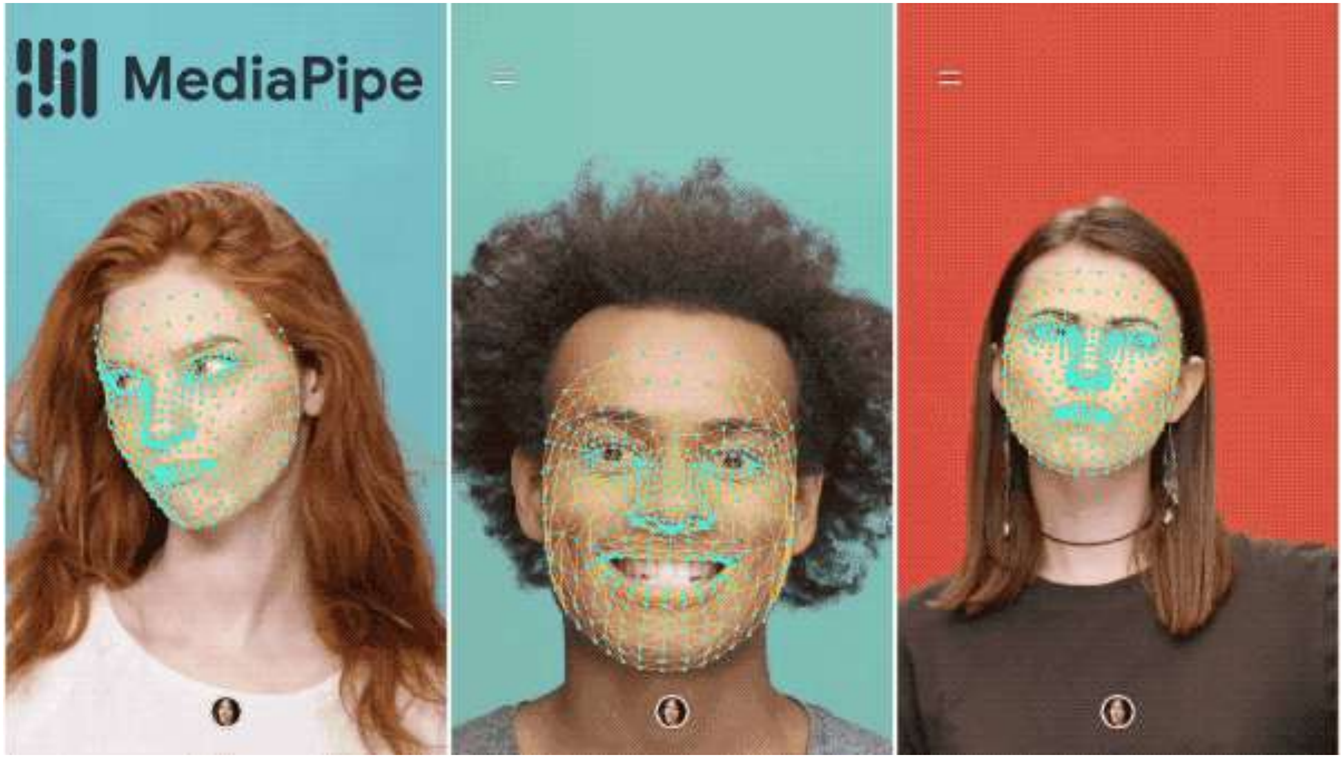
中国科学技术大学
University of Science and Technology of China

