

---

# Convex Optimization in Non-rigid Image Registration

---

**Xiaoran Zhang**  
Electrical and Computer Engineering  
University of California, Los Angeles  
xiaoran108@ucla.edu

## 1 Introduction

Image registration is the process of transforming different sets of data into one coordinate system and it is an important and fundamental task in image processing especially in many biomedical imaging applications. The registration task requires user to input a pair of images acquired from different devices, among which one denoted as *moving*, indicating the image that needs to be aligned, and the other denoted as *fixed*, indicating the target coordinate system of the alignment, and outputs the aligned image after transformation, commonly denoted as *moved*.

Registration techniques can be generally divided into rigid registration and non-rigid registration. Rigid registration transforms the moving image based on a  $3 \times 3$  affine transformation matrix and preserves the pixelwise relations. However, its affine assumption is limited in many real life applications. Non-rigid registration transforms an image based on an optical flow by assigning a displacement vector for each pixel as described by Zhang et al. [2020]. It is more general but challenging as it requires to warp the image based on a displacement vector field (DVF) on the same scale with the image.

Most non-rigid registration approaches optimize through an energy function comprised of a dissimilarity metric and a convex regularization on the deformation field to enforce smoothness. From Yuan and Fenster [2018], Rajchl et al. [2014], the problem could be formulated as a convex optimization and solved by augmented Lagrangian method (ALM). With the recent advances brought by deep-learning, the non-rigid registration could also be formulated using convolution neural network (CNN) to produce the DVF as presented by Balakrishnan et al. [2019]. Although it demonstrates improved performance, its interpretability remains hard to define as the architecture of the CNN is often data-driven and designed empirically. Thus, it is still necessary and important to analyze the non-learning based methods. This project aims to formulate the non-rigid registration problem as the convex optimization problem and then optimize it through an ALM-based algorithm. We also compare its difference with other non-learning based registration algorithms including SimpleElastix by Marstal et al. [2016], Oriented FAST and Rotated BRIEF (ORB) by Rublee et al. [2011] and intensity-based image registration by Matlab.

## 2 Problem Formulation

The non-rigid registration often optimizes through the energy function consists of a dissimilarity metric and a regularization term

$$\min P(I_m, I_f; \mathbf{u}) + R(\mathbf{u}) \quad (1)$$

where the  $P(I_m, I_f; \mathbf{u})$  represent the dissimilarity metric of the input image pair  $(I_m, I_f)$  under the deformation field  $\mathbf{u}$  in 2D.  $R(\mathbf{u})$  is the convex regularization function to enforce the smoothness prior on the deformation. In this project we select the  $L_1$ -norm on the gradient of the DVF,  $\mathbf{u} =$

$[\mathbf{u}_1, \mathbf{u}_2]$ .  $\mathbf{u}_i$  denotes the pixel-shift for each dimension.

$$R(\mathbf{u}) = \alpha \sum_{i=1}^2 \|\nabla \mathbf{u}_i\|_1 \quad (2)$$

Although the regularization is a convex term, the dissimilarity metric is usually highly non-convex and nonlinear. In this project, we select the sum of absolute difference (SAD) for the dissimilarity metric

$$P(I_m, I_f; \mathbf{u}) = \sum \sum |I_m \circ \mathbf{u} - I_f|. \quad (3)$$

To relax the above problem to a convex problem, we introduce the following sequential process for convexification and linearization. We first introduce a iterative incremental warping variable  $\mathbf{h}^k = [\mathbf{h}_1^k, \mathbf{h}_2^k]$  for each iteration  $k$ . We initialize the  $\mathbf{h}^0 = \mathbf{0}$ . At  $k^{\text{th}}$  iteration, we calculate the deformation field as

$$\tilde{\mathbf{u}}[x, y] = \sum_{i=0}^{k-1} \mathbf{h}^i[x, y] \quad (4)$$

and then we calculate the  $k^{\text{th}}$  incremental update  $\mathbf{h}^k$  by minimizing the following function

$$\mathbf{h}^k = \operatorname{argmin} \sum \sum |I_m \circ (\tilde{\mathbf{u}} + \mathbf{h}^k) - I_f| + R(\tilde{\mathbf{u}} + \mathbf{h}^k). \quad (5)$$

