文章编号: 1001- 2486(2007) 06- 0039- 05

一种视差域基于学习方法的图像合成算法*

吴琼玉, 周东翔, 刘云辉, 蔡宣平
(国防科技大学 电子科学与工程学院, 湖南 长沙 410073)

摘 要: 基于图像序列的图像合成往往需要定标和恢复场景三维结构, 为了避免这些复杂过程, 根据参考图像的视点离虚拟视点的远近关系, 将参考图像序列分为主、从参考图像, 并利用多极约束的原理和非线性校正, 将图像合成的最优化问题从深度域转化到视差域, 从而在不定标和不匹配的情况下, 直接从图像序列合成虚拟视图。实验结果证明算法是有效的, 并有一定的应用前景。

关键词: 弱定标; 校正; 图像合成
中图分类号: TP391. 4 文献标识码: A

View Synthesis Based on Learning in Disparity Field

WU Qiong-yu, ZHOU Dong-xiang, LIU Yun-hui, CAI Xuan-ping
(College of Electronic Science and Engineering, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: The process of calibration and 3D reconstruction of the scene are always required in view synthesis from image sequences. To avoid those complicated processes, the reference images are first classified into primary reference image and subordinate reference image according to the distance between the reference viewpoints and the novel viewpoint. Then the global optimization problem of view synthesis is transformed from the depth field to disparity field, using a process of nonlinear rectification and multi-epipolar technology. Finally, the novel view is synthesized from the image sequence without matching and calibration. Experimental results show that the method is effective and has potential in the future.

Key words: weak calibration; rectification; view synthesis

图像合成又称为视图合成, 是指通过两个或多个摄像机视点上所得到的同一场景的图像, 计算出其他各个摄像机视点或视线方向上要得到的该场景图像。过去的十年, 视图合成受到了广泛关注。关于视图合成的方法很多, 总结起来, 大致可以分为三大类: 一大类是恢复场景三维信息的方法[1]; 二大类是基于图像的视图合成方法[2-3]; 还有一种就是立体匹配的方法[4-7]。但是, 这些算法不是过于复杂, 就是生成的中间图像不具备照相真实感。庆幸的是, 在 2003 年的 ICCV 会议上, Fitzgibbon 等提出了一种全新的图像合成算法[8] ——基于图像先验知识的图像合成算法(又称为基于机器学习的图像合成方法), 并取得了比较好的效果。

Fitzgibbon 等[8] 模拟人的视觉反应过程, 将图像合成的问题归结为一个全局优化的问题, 在已知图像的基础上, 不需要任何人工交互, 就能得出最有可能的中间视图。但是, Fitzgibbon 等的方法有两大局限性: 一是要对摄像机进行自定标, 以获得参考视点的坐标和投影矩阵; 二是该方法要指定场景中每一点的深度变化范围。摄像机的定标是一个十分关键而又复杂的问题, 它需要专门用于定标的物体并且在特定的坐标系下才能完成。和相机定标的情况相比, 无相机定标的立体视觉具有更大的适应性。因此, 为了克服该方法的两大缺点, 我们提出了一种无需定标情况下的视差域的基于机器学习的图像合成算法。

* 收稿日期: 2007- 03- 28
基金项目: 国家自然科学基金资助项目(60334010; 60475029)
作者简介: 吴琼玉(1978—), 女, 博士生。

1 问题陈述

已知关于某一三维场景的 n 个二维图像 $\boldsymbol{I}_1$ 到 $\boldsymbol{I}_n$, $\boldsymbol{I}_i(x,y)$ 表示其中第 i 幅图像在图像坐标(x,y)处的像素值。一般情况,我们认为 $\boldsymbol{I}_i(x,y)$ 是一个三维的向量(RGB)。这些已知图像($\boldsymbol{I}_{1,2,\cdots,n}$)是相机在不同的位置拍摄得到的,称为参考图像,而待合成的图像称为虚拟图像。对虚拟图像中的任意一点,我们希望找到最有可能是三维场景中的同一点在其他图像中的投影点,这种点最大的可能性必须从已知的图像推导获得。

