Image Deformation Using Moving Least Squares

摘要：

我们提供了图像变形方法基于使用了不同类型的线性函数的移动最小二乘法，包括仿射，相似和刚性变换。这些变形十分逼真，给用户操作真实世界的物体的感觉。我们也允许用户使用特征点或特征线来指定变形，后者对控制图像中的曲线和轮廓是有用的。对于上述的每个技术，我们提供了封闭形式的解来得到可以实时进行的快的变形。

关键词：

图像变形，移动最小二乘，刚性变换

1.介绍

图像变形在动画领域有着大量的使用，比如渐变[Smythe 1990]和医疗成像 [Warren et al. 2003]。要执行这些变形，用户选择一些控制点来控制变形。这些控制点可以是点[Bookstein 1989]，线条[Beier and Neely 1992]，甚至是多边形网格[MacCracken and Joy 1996]。当用户在修改这些控制点的位置和方向时，图像应该以直观的方式变形。

我们可以将图像变形看成在变形函数f作用下由未变形图像到变形图像的过程。施加函数f到未变形图像中的每个点v以创建变形图像。考虑图像的变形前的控制点的集合p和变形后的控制点的集合q。函数f是一个有效的变形函数必须满足下列性质：

（1）插值性：在f作用下p点直接映射到q点。($f(p_i) = q_i$).

（2）光滑性：f应该产生光滑的变形。

（3）确定性：如果控制点p和q是同一点，f应该是确定的方程。(i.e; $q_i = p_i \Rightarrow f(v) = v$)

这些性质非常类似于散乱数据插值。前两个性质只是说明了函数f添加了散乱数据值以及光滑性。而最后一个性质有时代表了在近似情况下可以认为是线性精度的。它指出，如果数据是从线性函数中取样，则插值再现了线性函数。鉴于这些相似之处，许多变形方法借用了散乱数据插值的技术也就一点也不奇怪了。

先前的工作

图像变形之前的工作都集中于使用不同类型的控制点指定变形。基于网格技术如自由形式变形 [Sederberg and Parry 1986; Lee et al. 1995] 采用了二元三次样条函数来创建C2变形。通常这些方法需要对应于图像特征的控制点，这些控制点的设置对于用户来说十分麻烦。

贝尔等人[Beier and Neely 1992] 在这些基于网格的技术上更进一步，允许用户使用线段的集合来控制变形。此方法是基于Shepard的插值方法 [Shepard 1968]能够创建平滑的变形。然而，作者指出他们的方法回产生复杂的有时会”鬼“的变形，也就是在变形会产生不可取的折叠情况。Koba等人[Kobayashi and Ootsubo 2003] 后来把这种技术推广到了表面变形。

很少的变形方法在变形中对变形类型上的调查是可取的。一个值得注意的例外是基于薄板样条[Bookstein 1989]试图最小化变形弯曲量的工作。Bookstein呈现了一个使用最简单的控制点，一个点，使用薄板样条的径向基函数的变形算法。图2（左）显示了在图1的例子中使用薄板样条来创建变形的一个例子。变形结果显示和图1中的仿射方法十分的相似。在这两种情况下，测试形状都有局部非均匀的缩放和剪切，这在许多应用中都不希望看到。

我们的论文主要建立在Igarashi等人最近的一篇文章[Igarashi et al. 2005]即提出卡通类图像基于点的图像变形技术产生的变形”尽可能刚性“。这种变形有局部缩放和剪切的量最小化的性质。（”尽可能刚性“这个概念本身首先由Alexa[Alexa et al. 2000]提出）

为了善生尽可能刚性的变形，Igarashi等人三角化输入图像并且求解一个大小等同于三角顶点数目的线性方程组。相比之下，我们的方法在一个均匀网格下（详见第四节）解决对每个点来求解一个很小的（2*2）线性方程组。所以，我们求解小得多的方程组以创建网格数是Igarashi等人万倍的网格数的快速实时变形。Igarashi等人的报告说他们的方法在300顶点数左右1Gz的机器上十分耗时。如图2所示，Igarashi等人的方法在小数目顶点的变形中可能含有明显的不连续性。图7显示了相同情况下使用我们的技术的变形，变形的十分光滑。