数字图像处理
Digital Image Processing

Homework_01 Image Warping

https://github.com/YudongGuo/DIP-Teaching/tree/main/Assignments/01_ImageWarping

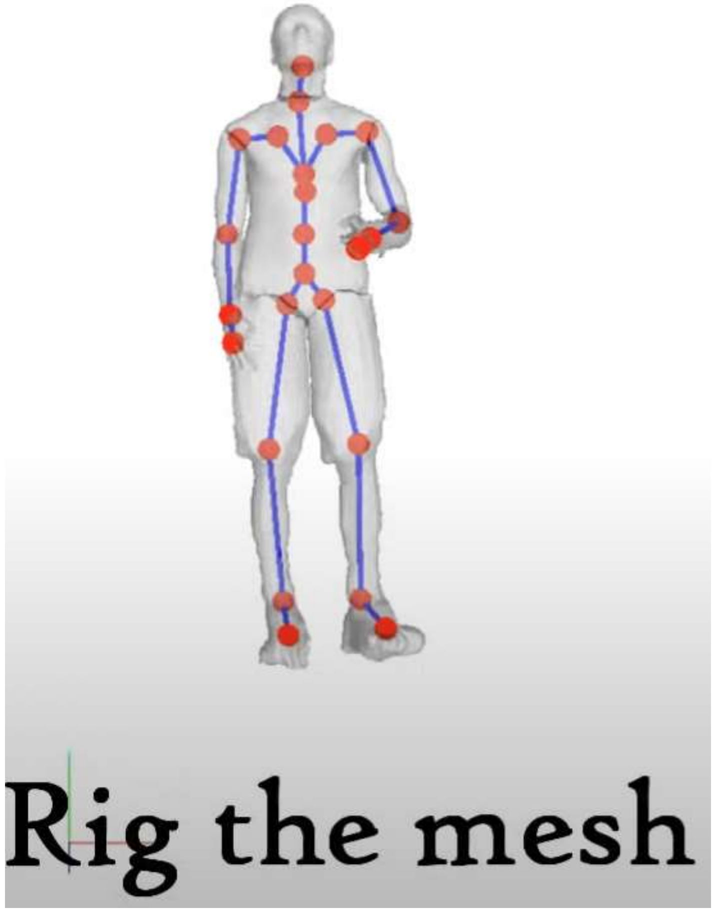
Why we need these transformations?



Rigid Transform
Rotation
Translation
Scaling

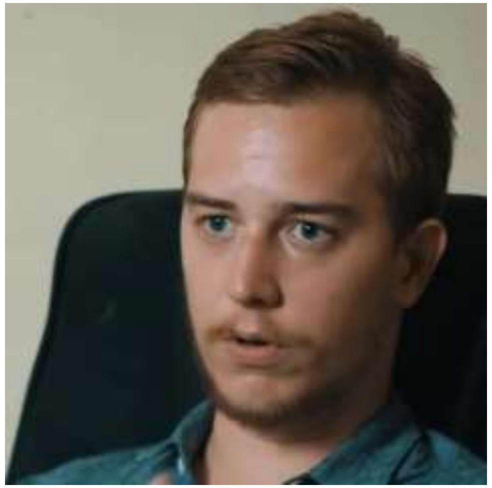
Why we need these transformations?

Why we need these transformations?