假设摄像机的投影矩阵为 $\boldsymbol{P}_{1,\cdots,n}$,投影矩阵 $\boldsymbol{P}$ 将三维空间的一点 $\boldsymbol{X}$(用齐次坐标表示)投影到二维点 $\boldsymbol{x}=(x,y,1)^{\mathrm{T}}$,即 $\boldsymbol{x}=\boldsymbol{PX}$。$\boldsymbol{I}_i(\boldsymbol{X})$ 表示三维点 $\boldsymbol{X}$ 映射到第 i 幅图像中的像素,那么

$$\boldsymbol{I}_i(\boldsymbol{X})=\boldsymbol{I}_i(\pi(\boldsymbol{P}_i\boldsymbol{X})),\quad \pi(x,y,w)=(x/w,y/w) \tag{1}$$

我们的任务就是在已知图像序列的情况下得到概率最大的虚拟视图 v。根据 Bayesian 理论,我们所要获得的 v 就是使后验概率 $p(v\mid \boldsymbol{I}_1\cdots\boldsymbol{I}_n)$ 最大的 v。根据 Bayesian 准则:

$$p(v\mid \boldsymbol{I}_1\cdots\boldsymbol{I}_n)=\frac{p(\boldsymbol{I}_1\cdots\boldsymbol{I}_n\mid v)p(v)}{p(\boldsymbol{I}_1\cdots\boldsymbol{I}_n)} \tag{2}$$

其中,$p(v)$ 是 v 的先验概率,$p(v\mid \boldsymbol{I}_1\cdots\boldsymbol{I}_n)$ 表示虚拟视点观测到的图像 v 的情况下图像序列 $\boldsymbol{I}_{1\cdots n}$ 发生的概率。因为我们要计算关于 v 的最大后验概率,所以不需要计算 $p(\boldsymbol{I}_1\cdots\boldsymbol{I}_n)$,只需要计算 $q(v)=p(\boldsymbol{I}_1\cdots\boldsymbol{I}_n\mid v)\cdot p(v)$ 的最大值。$q(v)$ 包括了两部分,图像连续性概率 $p(\boldsymbol{I}_1\cdots\boldsymbol{I}_n\mid v)$ 和先验概率 $p(v)$,我们把它又记做 $p_{\text{texture}}(v)$。下面分别讨论这两个部分。

2 图像连续性约束

从式(1)可知,要获得参考图像像素点的对应关系,投影矩阵 $\boldsymbol{P}$ 和对应点的三维坐标 $\boldsymbol{X}$ 必须已知。在投影矩阵已知的情况下,像素点的三维坐标可以通过它的深度 z 确定。然而,在很多情况下,摄像机标定都是非常困难和复杂的;另外,在摄像机获取场景的过程中,场景的深度往往是不断变化的,而且它的范围很难根据已有的图像直观获得。也就是说,一般情况下,很难获得参考图像的投影矩阵 $\boldsymbol{P}$ 和深度 z 的变换范围,而这两者正是图像连续性计算的前提条件[8]。但是如果将像素点的对应问题从深度域转化为视差域,就可以避免上述两个问题。

在未定标的情况下,极约束是图像间的唯一约束。设图像两两之间的极约束关系已知,并用基础矩阵 $\boldsymbol{F}_{ij}(1\leqslant i,j\leqslant n)$ 来表示。图 $\boldsymbol{I}_i$ 中的一点 V,图像 $\boldsymbol{I}_j$ 中和它对应的点在极线 $l=\boldsymbol{F}_{ij}\cdot V$ 上,沿极线 l 搜索,可以获得 V 在图像 $\boldsymbol{I}_j$ 中的对应点。极线是经过极点的一组线束,它的方向是不断变化的,为了使沿极线搜索对应点变得更加方便,可以通过图像校正使得两幅图像对应的极线投影到同一条水平或铅直线上,从而有效地提高搜索的速度。

2.1 图像校正

平面投影校正法是一种常用的校正算法。平面投影校正后的图像能够保持原有图像的特征,而且算法稳定,速度快,所以在图像合成中广泛运用。但是当视点包含的前向运动较大时,投影后的图像会变得非常大。由于图像序列包含的图像较多,很有可能包含前向运动,所以我们采用了一种对任意运动都有效的非线性校正算法[9]。

