

一种快速在线图形识别与规整化方法^{*}

孙建勇 金翔宇 彭彬彬 孙正兴 刘文印

(南京大学计算机软件新技术国家重点实验室 南京210093)

(南京大学计算机科学与技术系 多媒体技术研究所 南京210093)

A Method of Fast On-Line Graphics Recognition and Regularization

SUN Jian-Yong JIN Xiang-Yu PENG Bin-Bin SUN Zheng-Xing LIU Wen-Yin

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)

(Department of Computer Science and Technology, Institute of Multi-media Technology, Nanjing University, Nanjing 210093)

Abstract A novel and fast shape classification and regularization algorithm for on-line sketchy graphics recognition is proposed. We divide the on-line graphics recognition process into four stages: preprocessing, shape classification, shape fitting, and regularization. Attraction Force Model is employed to progressively combine the vertices on the input sketchy stroke and reduce the total number of vertices before the type of shape can be determined. After that, the shape is fitted and gradually rectified to a regular one, thus the regularized shape fits the user intended one precisely. Experimental results show that this algorithm can yield good recognition precision (averagely above 90%) and fine regularization effect but with fast speed. Consequently, it is especially suitable to computational critical environment such as PDAs, which solely depends on a pen-based user interface.

Keywords On-line graphics recognition, Shape classification, Shape fitting, Shape regularization, Sketch-based user interface

1. 引言

因特网的迅速发展以及个人电脑的广泛使用促成了数量惊人的电子文档的出现。相比纸质资料,数字媒体内容更为丰富,形式更为多样化。随着计算机的介入,传统的设计方式发生了革命性的变革。设计工作不再是单纯的人为任务,而是一个人机不断交互的过程。因此,需要有一种新的技术和工具,能将传统设计方法与计算机辅助设计的优点有机地结合起来。但是,目前的许多文档设计工具,诸如 Microsoft 的 office, PhotoDraw, Visio 以及大部分的 CAD 系统,摒弃了传统的用笔和纸进行设计的方法,要求用户用鼠标在大量的菜单和按钮中选择预定义的标准图形对象。这种方法看似快捷方便,实际仍存在着一些缺陷:

1. 输入不方便:用户发现画自己需要的图形时,往往要多次点击鼠标进行选择。特别当有许多预定义的图形(或对象)时,用户很难记住到哪儿选择自己需要的图形。例如:微软的 Visio 中有10000个预定义组合图形供用户选择,复杂的界面让用户感到难以适从。

2. 输入不自然:有些应用,例如方案设计,需要用户随时随地地快速记录其设计思路,而不是拘泥于某一细节。这与传统的以最终设计为目标的应用是截然不同的。在这些系统中,选择预定义标准对象的过程往往会打断用户的思路,从而阻碍用户平滑、流畅地体现其设计思想。

3. 不适合小屏幕掌上设备:对于 PDA 等仅有笔输入界面的小屏幕掌上设备来说,这种输入方式很不实际。因为,大量的菜单和按钮会使得原本拥挤的屏幕更加紧张,从而挤压用户的使用区域。

现在越来越多的情况要求用户能随时记录下脑中的想

法,这就要求一种新的工具能提供一个更方便自然的用户界面,帮助用户在小屏幕设备上完成这个工作。我们认为:最方便自然的方式就是提供基于草图的手绘界面,让用户用笔在界面上输入草图,就像用笔和纸一样。早在1963年, Sutherland 就设计出交互草图设计系统。Sketchpad^[1]用光笔直接在屏幕上输入图表,尽管受到当时机器性能、识别能力等的限制,效果不是很理想,却提供了一个有广阔前景的研究领域。今天,掌上设备逐渐盛行,也为这种草图界面的研究提供了很好的应用背景。