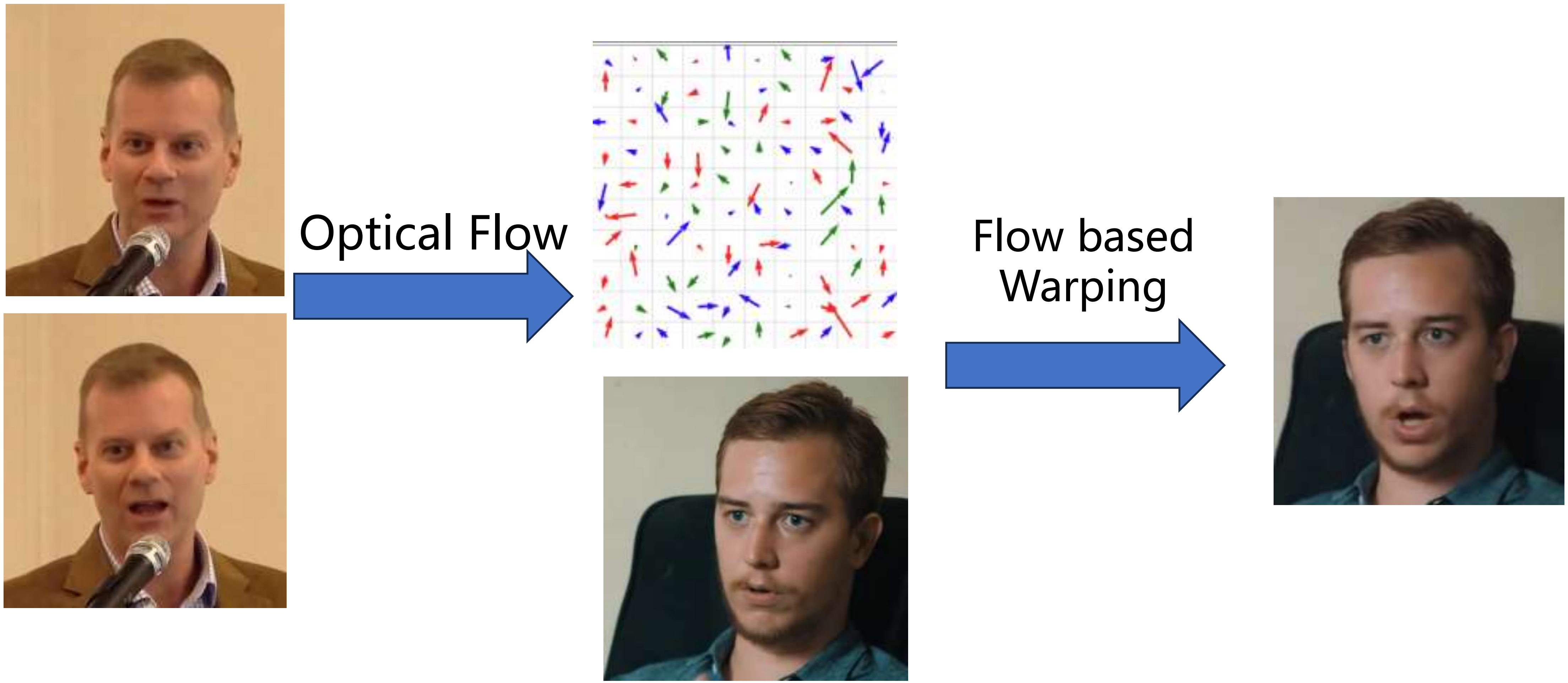


Point rigged deformation

Why we need these transformations?



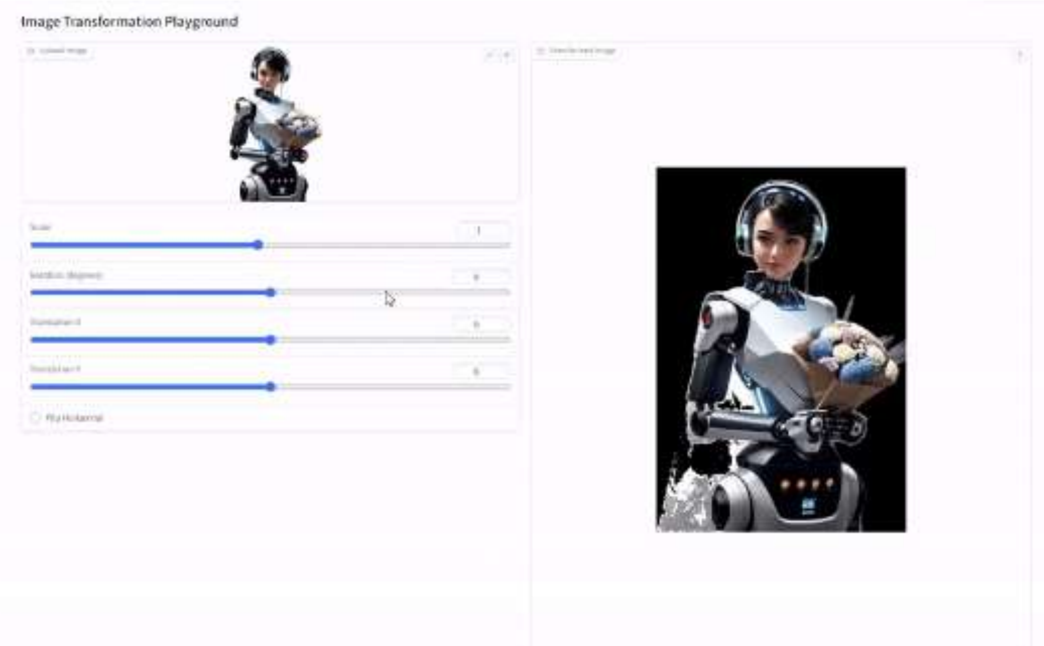
Why we need these transformations?



