

实验报告

魏远毅

2023 年 10 月 11 日

1 实验要求

移动最小二乘法并未对转换矩阵 M 进行条件限制, 如果添加其他限制条件后, 能得到不同形式的转换矩阵 M , 文章根据不同的转换矩阵 M 提出了三种变形方式:

仿射变换
相似变换
刚性变换

2 算法原理

2.1 移动最小二乘法原理

算法目标: $\min \sum_i w_i |l_v(p_i) - q_i|^2$ (1)

其中 l_v 为仿射变换, p_i 与 q_i 分别为拖动前后的点, 论文中采用权重 $w_i = \frac{1}{|p_i - v|^{2\alpha}}$ 。

因为 $l_v(x)$ 为仿射变换, 所以可以设 $l_v(x) = xM + T$ 。

将 (1) 式视为关于 T 的二次方程, 求导取零值得到 $T = q_* - p_*M$, 其中 $p_* = \frac{\sum_i w_i p_i}{\sum_i w_i}$, $q_* = \frac{\sum_i w_i q_i}{\sum_i w_i}$ 。

因此可以将 (1) 写成 $\min \sum_i w_i |\hat{p}_i M - \hat{q}_i|^2$ (2)

其中, $\hat{p}_i = p_i - p_*$ and $\hat{q}_i = q_i - q_*$ 。

而选取怎么样的 M 就是接下来三种方法的不同之处。

2.2 仿射变换

直接采用 (2) 的无约束解: $M = (\sum_i \hat{p}_i^T w_i \hat{p}_i)^{-1} \sum_j w_j \hat{p}_j^T \hat{q}_j$ 。

代入 $l_v(x)$, 有: $l_v(v) = (v - p_*)(\sum_i \hat{p}_i^T w_i \hat{p}_i)^{-1} \sum_j w_j \hat{p}_j^T \hat{q}_j + q_*$

2.3 相似变换

对 (2) 加上约束: $M^T M = \lambda I$ 。

设 $M = (M_1 \ M_2)$, 那么由约束条件, $M_1^T M_1 = M_2^T M_2 = \lambda$, $M_1^T M_2 = 0$ 。

那么 (2) 就等价于: $\min \sum_i w_i \left| \begin{pmatrix} \hat{p}_i \\ -\hat{p}_i^\perp \end{pmatrix} M_1 - \hat{q}_i^T \right|^2$

解出: $M = \frac{1}{\mu_s} \sum_i w_i \begin{pmatrix} \hat{p}_i \\ -\hat{p}_i^\perp \end{pmatrix} \begin{pmatrix} \hat{q}_i^T & -\hat{q}_i^{\perp T} \end{pmatrix}$

其中, $\mu_s = \sum_i w_i \hat{p}_i \hat{p}_i^T$ 。

那么, 有 $l_v(v) = \sum_i \hat{q}_i (\frac{1}{\mu_s} A_i) + q_*$ 。

其中, $A_i = w_i \begin{pmatrix} \hat{p}_i \\ -\hat{p}_i^\perp \end{pmatrix} \begin{pmatrix} v - p_* \\ -(v - p_*)^\perp \end{pmatrix}^T$ 。

2.4 刚性变换

刚性变换则是给一个更强的约束条件: $M^T M = I$ 。

按照论文中定理 2.1 的结论, 令 $f_r(v) = \sum_i \hat{q}_i A_i$, A_i 由相似变换中的式子给出。

则 $l_v(v) = |v - p_*| \frac{f_r(v)}{|f_r(v)|} + q_*$ 。

3 算法实现

该论文已经给了非常详细的实现公式, 只需要将公式翻译成代码即可, 图 1-3 即原理部分的代码。

实现时需要注意的几个细节:

1. matlab 中图像的坐标是以左上角为原点, y 轴向下, x 轴向右。
2. 读取的图像矩阵的行为 y 轴, 列为 x 轴。
3. 可以使用 `imwarp` 函数来变换图像, 该函数会自动帮助你进行插值操作。

```
function [im2] = Affine_Deformation(im, psrc, pdst)

[h, w, ~] = size(im);
vector_field = zeros(h,w,2);

for i = 1 : w
    for j = 1 : h
        v = [i j];
        weight = 1./((vecnorm((psrc - v)')).^3);
        p_star = weight * psrc / sum(weight);
        q_star = weight * pdst / sum(weight);
        psrc_hat = psrc - p_star;
        pdst_hat = pdst - q_star;
        M_1 = zeros(2,2);
        M_2 = zeros(2,2);
        for k = 1 : size(psrc, 1)
            M_1 = M_1 + weight(k) * psrc_hat(k,:)' * psrc_hat(k,:);
            M_2 = M_2 + weight(k) * psrc_hat(k,:)' * pdst_hat(k,:);
        end
        M = M_1 \ M_2;
        vector_field(j,i,:) = v - ((v - p_star) * M + q_star);
    end
end

im2 = imwarp(im,vector_field);
```

图 1: code of affine deformation

```

function [im2] = Similarity_Deformation(im,psrc,pdst)
[h, w, ~] = size(im);
vector_field = zeros(h,w,2);

for i = 1 : w
    for j = 1 : h
        v = [i j];
        weight = 1./(vecnorm((psrc - v)'));
        p = weight * psrc / sum(weight);
        q = weight * pdst / sum(weight);
        psrc_2 = psrc - p;
        pdst_2 = pdst - q;
        u = 0;
        M_2 = zeros(2,2);
        for k = 1 : size(psrc, 1)
            u = u + weight(k) * psrc_2(k,:) * psrc_2(k,:)';
            M_2 = M_2 + weight(k) * [psrc_2(k, :); psrc_2(k,2), -psrc_2(k,1)]...
                * [pdst_2(k,1), pdst_2(k,2);pdst_2(k,2),-pdst_2(k,1)];
        end
        M = M_2 / u;
        vector_field(j,i,:) = -(v - p) * M - q + v;
    end
end

im2 = imwarp(im,vector_field);

```

图 2: code of similarity deformation

```

function [im2] = Rigid_Deformation(im,psrc,pdst)
[h, w, ~] = size(im);
vector_field = zeros(h,w,2);

for i = 1 : w
    for j = 1 : h
        v = [i j];
        weight = 1./(vecnorm((psrc - v)'));
        p = weight * psrc / sum(weight);
        q = weight * pdst / sum(weight);
        psrc_2 = psrc - p;
        pdst_2 = pdst - q;
        v_vec = v - p;
        V = [v_vec;v_vec(2),-v_vec(1)];
        V = transpose(V);
        F = zeros(1,2);
        for k = 1 : size(psrc, 1)
            F = F + weight(k) * pdst_2(k,:) * [psrc_2(k, :); psrc_2(k,2), -psrc_2(k,1)] * V;
        end
        vector_field(j,i,:) = -norm(v-p) * F / norm(F) - q + v;
    end
end

im2 = imwarp(im,vector_field);

```

图 3: code of rigid deformation

关于 UI 界面的编写，主要采用了陈仁杰老师计算机图形学课程的 UI 框架，其使用说明如下：

1. 如果准备切换三个扭曲图像函数，请在回调函数 toolWorpCB 中进行切换。
2. 拖动点不要少于两个。
3. 点击红点进行选点拖动，点击蓝点运行结果。

4 实验结果

1. 姜饼人：



图 4: 仿射变换，相似变换，刚性变换

2. 蒙娜丽莎



图 5: 仿射变换，相似变换，刚性变换

从实验结果中可以发现，刚性变换对形态的保护最好，仿射变换最差。