

图像主特征直线的检测算法

陈仁杰 刘利刚 董光昌

(浙江大学数学系计算机图像图形研究所, 杭州 310027)

摘要 直线特征的检测是计算机视觉与图像处理的基本任务之一。主特征直线是图像中最显著而且被人们最为关心的一些直线。本文提出了从图像中自动检测主特征直线的算法。首先利用图像分割算法分析得到图像的特征边缘, 然后用短线段逼近这些边缘像素, 接着用动态聚类算法对这些线段进行聚类, 并用直线拟合聚类中的线段得到所有可能存在的直线, 最后根据直线有效性度量找到其中有效性最高的直线就是主特征直线。最后本文给出了主特征直线在图像自动修复和自动美学构图中的应用实例。

关键词 主特征直线 直线检测 线段 聚类 拟合 有效性

中图分类号: TP391.41 文献标志码: A 文章编号: 1006-8961(2010)03-403-06

Detection of Principal Lines in Images

CHEN Renjie LIU Ligang DONG Guangchang

(Institute of Computer Graphics and Image Processing, Zhejiang University, Hangzhou 310027)

Abstract The detection of linear feature is one of the most fundamental tasks in computer vision and image processing. Principal lines are the most visually prominent lines in the images that people mostly care for. This paper presents a novel algorithm for automatic detection of the principal lines in digital images. The feature edge pixels are first detected from the image with an image segmentation technique. Then they are approximated with short line segments which are further clustered using some similarity metric. Each cluster of segments is fitted with a straight line in an iterative manner. The principal lines are determined according to the validity measurement. We demonstrate two practical applications including automatic image recovering and automatic photo composition to show the applicability of the principal lines in image processing.

Keywords principal line, line detection, segment, clustering, fitting, validity

0 引 言

直线检测是计算机视觉和模式识别中最重要的任务之一。直线是图像的重要特征, 直线特征提取方法的研究对图像的理解、模式识别有重要的意义。例如可以用于自动识别图像中直线特征明显的人造目标(如河流、道路、桥梁等)。

直线检测的方法有很多, Hough 变换是其中最经典的算法。Hough 变换^[1]的基本原理是利用点和

线之前的对偶性, 把图像空间中直线检测问题转换为参数空间中点的检测问题, 通过在参数空间进行简单的累加统计完成检测任务。该算法的优点是抗噪能力强, 能在低信噪比下检测出直线, 其缺点是由于是全搜索, 计算量和存储量都很大。多年来研究者致力于发展各种有效方法解决这些问题。其中自适应 Hough 变换^[2]、快速 Hough 变换^[3]、多分辨率 Hough 变换^[4-5]等利用多分辨率方法把图像分成一系列独立的子图像分别进行 Hough 变换从而实现有效加速。而概率 Hough 变换^[6]、随机 Hough 变

基金项目: 国家自然科学基金委员会与微软亚洲研究院联合项目 (60776799)

收稿日期: 2008-10-06 改回日期: 2009-01-12

第一作者简介: 陈仁杰 (1983—), 男, 浙江大学数学系计算机图像图形研究所博士研究生。主要研究领域为计算机图形学、数字几何处理。E-mail: renjie@zju.edu.cn

通讯作者: 刘利刚。E-mail: ligangli@zju.edu.cn

换^[7-8]等则在尽可能保证和标准 Hough 变换得到相同结果的前提下对原始边缘图像进行随机抽样从而减少计算量实现加速。文献 [9] 则通过改进 Hough 变换的投票算法, 对近似共线的点进行加权投票, 实现实时而精确的直线检测。另外一类直线检测的方法是模糊聚类算法。基于聚类的直线检测方法近年来发展很快。模糊聚类传统上用于检测紧的或具有球对称的形状, 通过改变聚类元素并选择合适的相似性度量, 这类方法可用于检测其他形状。在文献 [10], [11] 中, 对所有边界像素直接应用聚类过程得到狭长的聚类, 然后合并共线的相邻聚类并用直线拟合整个聚类就得到了目标直线。文献 [12] ~ [14] 则先从图像中分析得到边缘线段, 然后重复寻找共线的两条线段合并成一条长线段, 从而得到目标直线。