The deformation applied to the *moving* image  $I_m$  using the  $\circ$  notation could be represented by the bilinear interpolation of the original image

$$(I_m \circ (\tilde{\mathbf{u}} + \mathbf{h}^k))[x, y] = I_m[x + \tilde{\mathbf{u}}_1[x, y] + \mathbf{h}_1^k[x, y], y + \tilde{\mathbf{u}}_2[x, y] + \mathbf{h}_2^k[x, y]]. \quad (6)$$

We will continue using the  $\circ$  for notation simplicity in the following report. To linearize this highly non-linear function, We introduce the first-order approximation

$$I_m \circ (\tilde{\mathbf{u}} + \mathbf{h}^k) \approx I_m \circ \tilde{\mathbf{u}} + \nabla(I_m \circ \tilde{\mathbf{u}}) \cdot \mathbf{h}^k \quad (7)$$

Thus, we could rewrite the  $\mathbf{h}^k$  update equation in Eq. 5 as

$$\mathbf{h}^k = \operatorname{argmin} \sum \sum |\tilde{P}_0^k + \nabla \tilde{P}^k \cdot \mathbf{h}^k| + R(\tilde{\mathbf{u}} + \mathbf{h}^k) \quad (8)$$

where  $\tilde{P}^k = I_m \circ \tilde{\mathbf{u}}$  and  $\tilde{P}_0^k = \tilde{P}^k - I_f$ . After the linearization, the next step is to solve the update using convex optimization. We firstly derive the dual of the minimization problem in Eq. 8 as

$$\begin{aligned} & \max_{|\mathbf{w}[x, y]| \leq 1, \mathbf{q}} \sum \sum (\mathbf{w}[x, y] \tilde{P}^k[x, y] + \sum_{i=1}^2 \tilde{\mathbf{u}}_i[x, y] \operatorname{div} \mathbf{q}_i[x, y]) - R_\alpha^*(\mathbf{q}) \\ & \text{s.t.} \quad \mathbf{w}[x, y] \partial_i P[x, y] + \operatorname{div} \mathbf{q}_i[x, y] = 0 \quad i = 1, 2 \end{aligned} \quad (9)$$

where  $R_\alpha^*(\mathbf{q})$  is defined in Eq. 22. The derivation for the dual in Eq. 9 is presented in appendix section. With Eq. 9, we have the convex formulation for the registration problem. The next section will introduce the sequential optimization algorithm with ALM to solve this dual problem.

### 3 Methods

#### 3.1 Coarse-to-fine Image Pyramid Construction

We first construct a coarse-to-fine image pyramid representation for the registration problem. The image pyramid will be helpful to improve the registration performance as the accuracy of displacement vector estimation critically depends on the magnitude of image motion by Ruhnau et al. [2005]. Let  $I_m^1, \dots, I_m^L$  be the L-level pyramid representation of the *moving* image from the coarsest resolution  $I_m^1$  to the finest resolution  $I_m^L$  and  $I_f^1, \dots, I_f^L$  be the pyramid representation of the *fixed* image. We optimize the deformation field  $\mathbf{u}$  for each pyramid level denoted as  $\mathbf{u}^l$ . We then optimize the deformation in the next pyramid level using the information provided by the previous level after resizing the  $\mathbf{u}^{l-1}$  to the level  $l$  by

$$\min \quad P(I_m^l \circ \mathbf{u}^{l-1}, I_f^l; \mathbf{t}^l) + R(\mathbf{u}^{l-1} + \mathbf{t}^l). \quad (10)$$

The  $\mathbf{t}^l$  is the total incremental deformation at pyramid level  $l$  given by

$$\mathbf{t}^l = \sum_{i=0}^N (\mathbf{h}^l)^i \quad (11)$$

where  $N$  is the total number of iterations to estimate the incremental deformation.

### 3.2 Sequential Optimization with ALM

Given the pyramid construction and the problem formulation introduced in the previous sections, we have the following optimization procedure.

