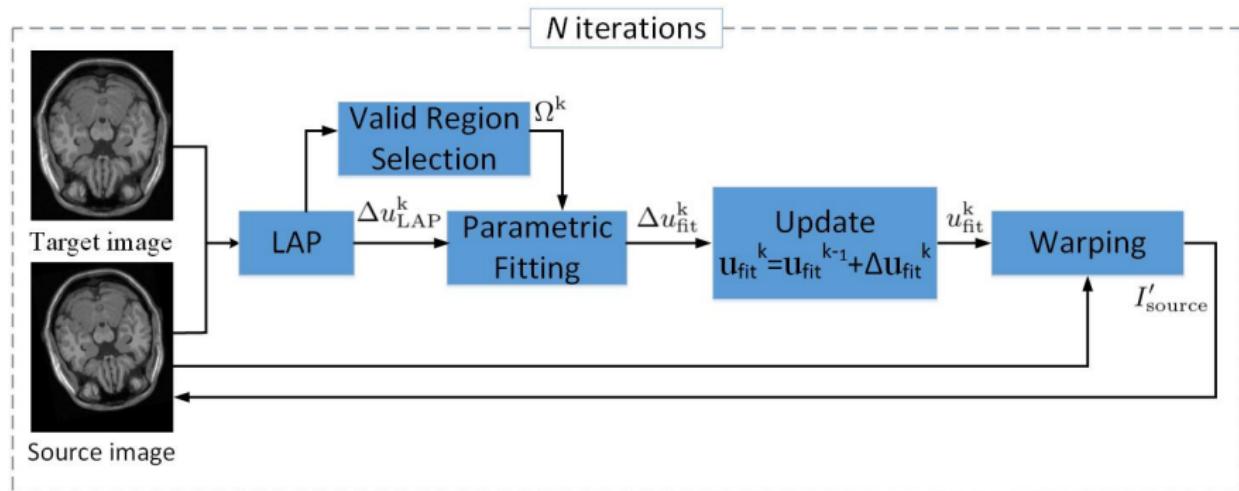


Target image



Valid region

Flow chart of the proposed method



- Δu_{LAP}^k : the displacement estimated by the Local All-Pass Filters (LAP) in the k th iteration.
- u_{fit}^k : the displacement estimated by polynomial fitting in the k th iteration.
- Δu_{fit}^k : the displacement increment estimated by polynomial fitting in the k th iteration.
- Ω^k : valid region in the k th iteration.

Experimental results

Comparison with the state-of-the-art

Synthetic images

Table: Error comparison for the iterative fitting method and the state-of-the-art image registration methods.

		Noiseless Image			Noisy Image (15dB)			Gaussian Blurry Image			Missing Information		
		E_{Med}	E_{Mean}	Time	E_{Med}	E_{Mean}	Time	E_{Med}	E_{Mean}	Time	E_{Med}	E_{Mean}	Time
Parametric algorithms	Ours	0	0	18.45	0.238	0.430	19.08	0.505	0.590	32.22	0	0	38.58
	AECC	0.851	1.155	9.10	0.908	1.229	9.522	1.001	1.289	8.10	0.907	1.206	9.32
Elastic algorithms	LAP	0.006	1.189	4.60	1.799	4.076	4.52	3.118	3.415	5.94	0.011	2.131	7.94
	Demons	5.114	32.874	37.15	7.601	10.241	22.13	6.066	7.497	31.75	5.700	10.575	20.81
	MIRT	3.232	9.931	75.00	7.976	12.065	52.51	7.420	11.467	65.67	7.764	12.661	80.00
	bUnwarpJ	1.3402	1.4107	14.64	1.6763	1.8072	25.07	3.346	4.512	23.80	1.924	7.220	325.38

(1) Bold values indicate the best results. (2) The size of images is 388 by 584 pixels. (3) PSNR between the noisy image and original noiseless image is 15dB. (4) Results averaged over 5 different parametric deformation fields (maximum displacement is 16 pixels).

Absolute Error: $E(j) = |u_{\text{GT}}(j) - u(j)|$, $j \in D$, D is the common region

Median Absolute Error: $E_{\text{Med}} = \text{Median}(E)$

Mean Absolute Error: $E_{\text{Mean}} = \text{Mean}(E)$

Reference:

AECC: G. D. Evangelidis and E. Z. Psarakis, 2008;

Demons: H. Lombaert, L. Grady, and X. Pennec et al., 2009;

MIRT: A. Myronenko and X. Song, 2010;

bUnwarpJ: I. Arganda-Carreras, C. O. S. Sorzano, and R. Marabini et al., 2006.

Noiseless images

I_{source} and I_{target}

AECC

Demons

bUnwarpJ

LAP

Ours

Noisy images (15dB)

I_{source} and I_{target}

AECC

Demons

bUnwarpJ

LAP

Ours

Gaussian blurry images

I_{source} and I_{target}

AECC

Demons

bUnwarpJ

LAP

Ours

Real images (400×400)

I_{source} and I_{target}

MIRT 78.74s

Demons 13.90s

AECC 6.67s

LAP 5.39s

Ours 21.32s

Real images (480×640)

I_{source} and I_{target}

MIRT 82.32s

Demons 8.75s

AECC 6.67s

LAP 18.82s

Ours 91.67s

Real images (480×640)

I_{source} and I_{target}

MIRT 104.36s

Demons 178.97s

AECC 14.96s

LAP 24.34s

Ours 64.85s

Real images (480×640)

I_{source} and I_{target}

Ours 50.24s

Video stabilization

Input

Output

Conclusion

Highlights

- 1** Accurate and fast.
- 2** Flexible and costless to add more parameters.
- 3** Robust to model mis-match (e.g. noise and blurring).
- 4** Robust to very large displacement.

Highlights

- 1** Accurate and fast.
- 2** Flexible and costless to add more parameters.
- 3** Robust to model mis-match (e.g. noise and blurring).
- 4** Robust to very large displacement.

Highlights

- 1** Accurate and fast.
- 2** Flexible and costless to add more parameters.
- 3** Robust to model mis-match (e.g. noise and blurring).
- 4** Robust to very large displacement.

Highlights

- 1** Accurate and fast.
- 2** Flexible and costless to add more parameters.
- 3** Robust to model mis-match (e.g. noise and blurring).
- 4** Robust to very large displacement.

Thanks for your attention.

