

与视点对应的视图插补*

吴福朝 于洪川 袁波 韦穗

(安徽大学人工智能研究所, 合肥 230039)

摘要 Image Morphing 是产生图象之间插补变换的一类被广泛使用的技术,所有的 Morphing 方法都是基于两幅图象之间的象素位置和颜色的内插。现行的 Morphing 技术不能保证所生成图象的真实性。引进视点 Morphing 技术,则能保持生成图象的真实性,但事先要知道源图象所对应的 3D 到 2D 的投影变换,即摄像机投影矩阵。用目前的技术,从图象获得摄像机矩阵是十分困难的,这极大地限制了视点 Morphing 技术的应用。本文提出的视图插补技术,使得所生成的插补图象对应于某个视点,且不需要知道源图象所对应的摄像机矩阵,因此计算简单。

关键词 图象形变 图象插补 Warp 函数

0 引言

图象形变(Image Morphing)是产生图象之间插补变换的一类被广泛使用的技术,所有的 Morphing 方法都是基于两幅图象之间的象素位置和颜色的内插。目前,用 Morphing 方法所得图象的质量和真实性没有判断标准。一个自然的问题是 Morphing 技术能否满足透视性质,即两个不同视点取得同一场景或物体的图象,通过 Morphing 技术获得的插补图象能否对应这个场景或物体在某个新视点下的图象。研究表明^[1,5]:现行的 Morphing 技术不满足透视性质。因此,文献[2]引进视点 Morphing 技术,这种技术能保持真实性,但事先要知道源图象所对应的 3D 到 2D 的投影变换即摄像机矩阵。用目前的技术,从图象获得摄像机投影矩阵是十分困难的,这极大地限制了视点 Morphing 技术的应用。本文提出的图象插补技术,在不知道源图象所对应的摄像机投影矩阵时,使生成的插补图象对应于某个视点,且计算简单。

1 图象形变

令 I_0, I_1 是同一场景 S 在两个不同视点下的图象,求中间图象 I_s 的过程,称为图象形变。此过程通常采用 Warp 技术,有以下 3 步:

(1) 建立对应关系

$$C_0: I_0 \rightarrow I_1, C_1: I_1 \rightarrow I_0$$

其中, $p_0 \in I_0, C_0(p_0) \in I_1$ 是同一场景点在 I_0 与 I_1 上的象点, $p_1 \in I_1, C_1(p_1) \in I_0$ 也有类似的性质;

(2) 构造 Warp 函数。通常使用线性插补来构造:

$$W(p_0, s) = sp_0 + (1-s)C_0(p_0)$$

$$W(p_1, s) = sp_1 + (1-s)C_1(p_1)$$

其中 $s \in [0, 1], p_0 = (x, y)^T \in I_0, p_1 = (x', y')^T \in I_1$, 显然

$W(I_0, s) = \{sp_0 + (1-s)C_0(p_0) | p_0 \in I_0\}$ (1) 是关于 s 的连续集值映射,且

$$W(I_0, 0) = C_0(I_0), W(I_0, 1) = I_0$$

所以 $W(I_0, s)$ 是 I_0 到 $C_0(I_0)$ 上的线性同伦映射。 $W(I_0, s)$ 与 I_0 和 $C_0(I_0)$ 具有同伦关系,是由 I_0 连续变换到 $C_0(I_0)$ 上的中间图象对应点集。

$W(I_1, s)$ 也可以作类似的解释。

(3) 灰度值的确定

令 $\bar{I}_0 = W(I_0, s), \bar{I}_1 = W(I_1, s)$, 并设 $p_0 \in I_0, p'_0 = C_0(p_0) \in I_1$ 的灰度值分别为 $r(p_0), r(p'_0)$ 。 \bar{I}_0 上的对应点 $W(p_0, s) = sp_0 + (1-s)C_0(p_0)$ 的灰度值定义为

$$r(W(p_0, s)) = sr(p_0) + (1-s)r(p'_0) \quad (2)$$

称 $I_{0s} = (\bar{I}_0, r(W(p_0, s)))$ 为 I_0, I_1 的中间 Warp 图象。

类似地可由 $W(p_1, s)$ 确定中间 Warp 图象 I_{1s} 。

* 本文研究得到国家自然科学基金和“863”计划资助

收稿日期:1998-04-02;收到修改稿日期:1998-10-09

2 建立 I_0 与 I_1 间的对应

在图象形变过程中较精确地建立 C_0 与 C_1 之间的对应是非常重要的。我们所使用的方法是:先确定 I_0 与 I_1 的 $N(\geq 4)$ 个匹配点,剩下的对应点,建立二维投影变换来自动实现。

