

网格提取的图像修复方法

沈玉峰^{1,2}, 汪继文^{1,2}, 林胜华^{1,2}, 苏守宝^{1,2}

SHEN Yu-feng^{1,2}, WANG Ji-wen^{1,2}, LIN Sheng-hua^{1,2}, SU Shou-bao^{1,2}

1.安徽大学 计算智能与信号处理教育部重点实验室, 合肥 230039

2.安徽大学 计算机科学技术学院, 合肥 230039

1.Ministry of Education Key Lab of IC & SP, Anhui University, Hefei 230039, China

2.School of Computer Science & Technology, Anhui University, Hefei 230039, China

E-mail: syfghz@163.com

SHEN Yu-feng, WANG Ji-wen, LIN Sheng-hua, et al. Approach for image inpainting extraction of fence. Computer Engineering and Applications, 2009, 45(20): 173-174.

Abstract: Image inpainting requires manual operations to specify the regions to be completed. This paper proposes a new method of signing the obstacles on a still image. The obstacles are regarded as the fence, and this method uses the obvious character that the fence consists of two sets of regular lines crossing each other in a space. Then the obstacles can be extracted by computation, and the method doesn't need manual operations. In order to ensure that the fence obtained can cover the obstacles on the image, we may dilate the lines of the fence. The experimental results show the method is effective.

Key words: obstacles; fence; Hough transform; image inpainting

摘要: 数字图像的修复, 一般都需要人们用手工来标记待修复的区域。提出了一种新的方法, 能够自动标记图像中的待修复区域——网状障碍物。首先将网状障碍物看作网格, 利用网格是由空间中有规律两组线交叉形成的这一显著特点, 然后通过计算来提取覆盖在图像上的网状障碍物, 这样就不需要人们用手工来标记。为了保证得到的网格线能够完全覆盖图像上的网状障碍物, 把提取的网格线进行适当的膨胀, 实验证明该方法是有效的, 且比较容易实现。

关键词: 网状障碍物; 网格; Hough 变换; 图像修复

DOI: 10.3778/j.issn.1002-8331.2009.20.051 文章编号: 1002-8331(2009)20-0173-02 文献标识码: A 中图分类号: TP391

1 引言

人们在进行图像修复的过程中, 经常遇到这样一些情况: 图像上有网格、栅栏、树枝等障碍物。特别是在一些设置了网状栅栏的地方, 如动物园、网球场和棒球场等, 拍摄的照片都有一些菱形的网格。对于这类的图像进行修复需要一些相关的检测技术。

在计算机视觉中已经研究了图像的填充过程, 给定图像的待修复区域后, 可以采用 PDE 方法^[1-3]、自相关性^[4]、水平集^[5]和变量方法^[6]来确定修复的计算方法。这些方法虽然应用很广, 但是它们只是手工地标识了一些图像上的待修复区域的临时信息^[7], 都没有考虑要自动提取待修复区域的问题。例如, 文献[8]中采用了 TV 模型^[9]的修复过程, 移除了人物面部的眼镜。文献[10]中则用此模型去除了照片上的水滴。本文利用网格变化的规律性提出了一种新的算法: 提取覆盖在图像上的网格, 然后修复图像的背景。

2 网格的提取方法

2.1 平行于图像平面的网格

通常来说, 平行于图像平面的网格是由两组平行线交叉形成, 这些网格大小可以由相对于水平线的角度 α 和平行线之间的间距 d 来定义, 如图 1 所示。这两个参数可以利用图像空间的自相关来估计, 图像 $u(x, y)$ 的自相关 u_a 表示为:

$$u_a = \mathbf{F}^{-1} \{ |\mathbf{F} \{u(x, y)\}|^2 \} \quad (\mathbf{F} \text{ 表示傅里叶变换}) \quad (1)$$

根据式(1)可以计算出图像 $u(x, y)$ 的自相关 u_a 如图 2 所示, 最亮的像素点代表了强相关性, 将其作为原点。一般来说, 距离原点越远相关性就会越低。除原点之外, 其周围有 4 个较亮点也很容易找到, 即峰值点, 将原点 o 到一个峰值点 p 的距离 op 记为 d , op 与 x 轴正方向所成的角记为 α , 如图 3 所示。