现有的直线检测算法试图检测图像中所有直线, 而不管直线的重要程度, 然而许多实际应用中可能只需要主特征直线, 即图像中最重要的一条或几条直线。为此, 提出了图像中主特征直线的检测算法, 并将此算法用于图像的自动修复和自动美学构图。

1 基于线段聚类的直线检测算法

和传统的直线检测算法一样, 先从图像中检测边缘像素, 然后从这些边缘像素中分析可能存在的直线。传统的边缘检测算法, 例如 Canny 算子^[15], 不能很好地找到所需要的边界点。虽然可以调整不同参数得到不同的检测结果, 但是如图 1 所示, 不同参数或者导致检测出过多与目标直线无关的边缘, 或者导致一些构成目标直线的重要边缘丢失。

注意到图像分割算法提供了很好的图像边缘, 如图 2 所示。因此以图像分割后不同块之间的边界像素为初始条件来进行直线检测, 采用基于图的分割算法^[16]。当然图像分割结果也和参数选取有关系, 采用比较保守的参数, 使得尽可能找到所有和目标直线相关的边缘像素。得到这些边缘像素后, 可以直接对这些边缘点进行聚类, 然而由于边缘像素很可能比较多, 所以点聚类过程会很慢, 不易收敛。注意到这些边界像素都是连续的, 并且有很多相邻像素位于同一直线, 因此在直线检测前, 先用短线段近似这些边缘像素, 近似结果如图 2(b) 所示, 从图中可以看出图像分割算法确实提供了更好更相关的边缘。得到短线段后, 采用线段聚类的方法从所有这些短线段中分析可能存在的直线。

由于图像中可能存在的直线条数是不确定的, 因此采用动态聚类方法, 并且不固定聚类个数, 而由聚类过程来最终确定。基本检测算法如下:

1) 随机选择一条线段, 作为第一个聚类

2) 对每条线段, 计算其到所有聚类中心的距离, 判断这些距离的最小值是否小于事先给定的阈值, 若是则将其归类至相应的聚类, 否则新建一个只包含它的聚类。

3) 对每个聚类, 重新计算聚类中心。若某个聚类不包含任何线段则删除此聚类。

4) 判断结果是否收敛, 若聚类个数或某个聚类内部成员有变化, 则返回步骤 1), 否则退出。

在点聚类算法中, 聚类中心一般取为聚类内所有元素的平均值, 因此数据元素和聚类中心均为低维或高维空间中的点, 故元素和聚类中心的距离可直接定义为空间中两点之间距离。然而在线段聚类中, 数据元素为线段, 而聚类中心为目标直线。故需要对线段到聚类中心的距离进行合理的定义, 并给出相应计算聚类中心的方法。

最简单的, 可以定义线段和直线的距离为线段中点到直线的距离, 或者为两者之间夹角。明显这两种定义都有缺陷, 并且两者数量上无法统一, 难以对两者进行适当组合得到较合理的定义。

定义线段和直线的距离为线段上所有点到直线距离之和。即给定线段 $AB: P(t) = (x(t), y(t)), 0 \leq t \leq 1$ 和直线 $L: x \sin \theta + y \cos \theta + c = Q$ 则

$$d(AB, L) = \int_0^1 d(p(t), L) dt \quad (1)$$

式中, $d(p(t), L)$ 表示点 $p(t)$ 到直线 L 的距离。例如, 在图 3 中, 红色线段和蓝色直线之间距离为灰色区域面积。设线段两端点 A, B 坐标分别为 $(x_1, y_1), (x_2, y_2)$, 则

$$x(t) = x_1 + (x_2 - x_1)t$$

$$y(t) = y_1 + (y_2 - y_1)t$$

故有

$$d(p(t), L) = |x(t) \sin \theta + y(t) \cos \theta + c| = \frac{|(x_2 - x_1)t \sin \theta + (y_2 - y_1)t \cos \theta + x_1 \sin \theta + y_1 \cos \theta + c|}{1} \quad (2)$$