2.1 二维投影变换

令 $u^0 = (x, y), u^1 = (x', y')$ 是同一 2D 场景点 X 分别在不同视点下视图 I_0, I_1 中的象,在象平面坐标下满足二维投影变换的关系式^[3]:

$$\begin{cases} x' = \frac{m_0x + m_1y + m_2}{m_6x + m_7y + 1} \\ y' = \frac{m_3x + m_4y + m_5}{m_6x + m_7y + 1} \end{cases} \quad (3)$$

求二维投影变换算法

由于我们已知道二维投影变换的形式,所以通过 2D 场景或物体两幅图象 I_0, I_1 之间的 $N(\geq 4)$ 对匹配点,可以确定 2D 投影变换。

(1) 求初值 m

令 $(x'_i, y'_i) \in I_0, (x_i, y_i) \in I_1$ 为一对匹配点, $i = 1, 2, \dots, N$ 。由每一对匹配点,根据(3)式可以得到 $Am = b$ 。其中: $m = (m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7)^T$ 。当 $N \geq 4$ 时,我们可以用最小二乘法求 m :

$$m = A^+ b$$

其中, A^+ 为 A 的伪逆阵。

(2) 求精 m

由于根据最小二乘法所求的 m 是下述意义

$$\min \|Am - b\|$$

下的解,因此将 $(x_i, y_i)^T$ 代入(3)式所得的 $(\bar{x}_i, \bar{y}_i)^T$ 与其匹配点 $(x'_i, y'_i)^T$ 仍有一定的误差。

下面我们给出求精 m 算法,使误差达到最小。令

$$\chi(m) = \sum_{i=1}^N \{ (x'_i - \bar{x}_i)^2 + (y'_i - \bar{y}_i)^2 \} = \sum_{i=1}^N d_i^2$$

其中, $\bar{x}_i = \frac{m_0x + m_1y + m_2}{m_6x + m_7y + 1}, \bar{y}_i = \frac{m_3x + m_4y + m_5}{m_6x + m_7y + 1}$ 。

显然求精 m , 就是求一具体的 m' , 使

$$\chi(m') = \min \chi(m) = \sum_{i=1}^N d_i^2(m')$$

这是非线性最小问题,可以利用 levenberg-marquardt 方法^[4]求解。

2.2 3D 场景两幅图象 I_0, I_1 之间对应的近似

对于 3D 场景或物体的两幅图象 I_0, I_1 , 我们可以用下面的步骤来近似它们之间的对应关系。

(1) 将 I_0, I_1 划分为不相重叠的区域 $\{p_0^0, p_1^0, \dots, p_m^0\}, \{p_0^1, p_1^1, \dots, p_m^1\}$, 使 p_i^0, p_i^1 对应于 3D 物体表面的同一区域 Σ_i , 且 Σ_i 近似于平面;

(2) 按前述 2.1 节中的方法计算 p_i^0, p_i^1 之间的 2D 投影变换 M_i , 由 M_i 近似 I_0, I_1 之间的对应关系。当 3D 物体表面由若干平面构成时,这个对应关系是精确的^[3];在一般情况下,该方法可以任意逼近精确对应关系。

3 与视点对应的 Warp 函数

3.1 线性 Warp 函数的缺点

利用通常的 Warp 函数,其严重缺点是所得图象 I_s 不是同一场景或物体在某视点下的透视象。我们以 2D 场景图象来说明这一点,令 $(x, y)^T \in I_0$, 则 $C_0(x, y)^T = \left(\frac{m_0x + m_1y + m_2}{m_6x + m_7y + 1}, \frac{m_3x + m_4y + m_5}{m_6x + m_7y + 1} \right)^T$ Warp 图象的对应点 $(x^s, y^s)^T \in W(I_0, s)$:

$$(x^s, y^s)^T = s(x, y)^T + (1 - s) \times$$

$$\left(\frac{m_0x + m_1y + m_2}{m_6x + m_7y + 1}, \frac{m_3x + m_4y + m_5}{m_6x + m_7y + 1} \right)^T$$