2. 在线图形识别

手绘图形往往很不精确,用户画的草图与他本来的意图往往差别很大。用户自己理解不会有太大困难,但其它用户却会误解用户的意图。因此,手绘图形人机界面的关键问题是图形的识别,即预测用户的输入意图并把初始的草图规整到标准的文档以便其它用户的理解。由于视觉误差的存在,这个过程是实现是很困难的,往往预测的结果与用户的意图大相径庭。这就需要和用户不断地交互,通过一定的学习来改进效果,我们称这种不断交互的过程为在线图形识别。

相比于其它模式识别问题,在线图形识别有其自身的特点:

1. 与在线字符识别比较:作为光笔输入的主要应用,在线字符输入已相当普遍。首先,图形对象比字符更为复杂;其次,图形对象是笔划数目和笔划顺序无关的,用户可以按照自己的偏好输入,这给识别带来困难。最后,需要对笔划进行分割和重新合并,根据一些特殊信息确定它属于哪个图形对象,这使得图形识别更加复杂。

2. 与离线图形识别比较:对于诸如 OCR(Optical Charac-

^{*} 本文得到国家自然科学基金项目(编号:69903006)资助。孙建勇 研究方向为机器视觉,在线图形识别。金翔宇 硕士生,研究方向为智能化人机界面。彭彬彬 研究方向为机器学习,数据挖掘。孙正兴 副教授,博士后,研究方向为:多媒体辅助工程、智能信息处理。

ter Recognition 光学字符识别)的离线识别,光栅矢量化是最为困难的一步,而对在线识别则不需要这一过程。离线情况下图形与背景联系紧密,将图形从繁冗的背景环境中分离异常苦难,而在线图形识别可以利用用户输入对象的时序相关性加以分割;其次,离线识别对笔划的规则性要求较高而在线识别对输入的约束少了很多。其输入的图形往往具有很大歧义,如下图1离线图形识别的样本取自非常规整的工程图纸,因而其图形识别相对容易。最后,离线样本点是等距采样而在线样本点是等时采样,因而可以根据笔划移动速度的变化信息^[8]来进行笔划分割和图形识别。



图1 在线识别中由于用户输入的随意性引起的歧义。灰线为原始图形,红线为最终结果

基本图形的识别与规整化是在线图形识别的基本问题^[9],基本图形是图表以及其它复杂图形对象的基本组成元素。通常是一些封闭的凸多边形(例如三角形,四边形,五边形,六边形),椭圆或者线/弧段。而对其的识别以及规整化问题可以看成是如何从封闭的笔划中得到用户希望的图形。我们认为:这个问题的解决可以分为三个关键阶段,即:

1. 图形分类:根据原始的笔划序列识别出所属的基本图形类,如三角形,四边形,椭圆等。
2. 图形拟合:对上阶段得到的基本图形进行拟合,估计出图形的具体参数,使得拟合后的图形与输入图形最为相似。
3. 图形规整:对拟合的图形进行调整,使之与用户的输入意图最为吻合。

在图形识别分类问题上,国内外研究者都投入了大量的精力。纵观他们的工作,可以分为以下主要四类:离线方法,基于过滤器的方法,能量最小化方法,以及机器学习算法。

1. 离线方法:Revanka 等人^[4]提出了基于决策树的方法。他们将手绘图形分割成几个部分,通过一些规则建立决策树,每个叶节点输出一种形状。由于针对离线图形,因此制定的规则都很简单。但对于在线图形来说,这种方法并不适合。受到输入的随意性以及视觉偏差的影响,不可能将图形分割成标准的笔划,而且简单的规则不能区分复杂的图形。

2. 基于过滤器的方法:Takayuki^[2]等人提出用过滤器进行分类。但是,除了圆和直线外,其它图形必须与 X 轴或者 Y 轴平行输入。这个要求对于在线输入过于苛刻。Fonseca, Jorge 以及他们的研究小组^[3]对这种方法进行了改进,增加了一些新的过滤器。他们首先计算图形的一些全局特征,如闭包^[5],最大内接平行四边形^[6]以及外接矩形^[7]等,在此基础上将过滤器组合起来进行识别。识别过程中运用模糊逻辑输出的是一个相似图形序列。这对在线图形识别来说是必要的,但是,这种方法无法区分彼此很接近的图形,如五边形和六边形,此外,扩展形较低。