经过简单计算可得

$$d(AB, L) = \begin{cases} \frac{|AB| (|d_1| + |d_2|)}{2} & d_1 \cdot d_2 \geq 0 \\ \frac{|AB| (d_1^2 + d_2^2)}{2(|d_1| + |d_2|)} & d_1 \cdot d_2 < 0 \end{cases}$$



图 1 对同一图像用不同参数进行 Canny 边缘检测^[14]的结果

Fig. 1 Apply the canny edge detector to the same image with different parameters

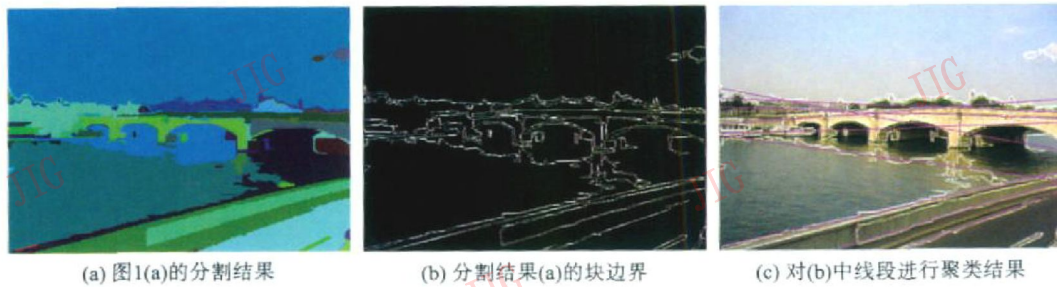


图 2 从图像中检测直线

Fig. 2 Detecting lines from image

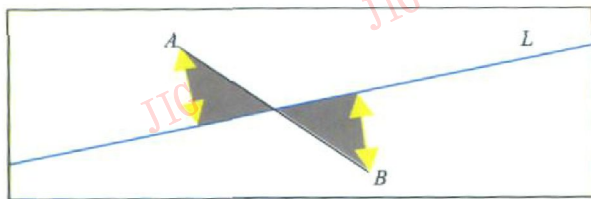


图 3 线段和直线之间距离

Fig. 3 The distance between a segment and a line

$$d_1 = x_1 \sin \theta + y_1 \cos \theta + c$$

$$d_2 = x_2 \sin \theta + y_2 \cos \theta + c$$

可以看出只有当线段 AB 和直线 L 完全重合时, 两者之前距离 $d(AB, L)$ 才为 0 因此上述定义是合理的。

在步骤 2) 中, 需要比较当前线段到所有聚类中心距离的最小值和给定阈值的大小, 然后再进行归类, 这是因为采用变化的聚类个数, 一开始聚类数为 1, 若不进行比较直接聚类将导致所有线段被聚为一类。这个阈值还会影响最终聚类个数, 阈值越大则聚类数越小。一般取此阈值为 10σ 这是因为图像中相邻直线之前平均距离不小于此数值。

在步骤 3) 中, 计算聚类中心也就是用直线拟合聚类内所有线段的过程。拟合直线应满足该聚类内所有线段到它的距离之和最小。因此, 和从点集拟合直线类似, 通过最小化以下目标函数实现对线段集合的拟合:

$$err(L) = \sum_i d(A_i B_i, L) = \sum_i \int d(p_i(t), L) dt$$

其中, $A_i B_i; p_i(t) = (x_i(t), y_i(t))$ 为聚类内第 i 条线段, 而 L 为目标拟合直线。然而根据式 (1) 和式 (2), 上述能量函数不能关于 θ 和 t 直接求导, 因而难于优化。因此修改目标函数为