通过上述计算, 可以粗略地得到网格中两组平行线的间距

基金项目: 安徽省高校自然科学基金项目(the Natural Science Research Program of Anhui Colleges and Universities under Grant No.2006KJ028B); 安徽省高校自然科学基金重点项目(the Grand Natural Science Research Program of Anhui Colleges and Universities under Grant No.KJ2007A087)。

作者简介: 沈玉峰(1978-), 女, 硕士研究生, 研究方向: 图像处理; 汪继文(1957-), 男, 教授, 博士生导师, 主要研究方向: 数值模拟方法, 偏微分方程数值方法及其应用; 林胜华(1981-), 男, 硕士研究生, 研究方向: 图像处理; 苏守宝(1965-), 男, 博士研究生, 副教授, 主要研究方向: 群智能与本体计算, 数值模拟等。

收稿日期: 2008-04-21 **修回日期:** 2008-07-09

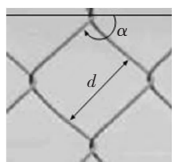


图1 间距和角度图



图2 自相关 u_α

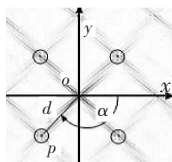


图3 与原点相邻的4个峰值

和角度, 利用 Hough 变换将这两个参数在 $\theta-\rho$ 参数空间中表示。这里, 规定 θ 的浮动范围为 $\alpha \pm \delta\theta$ 。这样, 在计算过程中图像上的平行线就被表示成沿着 ρ 的方向每隔一个间距就有一个极值点。

为了能准确地找到每条线的峰值点, 可以用 θ 的最大值 α 所对应的 ρ 来求峰值点的极值。然后, 在离 ρ 为 d 的区域, 即 $[\rho+d-\delta\rho, \rho+d+\delta\rho] \times [\alpha-\delta\theta, \alpha+\delta\theta]$ 内也可以找到极值。按照这种方式, 在区域 $[\rho+nd-\delta\rho, \rho+nd+\delta\rho] \times [\alpha-\delta\theta, \alpha+\delta\theta]$ 中都可以求出极值, 取 $\delta\theta=3, \delta\rho=5, n=\pm 1, \pm 2, \dots$ 。然后, 利用 Hough 变换将参数空间内找到的极值投影到图像上就形成了平行线。

2.2 非平行于图像平面的网格

平行于图像平面的网格提取方法与非平行于图像平面的网格提取方法完全不同。非平行于图像平面的网格也是由两组线交叉形成的: 一组投影到图像平面上形成了经过一个焦点的一组线, 以及另外一组投影到同一图像平面上形成了经过另外一个焦点的一组线。这样, 经过焦点就可以产生一条焦点线。

尽管现在已经有许多巧妙的算法^[1]可以找到焦点, 但是文中采用了 LMeds 算法来求焦点。这种方法要求用 GUI 工具来指出图像上网格线的角度范围。然后, 在 Hough 变换的参数空间中, 定义了这个范围内的一个区域。假设这个区域内有 n 条可供选择的线, 则从这 n 条线中任意选择 2 条, 可以得到一个交叉点为 (x_t, y_t) (t 表示第 t 次实验, $t=1, 2, \dots, s$)。从交叉点 (x_t, y_t) 到其余的 $n-2$ 条线的任意一条的距离记为 d_{ij} 。对于 s 次实验来说, 当 d_{ij} 的值最小时, 则此时交叉点 (x_t, y_t) 就是要求的焦点 $w=(x, y)$, 即

$$w=(x, y)=(x_{\min t}, y_{\min t}) \quad (2)$$

$$\min t = \arg \min_{1 \leq t \leq s} d_{ij}$$

用上述方法就可以得到焦点, 然后就可以计算出经过焦点的焦点线, 如图 4 所示。

然而, 上述算法选择的这 n 条线可能包含了不在网格上的线, 比如图像中背景本身的边界等等。为了能够准确找到网格上的线, 可以将得到的焦点线平移到 l' 的位置, 假设 k 为 l' 上的点, $m(k)$ 是经过 k 点和一个焦点 w 的一组线, 如图 5 所示。在 Hough 变换的参数空间中, 可以将 $m(k)$ 表示成正弦曲线 V , 即:

$$w \cdot (\cos\theta, \sin\theta) = \rho \quad (3)$$

将 $f(k)$ 记为沿曲线 V 的累积值。如果 k 是网格上的点, 那么 $m(k)$ 就表示网格上的线, 此时 $f(k)$ 的值比其他值要大。所以, $f(k)$ 的极值点对应着网格线上的点, 并且呈现周期性的变化。因此, 可以通过计算 $f(k)$ 的自相关性 f_n 来求相邻峰值点之间的间距 d' 。随后, 就可以用 d' 在 $f(k)$ 内得到其他的极值点。