3. 能量最小化方法:Liu 等^[9]提出一种基于模板的猜试方法进行图形的分类。首先假设原始图形为三角形,矩形或者椭圆。然后计算每种猜测图形和规则化后的输入图形之间的差别。具体是将模板缩放和旋转多次,取所有情况的距离平均值作为每次猜测的代价。最后取代价最小者作为最终的分类结果。但是,代价最小的情况只是对输入笔划的最相似逼近,却未必是用户希望输入的图形。而且,算法的总体复杂度较

高,不适合诸如 PDA 的掌上设备。

4. 机器学习算法:神经网络 NN,SVM^[9]。优点:识别率高,达到95%以上的精度;缺点:速度慢,内存消耗大,实现复杂,不适合 PDA 环境。

相比于图形识别,图形的拟合和规整方面的相关研究较少,通常被认为是图形识别问题的一部分。Revanka^[4]在他的工作中对识别出的图形细分,如三角形有等腰(边)三角形,直角三角形和一般三角形等。但是,正如前面提到的,这种方法适用于离线情况。而在线情况中,图形的拟合及规整是与时序有关的,每次校正的失误会向后累积最终导致整个图形的变形。因此,图形的拟合以及规整在基本图形识别中处于重要的地位。而在目前大部分的识别方法中却往往被忽视。

本文针对在线图形的识别,拟合以及规整问题提出了一个有效的解决方法。输入的笔划经过预处理,图形识别,图形拟合和图形规整等过程,最终得到用户希望的图形。具体的流程如图2。

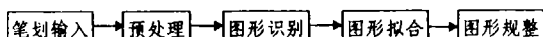


图2 基本图形的识别流程

3 图形识别与拟合

3.1 预处理

由于输入条件的限制和输入习惯的差异,用户输入的图形不可避免地受到噪音的影响。一般来讲,噪音包括笔划冗余点,曲线闭合误差。预处理就是要消除这些噪音的影响,便于以后的图形的识别以及拟合。预处理主要包括冗余点去除,聚点消除,端点校正和闭包计算。

3.1.1 冗余点消除 受采用频率和用户输入习惯的影响,采集图形中的节点数往往很多。过多的采样点会降低后面图形识别的性能。因此,在不改变原有图像外形轮廓的前提下,减少节点数量很有必要。实际上,输入图形是由大量折线段组成。有些线段与由其相邻结点构成的线段很相似,删除这些线段中的节点将不会影响最终的识别效果。如果某个节点到由其相邻节点构成的折线段的距离小于设定阈值 ϵ ,则该点是冗余点,可以去除。我们采用了由 Sklansky 和 Gonzalez 提出的算法实现该思想。算法复杂度为 $O(n)$,其中 n 为节点数。 ϵ 越小,保留点数目越多,曲线相对更平滑。在本文中,我们设定 $\epsilon=1$ 实验效果如图3。

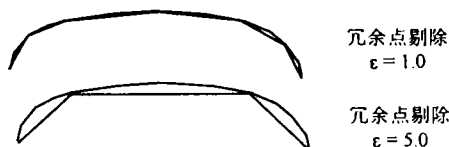


图3 冗余点去除

3.1.2 聚点消除 聚点(Agglomerate Points)一般有两类:钩子(hooklet)和圈(circulet),如下左图所示。这些点在冗余点消除中不会被去掉但是会影响后面图形的识别。考虑到转折处的点密度较大,我们定义长度为 L ,结点数为 N 的曲线的点密度为 N/L ,曲线中任一段长度为 l 结点数为 n 的折线的点密度为 n/l ,如果相对密度 Ln/Nl 大于设定阈值,则认为该字段上的点均是聚点,消除方法是用重心代替。其算法复杂度为 $O(n^2)$ 。实验效果如图4。

