Introduction to Dense Methods for Image Alignment

Ivan Gerasimov

August 28, 2023

Contents

1 Introduction

Image alignment is a fundamental task in computer vision that involves establishing correspondences between points, features, or pixels in multiple images. Among the various techniques available, dense methods for image alignment play a significant role. Dense methods aim to align images at a pixel-level granularity, providing high precision and enabling accurate spatial transformations. These methods consider the alignment of every pixel, which can be particularly useful when dealing with complex deformations or changes in lighting and perspective.

Dense image alignment methods seek to find a transformation that minimizes the difference between corresponding pixels in multiple images. This is achieved by optimizing a similarity metric that quantifies the similarity or dissimilarity between the pixel values of the images being aligned. Unlike sparse methods that focus on specific features or points, dense methods consider the entire image, allowing for more comprehensive alignment.

2 Problem Statement

2.1 Definitions

Def 2.1 (Image)**.** *We will consider image as function over coordinate space that for certain coordinate $\mathbf{x} \in \mathcal{R}^2$ it provides the value of luminous intensity. So the image is function:*

$$I(\mathbf{x}) : \mathcal{D} \to \mathcal{R}_{[0,1]},$$

where $\mathcal{D}$ is closed and bounded subset of Euclidean space in $\mathcal{R}^2$. Later, we will denote a template image as T and an input image as I.

Def 2.2 (Warp)**.** *The family of warps $\mathcal{F}$ parameterized by n parameters:*

$$\mathcal{W} : \mathcal{R}^2 \times \mathcal{R}^n \to \mathcal{R}^2$$

It takes coordinates $\mathbf{x}$ and set of parameters $\mathbf{p}$ and transform coordinated with some rule corresponding the parameters: $(\mathbf{x}, \mathbf{p}) \to \mathcal{W}(\mathbf{x}, \mathbf{p}) = \tilde{\mathbf{x}}$ ($\mathbf{x}, \tilde{\mathbf{x}} \in \mathcal{R}^2$ and $\mathbf{p} \in \mathcal{R}^n$).

Together with Warp we gonna use ***Warped Image*** *definition:*

$$I_p(\mathbf{x}) = I(\mathcal{W}(\mathbf{x}, \mathbf{p})),\ x \in \mathcal{D}$$

Warps examples:

- *Affine Transformations*
- *Rigid Body Transformations*
- *Geometric Transformations*

2.2 Image Alignment problem formulation

We gonna consider the Image Alignment problem as problem of estimating parameters $\mathbf{p}_I$ (for a given input image I) of the warp mapping $\mathcal{W}$ for the given reference image T and the input image I, such that:

$$T(\mathbf{x}) = I_p(\mathbf{x}) = I(\mathcal{W}(\mathbf{x}, \mathbf{p})),\ \forall \mathbf{x} \in \tilde{\mathcal{D}}, \tag{1}$$

where $\tilde{\mathcal{D}} \subset \mathcal{D}$ and $\neg \exists\ \mathcal{D}^* : \mathcal{D}^* \subset \mathcal{D}, |\mathcal{D}^*| > |\tilde{\mathcal{D}}|$ and (1) holds $\forall \mathbf{x} \in \mathcal{D}^*$.

But in real world we have a lot of uncertainties (noise for example) and we cannot guarantee "equal" sign in (1) but we want to overlay one image on top of another as accurately as possible. That brings us to optimization formulation.

2.3 Optimisation Framework

Let's rewrite the problem (1) as optimization problem, supposed that we want to align images as close as possible and find the set of parameters $\mathbf{p}_I$ for input image transformation:

$$\mathbf{p}_I = \arg\min_{\mathbf{p}\in\mathcal{R}^n} \sum_{\mathbf{x}\in\mathcal{D}} (I(\mathcal{W}(\mathbf{x},\mathbf{p})) - T(\mathbf{x}))^2 \tag{2}$$

Now we gonna work around the equation above and one of the most popular way is to try to update some set of parameters $\mathbf{p}_c$ (current) that reduce the goal function in (2). Based on that idea we gonna build a couple of methods for estimation warp parameters update $\delta\mathbf{p}$.