贡献

在本文中，我们提出了一种基于线性移动最小二乘的图像变形方法。为了最大限度的减少局部缩放和剪切，我们限制了使用移动最小二乘的变形的类以接近刚体变换。通过使用MLS，我们不再需要三角化输入图像（如Igarashi等人所做的）也可得到光滑变形。

接下来，我们推导出两者的相似性和刚性MLS变换的封闭形式的公式。这些公式很简单，易于实现，并且提供实时变形。这个推导依赖于相似变换和刚性变换的令人惊讶和不为人知的关系，可以最大限度的减少常见的最小二乘问题。相对于Igarashi等人，我们的公式不要求使用通用线性结算器。

作为我们基于点的方法的自然延伸，我们扩展了我们的MLS变形方法，由点的集合到线段的集合并且再次对于变形方法给出了封闭形式的表达。

2.移动最小二乘变形

这里我们考虑构造基于用户控制点的集合的图像变形。令p是变形前的控制点的集合，q是p变形后的控制点的集合。我们构造使用最小二乘法的变形函数f来满足在介绍中列出的三个属性。给定图像中的一个点v，存在最佳仿射变换lv(x)令下式取得最小值：

$$\sum_i w_i |l_v(p_i) - q_i|^2 \qquad (1)$$

其中pi和qi是控制点中像素点的坐标，用行向量来表示，wi为权重，表达式为：

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

由于权重wi的取值随着点v的变化而变化，所以我们将这种方法称为移动最小二乘法。因此，对于每个点v，我们得到不同的变换函数lv(x)。

现在我们定义我们的变形函数f，令f(v)=lv(v)。当v接近pi时，wi趋近于无穷大，f(pi)=qi。此外，如果qi=pi，那么对于所有的x有lv(x)=x，因此f是一个恒等变换f(v)=v。最后，变形函数f有着处处光滑的性质。（除了当α ≤ 1，在控制点pi处时）

一般的，变形函数lv(x)可以分为线性变换项和平移变换项，分别用2*2的矩阵M和长度为2的行向量T表示，则有：

$$l_v(x) = xM + T \qquad (2)$$

将公式（2）代回公式（1），并求最小值，即对lv(x)的变量求导数且等于零，有：

$$T = q_* - p_* M$$

p*和q*分别是加权重心，表达式为：

$$\begin{aligned} p_* &= \frac{\sum_i w_i p_i}{\sum_i w_i} \\ q_* &= \frac{\sum_i w_i q_i}{\sum_i w_i} \end{aligned}$$

变形函数的一般形式为：

$$l_v(x) = (x - p_*)M + q_* \qquad (3)$$

在此基础上，公式（1）可以改写为：

$$\sum_i w_i |\hat{p}_i M - \hat{q}_i|^2 \qquad (4)$$

其中$\hat{p}_i = p_i - p_*$，$\hat{q}_i = q_i - q_*$。注意在MLS这个框架中，矩阵M可以视为一般化的仿射变换，包含了缩放、剪切、旋转等变换成分，对这些成分进行组合分析，可以获得仿射变换（非均匀缩放、剪切和平移）、相似变换（同比例缩放、旋转和平移）、刚性变换（非均匀缩放、旋转和平移）情况下的变形函数。接下来我们构建具有相似情况下的变形并展示如何找到对于刚性变换的移动最小二乘的封闭形式的解。