---

**Algorithm 1:** Sequential optimization framework

---

```

1 Initialize:  $(\mathbf{h}^0)^0$  and  $\mathbf{u}^0$ 
2 for  $l = 1, 2, \dots, L$  do
3   for  $k = 1, 2, \dots, N$  do
4      $\tilde{\mathbf{u}}^{l-1} = \mathbf{u}^{l-1} + \sum_{i=0}^{k-1} (\mathbf{h}^l)^i$ 
5      $(\mathbf{h}^l)^k = \operatorname{argmin} \sum \sum |(\tilde{P}_0^l)^k + \nabla(\tilde{P}^l)^k \cdot (\mathbf{h}^l)^k| + R(\tilde{\mathbf{u}}^{l-1} + (\mathbf{h}^l)^k)$ 
6      $\mathbf{t}^l = \sum_{i=0}^k (\mathbf{h}^l)^i$ 
7      $\mathbf{u}^l = \mathbf{u}^{l-1} + \mathbf{t}^l$ 

```

---

From the algorithm, the key step is to optimize the incremental deformation vector  $(\mathbf{h}^l)^k$ . From the dual formulation given in Eq. 9, we denote the objective function as the energy function and the constraint function described as

$$E(\mathbf{w}, \mathbf{q}) = \sum \sum (\mathbf{w}[x, y] (\tilde{P}_0^{l-1})^k [x, y] + \sum_{i=1}^2 \tilde{\mathbf{u}}_i^{l-1} [x, y] \operatorname{div} \mathbf{q}_i [x, y]) - R_\alpha^*(\mathbf{q}) \quad (12)$$

$$F_i[x, y] = \mathbf{w}[x, y] \partial_i (\tilde{P}^{l-1})^k [x, y] + \operatorname{div} \mathbf{q}_i [x, y] \quad i = 1, 2. \quad (13)$$

We then construct the lagrangian function as

$$\mathcal{L}(\mathbf{h}, \mathbf{w}, \mathbf{q}) = E(\mathbf{w}, \mathbf{q}) + \sum_{i=1}^2 \mathbf{h}_i \cdot \mathbf{F}_i \quad (14)$$

and the augmented lagrangian function as

$$\mathcal{L}_c(\mathbf{h}, \mathbf{w}, \mathbf{q}) = \mathcal{L}(\mathbf{h}, \mathbf{w}, \mathbf{q}) - \frac{c}{2} \sum_{i=1}^2 \|\mathbf{F}_i\|_2^2 \quad (15)$$

where  $c > 0$  is the positive penalty for the  $L_2$  regularization introduced in the augmented lagrangian function. We follow the standard procedure of ALM and introduce the following optimization algorithm

---

**Algorithm 2:** Sequential optimization framework with ALM

---

```

1 Initialize:  $(\mathbf{h}^0)^0$  and  $\mathbf{u}^0$ 
2 For  $l = 1, 2, \dots, L$  do
3   For  $k = 1, 2, \dots, N$  do
4      $\tilde{\mathbf{u}}^{l-1} = \mathbf{u}^{l-1} + \sum_{i=0}^{k-1} (\mathbf{h}^l)^i$ 
5     Initialize:  $(\mathbf{h}^l)_j^k, \mathbf{w}^0$  and  $\mathbf{q}^0$ 
6     For  $j = 1, 2, \dots, M$  do
7        $\mathbf{w}^{j+1} = \underset{|\mathbf{w}[x, y]| \leq 1}{\operatorname{argmax}} \mathcal{L}_c((\mathbf{h}^l)_j^k, \mathbf{w}, \mathbf{q}^j)$  with fixed  $(\mathbf{h}^l)_j^k$  and  $\mathbf{q}^j$ 
8        $\mathbf{q}^{j+1} = \underset{\mathbf{q}}{\operatorname{argmin}} \mathcal{L}_c((\mathbf{h}^l)_j^k, \mathbf{w}^{j+1}, \mathbf{q})$  with fixed  $(\mathbf{h}^l)_j^k$  and  $\mathbf{w}^{j+1}$ 
9        $(\mathbf{h}^l)_{j+1}^k = (\mathbf{h}^l)_j^k - c(\mathbf{w}^{j+1} \cdot \partial_i P + \operatorname{div} \mathbf{q}^{j+1})$ 
10      update =  $c \sum \sum (\mathbf{w}^{j+1} \cdot \partial_i P + \operatorname{div} \mathbf{q}^{j+1})$ 
11      If update  $\leq$  tolerance
12        break
13       $\mathbf{t}^l = \sum_{i=0}^k (\mathbf{h}^l)^i$ 
14       $\mathbf{u}^l = \mathbf{u}^{l-1} + \mathbf{t}^l$ 