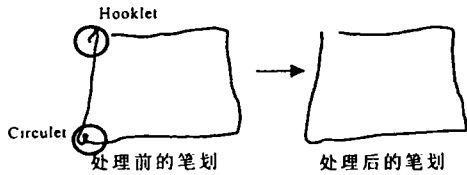


图4 聚点消除

3.1.3 端点校正 图形识别要求输入曲线的起点和终点重合,这对一般用户很难做到。通常会出图形端点分离或者交叉的情况。这会影响后面图形拟合的精度。我们将分离情况下的边反向延长转化为交叉类型,将其交点作为新的顶点。而如果交叉突出的部分与笔划长度的比例小于一个阈值,则认为用户的误输入,将之删除(如图5)。

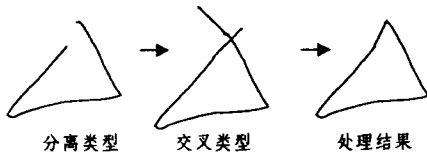
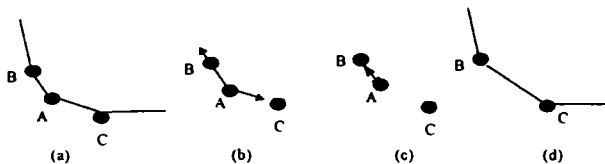


图5 端点校正过程

3.1.4 闭包计算 其目的是将待识别图形变为凸多边形,减小凹多边形给后面识别带来的噪声。我们采用由 Jarvi's March 提出的平面点集的闭包算法,其算法复杂度为 $O(n^2)$ 。

3.2 基本图形分类

我们设计了一种“引力模型”来识别基本图形。认为点与点之间有相互吸引的趋势,如果这种趋势大于某一个阈值的话,就认为这两个点之间的内部能量很大而受到外部的影响较小,可以将它们合为一个点。设曲线上任一点 A 的相邻点为 B 和 C, A 受到 B 和 C 的吸引。如果 B 对 A 的吸引力更强并且高于设置的阈值,则将 A 和 B 合并为一点,如图6所示。



(a) B, C 是 A 的相邻顶点; (b) A 受到 B 和 C 的吸引
(c) B 对 A 的吸引更强并且高于设定的阈值 (d) 将 A, B 合并

图6 顶点合并过程

给定 A, B 的转角分别为 α, β , $Dis(A, B)$ 是 A, B 两点的距离, $f(A, B)$ 是两顶点之间的引力。 $f(A, B)$ 应满足以下条件: ① α 越大, $f(A, B)$ 越大; ② β 越大, $f(A, B)$ 越小; ③ $Dis(A, B)$ 越大, $f(A, B)$ 越小。由此定义:

$$f(A, B) = \frac{\alpha}{\beta Dis^2(A, B)}$$

对于任意一顶点,计算两个相邻点对它的吸引,如果引力值超过了阈值,将该点和相邻点合并。整个算法描述如下:

顶点归并算法

1. 对每一顶点 A, 根据曲线旋转方向找到上邻点 B 和下邻点 C, 计算 $F(A) = f(A, B) - f(A, C)$;
2. 选择 $|F(A)|$ 最大的点 A;
3. 如果 $|F(A)|$ 小于设定的引力阈值, 则停止计算, 否则将 A 与 B 合并(如果 $F(A) > 0$) 或者与 C 合并(如果 $F(A) < 0$);
4. 如果剩余点的数目 ≤ 3 , 停止计算;
5. 返回第一步。

其中阈值的设定决定最终的点数目, 过大的阈值会增加最后点序列规模, 加大以后的计算量; 过小的阈值则会失掉关键点, 造成对形状的误判。一般情况根据不同的用户输入进行

调整。

经过聚点消除后, 设剩下点的数目为 V_{min} , 若 $V_{min} < 3$, 则最终形状为三角形; 若 V_{min} 为 4(5, 6), 则分别对应四(五, 六)边形; 若 $V_{min} > 6$, 则最终识别成椭圆。