2.1仿射变换

采用经典普通方程的解决方法直接最小化等式（4）可以得到仿射变换。

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

虽然这个解决方案需要一个矩阵转置，但是这个矩阵恒定大小（2*2）非常快速的可以转置。随着对于矩阵M的封闭形式的表达，我们可以写出变形幻术fa(v)的简易表达。

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

对于原图中的每个点调用该变形函数来获得一个全新的，变形后的图像。

当用户通过操作点集q创建这些变形时，点集p是固定的。由于p在变形过程中不变，等式（5）中的一部分可以通过预先计算得到，从而可以得到快速的变形。特别地，我们可以重写等式（5）

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

其中Aj是一个标量，由下式给出

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

注意到给定点v，Aj中的任何值都可以通过预先计算得到一个简单的加权和。表（1）中给出了在本文中的例子的运行时间结果，表明了在我们的例子中这些变换每秒可以执行超过500次。

图（1）b显示了该仿射移动最小二乘变换应用于测试图像的情况。不幸的是，这个变形并不可取因为手臂和躯干上出现的伸展情况。这些情况的出现是因为仿射变换包括变形中的不均匀缩放和剪切。为了消除这些不良的变形，我们需要考虑限制线性函数lv(x)。特别的是，我们改变变形函数lv(x)的类型通过限制变换矩阵M由完全线性到相似变换和刚性变换。

2.2 相似变换

仿射变换包括非均匀缩放和剪切，而在现实中许多对象甚至不发生这些简单的变换。相似变换是仿射变换的一个特殊子集，它只包括平移，旋转和均匀缩放。

要修改我们的变形只能使用相似变换，我们限制了矩阵有以下性质：对于某些标量λ 有 $M^TM=\lambda^2 I$ 。如果M是以下形式的分块矩阵：

$$M = \begin{pmatrix} M_1 & M_2 \end{pmatrix}$$

其中M1,M2是长度为2的列向量，如果我们限制M是一个相似变换，那么有$M_1{}^T M_1 = M_2{}^T M_2 = \lambda^2$ 和$M_1{}^T M_2 = 0$ 。这个限制说明了$M_2 = M_1{}^{\perp}$ 其中⊥ 是2D空间向量的操作符，$(x, y)^{\perp} = (-y, x)$ 。虽然受到限制，但是等式（4）中的最小化问题对于M1依旧有效并且重述为对于列向量M1最小化下式：

$$\sum_i w_i \left| \begin{pmatrix} \hat{p}_i \\ -\hat{p}_i^{\perp} \end{pmatrix} M_1 - \hat{q}_i^T \right|^2.$$

这个二次方程有独立最小解，我们可以得到最佳的变换矩阵M：

$$M = \frac{1}{\mu_s}\sum_i w_i \begin{pmatrix} \hat{p}_i \\ -\hat{p}_i^{\perp} \end{pmatrix} \begin{pmatrix} \hat{q}_i^T & -\hat{q}_i^{\perp T} \end{pmatrix} \qquad (6)$$

其中：

$$\mu_s = \sum_i w_i \hat{p}_i \hat{p}_i^T.$$

和仿射变换相似，用户操作q来产生变形而p保持不动。通过以上观察，我们可以得到最有益于预计算的变形函数fs(v)的形式：

$$f_s(v) = \sum_i \hat{q}_i \left(\frac{1}{\mu_s} A_i\right) + q_*$$

其中μ_s和A_i只依赖于pi和v，可以预计算得到，Ai的表达式如下式：

$$A_i = w_i \begin{pmatrix} \hat{p}_i \\ -\hat{p}_i^{\perp} \end{pmatrix} \begin{pmatrix} v - p_* \\ -(v - p_*)^{\perp} \end{pmatrix}^T. \qquad (7)$$

如预期所示，相似MLS变换比仿射MLS变换保存了更多原始图像的角度（严格保持角度的变换被称为共形变换，已被广泛的研究 [Gu and Yau 2003]）。尽管近似（或精确）保存角度在许多情况下是一个值得拥有的性质，但是由此产生的局部缩放经常导致不可接受的变形。图（1）c表示施加相似移动最小二乘变换到我们的测试图像中的实例。这个结果看起来比（b）更加接近真实。但是，这个变换缩放了上臂的形状好像被拉伸了一般。为了消除这种缩放，我们考虑只使用刚性变换来构造变形。