$$err(L) = \sum_i \int d^2(p_i(t), L) dt \quad (3)$$

根据 $err(L)$ 最小化条件, 对式 (3) 关于 θ 和 t 求导可得:

$$0 = K_1 \sin^2 \theta + K_2 \cos^2 \theta + K_3 \sin \theta \cos \theta + c(K_4 \sin \theta + K_5 \cos \theta) \quad (4)$$

$$0 = L_1 \sin \theta - L_2 \cos \theta + 2L_3 \quad (5)$$

式中, K_1, \dots, K_5 和 L_1, L_2, L_3 为关于聚类内线段端点坐标的表达式, 与 θ 和 t 均无关, 故从上述两式容易计算出 θ 和 t 这样就完成了拟合过程。由于采用变化的聚类个数, 所以在迭代过程中有些聚类可能会不包含任何线段, 因此在步骤 3) 最后, 将所有的空聚类删除。

聚类过程迭代结束, 就得到了图像中的所有直线, 也就是所有聚类中心。例如在图 2 中, 图 2 (c) 中紫色直线就是对图 2 (b) 中线段进行聚类最后得到的聚类中心, 可以看出这些紫色直线绝大多数都是图像上真实存在的直线。

2 直线有效性度量

在很多实际应用中, 事实上并不需要检测出图像中所有直线。这些应用最关注的是主特征直线,

即图像中最重要的一条或几条直线,然而现有的直线检测算法未曾考虑此问题。为此,提出了直线的有效性度量,然后根据有效性确定主特征直线。从以下几个因素考虑检测到的直线的有效性。

直线附近区域梯度变化。设直线附近区域梯度的平均值为 $grad(L)$, 则 $grad(L)$ 越大有效度越高。例如图 4(a) 为图 1(a) 的梯度图, 可以看出图中橘色区域像素灰度值的平均值很大, 因此红色直线极有可能是主特征直线。

直线的有效拟合长度。有效拟合长度是指用于拟合当前直线的线段集合在直线上投影区间的长度, 即 $len(L) = \|\bigcup_i proj(A_i B_i, L)\|$ 其中 $proj(A_i B_i, L)$ 表示线段 $A_i B_i$ 在当前直线 L 上的投影。显然 $len(L)$ 越大, 则直线有效性越高。如图 4(b) 所示, 紫色线段为蓝色线段在绿色拟合直线上的投影, 显然直线 ①比 ②和 ③有效拟合长度更长, 因此其有效性更高。

直线的拟合误差 $err(L)$ 。本文的线段聚类算法并不直接要求聚类内线段共线, 因此有可能很多细碎杂乱的短线段会聚成一类, 但是实际上这些短线段并不属于同一条直线, 例如图 4(b) 中的直线 ②。由于这些短线段和拟合直线之间夹角较大, 由式 (3) 可知拟合误差一定比较大。因此 $err(L)$ 越小, 则有效性越高。

直线被图像画面所截长度 $len2(L)$ 。当直线特别接近图像边框时, 例如图 4(b) 中的直线 ③在图像画面内的长度特别短, 因此其有效性很低。

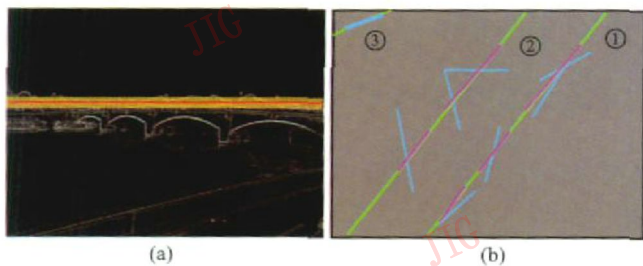


图 4 直线有效性度量

Fig 4 The validity measurement of lines

在确定任意直线的有效性时, 必须综合考虑以上因素。在本文实验中, 采用下式作为直线有效性度量标准

$$val(L) = a_1 \times grad(L) + a_2 \times len(L) - a_3 \times err(L) + a_4 \times len2(L)$$