Assignment_01

1. Basic Image Geometric Transformation (Scale/Rotation/Translation).

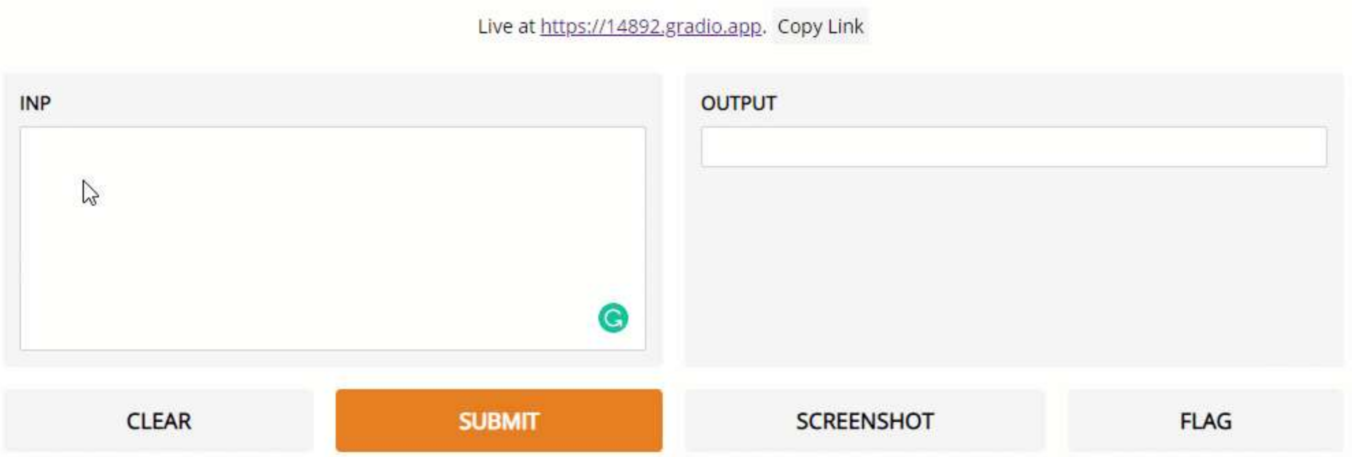
Fill the [Missing Part]() of 'run_global_transform.py'.



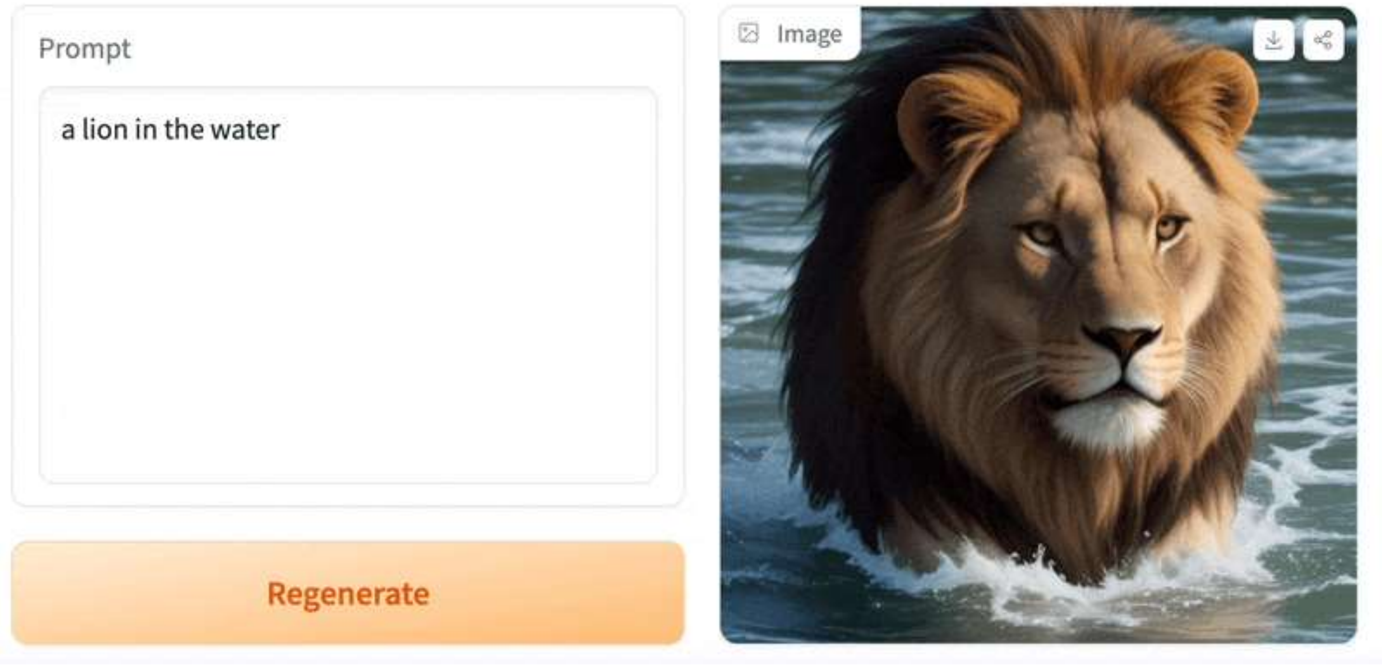
2. Point Based Image Deformation.