2.3刚性变换

近期，一些工作 [Alexa et al. 2000; Igarashi et al. 2005]已经表明对于现实形状，变形应该尽可能刚性；也就是说，变形中应该不包括均匀缩放。传统的变形研究人员一直不愿意直接解决这个问题，因为非线性约束$M^T M = I$. 但是我们从迭代最近点集 [Horn 1987]这个问题可以得到封闭形式的解决方案。Horn表明了最佳的刚体变换从特征值和涉及pi和qi的协方差特征向量的角度来发现。我们表明，这些刚性变形通过下面的定理可以和2.2节中的相似变形联系起来。

定理2.1：设C为使相似函数取得最小值的矩阵：

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

其中$C=\lambda R$，R是旋转矩阵λ的一个标量，则旋转矩阵R取得最小值：

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

证明：见附录A。

这个定理对于任意尺寸都有效，并且他很容易在二维空间中使用。使用这个定理，我们发现刚性变换和等式（6）完全相同除了我们使用了不同的常数μr：

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

不同于相似变形函数fs(v)，我们对刚性变形函数fr(v)不能预先计算出很多信息。但是，变形过程仍然可以是非常高效的。令

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

Ai在等式（7）中定义，并可以预计算得到。所以我们得到fr(v)的表达式：

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

这种方法由于归一化操作比相似变形更慢；但是，如表1所示，这些变形仍旧非常快。

图（1）（d）展现了对于测试图像（a）使用刚性变形的结果。相对于其他的方法，这种变形是很现实的，几乎感觉就像是用户在操纵一个真实的对象。图（3）和（4）显示了刚性变换的其他例子。在蒙娜丽莎图中，我们的变形创造了更薄的面部轮廓并使她微笑。在马图中，我们伸展了马的双腿和脖子来创造了长颈鹿。由于使用了刚性变换，变形保持了刚性并在局部缩放中使马的身体和头部保持它们的相对形状。

3基于线段的变形

至此，我们已经考虑了用点集使用移动最小二乘控制来创建变形。在需要例如图像轮廓曲线精确控制的应用程序中，点可能不足以用于指定变形。一个解决方案是将用户精确控制的曲线转换为密集的点集，然后应用于基于点集的变形[Wolberg 1998]。这种方法的缺点是，计算时间和控制点的数量成正比以及大量创建的控制点会影响性能。

另外地，我们希望第2节中的移动最小二乘变形能够推广到平面任意曲线。首先，令pi(t)是控制曲线，qi(t)是pi(t)变形后对应的控制曲线。令 t ∈ [0, 1]，对于每个控制曲线pi(t)归纳等式（1）中的二次函数，得到

$$\sum_i \int_0^1 w_i(t)\,|p_i(t)M + T - q_i(t)|^2 \qquad (9)$$

其中$w_i(t)$

$$w_i(t) = \frac{|p_i'(t)|}{|p_i(t) - v|^{2\alpha}}$$

等式（9）仍然适用：

$$T = q_* - p_* M$$

其中p_*和 q_* 的表达式：

$$p_* = \frac{\sum_i \int_0^1 w_i(t)p_i(t)dt}{\sum_i \int_0^1 w_i(t)dt}$$
$$q_* = \frac{\sum_i \int_0^1 w_i(t)q_i(t)dt}{\sum_i \int_0^1 w_i(t)dt} \qquad (10)$$

因此，我们重写等式（9）

$$\sum_i \int_0^1 w_i(t) |\hat{p}_i(t)M - \hat{q}_i(t)|^2 \qquad (11)$$

其中：

$$\begin{aligned} \hat{p}_i(t) &= p_i(t) - p_* \\ \hat{q}_i(t) &= q_i(t) - q_*. \end{aligned}$$