显然 $(x^s, y^s)^T$ 不是 $(x, y)^T$ 的分式线性函数,因为分子是 x, y 的二次函数,所以不满足 2D 投影关系,即 $(x^s, y^s)^T$ 不是 $(x, y)^T$ 对应的场景点在某个透视点下的象。

3.2 与视点对应的 Warp 函数

为了使图象是某个透视点下的象,必须对函数进行改进。事实上,只须将通常 Warp 函数改为齐次坐标即可。令

$u^0 = (x^0, y^0, w^0)^T \in I_0$ 与 $u^1 = (x^1, y^1, w^1)^T \in I_1$ 为一对对应点。定义 Warp 函数为

$$u^s = su^0 + (1 - s)u^1 \quad (4)$$

可以证明由此得到的图象是某个透视点下的图象。

(1) 2D 场景图象 Warp 函数的计算公式

将二维投影变换改写成齐次坐标的形式,我们有

$$\begin{pmatrix} x^1 \\ y^1 \\ w^1 \end{pmatrix} = \begin{pmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{pmatrix} \begin{pmatrix} x^0 \\ y^0 \\ w^0 \end{pmatrix}$$

其中 $u^0 = (x^0, y^0, w^0)^T$ 与 $u^1 = (x^1, y^1, w^1)^T$ 为一对对应点。

由 Warp 函数的定义

$$\begin{pmatrix} x^s \\ y^s \\ w^s \end{pmatrix} = \left[sI + (1-s) \begin{pmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{pmatrix} \right] \begin{pmatrix} x^0 \\ y^0 \\ w^0 \end{pmatrix}$$

$$= \begin{pmatrix} (s+(1-s)m_0)x^0 + (1-s)m_1y^0 + (1-s)m_2w^0 \\ (1-s)m_3x^0 + (s+(1-s)m_4)y^0 + (1-s)m_5w^0 \\ (1-s)m_6x^0 + (1-s)m_7y^0 + w^0 \end{pmatrix}$$

所以在直角坐标系下,有 Warp 函数计算公式:

$$x^s = \frac{(s+(1-s)m_0)x^0 + (1-s)m_1y^0 + (1-s)m_2w^0}{(1-s)m_6x^0 + (1-s)m_7y^0 + 1}$$

$$y^s = \frac{(1-s)m_3x^0 + (s+(1-s)m_4)y^0 + (1-s)m_5w^0}{(1-s)m_6x^0 + (1-s)m_7y^0 + 1}$$

这里 $(x^s, y^s)^T, (x^0, y^0)^T$ 是直角坐标。

(2) 3D 场景图像 Warp 函数的计算公式

令 $\{p_0^0, p_1^0, \dots, p_m^0\}, \{p_0^1, p_1^1, \dots, p_m^1\}$, 为满足 2.2 节中的条件 I_0, I_1 不相重叠区域的划分, p_i^0, p_i^1 之间

的 2D 投影变换矩阵为 $M_i = \begin{pmatrix} m_0^i & m_1^i & m_2^i \\ m_3^i & m_4^i & m_5^i \\ m_6^i & m_7^i & 1 \end{pmatrix}$, Warp

函数可按下式计算:

$$\forall (x^0, y^0)^T \in p_i^0, i=1, 2, \dots, m$$

$$x^s = \frac{(s+(1-s)m_0^i)x^0 + (1-s)m_1^iy^0 + (1-s)m_2^iw^0}{(1-s)m_6^ix^0 + (1-s)m_7^iy^0 + 1}$$

$$y^s = \frac{(1-s)m_3^ix^0 + (s+(1-s)m_4^i)y^0 + (1-s)m_5^iw^0}{(1-s)m_6^ix^0 + (1-s)m_7^iy^0 + 1}$$

4 实验结果

图 1 中上行带方框的图象是盒子在两个不同视点下的源图象,使用本文第 2 节的方法建立起它们之间的对应关系,下行的图象依次(从左向右)是在 $s=0.25, 0.5, 0.75$ 时所获得的插补图象,它们准确地反映了从开始视点到结束视点之间的视图变化。