Implement MLS or RBF based image deformation in the [Missing Part]() of 'run_point_transform.py'.

Gradio



建议熟练掌握，后续用来交互调试参数、测试与演示，无需重复编译运行



Point based warping Paper

局部变形

绝大部分不动

局部考虑（逐像素）

Point based warping Paper

We view this deformation as a function f that maps points in the undeformed image to the deformed image. Applying the function f to each point v in the undeformed image creates the deformed image. Now consider an image with a set of handles p that the user moves to new positions q. For f to be useful for deformations it must satisfy the following properties:

- *Interpolation*: The handles p should map directly to q under deformation. (i.e; $f(p_i) = q_i$).
- *Smoothness*: f should produce smooth deformations
- *Identity*: If the deformed handles q are the same as the p, then f should be the identity function. (i.e; $q_i = p_i \Rightarrow f(v) = v$).

每一个像素v对应的变换f需要满足的性质

Point based warping Paper

2 Moving Least Squares Deformation

Here we consider building image deformations based on collections of points with which the user controls the deformation. Let p be a set of control points and q the deformed positions of the control points p. We construct a deformation function f satisfying the three properties outlined in the introduction using Moving Least Squares [Levin 1998]. Given a point v in the image, we solve for the best affine transformation $l_v(x)$ that minimizes

$$\sum_i w_i |l_v(p_i) - q_i|^2 \tag{1}$$

where p_i and q_i are row vectors and the weights w_i have the form

$$w_i = \frac{1}{|p_i - v|^{2\alpha}}.$$

Because the weights w_i in this least squares problem are dependent on the point of evaluation v, we call this a *Moving Least Squares* minimization. Therefore, we obtain a different transformation $l_v(x)$ for each v.

论文提出的一种满足以上约束的变换形式

Point based warping Paper

2.1 Affine Deformations

Finding an affine deformation that minimizes equation 4 is straightforward using the classic normal equations solution.

$$M = \left(\sum_i \hat{p}_i^T w_i \hat{p}_i\right)^{-1} \sum_j w_j \hat{p}_j^T \hat{q}_j.$$

Though this solution requires the inversion of a matrix, the matrix is a constant size (2×2) and is fast to invert. With this closed-form solution for M we can write a simple expression for the deformation function $f_a(v)$.

$$f_a(v) = (v - p_*)\left(\sum_i \hat{p}_i^T w_i \hat{p}_i\right)^{-1} \sum_j w_j \hat{p}_j^T \hat{q}_j + q_*. \tag{5}$$

Applying this deformation function to each point in the image creates a new, deformed image.