至此，pi(t)和qi(t)已经可以是任意曲线了。然而，等式（11）中的积分难以评估为任意函数。相反，我们限制这些函数为线段，并从这些段的端点推导出变形函数封闭形式的解。类似第2节，我们首先考虑仿射变换，因为仿射变换较为简单的推导，然后推导到相似变换以创建封闭形式的解决方案，使用刚体变换的等效问题。

3.1仿射变换

$\hat{p}_i(t)$, $\hat{q}_i(t)$ 是线段，我们可以将它们以矩阵形式表示

$$\begin{aligned} \hat{p}_i(t) &= \begin{pmatrix} 1-t & t \end{pmatrix} \begin{pmatrix} \hat{a}_i \\ \hat{b}_i \end{pmatrix} \\ \hat{q}_i(t) &= \begin{pmatrix} 1-t & t \end{pmatrix} \begin{pmatrix} \hat{c}_i \\ \hat{d}_i \end{pmatrix} \end{aligned}$$

等式（11）可以被重写为

$$\sum_i \int_0^1 \left| \begin{pmatrix} 1-t & t \end{pmatrix} \left(\begin{pmatrix} \hat{a}_i \\ \hat{b}_i \end{pmatrix} M - \begin{pmatrix} \hat{c}_i \\ \hat{d}_i \end{pmatrix} \right) \right|^2 \qquad (12)$$

最小化得到：

$$M = \left(\sum_i \begin{pmatrix} \hat{a}_i \\ \hat{b}_i \end{pmatrix}^T W_i \begin{pmatrix} \hat{a}_i \\ \hat{b}_i \end{pmatrix} \right)^{-1} \sum_j \begin{pmatrix} \hat{a}_j \\ \hat{b}_j \end{pmatrix}^T W_j \begin{pmatrix} \hat{c}_j \\ \hat{d}_j \end{pmatrix}$$

其中Wi是权重矩阵：

$$W_i = \begin{pmatrix} \delta_i^{00} & \delta_i^{01} \\ \delta_i^{01} & \delta_i^{11} \end{pmatrix}$$

$$\begin{aligned} \delta_i^{00} &= \int_0^1 w_i(t)(1-t)^2 dt \\ \delta_i^{01} &= \int_0^1 w_i(t)(1-t)t dt \\ \delta_i^{11} &= \int_0^1 w_i(t)t^2 dt \end{aligned}$$

上述积分对于不同α的值有着不同形式的封闭形式解。在附录B，我们提供了α=2的封闭形式解，虽然其他的方案可以通过符号积分包的帮助下求解。请注意，这些积分也可以用于计算等式（10）中的p_*和q_*。

$$p_* = \frac{\sum a_i(\delta_i^{00}+\delta_i^{01})+b_i(\delta_i^{01}+\delta_i^{11})}{\sum \delta_i^{00}+2\delta_i^{01}+\delta_i^{11}}$$
$$q_* = \frac{\sum c_i(\delta_i^{00}+\delta_i^{01})+d_i(\delta_i^{01}+\delta_i^{11})}{\sum \delta_i^{00}+2\delta_i^{01}+\delta_i^{11}}$$

像之前一样，我们有变形函数fa(v)：

$$f_a(v) = \sum_j A_j \begin{pmatrix} \hat{c}_j \\ \hat{d}_j \end{pmatrix} + q_*$$

其中Aj是一个1*2的矩阵：

$$A_j = (v - p_*)\left(\sum_i \begin{pmatrix} \hat{a}_i \\ \hat{b}_i \end{pmatrix}^T W_i \begin{pmatrix} \hat{a}_i \\ \hat{b}_i \end{pmatrix}\right)^{-1} \begin{pmatrix} \hat{a}_j \\ \hat{b}_j \end{pmatrix}^T W_j.$$