假设 $\boldsymbol{I}_{n1}$、$\boldsymbol{I}_{n2}$ 为待校正的图像,它们之间的极约束可以由基础矩阵 $\boldsymbol{F}$ 表示,由 $\boldsymbol{F}$ 可以算出两个极点坐标 $\boldsymbol{e}_{n1n2}(x,y)$ 和 $\boldsymbol{e}_{n2n1}(x,y)$。由于极线对应的模糊度由整条极线减少到了半条极线[9],所以只需考虑半径为正值的情况。对图像 $\boldsymbol{I}_{n1}$,以极点 $\boldsymbol{e}_{n2n1}(x,y)$ 为原点建立坐标系,如图 1 所示,x 轴和 y 轴分别和图像的一条边平行。校正的过程就是从平面坐标到极坐标的转换过程。图像中的一点(x,y),在校正图像中的坐标为(r,θ),

$$r=\sqrt{x^2+y^2} \tag{3}$$

$$\theta = \arctan(y/x) \tag{4}$$

其中, r 表示点到极点的距离, θ 表示点所在的角度。图像的校正是按极线的排列依次进行的, 其中每一条极线对应极坐标中的一个角度, 该角度对应的横坐标长度就是极线的长度。图 1 给出了校正过程的示意图。注意到从式(3)、(4)中, 容易计算出校正后图像的最大尺寸。设图像的两边长分别为 W、H, 校正后图像的高度最大值为原图像的边长 $2(W+H)$, 宽度最大值为原图像对角线的长度 $\sqrt{W^2+H^2}$。算法没有改变极线的长度, 也没有降低极线束采样的精度, 所以它保证了校正后的图像和原图像相比没有失真。

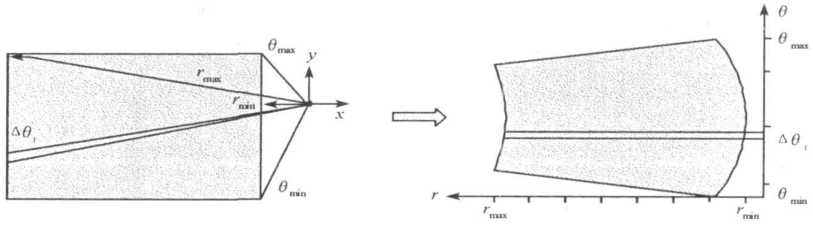


图 1 校正示意图

Fig. 1 The process of rectification

2.2 连续性约束的计算

设获取图像序列 $\boldsymbol{I}_{1\cdots n}$ 的视点为 $\boldsymbol{E}_{1\cdots n}$, 待合成图像的虚拟视点为 $\boldsymbol{E}_0$, 设 $\boldsymbol{E}_{n1}$ 和 $\boldsymbol{E}_{n2}(1\leqslant i,j\leqslant n)$ 为离虚拟视点最近的 2 个参考视点, 我们认为离虚拟视点最近的参考图像具有最大的参考作用, 所以选定 $\boldsymbol{I}_{n1}$、$\boldsymbol{I}_{n2}$ 作为主要的参考图像, 简称主参考图像, 其他的参考图像为次参考图像。用 2.1 节描述的算法校正图像对 $\boldsymbol{I}_{n1}$ 和 $\boldsymbol{I}_{n2}$, 可以得到校正后的主参考图像 $\boldsymbol{I}'_{n1}$、$\boldsymbol{I}'_{n2}$。

假设场景是漫反射不透明的物体, 对虚拟图像 ν 中的每一点 $\boldsymbol{V}(x,y)$, 由式(3)、(4)可以计算出它在虚拟校正图像中的坐标 $\boldsymbol{V}'(r_0,\theta_0)$。设图像 $\boldsymbol{I}'_{n1}$、$\boldsymbol{I}'_{n2}$ 之间的视差 l 可能的变化范围为 $l_{\min}<l<l_{\max}$, 对给定的视差 l, 可以计算出 $\boldsymbol{V}(x,y)$ 在图像 $\boldsymbol{I}'_{n1}$、$\boldsymbol{I}'_{n2}$ 中的对应点分别为 $\boldsymbol{V}'(r_1,\theta_1)$ 和 $\boldsymbol{V}'(r_2,\theta_2)$。由于图像 $\boldsymbol{I}'_{n1}$、$\boldsymbol{I}'_{n2}$ 的极线呈一一对应的关系, 所以 $\theta_2=\theta_1=\theta_0$。根据线性插值的原理: $r_1\cdot\alpha+(1-\alpha)\cdot r_2=r_0$, 其中, $\alpha(0\leqslant\alpha\leqslant1)$ 为插值系数, 可以计算出

$$\begin{cases} r_1 = r_0 + (1-\alpha)\cdot l \\ r_2 = r_0 - \alpha\cdot l \end{cases} \tag{5}$$