3 First Order Methods

3.1 Forward Additive Method

Let's decompose equation for warped input image around the neighborhood of the point $\mathbf{p}_c$:

$$F(\delta\mathbf{p}) = \sum_{\mathbf{x}\in\mathcal{D}} (I(\mathcal{W}(\mathbf{x},\mathbf{p}_c+\delta\mathbf{p})) - T(\mathbf{x}))^2 \tag{3}$$

$$\approx \sum_{\mathbf{x}\in\mathcal{D}} (I(\mathcal{W}(\mathbf{x},\mathbf{p}_c)) + J(\mathbf{x},\mathbf{p}_c)\delta\mathbf{p} - T(\mathbf{x}))^2 \tag{4}$$

where $J(\mathbf{x},\mathbf{p})$ is calculated by the rule for differentiation of a complex function $\frac{\partial I(\mathcal{W}(\mathbf{x},\mathbf{p}))}{\partial\mathbf{p}}$:

$$J(\mathbf{x},\mathbf{p}_c) = \frac{\partial I(\mathcal{W}(\mathbf{x},\mathbf{p}))}{\partial\mathbf{p}}\Big|_{\mathbf{p}=\mathbf{p}_c} \tag{5}$$

$$= \nabla I(\mathbf{x}) \frac{\partial\mathcal{W}(\mathbf{x},\mathbf{p})}{\partial\mathbf{p}}\Big|_{\mathbf{p}=\mathbf{p}_c} \tag{6}$$

where $\nabla I(\mathbf{x})$ is the vector (with shape 1×2) of image gradient at point $\mathbf{x}$, $\frac{\partial\mathcal{W}(\mathbf{x},\mathbf{p})}{\partial\mathbf{p}}$ is matrix (with shape $2\times n$) with structure:

$$\frac{\partial\mathcal{W}(\mathbf{x},\mathbf{p})}{\partial\mathbf{p}} = \begin{bmatrix} \frac{\partial\mathcal{W}_x}{\partial\mathbf{p}_1} & \cdots & \frac{\partial\mathcal{W}_x}{\partial\mathbf{p}_n} \\ \frac{\partial\mathcal{W}_y}{\partial\mathbf{p}_1} & \cdots & \frac{\partial\mathcal{W}_y}{\partial\mathbf{p}_n} \end{bmatrix}$$

The equation for $F(\delta\mathbf{p})$ (4) is a quadratic form relative to $\delta\mathbf{p}$ and moreover it is convex, that means we may write necessary condition for local minimum:

$$\frac{\partial F(\delta\mathbf{p})}{\partial\mathbf{p}} = 0 \Rightarrow \left\langle \sum_{\mathbf{x}\in\mathcal{D}} 2J(\mathbf{x},\mathbf{p}_c)^\top (I(\mathcal{W}(\mathbf{x},\mathbf{p}_c)) + J(\mathbf{x},\mathbf{p}_c)\delta\mathbf{p} - T(\mathbf{x})), \delta\mathbf{p} \right\rangle = 0$$

By opening the brackets, we can explicitly get an update for our parameters $\delta\mathbf{p}$:

$$\delta\mathbf{p} = \left(\sum_{\mathbf{x}\in\mathcal{D}} J(\mathbf{x},\mathbf{p}_c)^\top J(\mathbf{x},\mathbf{p}_c)\right)^{-1} \left(\sum_{\mathbf{x}\in\mathcal{D}} J(\mathbf{x},\mathbf{p}_c)^\top (T(\mathbf{x}) - I(\mathcal{W}(\mathbf{x},\mathbf{p}_c)))\right)$$

or:

$$\delta\mathbf{p} = (H(\mathbf{x},\mathbf{p}_c))^{-1} \left(\sum_{\mathbf{x}\in\mathcal{D}} J(\mathbf{x},\mathbf{p}_c)^\top (T(\mathbf{x}) - I(\mathcal{W}(\mathbf{x},\mathbf{p}_c)))\right), \tag{7}$$

where

$$H(\mathbf{x},\mathbf{p}_c) = \sum_{\mathbf{x}\in\mathcal{D}} J(\mathbf{x},\mathbf{p}_c)^\top J(\mathbf{x},\mathbf{p}_c) \tag{8}$$

