基于滴水算法的验证码中粘连字符分割方法

李兴国[1,2],高 炜[1]
LI Xingguo[1,2], GAO Wei[1]

1.合肥工业大学 管理学院,合肥 230009
2.过程优化与智能决策教育部重点实验室,合肥 230009
1.School of Management, Hefei University of Technology, Hefei 230009, China
2.Key Laboratory of Process Optimization and Intelligent Decision-making, Ministry of Education, Hefei 230009, China

LI Xingguo, GAO Wei. Segmentation method for merged characters in CAPTCHA based on drop fall algorithm. Computer Engineering and Applications, 2014, 50(1):163-166.

Abstract: Many researches demonstrate that good result can be gained by existing machine learning algorithms in the recognization of CAPTCHA(Completely Automated Public Turing test to tell Computers and Humans Apart) if single characters can be split. A method is presented to segment the merged characters in the recognization of CAPTCHA with touching characters. It seeks division points by combining the statistics of character width and the vertical histogram projection minimums, and then uses these points as the starting points of the drop fall algorithm to segment merged characters in CAPTCHA. The experiments show that it is a general method and can improve the recognization rate.
Key words: drop fall algorithm; CAPTCHA; merged characters; segmentation; vertical histogram projection

摘 要: 众多研究表明,如果能将验证码中的字符分割开来,用现有的机器学习算法一般都能取得比较好的识别效果。针对字符粘连情况下的验证码的识别问题,提出了一种粘连字符的分割方法。该方法将字符的宽度统计值和竖直投影直方图中的投影极小值点相结合找到分割点,以这些分割点作为滴水算法的起始滴落点对粘连字符进行分割。实验结果证明,该方法用于分割验证码中的粘连字符具有一般性,能够提高验证码识别率。
关键词: 滴水算法;验证码;粘连字符;分割;竖直投影
文献标志码: A **中图分类号:** TP391 doi:10.3778/j.issn.1002-8331.1208-0310

1 引言

验证码(Completely Automated Public Turing test to tell Computers and Humans Apart,CAPTCHA),即全自动区分计算机和人类的图灵测试,是一种区分用户是计算机还是人类的公共全自动程序[1]。当前,为了保障网络应用的安全,防止计算机自动恶意程序对网络的攻击以及对网络资源的恶意占用等,验证码技术得到了广泛的应用。目前最常见的一种形式是让用户输入一幅图片上的字符,以证明用户的人类身份。本文以下所说的皆是指这种类型的字符验证码。

国内外已经有越来越多的学者和机构对验证码的识别进行研究。一方面,对验证码识别的研究可以使人们发现当前所使用的验证码的缺陷和不足之处,从而设计出更好、更安全可靠的验证码,以保障网络应用的安全;另一方面,当某种验证码被破解时,也是人工智能领域的一大进步[2]。

验证码的识别一般分为验证码图片的预处理阶段和识别阶段。首先将图片二值化、去除噪点,然后将单个的字符分割出来,最后进行识别。目前绝大多数的识别算法都需要验证码字符的分割这一步骤。文献[3]的研究表明,只要能将验证码上的字符分割出来,用现有的机器学习算法一般都可以取得比较好的识别效果。机器学习算法可以有效地解决字符识别问题,故字符的有效分割对验证码的识别至关重要,然而当前还没有一种通用、有效的字符分割算法[4]。非粘连字符的分割可以很容易地通过二值化后的图片的竖直投影直方图来

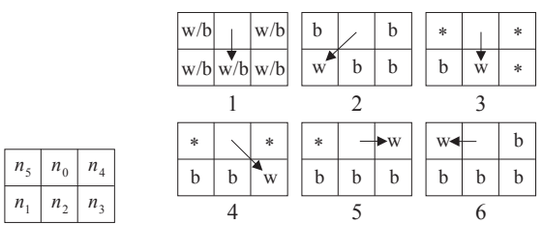
作者简介: 李兴国(1963—),男,教授,硕士生导师,研究方向为信息资源管理,信息系统,电子商务和电子政务。E-mail:lxglzz@sina.com
收稿日期: 2012-08-27 **修回日期:** 2012-12-25 **文章编号:** 1002-8331(2014)01-0163-04
CNKI **网络优先出版:** 2013-01-22, http://www.cnki.net/kcms/detail/11.2127.TP.20130122.1437.007.html