3.3 图形拟合

拟合的目的是寻找与输入曲线最为相似的图形。以下分两类情况讨论:

3.3.1 多边形的拟合

多边形拟合主要分两个步骤: 首先要获得曲线的最佳分割, 使得每个子段都能最大程度逼近目标多边形的各边; 接着对各段曲线进行线性拟合, 通过拟合直线的交点确定最终的目标图形。其中第二步的拟合比较简单。所以, 多边形拟合关键是如何得到曲线的最优分割。我们采用搜索的方法寻找最优分割, 设曲线的节点数为 m , 拟合多边形的顶点数为 n 。设曲线的点序列为 S , 其中节点为 $P_i (i=0, 1, 2, \dots, m)$; 拟合后的多边形顶点序列为 T , 其中各个顶点为 $q_j (j=0, 1, 2, \dots, n)$ 。以下定义:

$$D1 = \sum Dis(p_i, T) / m \text{ 为 } S \text{ 到 } T \text{ 的距离;}$$

$$D2 = \sum Dis(q_j, T) / n \text{ 为 } T \text{ 到 } S \text{ 的距离;}$$

$D = D1 + D2$ 为曲线拟合成多边形的代价。 D 越大, 表明曲线与多边形的相似度越低。

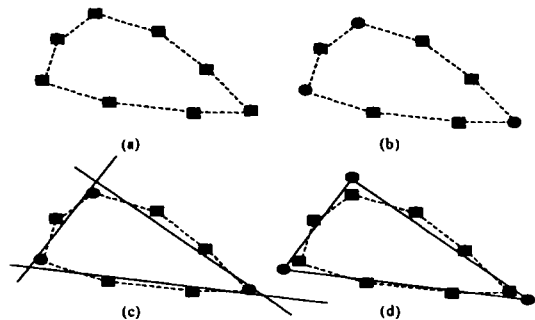
搜索算法如下:

最佳分点搜索算法

1. 在 m 点中任取 n 点;
2. 根据上面所取的 n 个分点将曲线拟合成 n 边形;
3. 计算曲线与多边形之间的拟合代价 D ;
4. 如已列举所有情况, 跳至下一步, 否则跳至第一步;
5. 取拟合代价最小的情况作为最终的拟合结果。

由于第一步中是任意选取 n 个分点, 因此, 会有多种选择。如果在 S 中穷举所有可能的情况, 不仅增加了运算复杂度, 由于一些非关键点的存在, 也会影响拟合精度。因此, 需要对算法优化。在聚点消除阶段, 我们通过阈值来控制最终点序列的规模。因此, 我们可以首先用较大的阈值获得候选点序列 S' , 用较小的阈值在 S' 中确定曲线的最终形状 n 。在拟合阶段, 我们用 S' 代替 S 作为源点序列, 这样, 不仅降低了运算复杂度, 而且 n 是由 S' 确定的, 因此, 在 S' 中寻找分点进行拟合又可以保证最终的效果可以最大程度地逼近理想情况。

$D1 (D2)$ 是 $S(T)$ 中每个点到 $T(S)$ 的有效距离 $\min\{d1, d2\}$ 之和, 其中 $d1$ 是 $P_i(q_j)$ 与 $T(S)$ 中点距离的最小值, $d2$ 是 $P_i(q_j)$ 到 $T(S)$ 中相邻两点连接的线段的有效距离(即点到线段的投影点在线段内部)的最小值。根据第六步确定最佳分点后, 就可以进行多边形的拟合。过程如下:



其中 (a) 图是获得的候选点序列 S' ;
(b) 中黑点是根据拟合代价确定的最佳分点
(c) 是对各个分段进行线性拟合; (d) 是最终的拟合效果

图7 多边形拟合

3.3.2 椭圆的拟合

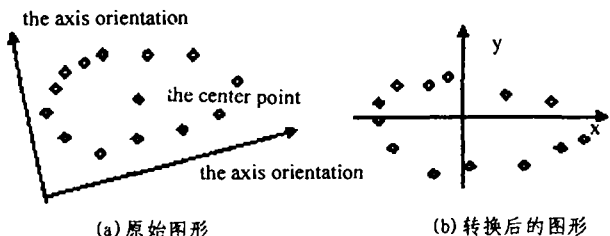


图8 椭圆拟合

椭圆拟合实际上就是要获得椭圆的参数化表示,并使得拟合的椭圆到用户输入的椭圆点序列之间的平均代价最小,如图8(a)。一般地,椭圆方程表示为:

$$\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1$$

因此,需要确定四个参数:长短轴 a, b, 中心点坐标 (x₀, y₀), 此外,还有坐标轴的方向。其中,中心点可以通过求已知样本点的重心得到。轴方向则是通过协方差矩阵的特征向量得到。得到中心点和坐标轴方向后,可以通过坐标平移,旋转得到椭圆的标准方程:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1.$$

此时,还要求出参数 a 和 b, 求出参数 a, b 使得平均代价最小不但难于实现而且没有必要,我们仅仅给出 a, b 的近似估计。

令 $Y = b^2 - \frac{b^2}{a^2} X$, 其中 $Y = y^2, X = x^2$,

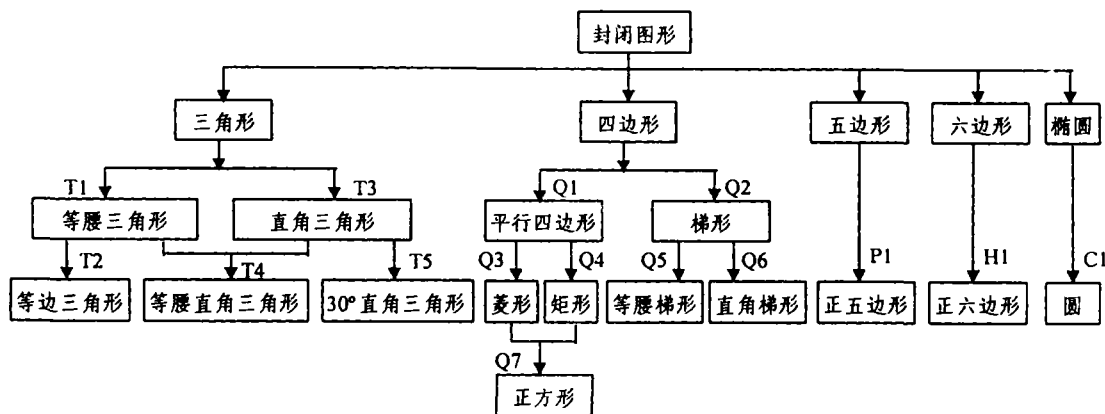
模仿线性拟合的做法,将所有样本点的 x² 和 y² 代入方程可以确定 a 和 b。转换后的图形见图8(b)。

4 图形规整化

图形的识别以及拟合只是使得图形最贴近用户的输入形状,但可能并不是用户希望得到的图形。规整化的目的是将拟合后的图形调整到最规整的形状,使它与用户的输入意图相吻合。规整化过程分为图形内规整化与图形间规整化。图形内规整只依据图形自身的信息对其规整,如三角形有两条边近似等长,则将它规整为等腰三角形;而图形间规整则利用了相邻图形的信息,如两个相邻矩形形状接近,则将它们调整到同等大小。