```

---

where the tolerance is chosen as a small positive parameter.

## 4 Experiment

### 4.1 Data generation

The data set used in this project is generated from the Kaggle Dogs vs Cats competition (<https://www.kaggle.com/c/dogs-vs-cats/>). We downloaded 200 images for testing. These downloaded images are considered as *moving* images. The *fixed* images in the testing set are deformed using manually created translation map. The pipeline is shown in Fig. 1. The transformation matri-

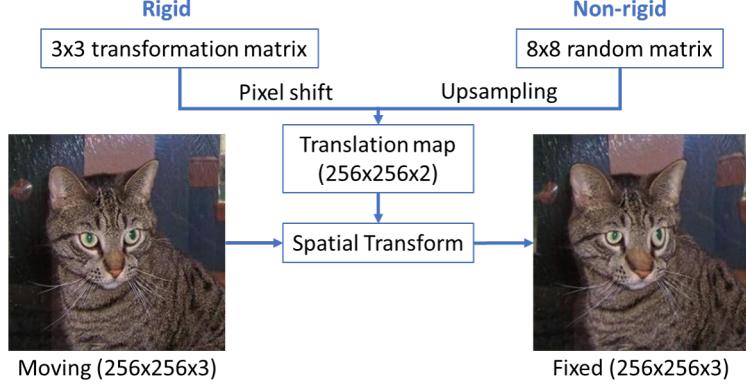


Figure 1: Data generator for *fixed* image from *moving* image.

ces and its random entries for each pair used in the generator are listed in Table 1.

Table 1: Transformation and random matrix used in the data generator.

Type	Matrix	Random Seed
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$t_x, t_y \in [-5, 5]$
Shearing	$\begin{bmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$sh_x, sh_y \in [0, 0.15]$
Scaling	$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$s_x, s_y \in [0.9, 1]$
Rotation	$\begin{bmatrix} \cos(q) & \sin(q) & 0 \\ -\sin(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$q \in [-5, 5]$
Pixelwise	$\begin{bmatrix} p_{11} & \dots & p_{18} \\ \vdots & \vdots & \vdots \\ p_{81} & \dots & p_{88} \end{bmatrix}$	$p_{ij} \in [-5, 5]$

The rigid transformation matrix  $T$  is a  $3 \times 3$  matrix. We take the pixel shift in Cartesian coordinate system to calculate the translation map from  $T$ . Let  $[x, y, 1]^T$  denote the homogeneous coordinate in *moving* image and  $[x', y', 1]^T$  denote the coordinate in the *fixed* image, we have

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (16)$$

and pixel shift can be calculated as

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} x' - x \\ y' - y \end{bmatrix}. \quad (17)$$

Thus we could have the ground truth translation map with shape  $256 \times 256 \times 2$ , where the first channel represents the pixel shift in  $x$  and second represents  $y$  for each pixel. For non-rigid transformation, we produce a  $8 \times 8$  random matrix and upsample it to  $256 \times 256$  for  $x$  and  $y$  and concatenate the two channels to generate the ground truth translation map. Random seed in Table 1 denotes the random entries generated in the transformation matrices for each image pair. For instance, in translation transformation, the random entries in  $T$  are  $t_x$  and  $t_y$  in range  $[-5, 5]$ .

The testing set is separated into 5 types, which are translation, rotation, scaling, shearing, pixelwise (nonrigid), and each contains 40 images.