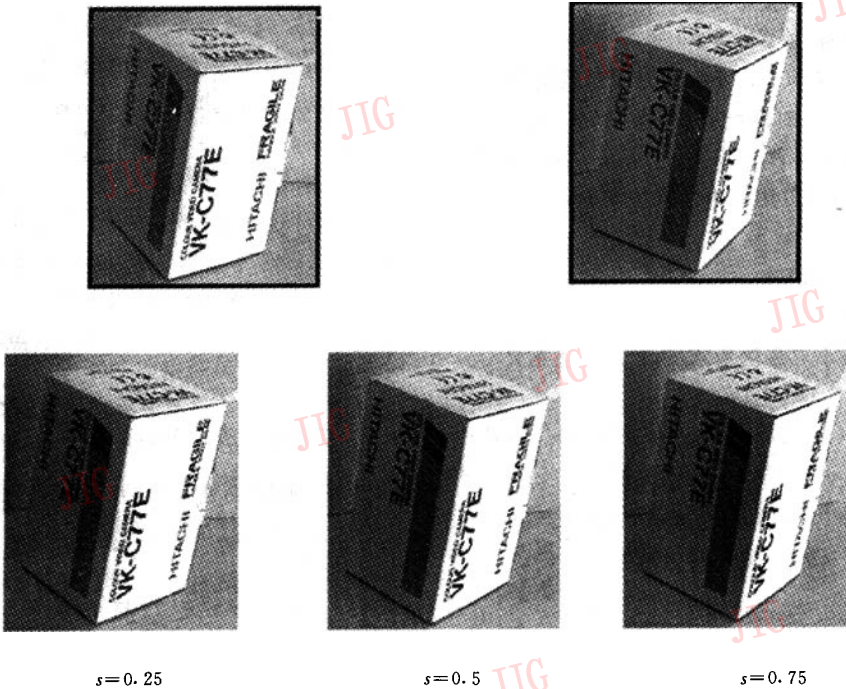


图 1 两个不同视点下的视图插补

参考文献

1 Debevec P E, Taylor J C, and Malik J. Modeling and rendering architecture from photographs: A hybrid geometry-and image-

based approach. In: Proc SIGGRAPH'96, 11~20.

2 Seitz S M, Dyer R. View Morphing. In: Proc SIGGRAPH'96, 1996, 21~30.

3 Irani M, Anandan P et al. Mosaic based representations of video sequences and their applications. In: Proc 5th Int Conf Computer

Vision, 1995, 605~611.

- 4 Press W H, Teukolsky S A *et al.*, Numerical Recipes in C: The Art of Scientific Computing (2nd Edition). Cambridge University Press, 1992.



吴福朝 安徽大学人工智能研究所教授。主要从事人工智能、计算机视觉、神经网络等方面的科研教学工作。现已在国内外学术刊物上发表四十多篇论文。

- 5 Werner T, Hersch R D, Hlavac V. Rendering real-world objects using view interpolation. In Proc 5th Int Conf Computer Vision, 1995, 957~962.



于洪川 中国科学院合肥智能研究所博士研究生。主要从事计算机视觉、虚拟现实技术等方面的研究工作。现已在国内外学术刊物和会议上发表十多篇论文。

袁波 博士,安徽大学人工智能研究所副教授。主要从事计算机视觉、计算机应用等方面的科研教学工作。现已在国内外学术刊物和会议上发表十多篇论文。



韦穗 教授,博士生导师,安徽大学副校长。主要从事图象处理、计算机视觉等方面的科研工作,先后主持多项国家自然科学基金和“863”计划项目,并多次获得中国科学院自然科学奖和国家教委科技进步奖。

View Interpolation Correspondence with Viewpoint

Wu Fuchao, Yu Hongchuan, Yuan Bo and Wei Sui

(Institute of Artificial Intelligence, Anhui University, Hefei 230039)

Abstract Image morphing method, which is based on the pixels position and color interpolation between two images, has been widely used in image interpolation. The former methods can't keep 3D shape, even though 2D shape. A new morphing method adopted viewpoint morphing technique. The 3D shape of the object can be kept if the projection transformation (called camera matrix) from 3D to 2D can be predetermined. It is difficult to gain camera matrix from the two images by the existing methods and techniques. This drawback confines to the applications of the view morphing. In this paper, a new morphing method is presented. We proved that the interpolatory view is correspondent to a viewpoint. The method needn't predetermine the camera matrix. And it is of simple computation and easy realization.

Keywords Image morphing, View interpolation, Warp function