表1 图形规整规则

<p>三角形规整规则:</p> <p>T1. 等腰三角形: 两条边近似等长</p> <p>T2. 等边三角形: 三条边近似等长</p> <p>T3. 直角三角形: 一个内角接近90°</p> <p>T4. 等腰直角三角形: 满足 T1 和 T3</p> <p>T5. 30°直角三角形: 满足 T3 并且有一个内角为30°或60°</p> <p>四边形规整规则:</p> <p>Q1. 平行四边形: 两组对边接近平行</p> <p>Q2. 梯形: 只有一组对边接近平行</p> <p>Q3. 菱形: 满足 Q1 并且相邻边近似等长</p> <p>Q4. 矩形: 满足 Q1 并且有一个内角为90°</p> <p>Q5. 等腰梯形: 满足 Q2 并且不平行的对边近似等长</p> <p>Q6. 直角梯形: 满足 Q2 并且有一个内角为90°</p> <p>Q7. 正方形: 满足 Q4 和 Q3</p> <p>椭圆规整规则:</p> <p>C1. 圆: 两个轴近似等长</p> <p>五边形规整规则:</p> <p>P1. 正五边形: 每条边近似等长</p> <p>六边形规整规则:</p> <p>H1. 正六边形: 每条边近似等长</p>
--



图中每个节点代表一个规则,箭头方向代表匹配的顺序。

图9 图形内规整化过程

图形内规整的过程分以下几步:

1. 等边校正: 对于多边形,如果两条边的长度相差不多,则将它们调整为相等;
2. 平行校正: 如果多边形的两条边方向接近平行,则将它们调整为平行;
3. 角度校正: 如果多边形的内角接近90°,将它调整为直角;如果内角的某一边垂直或水平,则将它调整为30°(150°)或45°(135°)或60°(120°),具体情况视角的大小而定;
4. 水平/垂直校正: 如果多边形的边,椭圆的长(短)轴,菱形的对角方向接近于水平或垂直,则将它们调整为水平或垂直。

其中1,2,3步是根据一些预定义的规则对图形不断进行形状规整,4则在保持图形形状的基础上进行方向的调整。表1给出了对图形规整采用的规则;图9给出了从最初的封闭图形调整到最终结果的规则树。

图形间规整过程又可以分为以下几个步骤:

1. 形状调整: 如果相邻图形形状相同(形状信息来自于图形内规整),并且大小接近,那么将它们规整为同等大小,见图10(a)。
2. 位置调整: 如果相邻多边形的边(或椭圆的中心点)几乎在同一个水平/垂直线上,则对图形进行调整使之在同一条直线上,见图10(b)。

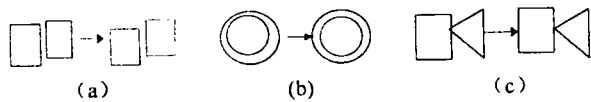
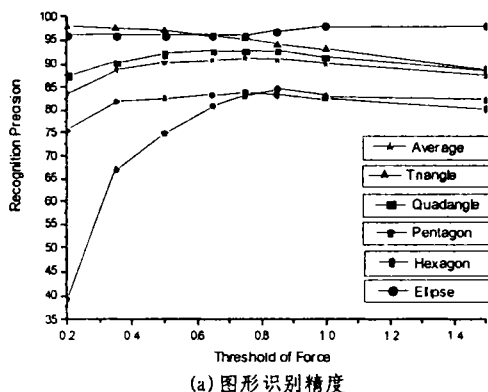


图10 图形间规整



3. 关键点调整:我们把椭圆的中心点,多边形的顶点以及边的中点称为关键点。如果相邻图形关键点的位置接近,则在保持图形性状的基础上,通过旋转,平移,缩放等操作将关键点重合,见图10(c)。

5 实验结果

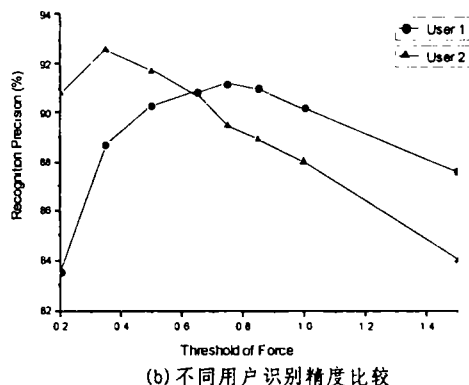


图11

我们让两个用户各自输入不同的基本图形,在 Pentium III650, Windows 2000下进行试验。识别结果如图11(a),图中横坐标是引力算法中设置的阈值,纵坐标是识别精度。从图中可以看出:

1. 不同图形识别精度的峰值出现在不同的位置。
2. 最初提高阈值,识别精度变化较为明显,到了某个值后,识别精度趋于平稳。
3. 对于三角形,阈值越低,识别效果越好,而对于其它多边形则是阈值越大,效果较好。椭圆的情况比较特殊,在某个阈值下出现最低峰值。
4. 平均识别精度高于90%,具有较好的效果。

图11(b)则给出了不同用户在不同引力阈值下的识别精度,可以看出,由于用户输入习惯的差别,相同的阈值下的识别效果不尽相同。而各自的平均识别精度也会受到输入习惯的影响而有所差异。

图12的样例给出一些基本图形的最终识别效果:



图12 基本图形识别结果

Smart Sketchpad^[9]系统使用本文提出的方法对在线绘制的草图进行识别。用户用光笔(或鼠标)在输入板上绘制草图,光笔在输入板上移动时,系统对轨迹进行采样,得到的采样点序列作为原始的笔划,到光笔离开输入板时,系统随即给出对原始笔划识别的结果。图13给出了绘制草图的识别效果。

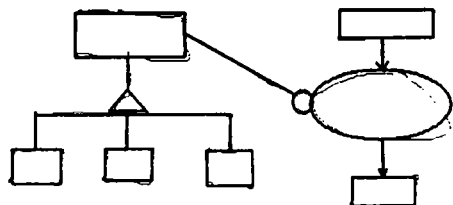


图13 手绘草图的识别

结束语 本文针对在线图形识别的特点给出了一种简单、快速、有效的简单图形识别方法。并在识别精度、计算量以及图形数量上进行了合理的平衡,经分组试验取得令人满意的效果。图形识别不仅是模式识别的问题,与人工智能、人的认知原理以及心理学等多个领域相关,为了进一步地提高识别效果,应利用笔划输入的上下文关系并提高系统的学习能力,这将是后续研究的重点。

参考文献

- 1 Sutherland I E. Sketchpad: A Man-Machine Graphical Communication System. Spring Joint Comp. Conf., AFIPS Press, 1963. 2~19
- 2 Takayuki D K, Ajay A, Van V. Recognizing Multistroke Geometric Shapes: An Experimental Evaluation. In: Proc. of the 6th Annual ACM Symposium on User Interface Software and Technology, 1993. 121~128
- 3 Fonseca M J, Jorge J A. Using Fuzzy Logic to Recognize Geometric Shapes Interactively. In: Proc. 9th IEEE Conf. on Fuzzy Systems, Vol. 1, 2000. 291~296
- 4 Revankar S, Yegnanarayana B. Machine Recognition and Correction of Freehand Geometric Line Sketches. In: Proc. of IEEE Intl. Conf. on Systems, Man, and Cybernetics, vol. 1, 1991. 87~92
- 5 Joseph O R. Computational Geometry. C. Cambridge Univ. Press, 2nd edition, 1998
- 6 Boyce J E, Dobkin D P. Finding Extremal Polygons. SIAM Journal on Computing, 1985, 14(1): 137~147
- 7 Freeman H, Shapira R. Determining the Minimum-area Enclosing Rectangle for an Arbitrary Closed Curve. Communication of the ACM, 1975, 18(7): 409~413
- 8 Sezgin M T, Stahovich T, Davis R. Sketch Based Interface: Early Processing for Sketch Understanding. In: Proc. of PUI-2001, Nov. 2001
- 9 Liu W, Qian W, Xiao R, Jin X. Smart Sketchpad-An On-line Graphics Recognition System. In: Proc. of ICDAR2001, Seattle, Sep. 2001