## 4.2 Experiment setup

The SimpleElastix, ORB are implemented in Python and Intensity-based registration and the proposed algorithm are implemented in Matlab 2017b on a desktop with CPU Intel(R) Xeon(R) W-2195 CPU @ 2.30GHz.

## 4.3 Evaluation metrics

The quantitative evaluation is conducted by calculating root-mean-square error (RMSE) and mean absolute error (MAE) between the estimated translation map and ground truth translation map, each with a size of  $256 \times 256 \times 2$ .

### 4.3.1 Root mean square error

Let  $\hat{t}_{ij}$  denotes the element in estimated translation map and  $t_{ij}$  denotes the element in ground truth translation map. The RMSE is calculated as

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^{N_{col}} \sum_{i=1}^{N_{row}} (\hat{t}_{ij} - t_{ij})^2}, \quad (18)$$

where  $N$  denotes the total number of points,  $N_{col}$  denotes number of column pixels,  $N_{row}$  denotes number of row pixels. In our case,  $N_{col} = N_{row} = 256$ .

### 4.3.2 Mean absolute error

The MAE is calculated as

$$MAE = \frac{1}{N} \sum_{j=1}^{N_{col}} \sum_{i=1}^{N_{row}} |\hat{t}_{ij} - t_{ij}|. \quad (19)$$

## 5 Results

### 5.1 Quantitative assessment

The quantitative assessment using RMSE metric in Cartesian  $x$  and  $y$  are reported in Table 2 and Table 3 respectively. The MAE in  $x$  and  $y$  are reported in Table 4 and Table 5. The best performance observed in rigid transformation testing is implemented by SimpleElastix. It achieves a high score in both RMSE and MAE with an average of 0.11 in  $x$ , 0.11 in  $y$  reported in RMSE and 0.09 in  $x$ , 0.09 in  $y$  reported in MAE. Our proposed method achieves a similar performance in both metrics. In non-rigid transformation, Intensity-based method achieves the best score with 2.83 in  $x$  and 3.23 in  $y$  using RMSE metric and our proposed method achieves the best score with 2.06 in  $x$  and 2.10 in  $y$  using MAE metric.

Table 2: RMSE error for x coordinate in pixel(px).

RMSE(px)	SimpleElastix	ORB	Intensity-based	Proposed
Translation	<b>0.11 ± 0.08</b>	0.26 ± 0.21	0.28 ± 0.18	0.87 ± 1.30
Rotation	<b>0.13 ± 0.09</b>	0.31 ± 0.23	0.28 ± 0.13	1.52 ± 2.38
Scaling	<b>0.09 ± 0.08</b>	0.48 ± 0.66	1.11 ± 1.21	1.01 ± 0.91
Shearing	<b>0.11 ± 0.10</b>	0.45 ± 0.31	4.80 ± 3.20	1.44 ± 1.01
Pixelwise	3.91 ± 3.60	4.60 ± 3.40	<b>2.83 ± 1.98</b>	3.10 ± 1.87

Table 3: RMSE error for y coordinate in pixel(px).

RMSE(px)	SimpleElastix	ORB	Intensity-based	Proposed
Translation	<b>0.10 ± 0.10</b>	0.26 ± 0.21	0.30 ± 0.20	1.53 ± 2.91
Rotation	<b>0.11 ± 0.10</b>	0.29 ± 0.26	0.26 ± 0.13	1.61 ± 2.22
Scaling	<b>0.11 ± 0.08</b>	0.47 ± 0.56	1.78 ± 1.58	1.35 ± 2.78
Shearing	<b>0.13 ± 0.13</b>	0.43 ± 0.34	7.80 ± 4.43	1.39 ± 1.80
Pixelwise	3.87 ± 2.27	4.24 ± 2.64	<b>3.23 ± 2.31</b>	3.29 ± 2.29

Table 4: MAE error for x coordinate in pixel(px).