首先, 将 $f(k)$ 的最大值传送给参数 ρ' ; 其次, 沿着曲线 V 在距离 ρ' 为 d' 的范围内得到下一个极值点。然而, 由于考虑到噪声的影响, 极值点可能不在曲线 V 上, 而在距离 ρ' 为 nd' ($n=\pm 1, \pm 2, \dots$) 的范围内的一个 $\delta\theta \times \delta\rho$ 的矩形上, 按照上述方法在曲线 V 上就可以得到所有的极值点。随后, 将得到的极值点

投影到图像上得到估计的网格线。

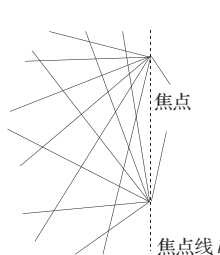


图4 焦点和焦点线

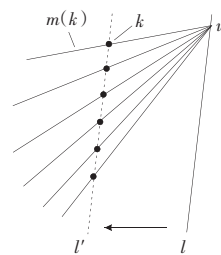


图5 经过焦点 w 的网格线

3 图像修复的步骤

为了保证所得到的网格线(包括平行于图像平面的网格和非平行于图像平面的网格)能够完全覆盖图像上需要修复的整个网格区域, 可以将得到的这些网格线进行膨胀。以网格上的点为圆心, 以 ε 为半径的圆作为膨胀算子(根据待修复图像的不同特点来选择 ε 的具体值), 将得到的网格线进行膨胀。

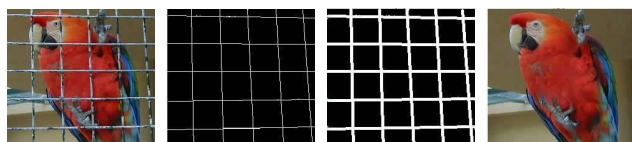
由膨胀后的网格线对要修复的图像做掩模, 得到掩模图像, 再利用一般图像修复算法(如基于 PDEs 的图像修复方法)来修复图像, 从而去除了图像中的网状障碍物, 具体步骤如下:

- (1) 判断待修复图像的网格是平行于图像平面的, 还是非平行于图像平面的;
- (2) 根据(1)的结果来选择网格的提取方法, 即文中 2.1 节或者 2.2 节所讲述的方法;
- (3) 将提取的网格根据具体特点选择适当的膨胀半径 ε , 对其进行膨胀;
- (4) 使用膨胀后的网格与待修复图像得到待修复图像的掩模图像;
- (5) 利用(1)中所讲的任意一种图像修复算法完成掩模图像的修复, 从而去除了待修复图像中的网格。

4 实验结果

在本次实验里, 用两类带有网状障碍物(平行与非平行的网状障碍物)的图像进行测试。在实验的过程中, 要求图像上的网格既不能弯曲也不能缺损。本次实验所用的运行环境是基于 WindowsXP 操作系统上的 Matlab7.0, 采用整体变分的图像修复模型^[3]。

图 6(a)是由平行于图像平面的网格所覆盖的图像, 即待修复的图像; 图 6(b)是用上述 2.1 节中的方法提取的网格; 图 6(c)是(b)的网格用 $\varepsilon=2$ 膨胀后的网格; 图 6(d)是用 BSCB 模型^[12]修复后的结果。



(a)待修复图像 (b)提取的网格 (c)膨胀后的网格 (d)修复后的图像

图6 平行于图像平面的网格图像修复

图 7(a)是由非平行于图像平面的网格所覆盖的图像, 即待修复的图像; 图 7(b)是用上述 2.2 节中的方法提取的网格; 图 7(c)是(b)的网格用 $\varepsilon=1$ 膨胀后的网格; 图 7(d)是用 TV 模型修复后的结果。