完成,而粘连字符的分割则很困难,是导致识别错误的最主要的原因[5]。事实上,粘连字符的设计也是保障验证码的安全的最有效的手段[6-7]。本文通过验证码字符的宽度统计值及竖直投影直方图的局部极小值点相结合的方式来为滴水算法找到滴落的起始点,然后用滴水算法来对粘连字符进行分割,通过两种不同的验证码的分割实验证明了该方法的有效性。

2　滴水算法

文献[8]针对手写数字字符的分割提出了一种分割算法:滴水算法。滴水算法根据水滴滴落的方向的不同可分为向上滴落和向下滴落。以向下滴落为例,该算法模拟水滴从高处向低处滴落的过程来对粘连字符进行切分。水滴从字符串顶部在重力的作用下,沿着字符轮廓向下滴落或水平滚动,当水滴陷在轮廓的凹处时,将渗漏到字符笔划中,经穿透笔划后继续滴落,最终水滴所经过的轨迹就构成了字符的分割路径[9]。

滴水算法的滴落规则如图1所示。图1(a)中 n_0 表示水滴当前的位置,水滴的下一步的滴落位置由它下方的三个像素点和它左右两个像素点共五个像素点的情况决定,它们的编号如图1所示。图1(b)显示了水滴的下一滴落位置的选择规则,其中w表示白点,b表示黑点,*表示既有可能是白点也有可能是黑点。以1为例,当水滴当前的位置上邻近的五个像素点全是白点或黑点时,水滴将向下滴落。值得注意的是,当水滴落入字符轮廓的凹陷处时,情况5和情况6将会交替出现,即水滴将左右滚动,此时采取的策略是水滴在凹陷处右侧向下竖直渗漏。



(a)水滴周围像素编号　(b)水滴下一滴落位置的选择

图1　滴水算法的滴落规则

文献[10]给出了滴水算法的数学描述。设所要分割的图片是 I,I 是经过二值化后的,它的尺寸是 $N\times M$,N 是图片的高,M 是图片的宽,建立坐标系如图2。设水滴的当前坐标为 (x_i,y_i),水滴的滴落路径为 T,则 $T(x_{i+1},y_{i+1})=f(x_i,y_i,W_i),i=0,1,\cdots$。其中 (x_{i+1},y_{i+1}) 表示水滴下一步滴落点的坐标,(x_0,y_0) 是起始滴落点,W_i 则是水滴在当前位置上的重力势能的衡量。W_i 的值由式(1)决定:

$$W_i=\begin{cases}4, & \sum=0 \text{ or } 15\\ \max\limits_{j=1}^{5} z_j w_j, & \text{其他}\end{cases} \tag{1}$$

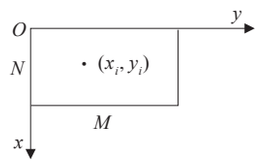


图2　I的坐标系

其中,$\sum=\sum_{j=1}^{5} z_j w_j$,$z_j$ 表示 n_j 点的像素值,0表示黑点,1表示白点。w_j 表示 n_j 点被选为下一滴落点的权重大小,$w_j=6-j$。那么:

$$T(x_{i+1},y_{i+1})=f(x_i,y_i,W_i)=\begin{cases}(x_i,y_i-1), & W_i=1\\ (x_i,y_i+1), & W_i=2\\ (x_i+1,y_i+1), & W_i=3\\ (x_i+1,y_i), & W_i=4\\ (x_i+1,y_i-1), & W_i=5\end{cases} \tag{2}$$

