

基于线匹配的图像拼接

张显全 唐振军 卢江涛

(广西师范大学计算机科学系 桂林541004)

摘要 提出一种图像拼接算法,通过确定两幅图像的最佳拼接线,不需要手工指定特征点,可对图像进行快速、自动无缝拼接。实验结果证明了该算法的有效性。

关键词 图像拼接,线匹配,像素

Image Stitching Based on Line Matching

ZHANG Xian-Quan TANG Zhen-Jun LU Jiang-Tao

(Department of Computer Science, Guangxi Normal University, Guilin 541004)

Abstract An algorithm was proposed for the image stitching. It can establish the prime stitching line without special features given by man and stitch the images quickly and perfectly. The experimental results showed that it was effective.

Keywords Image stitching, Line matching, Pixel

1 前言

图像拼接在虚拟现实、地理信息系统(GIS)、视频压缩和医学图像处理等领域都有重要的应用。利用现有图像采集设备获取单幅图像的实际视区(field of view)范围通常是有限的,虽然利用图像处理工具通过剪切等功能可以实现图像的手动拼接,但这种方法精度不高且不实用,不能满足人们的需求,因此利用计算机实现自动拼接图像显得十分必要。通常图像拼接是通过搜索相邻两幅图像中相同的内容,再确定其相对位置,然后平滑连接就可实现自动拼接了。

目前,拼接通常采用的方法有两种:一种是基于两图像重叠部分在RGB或CMYK颜色系统中所对应的灰度级的相似性,自动寻找两幅图像的最佳匹配位置,这种方法常用的算法有以下几种:(1)基于块的匹配,取第一幅图像重叠部分的一块作为模板,在第二幅图像中搜索具有相同(或相似)值的块,从而确定重叠位置,进一步确定拼接点,当模板较大时算法的精度比较高,但模板越大计算量就越大。(2)比值匹配^[1]是在第一幅图像的重叠区域内,取相邻两列上的部分像素,用它们的比值作为模板,在另一幅图像中搜索最佳匹配,这种算法计算量较前一种算法有所减少,但由于所取的像素减少,所以精度也有所降低。(3)基于网格的快速匹配^[2],该算法是基于块匹配的,在搜索过程中,首先进行粗略匹配,每次移动一个步长,计算对应像素点RGB值的差的平方和,记录最小值的网格位置为目前最佳匹配位置;接下来以该位置为中心进行精确匹配,每次步长减半,再次搜索最小值,循环这个过程直到步长为0,最后就得到了最佳拼接位置;该算法在一定程度上减小了运算量,但在粗略匹配过程中,由于移动的步长较大,所以很有可能无法得到两幅图像完全匹配的最佳网格,而达不到精确拼接。另一种是基于特征的拼接方法^[3],首先从两幅图中选择一系列特征,然后根据相似性原则进行图像间的特征匹配,最后采用优化过程计算从一幅图像到另一幅图像的变换。特征点可以是图像的角点、封闭轮廓线的中心、纹理特

征点或其它突出点。如果在前一幅图像中以特征点A为中心的窗口与后一幅图像中以特征点B为中心的同样大小窗口的内容相似,那么,两个特征点被认为是匹配的,这种特征匹配方法称为模板匹配,其它类似的可以比较的信息包括:灰度分布、Fourier频谱、小波系数、直方图以及链码等。针对图像拼接过程中计算量较大的问题,本文提出一种基于线匹配的图像拼接算法,该算法在有效地降低计算量的同时能保证拼接的准确性。

2 图像拼接

一个能发光的物体称为有源物体,其颜色由它发出的光波决定,可使用RGB相加混色模型表示。计算机显示器是一个有源物体,因而计算机图像通常可用RGB相加混色模型来描述颜色,如位图是Windows系统中最常用的图像,其格式采用的就是RGB颜色模型,其他模型可转换成RGB模型,本文采用的颜色模型是RGB模型,在这个模型中,彩色图像中的每个像素值表示特定颜色的强度,一个像素用R、G、B三个分量表示,即:

$$C(\text{像素值}) = R(\text{红色的百分比}) + G(\text{绿色的百分比}) + B(\text{蓝色的百分比})$$

2.1 差异度的计算

设两幅图像P1和P2高度相同,为同一水平的两幅图像,即拍摄时不存在平移等视点变化。P1为M×L像素的图像,P2为N×L像素的图像,P1和P2为左右重叠关系(上下重叠关系可用同样的方法实现),P1在P2的左边;图像P1、P2可用如下的矩阵形式表示:

$$P1 = \left[\begin{array}{cccccc} * & * & * & \cdots & * \\ * & * & * & \cdots & * \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ * & * & * & \cdots & * \\ 1 & 4 & 4 & 2 & 4 & \cdots & 4 & 4 & 3 \end{array} \right] \left. \vphantom{\begin{array}{c} * \\ * \\ \cdots \\ * \\ 1 \end{array}} \right\} L$$