Now, we have explicit formula for parameters update and as a result new estimation for parameters:

$$\mathbf{p}_{c+1} = \mathbf{p}_c + \delta\mathbf{p}$$

But it was found that to make the step on the $\delta\mathbf{p}$ is too much (in case of large noise) and therefore we proposed to do steps somewhat scaled by a factor α:

$$\mathbf{p}_{c+1} = \mathbf{p}_c + \alpha \cdot \delta\mathbf{p}$$

That brings us to the formulation of the Forward Additive algorithm:

Algorithm 1: Forward Additive Algorithm

Input: Input Image $I(\mathbf{x})$, Template Image $T(\mathbf{x})$, warp transformation $\mathcal{W}(\mathbf{x},\mathbf{p})$, its derivative $\frac{\partial\mathcal{W}}{\partial p}$ and initial parameters $\mathbf{p}_{init}$, α, maximum number of optimization steps N, ϵ

Output: Vector of parameters $\tilde{\mathbf{p}}$ for the given warp $\mathcal{W}(\mathbf{x},\mathbf{p})$

$p_c = p_{init}$

for $i = 1,\ i \leq N,\ i+ = 1$ **do**

  Compute $J(\mathbf{x},\mathbf{p}_c)$ for all $\mathbf{x} \in \mathcal{D}$ with equation (6)

  Compute $H(\mathbf{x},\mathbf{p}_c)$ with equation (8)

  Compute $\delta\mathbf{p}$ with equation (7) Update parameters with $\mathbf{p}_c = \mathbf{p}_c + \alpha \cdot \delta\mathbf{p}$

  if $\|\delta\mathbf{p}\|_2 < \epsilon$ **then**

    output $\mathbf{p}_c$

output $\mathbf{p}_c$

3.2 Another First Order Methods

Developing the idea described above, you can build a number of methods that offer a different view of the shape of the objective function and the way to update the parameters, but they all start from the idea of decomposing the transformed image in the vicinity of the current estimate of the transformation parameters and then everything is as in the algorithm described above. More detailed details about these algorithms and their complexity depending on the size of the input data can be found here [2], [1], [5].

3.3 The Effect of Noise

This section is devoted to what can be expected in the presence of noise in the images (T and I). For ease of understanding, white noise was taken, which was added to the pictures and the objective function of the mimization problem (2) was analyzed.

Considering two images T and I we apply both of them white noise with σ and we observed the convergence of the Forward Additive Algorithm 3.1 for the set of parameters for Affine Warp.