对于粘连字符的分割,滴水算法可以获得比直接竖直进行分割更好的效果,特别是在字符存在着倾斜、扭曲等的情况下[11]。如图3所示,图(a)是某验证码粘连字符片段,图(b)是找到一个合适的位置直接竖直分割的效果,图(c)是用滴水算法分割的效果。可以看出,滴水算法沿着字符的轮廓进行分割,可以有效地避免直接竖直分割造成的过分分割的问题。这也是本文采用滴水算法进行分割粘连字符的原因。

(a)某验证码片段　(b)竖直分割　(c)滴水算法分割

图3　竖直分割与滴水算法分割对比示意图

3　粘连字符的分割

传统滴水算法的起始滴落点的确定规则是这样的:由上到下由左到右扫描图片中的每一行,选择第一个满足像素分布情况为(…0*1…10)的白色像素点(*)为起始点。其中0表示黑色像素点,1表示白色像素点。这种适合手写数字字符特征的规则却并不适合验证码中的字符的情况。例如字母“X”和“Y”等,按照这种规则,起始的滴落点就会落入笔画的凹陷处,这显然是不合适的,会造成错误的分割。而基于投影的分割算法[1,4]根据投影直方图的特征能够找到较为合适的分割点,但它找出分割点后直接竖直地分割,如前所述,很容易造成过分割的现象。因此,本文提出利用基于投影的方法找出分割点,然后以这些分割点作为滴水算法的起始滴落点,用滴水算法进行分割。

本文提出的对粘连字符的分割方法要使用字符的宽度统计值,因此事先要取得验证码字符的一些宽度统计值。随机选取 n 张验证码图片,二值化后手工分割进行统计,n 应足够大,字符宽度统计值包括最小字符宽度

D_{min}、最大字符宽度 D_{max} 以及平均字符宽度 D_{mean}。

在对验证码字符分割之前，需要对验证码图片进行预处理。首先将图片二值化，本文采用最大类间方差法[12]。二值化的图像往往会有许多噪音，造成图像字符分割困难，也不容易提取字符特征[13]。因此如果二值化后，验证码图片上有干扰噪点，则还要进行去除噪点的处理。本文采用的是文献[14]提出的算法，统计每个黑色像素点周围八个方向上的黑色像素点的个数，如果低于某个阈值则认为是离散点，将其去除。

验证码字符分割的流程是：首先用CFS算法确定字符区块；然后对各个区块判断是否含有粘连字符，对粘连字符区块做竖直投影统计，找出投影极小值点；最后结合字符宽度统计值和这些极小值点找出分割点，以这些分割点作为滴水算法的起始滴落点进行分割。具体的算法流程如图4所示。

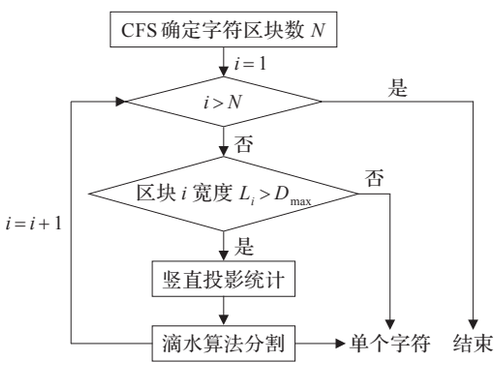
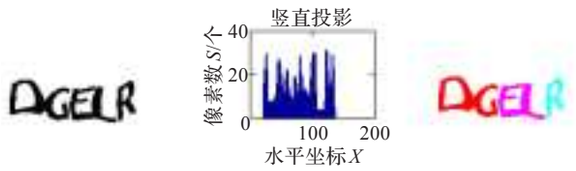