M

$$P2 = \left[\begin{array}{cccccc} * & * & * & \cdots & * \\ * & * & * & \cdots & * \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ * & * & * & \cdots & * \\ 1 & 4 & 4 & 4 & 2 & 4 & \cdots & 4 & 4 & 3 \end{array} \right] \left. \vphantom{\begin{array}{c} * \\ * \\ \cdots \\ * \\ 1 \end{array}} \right\} L$$

N

由于重叠区域的内容是相同的,其差异性主要是亮度和饱和度,因此它们在灰度级上是相似的,根据它们在灰度级上的差异度可以确定拼接线的位置。

由于两幅图像之间有一定的重叠区域,如果用整个重叠区域对图像进行匹配,运算量较大,同时又无法确定整个重叠区域的大小,为了在减少运算量的同时确保拼接点的准确性,从图像 $P2$ 的左边起,取第 i 列(其中 $k+1 \leq i \leq N-k-1$, 本文的实验取 $k=5, i=6$) 像素为模板线,图像 $P1$ 的第 j (其中 $k+1 \leq j \leq M-k-1$) 列与图像 $P2$ 第 i 列的差异度:

$$d_{ij} = \sum_{m=1}^L (C_m^i - C_m^j)^2$$

其中 C_m^i 为图像 $P2$ 第 i 列中的第 m 个像素的像素值, C_m^j 为图像 $P1$ 中第 j 列中的第 m 个像素的像素值, d_{ij} 表示图像 $P2$ 第 i 列与图像 $P1$ 第 j 列的差异度, L 为图像的高度。从差异度的计算公式可得 $d_{ij} \geq 0$; d_{ij} 值的大小反映出两列像素的差异程度,值小说明差异越小。

因为拼接线处两幅图像的像素比较接近,所以在拼接线处差异度的值较小,于是差异度最小两列像素所在的位置为图像的拼接线。然而由于伪匹配的存在,差异度最小处不一定是真实的拼接线所在,因此必须对拼接线进行确认。

2.2 拼接线的确认

为了使图像准确拼接,必须判断最小的差异度 d_{ij} 是否是真正的拼接位置。由图像的连续性可知,若是拼接位置,则在两幅图像拼接线附近区域中对应像素的差异度较小,因此可通过计算拼接线的某个区域中所有对应列的差异度来确定真实的拼接线;把拼接线的某个区域中所有对应列的差异度的和称为区域差异度。差异度为 d_{ij} 的拼接线的区域差异度计算如下:

分别以图像 $P2$ 第 i 列和图像 $P1$ 中第 j 列为中线左右各取 k 个像素宽度,以此构成两个对应的区域(区域宽度 $2k+1$),区域差异度是:

$$RegionD_{ij} = \sum_{i=1}^k \sum_{m=1}^L [(C_m^{i-1} - C_m^{i-1}')^2 + (C_m^{i+1} - C_m^{i+1}')^2] + d_{ij}$$

其中 $RegionD_{ij}$ 表示以差异度为 d_{ij} 的拼接线的区域差异度。通常伪匹配线出现的位置不会很多,所以只要对最小的 n 个差异度进行区域差异度的计算即可,最小区域差异度对应的列为图像的拼接线。

2.3 线匹配算法

(1) $i=k+1, y=1$;

(2) 计算图像 $P2$ 中第 i 列像素在图像 $P1$ 中的差异度 d_{ij} , 取最小的前 n 个差异度;

(3) 计算最小的前 n 个差异度的区域差异度 $RegionD_{ij}$, 设区域差异度的最小值为 $D(s_i, s_j)$;

(4) $y=y+1$, 如果 $y \leq T$ 则执行(5), 否则执行(6);

(5) $i=i+x$ 转入(2);

(6) 计算 $D(s_i, s_j)$ 中的最小值, 由最小的区域差异度确定图像的拼接线位置;

(7) 由(6)确定的拼接线对图像进行拼接。

算法中 x 为图像 $P2$ 中列像素的间隔, y 为拼接线的条

数, T 为拼接线总的条数, 本文的实验取 $x=3, T=10, n=10, k=5$ 。

2.4 拼接后处理

如果将两幅亮度、对比度有差异的图像拼接起来而不进行任何处理, 图像的颜色可能会出现明显的不协调。通常情况下, 相邻像素的像素值是相近的, 为了改善图像颜色的不协调可选择以拼接线为中心线的一个矩形小邻域对 R, G, B 颜色分量采取线性插值, 以达到颜色的平滑过渡。