变形过程中，当用户在操作qi(t)的两个端点时，pi(t)的两个短信是固定不动的。所以Aj对于ci,di独立，可以预计算得到。

图（5）显示了一个基于线段的变形例子，我们修改了比萨斜塔让他倾斜向相反的方向，并缩小了塔身。仿射MLS变换剪切到了塔侧，而不是转动，这种情况在现实中是不会出现的。要消除这种剪切效应，我们限制等式（11）中的矩阵是一个相似或者刚性变换。

3.2相似变换

对等式（12）中限制存在λ 有$M^T M = \lambda^2 I$。如2.2节中，矩阵M可以被写作

$$\sum_i \int_0^1 \left| \begin{pmatrix} 1-t & 0 & t & 0 \\ 0 & 1-t & 0 & t \end{pmatrix} \left(\begin{pmatrix} \hat{a}_i \\ -\hat{a}_i^{\perp} \\ \hat{b}_i \\ -\hat{b}_i^{\perp} \end{pmatrix} M_1 - \begin{pmatrix} \hat{c}_i^T \\ \hat{d}_i^T \end{pmatrix} \right) \right|^2$$

$$M = \frac{1}{\mu_s} \sum_j \begin{pmatrix} \hat{a}_j \\ -\hat{a}_j^{\perp} \\ \hat{b}_j \\ -\hat{b}_j^{\perp} \end{pmatrix}^T W_j \begin{pmatrix} \hat{c}_j^T & \hat{c}_j^{\perp T} \\ \hat{d}_j^T & \hat{d}_j^{\perp T} \end{pmatrix} \qquad (13)$$

其中Wj是权重矩阵：

$$W_j = \begin{pmatrix} \delta_j^{00} & 0 & \delta_j^{01} & 0 \\ 0 & \delta_j^{00} & 0 & \delta_j^{01} \\ \delta_j^{01} & 0 & \delta_j^{11} & 0 \\ 0 & \delta_j^{01} & 0 & \delta_j^{11} \end{pmatrix}$$

μs 是固定系数：

$$\mu_s = \sum_i \hat{a}_i \hat{a}_i^T \delta_i^{00} + 2\hat{a}_i \hat{b}_i^T \delta_i^{01} + \hat{b}_i \hat{b}_i^T \delta_i^{11}.$$

变形函数的形式和基于点的变形函数非常相似：

$$f_s(v) = \sum_j \begin{pmatrix} \hat{c}_j & \hat{d}_j \end{pmatrix} \left(\frac{1}{\mu_s} A_j\right) + q_*$$

其中Aj是一个4*2的矩阵：

$$A_j = W_j \begin{pmatrix} \hat{a}_j \\ -\hat{a}_j^\perp \\ \hat{b}_j \\ -\hat{b}_j^\perp \end{pmatrix} \begin{pmatrix} v - p_* \\ -(v - p_*)^\perp \end{pmatrix}^T \qquad (14)$$

图（5）展示了使用这种基于相似性的方法的变形。和仿射方法相反，塔表现出了旋转而不是剪切，保留了更加逼真的变形。相似变换含有均匀缩放，从塔的线段的收缩方式中显而易见。刚性变换将消除这种均匀缩放。

3.3刚性变换

使用3.2节的结论和定理2.1，我们直接得到了刚性变换的封闭形式的解。当然我们要选择一个不同的固定系数μr：

$$\mu_r = \left| \sum_j \begin{pmatrix} \hat{a}_j^T & -\hat{a}_j^{\perp T} & \hat{b}_j^T & -\hat{b}_j^{\perp T} \end{pmatrix} W_j \begin{pmatrix} \hat{c}_j^T \\ \hat{d}_j^T \end{pmatrix} \right|$$

变形函数和等式（8）的形式相同，其中：

$$\vec{f}_r(v) = \sum_j \begin{pmatrix} \hat{c}_j & \hat{d}_j \end{pmatrix} A_j$$