图4　粘连字符分割流程图

3.1　CFS分割

文献[15]提出了CFS(Color Filling Segmentation)分割算法。该算法的工作原理是：探测验证码图片上的黑色像素点，当探明一个黑色像素点后，以它为中心在它的邻近八个方向上继续探测黑点，每探测到一个黑点就继续在它的邻近方向上探测，这个过程不断重复直至检测不到新的黑点。这样这一过程结束之后，就探明了一个字符区块。在这个区块之外的另一个黑点开始，继续执行以上过程直到图片上的所有黑点都被检测到，这样就确定了若干个字符的区块。该算法就好像是用不同的颜色填充字符块一样，因此被称为Color Filling Segmentation(CFS)。

之所以首先使用CFS分割，是因为考虑到很多粘连字符的验证码存在部分粘连的情况，故通过这种方法先确定粘连区块，可以提高后续粘连字符的分割准确率。采用该算法而不使用竖直投影直方图则是因为即使字符存在倾斜，只要字符间不存在直接的粘连，该算法仍可以有效地分割。如图5所示，因为该验证码字符间存在倾斜，通过竖直投影直方图无法确定字符区块，而CFS算法成功地确定了三个区块。



(a)某验证码二值图　(b)竖直投影直方图　(c)CFS分割图

图5　CFS分割示意图

3.2　竖直投影统计

经过CFS分割确定了若干个字符区块后，首先考察各个区块的宽度 L，如果大于最大字符宽度 D_{max} 则认为是粘连字符区块，否则认为是单个字符。然后对粘连字符区块做竖直投影，统计投影值的局部极小值点。投影的局部极小值点的确定如图6所示，选择的是情况1中的 A 点以及情况2中的 B、C 点。要注意的是在情况2下的投影平坦区域只选择两端的端点。设 $\mathrm{d}(y)$ 表示 y 点的竖直投影值，$H(y)=\mathrm{d}(y+1)-\mathrm{d}(y)$，则投影极小值点满足式(3)或式(4)。图6中的 A 点即满足式(3)中的 y 点，B 点即满足式(4)中的 y 点，C 点即满足式(4)中的 $y+j$ 点。

$$\begin{cases} H(y-1)<0 \\ H(y)>0 \end{cases} \tag{3}$$

$$\begin{cases} H(y-1)<0 \\ H(y+i)=0 \\ H(y+j)>0 \\ i=0,1,\cdots,j-1 \\ 0<j\leqslant L-2 \end{cases} \tag{4}$$

(a)情况1　(b)情况2

图6　竖直投影局部极小值点的确定

3.3　滴水算法分割

设粘连字符区块在竖直投影统计中得到的投影极小值点的个数为 s，字符区块宽度为 L，则该区块粘连字符的个数 $num=round(L/D_{mean})$，$round$ 为四舍五入操作，那么该区块要确定的分割点的个数为 $num-1$。设 y_p $(p=0,1,\cdots,num-1)$，$p=0$ 时 y_p 表示区块的起始点，$p>0$ 时 y_p 表示第 p 个分割点的位置。则第 p 个分割点 y_p 为满足式(5)中的最后一个 y_i，y_i 是竖直投影统计中得到的投影极小值点。其中，$D(y_i,y_j)$ 表示 y_i、y_j 两点之间的距离，y_L 表示区块的结束点。当得到这 $num-1$ 个分割点后，以这些分割点作为滴水算法的起始滴落点进行分割，为避免水滴无限制地向右滚动而有可能造成的过分割问题，规定水滴在 y 轴上的坐标不得超过水滴的起始点，即在根据式(2)计算得出 y_{i+1} 后，最终

$y_{i+1}=\min(y_{i+1},y_0)$。此处 y_0 是水滴起始点的 y 坐标，min为取最小值的操作。

$$\begin{cases} D_{\min}\leqslant D(y_{p-1},y_i)\leqslant D_{\max} \\ D_{\min}\times(num-p)\leqslant D(y_i,y_L)\leqslant D_{\max}\times(num-p) \\ i=1,2,\cdots,s \\ p=1,2,\cdots,num\ -1 \end{cases} \tag{5}$$