$\boldsymbol{V}'(r_1,\theta_1)$ 和 $\boldsymbol{V}'(r_2,\theta_2)$ 通过反校正, 得到 $\boldsymbol{V}(x,y)$ 在 $\boldsymbol{I}_{n1}$ 和 $\boldsymbol{I}_{n2}$ 中对应点分别为 $\boldsymbol{V}_1(x,y)$ 和 $\boldsymbol{V}_2(x,y)$。

为了获得最优的结果, 我们还要充分利用次参考图像中的信息。在未定标的条件下, 无法直接获取 $\boldsymbol{V}(x,y)$ 在其他图像 $\boldsymbol{I}_i(i\neq n_1,n_2)$ 中的对应位置。如果仅仅用两幅图像来进行匹配, 很难获得非常满意的结果, 所以本文采用多幅图像立体匹配的方法来降低匹配的不确定性。根据多极线的匹配原理, 将 $\boldsymbol{V}_1(x,y)$ 和 $\boldsymbol{V}_2(x,y)$ 投影到第三幅图像 $\boldsymbol{I}_i(i\neq n_1,n_2)$ 中, 来降低匹配的模糊度。这样不仅可以充分利用多幅图像的信息, 还可以提高算法的精度。如果已知三个弱定标摄像机和空间某点的两个像点, 总可以通过将相应的外极线取交来预测该点在第三幅图像中的位置, 所以我们用极线取交的方法计算 $\boldsymbol{V}(x,y)$ 在视差为 l 的情况下在任意一幅次参考图像 $\boldsymbol{I}_i(i\neq n_1,n_2)$ 中的对应点。

由极约束的原理可知, $\boldsymbol{V}(x,y)$ 在 $\boldsymbol{I}_i(i\neq n_1,n_2)$ 的极线 $l_1=\boldsymbol{F}_{n1,i}\cdot\boldsymbol{V}_1(x,y)$ 上, $\boldsymbol{V}(x,y)$ 在 $\boldsymbol{I}_i(i\neq n_1,n_2)$ 的极线 $l_2=\boldsymbol{F}_{n2,i}\cdot\boldsymbol{V}_2(x,y)$ 上, 那么 $\boldsymbol{V}(x,y)$ 在图像 $\boldsymbol{I}_i$ 上的对应点为 l_1、l_2 的交点, 即

$$C(i,l)=l_1\times l_2 \quad (i\neq n_1,n_2) \tag{6}$$

如果 l_1、l_2 平行或者接近平行, 那么在计算图像连续性的过程中, 我们不考虑 $C(i,l)$, 即认为次参考图像 $\boldsymbol{I}_i$ 不具备参考作用。用同样的计算方法, 可以得到 $\boldsymbol{V}(x,y)$ 在各个参考图像(包括主、次参考图像)上像素点的集合, 可以表示为 $C(:,l)=\{C(i,l)\}_{i=1}^{n}$。

对于点 $V(x,y)$,考虑所有可能的 l 值,可以得到

$$p(C \mid \boldsymbol{V}) = \int p(C \mid \boldsymbol{V}, l)\,\mathrm{d}l = \int p(C(:,l) \mid \boldsymbol{V}, l)\,\mathrm{d}l \tag{7}$$

输入图像像素 $\boldsymbol{V}(x,y)$ 的噪声可以用概率密度函数表达为 $\exp[-\beta\rho(t)]$,它以 $\boldsymbol{V}$ 为中心,β 是一个常数,表示分布的宽度。这样,概率密度可以写成

$$p(C(:,l) \mid \boldsymbol{V}, l) = \prod_{i=1}^{n} \exp-\beta\rho[\|\boldsymbol{V} - c(i,l)\|] \tag{8}$$