While the user creates these deformations by manipulating the points q, the points p are fixed. Since the p do not change during deformation, much of equation 5 can be precomputed yielding very fast deformations. In particular, we can rewrite equation 5 in the form

$$f_a(v) = \sum_j A_j \hat{q}_j + q_*.$$

where A_j is a single scalar given by

$$A_j = (v - p_*)\left(\sum_i \hat{p}_i^T w_i \hat{p}_i\right)^{-1} w_j \hat{p}_j^T.$$

Notice that, given a point v, everything in A_j can be precomputed yielding a simple, weighted sum. Table 1 provides timing results for the examples in this paper, which shows that these deformations may be performed over 500 times per second in our examples.

Figure 1 (b) illustrates this affine Moving Least Squares deformation applied to our test image. Unfortunately, the deformation does not appear very desirable due to the stretching in the arms and torso. These artifacts are created because affine transformations include deformations such as non-uniform scaling and shear. To eliminate these undesirable deformations we need to consider restricting the linear transformation $l_v(x)$. In particular, we modify the class of deformations $l_v(x)$ produces by restricting the transformation matrix M from being fully linear to similarity and rigid-body transformations.

论文推导的实现形式

2.3 Rigid Deformations

Recently, several works [Alexa et al. 2000; Igarashi et al. 2005] have shown that, for realistic shapes, deformations should be as rigid as possible; that is, the space of deformations should not even include uniform scaling. Traditionally researchers in deformation have been reluctant to approach this problem directly due to the non-linear constraint that $M^T M = I$. However, we note that closed-form solutions to this problem are known from the Iterated Closest Point community [Horn 1987]. Horn shows that the optimal rigid transformation can be found in terms of eigenvalues and eigenvectors of a covariance matrix involving the points p_i and q_i. We show that these rigid deformations are related to the similarity deformations from section 2.2 via the following theorem.

Theorem 2.1 *Let C be the matrix that minimizes the following similarity functional*

$$\min_{M^T M = \lambda^2 I} \sum_i w_i |\hat{p}_i M - \hat{q}_i|^2.$$

If C is written in the form λR where R is a rotation matrix and λ is a scalar, the rotation matrix R minimizes the rigid functional

$$\min_{M^T M = I} \sum_i w_i |\hat{p}_i M - \hat{q}_i|^2.$$

Proof: See Appendix A.

This theorem is valid in arbitrary dimension, however, it is very easy to apply in $2D$. Using this theorem, we find that the rigid transformation is exactly the same as equation 6 except that we use a different constant μ_r in the solution so that $M^T M = I$ given by

$$\mu_r = \sqrt{\left(\sum_i w_i \hat{q}_i \hat{p}_i^T\right)^2 + \left(\sum_i w_i \hat{q}_i \hat{p}_i^{\perp T}\right)^2}.$$

Unlike the similarity deformation $f_s(v)$, we cannot precompute as much information for the rigid deformation function $f_r(v)$. However, the deformation process can still be made very efficient. Let

$$\vec{f}_r(v) = \sum_i \hat{q}_i A_i$$

where A_i is defined in equation 7, which may be precomputed. This vector $\vec{f}_r(v)$ is a rotated and scaled version of the vector $v - p_*$. To compute $f_r(v)$ we normalize $\vec{f}_r$, scale by the length of $v - p_*$ (which also can be precomputed), and translate by q_*.

$$f_r(v) = |v - p_*| \frac{\vec{f}_r(v)}{|\vec{f}_r(v)|} + q_*. \tag{8}$$

Point based warping Paper

2.3 Rigid Deformations

Recently, several works [Alexa et al. 2000; Igarashi et al. 2005] have shown that, for realistic shapes, deformations should be as rigid as possible; that is, the space of deformations should not even include uniform scaling. Traditionally researchers in deformation have been reluctant to approach this problem directly due to the non-linear constraint that $M^T M = I$. However, we note that closed-form solutions to this problem are known from the Iterated Closest Point community [Horn 1987]. Horn shows that the optimal rigid transformation can be found in terms of eigenvalues and eigenvectors of a covariance matrix involving the points p_i and q_i. We show that these rigid deformations are related to the similarity deformations from section 2.2 via the following theorem.

Theorem 2.1 *Let C be the matrix that minimizes the following similarity functional*

$$\min_{M^T M = \lambda^2 I} \sum_i w_i |\hat{p}_i M - \hat{q}_i|^2 .$$

If C is written in the form λR where R is a rotation matrix and λ is a scalar, the rotation matrix R minimizes the rigid functional

$$\min_{M^T M = I} \sum_i w_i |\hat{p}_i M - \hat{q}_i|^2 .$$

Proof: See Appendix A.