MAE(px)	SimpleElastix	ORB	Intensity-based	Proposed
Translation	<b>0.09 ± 0.07</b>	0.21 ± 0.18	0.23 ± 0.15	0.42 ± 0.58
Rotation	<b>0.11 ± 0.07</b>	0.26 ± 0.19	0.23 ± 0.10	0.87 ± 1.40
Scaling	<b>0.08 ± 0.07</b>	0.41 ± 0.55	0.93 ± 1.02	0.57 ± 0.49
Shearing	<b>0.09 ± 0.09</b>	0.37 ± 0.26	3.99 ± 2.65	0.84 ± 0.59
Pixelwise	3.23 ± 2.95	3.84 ± 2.83	2.36 ± 1.62	<b>2.06 ± 0.97</b>

Table 5: MAE error for y coordinate in pixel(px).

MAE(px)	SimpleElastix	ORB	Intensity-based	Proposed
Translation	<b>0.09 ± 0.08</b>	0.22 ± 0.17	0.25 ± 0.17	0.72 ± 1.37
Rotation	<b>0.09 ± 0.08</b>	0.24 ± 0.21	0.23 ± 0.11	0.84 ± 0.97
Scaling	<b>0.10 ± 0.07</b>	0.40 ± 0.47	1.51 ± 1.34	0.75 ± 1.47
Shearing	<b>0.11 ± 0.11</b>	0.36 ± 0.28	6.58 ± 3.78	0.80 ± 0.88
Pixelwise	3.22 ± 1.91	3.53 ± 2.17	2.70 ± 1.90	<b>2.10 ± 1.23</b>

## 5.2 Visual assessment

The visual assessment is demonstrated in Fig. 2 and Fig. 3. In this project, the input is the *moving* image and the ground truth image is the image warped by the ground truth translation map. Pixelwise 1-4 in Fig. 3 denote different types of translation map generated by different random matrices.

From Fig. 2, we can see that rigid transformation methods based on SimpleElastix, ORB and Intensity-based produce a black boundary on *moved* images and lose some information. The reason is that the warping is performed for the entire image instead of each pixel. In column 5, our proposed method produces a more consistent result as it optimizes the displacement vector for each pixel instead of the entire image.

From Fig. 3, we can see that ORB and Intensity-based approach fail to produce pixelwise *moved* image. For instance in Pixelwise 4, the box lines are still straight in these two methods as the rigid transformation considers a linear transformation instead of a pixelwise warping. SimpleElastix demonstrates a impressive result in non-rigid transformation. Our proposed method demonstrates a comparable performance in Column 5.



Figure 2: Visual assessment for testing on rigid transformation.