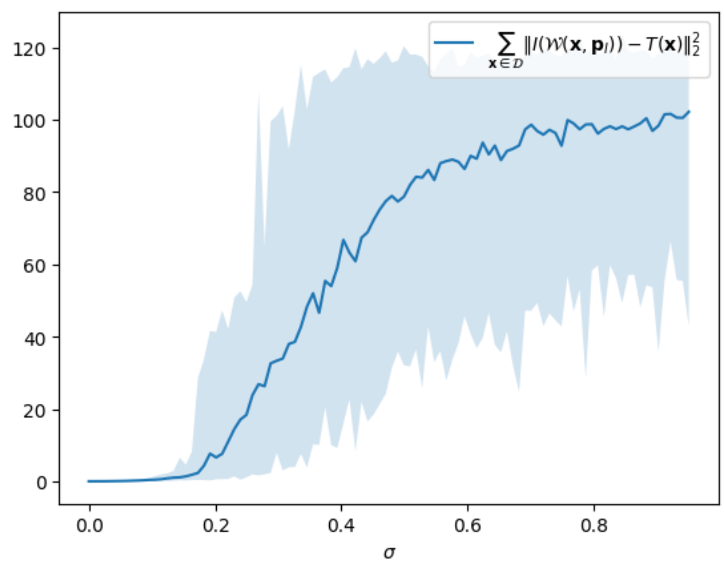

Figure 1: The effect of white noise (applied to both T and I images) on the value of the optimization objective function when using Forward Additive Algoritm 1 for the following parameters: $N = 50$, $\epsilon = 0.001$, $\alpha = 1.0$.

For each sigma value, we conducted 100 experiments after that, the result was averaged. Each experiment consists of the next steps:

1. Read T and I images.

2. Add Gauss Noise with parameters $\mathcal{N}(0, \sigma)$ to I and T.
3. Find optimal parameters $\mathbf{p}_I$ for Affine Transformations.
4. Apply Affine Transformations with parameters $\mathbf{p}_I$ to image I and calculate the objective function

$$\sum_{\mathbf{x} \in \mathcal{D}} \|I(\mathcal{W}(\mathbf{x}, \mathbf{p}_I)) - T(\mathbf{x})\|_2^2$$

4 Pros and Cons of Dense Methods for Image Alignment

4.1 Advantages

1. **Pixel-Level Accuracy:** Dense methods offer high accuracy since they align images at the pixel level. This level of precision is especially beneficial for applications that demand exact alignment (exactly our case with images from microscope).
2. **Global Information:** Dense methods consider the entire image, capturing global information about the alignment. This is advantageous when dealing with complex deformations or changes in perspective that might affect the entire image.
3. **Smoothness and Continuity:** Due to their pixel-level alignment, dense methods tend to produce smoother and more visually coherent results. This is particularly useful for tasks like video stabilization, super-resolution.
4. **Interpolation and Synthesis:** Dense methods can be used for tasks beyond alignment, such as image interpolation and synthesis. They can generate high-resolution images by aligning and combining lower-resolution frames. That is also useful for our images from microscope because we may work with low resolution of frames.
5. **Robustness to Outliers:** Dense methods can handle outliers and occlusions better than some sparse methods because they take into account a broader context of pixels in the alignment process. During our experiments with sparce methods (SuperPoint [3] + LightGlue [4]) and dense methods on frames from microscope we were convinced of absolute helplessness of sparse methods because they provide not enough number of sparse features and they are too similar for matching.

4.2 Disadvantages

1. **Computational Intensity:** Dense methods involve aligning every pixel in the images, which can be computationally intensive and time-consuming. Real-time performance can be a challenge, especially for large images or video sequences.

2. **Sensitive to Noise:** Dense methods can be sensitive to noise, as the alignment relies on pixel values. Noise or artifacts in the images can affect the accuracy of the alignment.

3. **Memory Requirements:** Aligning large images can require substantial memory, both for storing the images and the intermediate computations. This can be a limitation on resource-constrained systems.

4. **Limited to Similarity Measures:** Dense methods often rely on pixel-wise similarity measures, which might not capture more complex relationships between image features. This can limit their effectiveness in cases where structural information is important.

5. **Non-Rigid Transformations:** While some dense methods can handle non-rigid deformations, they might struggle with highly complex or extreme deformations due to the limitations of pixel-based alignment.

References

[1] Simon Baker and Iain Matthews. Equivalence and efficiency of image alignment algorithms. volume 1, pages I–1090, 02 2001.

[2] Alberto Crivellaro, Pascal Fua, and Vincent Lepetit. Dense methods for image alignment with an application to 3d tracking optimization framework, 01 2014.

[3] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description, 2018.

[4] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023.

[5] BD Lucas and Takeo Kanade. An iterative image registration technique with an application tostereo vision. volume 2, pages 674–679, 08 1997.