函数 ρ 是一个分布函数,在实验过程中取 $\rho(x) = x^2$。由于 β 的值很难获得,所以我们用如下的公式来近似表达:

$$p_{\text{photo}}(\boldsymbol{V}(x,y)) \approx \max_{l}? \; p(C(:,l) \mid \boldsymbol{V}, l) \tag{9}$$

在实现过程中,l 的值通过采样获得,典型数据是采样 $k(= l_{\max} - l_{\min})$ 个点。如果需要提高精度,可以提高采样密度。

3 纹理先验概率

$p_{\text{photo}}(\nu)$ 往往都是多态的[8](即往往有多个极值点),造成的原因可能是遮挡部分像素效应和图像形成模型的不完善等。为了消除这种多态性,我们采用文献[8]的方法来计算纹理概率。设 $\boldsymbol{V}(x,y)$ 为生成图像 ν 中的任意一点,它的先验概率可以写成:

$$p_{\text{texture}}(\boldsymbol{v}) = \prod_{x,y} p_{\text{texture}}(\boldsymbol{N}(x,y)) \tag{10}$$

$\boldsymbol{N}(x,y)$ 是 (x,y) 邻域内的点,本文采用 5×5 的邻域:

$$\boldsymbol{N}(x,y) = \{\boldsymbol{V}(x+i, y+j) \mid -2 \leqslant i,j \leqslant 2\} \tag{11}$$

4 图像连续性和纹理的结合

结合图像连续性概率和纹理概率:

$$q(\boldsymbol{v}) = \prod_{x,y} p_{\text{photo}}(\boldsymbol{V}(x,y))\, p_{\text{texture}}(\boldsymbol{N}(x,y)) \tag{12}$$

在实现过程中,对上式两边取对数后取负,求最小值。取对数后的能量公式:

$$E(\boldsymbol{v}) = \sum_{x,y} E_{\text{photo}}(\boldsymbol{V}(x,y)) + \sum_{x,y} E_{\text{texture}}(\boldsymbol{N}(x,y)) \tag{12}$$

图像合成的任务就转化成了在整个图像空间内将 E 最小化的问题。

5 实验及结果

图2给出了视点包含前向运动时的图像校正结果,其中,(a)、(b)为原图像,(c)、(d)为校正后的图像。比较两幅校正后的图像不难看出,校正后图像的极线变为水平方向,并且是一一对应的,这正好验证了校正算法的有效性。

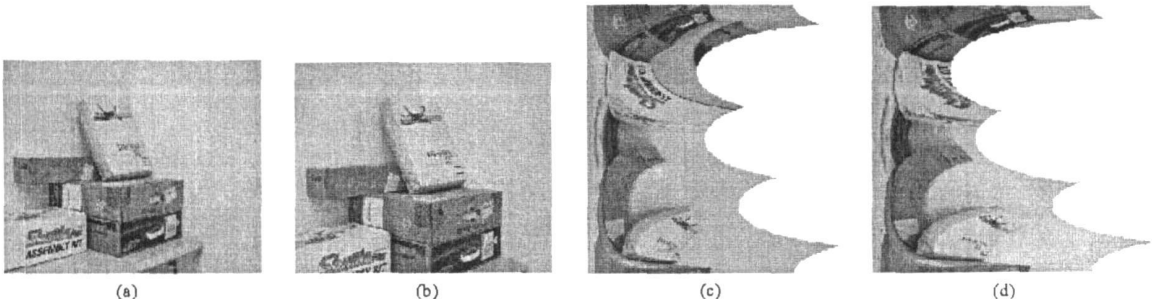


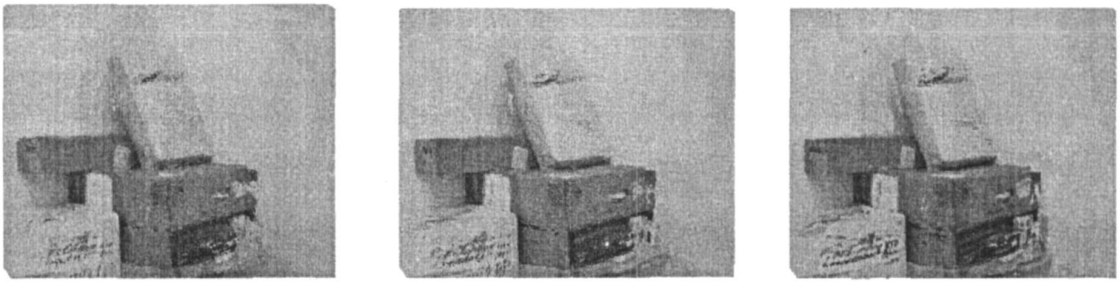
图2 视点包含前向运动校正结果:(a)、(b)为原始图像,(c)、(d)为校正后的图像