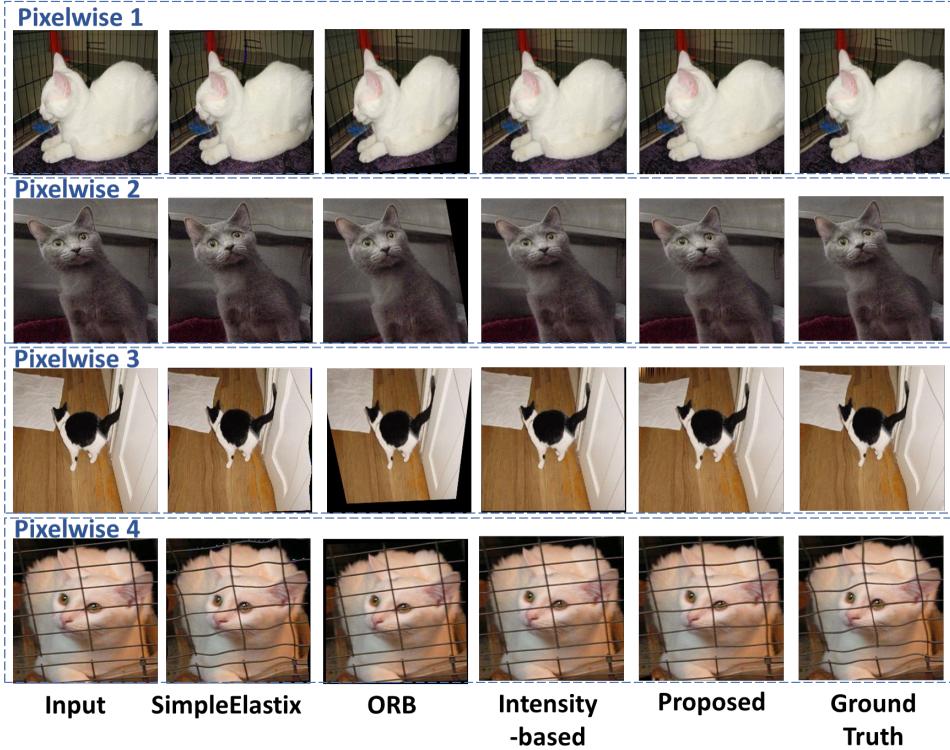


Figure 3: Visual assessment for testing on non-rigid transformation.

## 6 Conclusion

In this report, we present a non-rigid image registration algorithm using convex optimization. We first formulate the problem as a convex problem and then introduce a sequential optimization framework using augmented lagrangian method. We also compare our proposed algorithm with other non-learning based registration approaches including SimpleElastix, ORB and Intensity-based registration. Our results are evaluated quantitatively using RMSE and MAE and we observe that our proposed algorithm demonstrates a comparable performance with other non-learning based approaches with improved performance on non-rigid transformation reported in MAE. Although the algorithm shows a satisfying result, the computation time for our proposed method is relatively long. In the future, we will try to improve the method with a more efficient optimization procedure.

## 7 Acknowledgement

The author would like to thank Prof. Vandenberghe for the great series from 236B to 236C. Like other students who really enjoy this journey, the author sincerely wish Prof. Vandenberghe would open more related courses in the future.

## References

- Guha Balakrishnan, Amy Zhao, Mert R Sabuncu, John Guttag, and Adrian V Dalca. Voxelmorph: a learning framework for deformable medical image registration. *IEEE transactions on medical imaging*, 38(8):1788–1800, 2019.
- Kasper Marstal, Floris Berendsen, Marius Staring, and Stefan Klein. Simpleelastix: A user-friendly, multi-lingual library for medical image registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 134–142, 2016.
- Martin Rajchl, John SH Baxter, Wu Qiu, Ali R Khan, Aaron Fenster, Terry M Peters, and Jing Yuan. Rancor: Non-linear image registration with total variation regularization. *arXiv preprint arXiv:1404.2571*, 2014.
- Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, volume 11, page 2. Citeseer, 2011.
- Paul Ruhnau, Timo Kohlberger, Christoph Schnörr, and Holger Nobach. Variational optical flow estimation for particle image velocimetry. *Experiments in Fluids*, 38(1):21–32, 2005.
- Jing Yuan and Aaron Fenster. Modern convex optimization to medical image analysis. *arXiv preprint arXiv:1809.08734*, 2018.
- Xiaoran Zhang, Hexiang Dong, Di Gao, and Xiao Zhao. A comparative study for non-rigid image registration and rigid image registration. *arXiv preprint arXiv:2001.03831*, 2020.

## Appendix

### List of notations

Suppose the image size is  $M \times N \times C$  and pyramid decay factor is 0.5

Notation	Symbol	Dimension
moving image	$I_m$	$M \times N \times C$
fixed image	$I_f$	$M \times N \times C$
pyramid level	$L$	-
DVF at pyramid level $l$	$\mathbf{u}^l$	$2^{l-L}M \times 2^{l-L}N \times 2$
incremental DVF at level $l$	$\mathbf{h}^l$	$2^{l-L}M \times 2^{l-L}N \times 2$
total incremental at level $l$	$\mathbf{t}^l$	$2^{l-L}M \times 2^{l-L}N \times 2$
dual variable for dissimilarity	$\mathbf{w}$	$2^{l-L}M \times 2^{l-L}N$
dual variable for regularization	$\mathbf{q}$	$2^{l-L}M \times 2^{l-L}N \times 2$
image warping	$\circ$	-
total variation penalty	$\alpha$	-
ALM penalty	$c$	-