以拼接线为中心线左右各取宽度为 w 的区域, 那么区域的总宽度为 $2w+1$ 。设拼接线左边第 w 个像素的颜色分量为 $c1$, 右边第 w 个像素的颜色分量为 $c2$, 则插值步长计算如下:

$$\text{delta} = (c2 - c1) / (2w + 1)$$

从左边第 w 个像素到右边第 w 个像素, 它们的位置分别记为 $1, 2, \dots, 2w+1$; 那么第 i 个位置的颜色分量的值为 $c1 + (i-1)\text{delta}$ ($1 \leq i \leq 2w$)。

3 实验结果

对亮度、对比度差异不大的两幅图像做本文的方法进行拼接实验, 拼接后的图像不存在颜色不协调现象, 效果很好(实验1); 对亮度、对比度差异较大的两幅图像用本文的方法进行拼接实验, 拼接的位置准确, 但拼接后的图像在拼接处出现了(尤其是图像亮度对比明显的上部分)比较明显的不协调(实验2), 必须对图像进行进一步处理, 使整个图像协调。

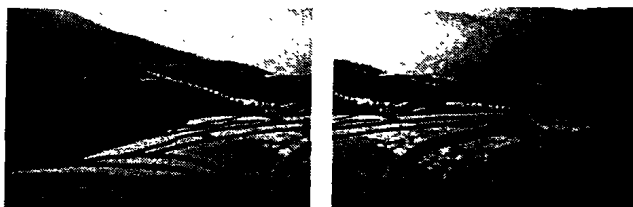
实验1 亮度、对比度差异不大的图像拼接



拼接后的图像:



实验2: 亮度、对比度差异明显的图像拼接



拼接后的图像:



总结 本文提出了一种基于线匹配的图像拼接算法, 用线差异度进行匹配, 有效地减少了计算量, 同时通过区域差异度确保拼接线的可靠性。实验结果表明该方法运算速度快、拼

接位置准确,算法有效可行。下一阶段我们将研究存在位移和旋转的图像拼接;同时寻找校正亮度和对比度的方法,以解决拼接后的图像颜色不协调问题。

参考文献

- 1 钟力,胡晓峰. 重叠图像拼接算法. 中国图像图形学报, 1998, 3(5): 367~370
- 2 李志刚. 边界重叠图像的一种快速拼接算法. 计算机工程, 2000, 26(5): 37~38
- 3 孟祥旭,杨承磊. 基于未校准照片构造全景图的新方法. 工程图学学报, 2002, 23(3): 100~114

- 4 王玉珍. 边缘重叠图像拼接中的特征块选取. 西北师范大学学报, 2002, 38(2): 38~40
- 5 李胜睿,等. 基于梯度滤波的彩色图像快速拼接与实现技术. 厦门大学学报, 2003, 42(1): 29~34
- 6 丁迎,洪继光. 图像拼接中伪匹配的判别和消解. 中国图像图形学报, 1999, 4(10): 886~890
- 7 李红,等. 向量小波在图像拼接中的应用. 华中理工大学学报, 1999, 27(11): 21~22
- 8 孙庆杰,吴恩华. 一种新的用于图像拼接的颜色融合方法. 工程图学学报, 1998, 19(4): 65~74

(上接第211页)

指标来评价不同的多尺度去噪方法产生的块效应,能与我们的主观视觉很好地吻合。从实验中也可以看出, HMOLT 及基于 HMOLT 的图像去噪算法不仅能有效地减轻重构图像的

块效应,而且在 MSE 方面和主观视觉方面也比 HMT 方法要好。最后,有必要指出,我们的块效应评价准则虽然基于图像去噪,但 HMOLT 本身是一种更为一般的图像或信号模型,可以应用于图像分割等其它场合。