Fig. 2 Rectification results of image pair with forward motion:

(a), (b) are original image pair and (c), (d) are rectified results of (a) and (b)

我们采用一种计算量较低的求最大后验概率估计的算法, 即迭代条件模型(ICM) 算法来计算式(13) 的最优值。首先找到 $p((C: l) \mid \boldsymbol{V}, l)$ 关于 l 的局部最大值, 它可能有几个极值点, 取这几个极值点的最大值作为 ICM 的初始估计; 在这几个 l 值点再计算先验概率密度函数, 结合起来, 判断最优解。

我们用上面描述的方法做了一个插值试验。给出参考图像序列, 生成了一幅视点位置和方向介于两幅原始的参考图像之间的图像。我们采用的参考图像序列的长度为 11, 图 2(a) 、(b) 为原图像序列中的两幅参考图像, 图 3(a) 为插值后(未经纹理学习) 生成的图像, 图 3(b) 为插值图像经过一次纹理学习(即 ICM 算法循环一次) 后获得的图像, 图 3(c) 为插值图像经过 5 次学习后获得的图像。和图像 3(a) 相比, 图 3(b) 在很多细节方面得到了改善, 但是真实感还不够, 图 3(c) 的真实感强得多。以图中白色的纸袋为例, 在没有学习前, 仅考虑图像连续性所获得的合成图像(图 3(a)) 中可以看到, 纸袋上的字非常模糊; 经过一次学习后, 纸袋上的字(见图 3(b)) 稍微清晰一些, 但是变化不明显; 经过 5 次学习后, 纸袋上的字(见图 3(c)) 明显清晰了许多。



(a) 插值后生成的图像　(b) 纹理学习1次后的合成图　(c) 纹理学习5次后的合成图

图 3　图像合成实验结果

Fig. 3　Results of view synthesis

6　结 论

本文的算法还有待于进一步改进。基于学习的图像合成算法的计算量比较大, 本文只探讨了一种最优化的方法来解决全局优化的问题, 在以后的研究中, 我们将比较不同的优化方法, 希望能找出一种更快更好的计算方法。同时, 合成图像的质量和原图像相比还有一点差距, 实验分析表明, 造成这种差距的原因主要在于弱定标结果的不准确性, 所以将来我们还要找到一种更加准确的弱定标算法, 提高合成图像的质量, 使合成图像具备完全的照相真实感。

参 考 文 献:

[1] Pollefeys M, Koch R, Vergauwen M, et al. Hand-held Acquisition of 3D Models with a Video Camera [C]// Proc. 2nd International Conference on 3-D Digital Imaging and Modeling, 1999: 14- 23.

[2] Gortler S J, Grzeszczuk R, Szeliski R, et al. The Lumigraph [C]// SIGGRAPH, 1996: 43- 54.

[3] Levoy M, Hanrahan P. Light Field Rendering [C]// SIGGRAPH, 1996: 31- 42.

[4] Koch R. 3D Surface Reconstruction from Stereoscopic Image Sequences [C]// ICCV, 1995: 109- 114.

[5] Scharstein D, Szeliski R. A Taxonomy and Evaluation of Dense Two-frame Stereo Correspondence Algorithms [J]. Computer Vision, 2002, 47(1): 7- 42.

[6] Sun J, Li Y, Kang S B, et al. Symmetric Stereo Matching for Occlusion Handling [C]// ICCV, 2005: 384- 390.

[7] Veksler O. Stereo Correspondence by Dynamic Programming on a Tree [C]// ICCV, 2005: 384- 390.

[8] Fitzgibbon A, Wexler Y, Zisserman A. Image-based Rendering Using Image-based Priors [C]// ICCV, 2003.

[9] Pollefeys M, Koch R, Gool L V. A Simple and Efficient Rectification Method for General Motion [C]// Proc. International Conference on Computer Vision, 1999: 496- 501.