Table 6: Essential notations used in this report

### Dual derivation

To derive the dual of the optimization problem described in Eq. 8, we firstly formulate the dissimilarity part of the problem using the conjugate function of  $L_1$  norm, we have

$$\sum \sum |\tilde{P}_0^k + \nabla \tilde{P}^k \cdot \mathbf{h}^k| = \max_{|\mathbf{w}[x,y]| \leq 1} \sum \sum \mathbf{w}[x,y] (\tilde{P}_0^k[x,y] + (\nabla \tilde{P}^k \cdot \mathbf{h}^k)[x,y]) \quad (20)$$

where  $\mathbf{w}$  is the dual variable. For the second convex regularization term, given the obvious equivalence  $q^T \nabla f = f \text{div } q$  and the conjugate function of  $L_1$  norm, we have the following:

$$\alpha \sum_{i=1}^2 \left( \sum \sum |\nabla(\tilde{\mathbf{u}}_i + \mathbf{h}_i^k)| \right) = \max_{\mathbf{q}} \sum_{i=1}^2 \left( \sum \sum \text{div} \mathbf{q}_i (\tilde{\mathbf{u}}_i + \mathbf{h}_i^k) \right) - R_\alpha^*(\mathbf{q}) \quad (21)$$

where  $\mathbf{q}_i = [\mathbf{q}_{ix}, \mathbf{q}_{iy}]$  is the dual variable corresponds to the each dimension in  $\nabla(\tilde{\mathbf{u}}_i + \mathbf{h}_i^k)$  and  $R_\alpha^*$  is the reciprocal of the standard indicator function defined as

$$R_\alpha^*(q) = \begin{cases} 0, & q \leq \alpha \\ \infty, & \text{otherwise} \end{cases}. \quad (22)$$

Combining Eq. 20 and Eq. 21, we have the following

$$\max_{|\mathbf{w}[x,y]| \leq 1, \mathbf{q}} \sum \sum \mathbf{w}[x,y] (\tilde{P}_0^k[x,y] + (\nabla \tilde{P}^k \cdot \mathbf{h}^k)[x,y]) + \sum_{i=1}^2 \left( \sum \sum \text{div} \mathbf{q}_i (\tilde{\mathbf{u}}_i + \mathbf{h}_i^k) \right) - R_\alpha^*(\mathbf{q}) \quad (23)$$

After rearranging the variable  $\mathbf{h}^k$ , we have the following primal-dual formulation

$$\begin{aligned} \min_{\mathbf{h}^k} \max_{|\mathbf{w}[x,y]| \leq 1, \mathbf{q}} & \sum \sum (\mathbf{w}[x,y] \tilde{P}_0^k[x,y] + \sum_{i=1}^2 \tilde{\mathbf{u}}_i[x,y] \text{div} \mathbf{q}_i[x,y]) \\ & + \sum_{i=1}^2 \left( \sum \sum \mathbf{h}_i^k[x,y] (\mathbf{w}[x,y] \partial_i \tilde{P}^k[x,y] + \text{div} \mathbf{q}_i[x,y]) \right) - R_\alpha^*(\mathbf{q}). \end{aligned} \quad (24)$$

The minimization over  $\mathbf{h}^k$  results in the following linear equality constraints

$$w[x,y] \partial_i \tilde{P}^k[x,y] + \text{div} \mathbf{q}_i[x,y] = 0 \quad i = 1, 2 \quad (25)$$

Thus, from Eq. 24 and Eq. 25, we have the following dual formulation:

$$\begin{aligned} & \max_{|\mathbf{w}[x,y]| \leq 1, \mathbf{q}} \sum \sum (\mathbf{w}[x,y] \tilde{P}_0^k[x,y] + \sum_{i=1}^2 \tilde{\mathbf{u}}_i[x,y] \operatorname{div} \mathbf{q}_i[x,y]) - R_\alpha^*(\mathbf{q}) \\ \text{s.t.} \quad & \mathbf{w}[x,y] \partial_i \tilde{P}^k[x,y] + \operatorname{div} \mathbf{q}_i[x,y] = 0 \quad i = 1, 2 \end{aligned} \quad (26)$$

thereby proving Eq. 9