Aj从等式（14）中得到。

图（5）右展示了使用刚性变换的变形例子。在这种变形里，塔产生了旋转但是并没有如相似变换一样产生缩放，替代的是沿线段方向上的非均匀缩放的效果。

图（6）还展示了刚性变换技术（右）和贝尔等人的[Beier and Neely 1992] 的线段变形方法（左）的比较。贝尔等人的方法创建的线段产生了折叠和不切实际的拉动，刚性方法则不会有这些相同的问题。

4实践

为了实现这些变形，对于变形函数f(v)我们预先计算出尽可能多的信息。当我们应用变形函数到图像时，一般不对图像中的每个像素都做处理。我们近似的得到网格图像并应用变形函数到网格上的每个顶点。然后，我们使用双线性内插得到的四边形（见图（7））。

在实践中，这种近似技术产生的变形和对图像的每个像素施加变形这种昂贵的方法基本没哟区别。对于本文中的所有例子，图像大小约为500*500像素。为了计算变形，我们使用了100*100的网格。如果需要更密集的网格来得到更准确的变形，变形时间和网格点的数量是呈线性上升的关系。

表（1）展示了在3GHz的英特尔机器上使用各种变形方法的时间。每种变形使用了100*100大小的网格。刚性变换花费最长的时间由于变形函数中的开方操作，但仍旧非常的快。

5结论以及未来的工作

我们提供了使用点集或线段集作为控制点创建图像的光滑变形的方法。使用移动最小二乘法，我们创建了基于仿射、相似和刚性的变换并且每种技术都提供了封闭形式的解决方案。尽管刚性变换的最小平方的最小化导致了非线性的最小化，我们也展示了这些解决方案是如何使用相似变换得到封闭形式的解从而绕过了非线性最小化的计算。

在限制方面， 我们的方法和大多数其他的空间翘曲方法一样，受到折叠的影响。当f发生雅克比变换时这些情况就会发生。对于许多变形，这些折叠可能并不是非常明显；虽然极端变形是可定会造成这样的折叠情况的发生（见图（8））。对于一些变形，折叠是可以接受的，因为这些2D图像是为了表现3D对象。Igarashi等人的方法在获取图像明确的拓扑上有优势并且为呈现这些变形提供了一个简单的方法。由于缺乏拓扑，这种技术难以实现，尽管拓扑信息可以被添加到我们的方法中。

在其他应用中，折叠情况是不希望的，必须予以消除。Tiddeman等人提供了可行的用于解决折叠的通用方法[Tiddeman et al. 2001]。给定变形后，Tiddeman等人创建之后的变形使得两个变形的结果是非负的雅克比矩阵。由于我们的变形的公式非常简单，我们打算探索使用Tiddeman等人的方法构建使用雅克比的封闭形式的公式的可能性。

我们的变形技术也同时对图像所在的整个平面进行了变形而不考虑图像的形状。这种拓扑性的缺乏既是好处也是限制。我们的方法的一个优点是缺乏拓扑结构来创建简单的变形函数。其他的技术如Igarashi等人[Igarashi et al. 2005]依赖于指定的拓扑构建轮廓形状的边界并三角化。此拓扑信息可以通过分离部分图像，如图（4）中的马的腿在几何上合拢从而创建更好的变形。注意到我们的方法足够通用以适应取决于形状的不同的距离度量而不是简单的欧几里得距离作为我们的权重因子。我们打算在以后的工作中探讨这个问题。

最后，在以后我们想探讨将这些方法推广到3D对表面进行变形。这种一般化在动作捕获领域很有潜力，其中每一帧的动画的数据可以是空间中的点的形式。然而，2.2节中的相似变换不再是一个二次的最小化问题，而是一个特征向量的问题。我们正在研究一个方法以有效的计算解决这个最小化问题。