其中, a_1, \dots, a_4 均为正数。对于不同类型图像, a_1, \dots, a_4 可能稍有不同。根据对大量图像的实验结果, 分别取 $a_1 = 2, a_2 = 2, a_3 = 2, a_4 = 1$ 。在计算总有

效性之前, 将 $grad(L), len(L), err(L)$ 和 $len2(L)$ 均归一化至 $[0, 1]$ 区间。根据所有直线的有效性 $val(L)$, 可以从中确定出主特征直线, 即 $val(L)$ 最大的几条。但是由于一般图像中主特征直线的条数并不确定, 所以给出一个经验阈值 val_0 (本文实验图像均取 2.5), 根据 $val(L)$ 是否大于 val_0 判断其是否为主特征直线。表 1 列出了图 2(c) 中总体有效性最高的四条直线的各有效因素得分。

表 1 图 2(c) 中直线有效性分数

Tab. 1 Validity measurement of lines in Fig. 2(c)

	$grad$	len	err	$len2$	val
直线 1	0.194	0.904	0.240	0.993	2.709
直线 2	0.307	0.771	0.261	1.000	2.634
直线 3	0.212	0.722	0.256	1.000	2.356
直线 4	0.256	0.573	0.342	1.000	1.974

3 实验结果

为了验证新算法的性能, 对大量图像进行了测试。结果如图 5(c), 图 6 和图 7 所示, 其中彩色直线为本文算法检测到的主特征直线, 并按不同的有效性进行着色, 其中红、绿、蓝 3 条直线分别对应有效性最高的 3 条, 另外图 6(b) 中白色像素为用短线段逼近图像分割边界的结果。从结果可以看出给出的有效性度量标准基本符合人眼视觉观察, 检测结果是接受的。

在图 5 中, 还与传统 Hough 变换进行了对比, 图 5(a) 和图 5(b) 中直线分别为用 Hough 变换^[1] 和概率 Hough^[6] 变换检测得到的直线。同样分别用红、绿、蓝 3 种颜色给 Hough 变换返回的前 3 条直线着色。注意到图 5(a) 中最重要的 3 条直线都位于第 2 条主特征直线附近, 而图 5(b) 中第 2 条直线完全不对应任何主特征直线。而本文算法正确检测出了两条主特征直线, 并且有效度量了每条直线的正确性。

从所有结果可以看出, 本文算法对直线斜率和连续性没有特别要求, 可以检测出任意角度的主特征直线。主特征直线的确定及其重要性和人的主观感觉及具体应用有关。例如图 7(a) 中位于桥塔上的蓝色和紫色两条直线的有效性和本文使用的阈值 val_0 非常接近, 事实上也可以看成非主特征直线。而图 7(b) 中旗杆所在直线由于有效拟合长度过短, 所以被判断为非主特征直线, 当然可以通过调整阈