This theorem is valid in arbitrary dimension, however, it is very easy to apply in $2D$. Using this theorem, we find that the rigid transformation is exactly the same as equation 6 except that we use a different constant μ_r in the solution so that $M^T M = I$ given by

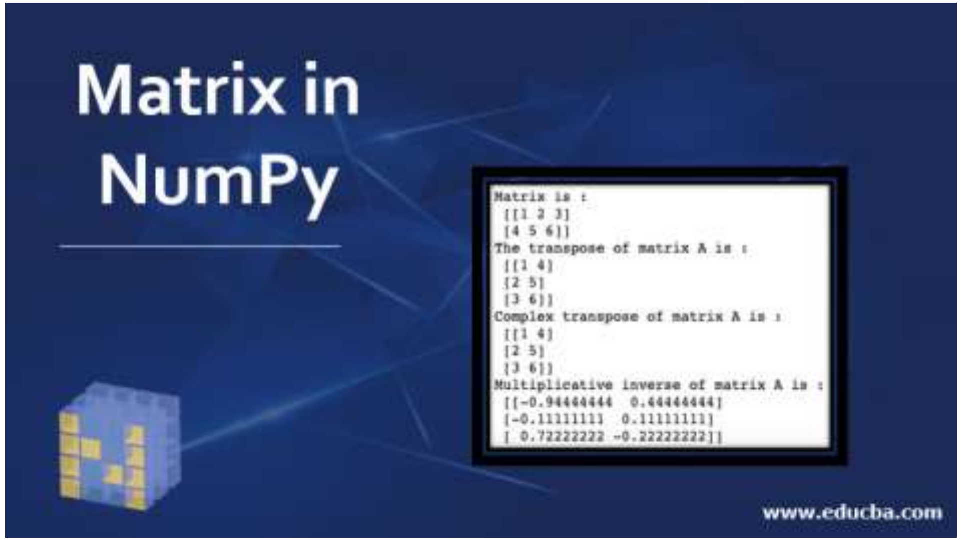
$$\mu_r = \sqrt{\left(\sum_i w_i \hat{q}_i \hat{p}_i^T\right)^2 + \left(\sum_i w_i \hat{q}_i \hat{p}_i^{\perp T}\right)^2} .$$

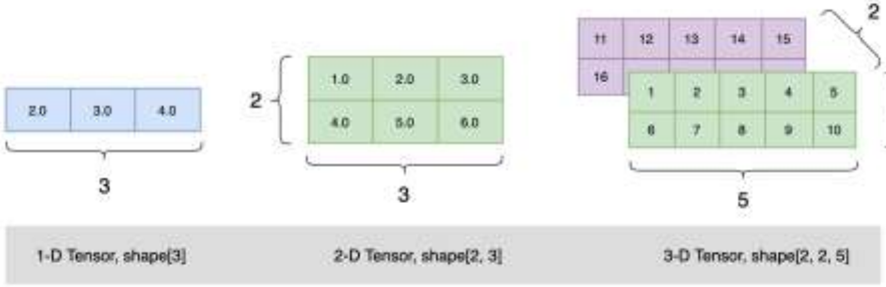
Unlike the similarity deformation $f_s(v)$, we cannot precompute as much information for the rigid deformation function $f_r(v)$. However, the deformation process can still be made very efficient. Let

$$\vec{f}_r(v) = \sum_i \hat{q}_i A_i$$

where A_i is defined in equation 7, which may be precomputed. This vector $\vec{f}_r(v)$ is a rotated and scaled version of the vector $v - p_*$. To compute $f_r(v)$ we normalize $\vec{f}_r$, scale by the length of $v - p_*$ (which also can be precomputed), and translate by q_*.

$$f_r(v) = |v - p_*| \frac{\vec{f}_r(v)}{|\vec{f}_r(v)|} + q_* . \qquad (8)$$





大概的实现流程（演示）

中国科学技术大学
University of Science and Technology of China

Q & A