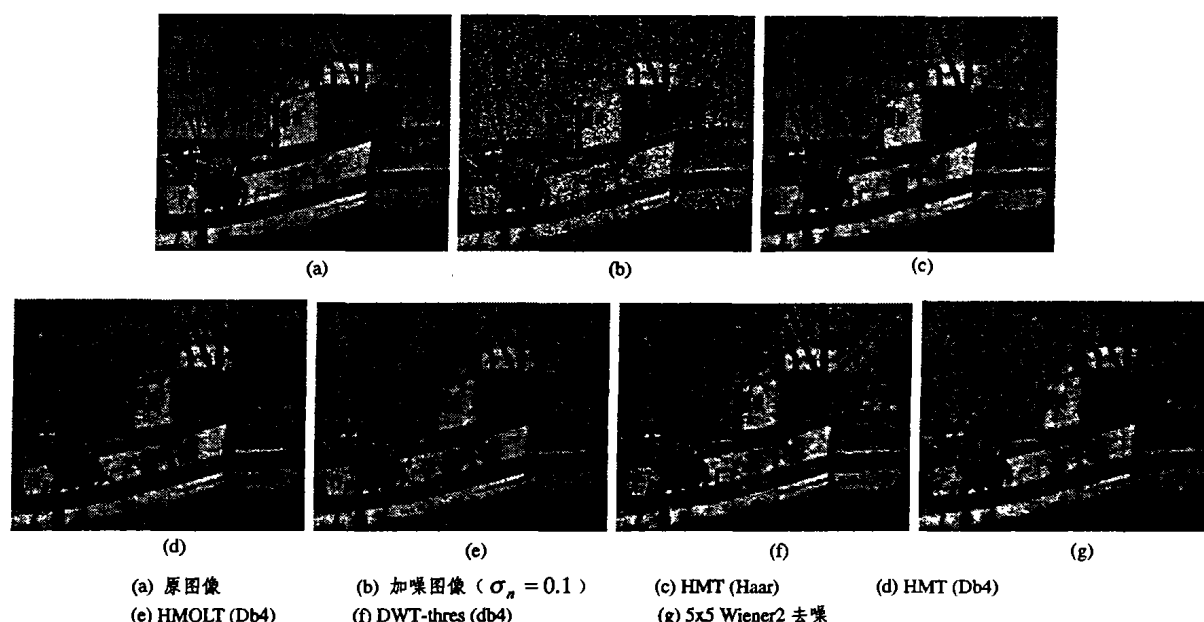


图5 几种不同方法对“Boats”图像的去噪结果,其中,(a)原图像,(b)加噪图像($\sigma_n=0.1$), (c)HMT(Haar), (d)HMT(D64), (e)HMOLT(D64), (f)DWT-thres(db4), (g)5×5 Wiener2去噪。

参考文献

- 1 Basseville M, et al. Modeling and estimation of multiresolution stochastic processes. IEEE Trans. Inform. Theory, 1992, 38: 766~784
- 2 Chou K C, Golden S A, Willsky A S. Multiresolution stochastic models, data, fusion, and wavelet transforms. Signal Process, 1993, 34: 257~282
- 3 Bouman C A, Shapiro M. A multiscale random field model for Bayesian image segmentation. IEEE Trans. Image Processing, 1994, 3: 162~177
- 4 Cheng H, Bouman C A. Multiscale Bayesian segmentation using a trainable context model. IEEE Trans. Image Processing, 2001, 10: 511~525
- 5 Crouse M S, Nowak R D, Baraniuk R G. Wavelet-Based Statistical Signal Processing Using Hidden Markov Models. IEEE Trans. Signal Processing, 1998, 46: 886~902
- 6 Romberg J K, Choi H, Baraniuk R G. Bayesian tree-structured image modeling using wavelet-domain hidden Markov models. IEEE Trans. Image Processing, 2001, 10: 1056~1068
- 7 Choi H, Baraniuk R G. Multiscale image segmentation using wavelet-domain hidden Markov models. IEEE Trans. Image Processing, 2001, 10: 1309~1321
- 8 Luetttgen M R, Karl W C, Willsky A S. Efficient multiscale regularization with applications to the computation of optical flow.

- IEEE Trans. Image Processing, 1994, 3: 41~64
- 9 Irving W, Fieguth P, Willsky A. An overlapping tree approach to multiscale stochastic modeling and estimation. IEEE Trans. Image Processing, 1997, 6: 1517~1529
- 10 Chipman H, Kolaczyk E, McCulloch R. Adaptive Bayesian wavelet shrinkage. J. Amer. Stat. Assoc., 1997, 92
- 11 Simoncel E P. Statistical models for images: Compression, restoration and synthesis. In: Proc. 31st Asilomar Conf. Pacific Grove, CA, 1997. 673~678
- 12 Mallat S, Zhong S. Characterization fo signals from multiscale edges. IEEE Trans. Pattern Anal. Machine Intell., 1992, 14: 710~732
- 13 Shapiro J. Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans. Signal Processing, 1993, 41: 3445~3462
- 14 Pesquet J, Krim H, Hamman E. Bayesian approach to best basis selection. In: IEEE Int. Conf. Acoust., Speech, Signal Process, ICASSP. Atlanta, GA, 1996. 2634~2637
- 15 Fan Guoliang, Xia Xianggen. Improved hidden Markov models in the wavelet-domain. IEEE Trans. Signal processing, 2001, 49: 115~120
- 16 McLachlan G, Peel D. Finite mixture models. New York: John Wiley & Sons, 2001
- 17 Donoho D L. De-Noising by Soft-Thresholding. IEEE Trans. on Information Theory, 1995, 41: 613~627