值 val_0 使结果更符合实际需要。图 7(a)和 7(b)中地平线都不是最重要的主特征直线, 对于地平线敏

感的应用可以在直线有效性度量中增加直线斜率因素从而使结果更符合实际应用的需要。



图 5 图 1(a) 直线检测结果及比较

Fig. 5 The comparison between different line detecting approaches

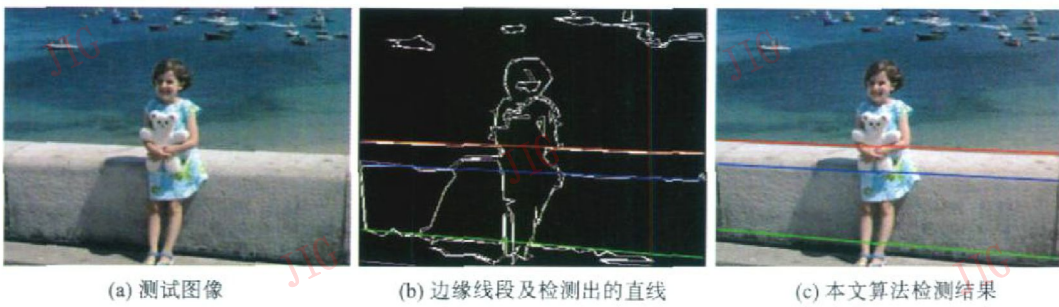


图 6 第 2 个例子

Fig. 6 Another result of principal line detection

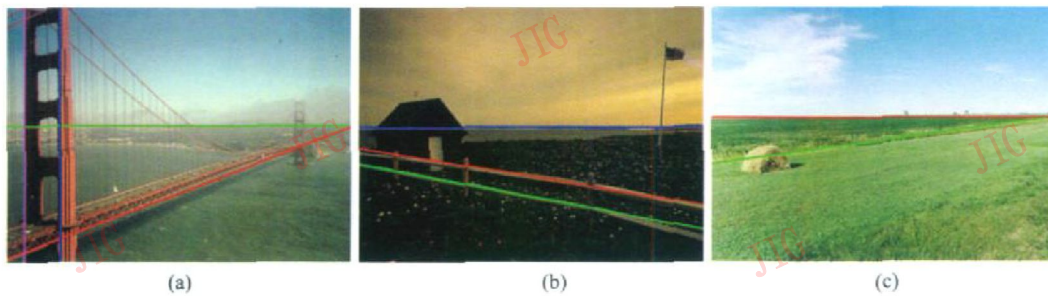


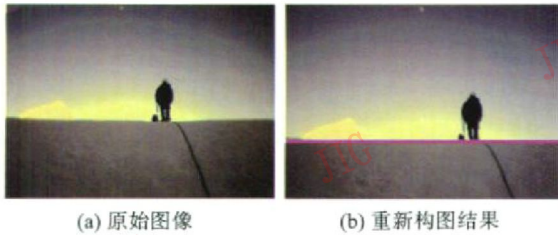
图 7 更多检测结果

Fig. 7 More results of principal line detection

4 主特征直线的一些应用

主特征直线还可用于实现数码照片的自动构图。直线是美学构图中最基本元素之一^[17], 图像中的主特征直线往往位于画面的三分线或黄金分割线附近。利用本文算法检测到主特征直线后, 可以对原图进行裁剪来改变其在图像中的相对位置, 从而使其更符合美学构图规则。例如在图 8(b)中, 紫色直线为本文算法检测到的主特征直线。为了便于比较, 将裁剪结果放大至和原图相同尺寸。

图像修复技术被广泛地用于重建旧照片和移除一些图片背景中不想要的物体。为了得到较好的修复结果, 需要指定关于待修复区域的结构特征。例如在文献 [18]中, 就是由用户手工指定一些直线或曲线特征。利用本文算法, 可以自动检测出主特征直线, 从而使整个修复过程无须人工交互自动完成。例如在图 9中, 图 9(b)为从原始图片图 9(a)中直接擦除南瓜得到的不完整图像, 先对图 9(b)进行主特征直线检测, 得到重要的直线结构, 如图 9(c)中紫色直线所示, 然后根据这些直线结构应用文献 [18]给出的修复算法, 得到最终修复结果图 9(d)。



(a) 原始图像

(b) 重新构图结果

图 8 自动构图应用

Fig. 8 Application of automatic image composition



(a) 原始图像

(b) 删除南瓜后的不完整图像

(c) 自动检测主特征直线 (d) 利用(c)中指定特征进行修复

图 9 图像修复应用

Fig. 9 Application of image recovering

5 结 论

直线检测一直是模式识别与计算机视觉的基本问题之一。给出了一种基于线段聚类的快速直线检测算法,并提出了直线有效性度量用于从中找出主特征直线。实验结果表明新算法检测结果准确度高、速度快、实用性强。今后工作主要是算法进一步加速,实现实时检测。目前由于采用基于图的分割算法,所以预处理分析比较慢。事实上分割结果不需要非常精确,故可以考虑对分割算法进行加速,或者采用其他更快的算法。另外关于直线有效性的检验,本文标准尚不适用于所有图像,必须根据具体的实际应用考虑更多因素。

参考文献 (References)

[1] Duda R O, Hart P E. Use of the Hough transformation to detect lines and curves in pictures [J]. *Communications of the ACM*, 1972, 15(1): 11-15.

- [2] Illingworth J, Kittler J. The adaptive Hough transform [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1987, 9(5): 690-698.
- [3] Li H W, Lavin M A, Master R J L. Fast Hough transform: A hierarchical approach [J]. *Computer Vision, Graphics, and Image Processing*, 1986, 36(2-3): 139-161.
- [4] Atiquzzaman M. Multiresolution Hough transform—an efficient method of detecting patterns in images [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992, 14(11): 1090-1095.
- [5] Yue Dongxue, Huang X insheng. Method of line detection based on improved multiresolution Hough transform [J]. *Computer Simulation*, 2006, 23(9): 76-79. [岳冬雪, 黄新生. 基于改进的多分辨率 Hough 变换的直线检测方法 [J]. *计算机仿真*, 2006, 23(9): 76-79.]
- [6] Kiriya N, Eldar Y, Bruckstein A M. A probabilistic Hough transform [J]. *Pattern Recognition*, 1991, 24(4): 303-316.
- [7] Kultanen P, Xu L, Oja E. A new curve detection method Randomized Hough transform [J]. *Pattern Recognition Letters*, 1990, 11(5): 331-338.
- [8] Li Quanlin, Zhou Yuan. Probability model of randomized Hough transform: limited data [J]. *Chinese Journal of Computers*, 2002, 25(3): 238-246. [李泉林, 周渊. 随机 Hough 变换的概率模型: 有限数据点 [J]. *计算机学报*, 2002, 25(3): 238-246.]
- [9] Leandro A, Fernandes F, Oliveira M M. Real-time line detection through an improved Hough transform voting scheme [J]. *Pattern Recognition*, 2008, 41(1): 299-314.
- [10] Krishnapuram R, Freg C P. Fitting an unknown number of lines and planes to image data through compatible cluster merging [J]. *Pattern Recognition*, 1992, 25(4): 385-400.
- [11] Dave R N. Use of the adaptive fuzzy clustering algorithm to detect lines in digital images [J]. *Intelligent Robots Computer Vision VIII*, 1989, 1192(2): 600-611.
- [12] Bami M, Cappellini V, Mecocci A, et al. Unsupervised detection of straight lines through possibilistic clustering [C] // *Proceedings of International Conference on Image Processing Lausanne Switzerland IEEE*, 1996, 963-966.
- [13] Scher A, Shneier M, Rosenfeld A. Clustering of colinear line segments [J]. *Pattern Recognition*, 1982, 15(2): 85-91.
- [14] Tsuda K, Minoh M, Ikeda K. Extracting straight lines by sequential fuzzy clustering [J]. *Pattern Recognition Letters*, 1996, 17(6): 643-649.
- [15] Canny J F. A computational approach to edge detection [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986, 8(6): 679-698.
- [16] Felzenszwalb P F, Huttenlocher D P. Efficient graph-based image segmentation [J]. *International Journal of Computer Vision*, 2004, 59(2): 167-181.
- [17] Banerjee S, Brian L E. Unsupervised automation of photographic composition rules in digital still cameras [J]. *Proceedings of SPIE*, 2004, 5301: 364-373.
- [18] Sun J, Yuan L, Jia J Y, et al. Image completion with structure propagation [J]. *ACM Transactions on Graphics*, 2005, 24(3): 861-868.