4　实验与分析

本文选择两个不同网站上的含粘连字符的验证码进行对比分割实验，如图7所示，实验结果如表1所示。凤鸣轩网站的验证码大多是三四个字符粘连，少数是两两粘连，字体固定。太平洋网络的验证码大多数是两三个字符粘连，少数是四五个字符粘连，字体有一些变化并存在扭曲。验证码字符分割实验的步骤如下：

（1）随机选取100张验证码图片，二值化、去噪点后手工分割，统计最小字符宽度 $D_{\min}$、最大字符宽度 $D_{\max}$，以及平均字符宽度 D_{mean}。

（2）随机选取300张验证码图片，分别采用基于投影的分割算法、传统的滴水算法和本文提出的算法分割，统计正确率。

（a）凤鸣轩　（b）太平洋网络

图7　两种验证码示例

表1　两种验证码分割实验结果

验证码	图片个数	基于投影的分割/(%)	传统滴水算法分割/(%)	本文算法分割/(%)
凤鸣轩	300	77.3	22.0	96.0
太平洋网络	300	51.0	13.0	78.7

经统计，凤鸣轩验证码的最小字符宽度 $D_{\min}=5$，最大字符宽度 $D_{\max}=13$，平均字符宽度 $D_{\mathrm{mean}}=10$。太平洋网络验证码的最小字符宽度 $D_{\min}=15$，最大字符宽度 $D_{\max}=36$，平均字符宽度 $D_{\mathrm{mean}}=23$。从实验结果中可以看出，本文提出的分割算法与基于投影的分割算法以及传统的滴水算法相比，可以极大地提高粘连字符的分割准确率。可以看到该方法对于像凤鸣轩这样字体固定，字符大小差别不大，没有扭曲和倾斜的验证码分割准确率较高，而对于像太平洋网络这样字体有变化，字符大小相差较大，存在扭曲，倾斜的验证码的正确率则有所下降，而分割错误的主要原因是确定了错误的粘连位置，即不合适的滴水算法的起始滴落点。

5　结束语

本文提出了一种利用滴水算法对验证码中粘连字符进行分割的方法，该分割方法具有一般性，可以解决一些验证码的粘连字符的分割问题，然而对于字符的宽度差别很大，字符存在扭曲和倾斜的情况其正确率会有所下降，难点在于确定字符粘连的位置。如何提高这种情况下的粘连字符的分割准确率，则是下一步的研究方向。

参考文献：

[1] 王璐，张荣，尹东，等.粘连字符的图片验证码识别[J].计算机工程与应用，2011，47(28)：150-153.

[2] 吕刚，郝平.基于神经网络的数字验证码识别研究[J].浙江工业大学学报，2010，38(4)：433-436.

[3] Chellapilla K，Larson K，Simard P，et al.Computers beat humans at single character recognition in reading-based Human Interaction Proofs[C]//Proceedings of the 2nd Conference on Email and Anti-Spam，2005.

[4] Huang Shihyu，Lee Yeuankuen，Bell G，et al.A projection-based segmentation algorithm for breaking MSN and YAHOO CAPTCHAs[C]//Proceedings of the World Congress on Engineering，2008.

[5] Yan J，El Ahmad A S.Breaking visual CAPTCHAs with naive pattern recognition algorithms[C]//Proceedings of 23rd Annual Computer Security Applications Conference，2007：279-291.

[6] Chellapilla K，Simard P Y.Using machine learning to break visual hips[C]//Proceedings of Conference on Neural Information Processing Systems，2004.

[7] Bursztein E，Martin M，Mitchell J C.Text-based CAPTCHA strengths and weaknesses[C]//Proceedings of ACM Computer and Communication Security，2011.

[8] Congedo G，D imauro G，Impedovo S，et al.Segmentation of numeric strings[C]//Proceedings of the 3rd International Conference on Document Analysis and Recognition，1995：1038-1041.

[9] 马瑞，杨静宇.一种用于手写数字分割的滴水算法的改进[J].小型微型计算机系统，2007，28(11)：2110-2112.

[10] Wang Xiujuan，Zheng Kangfeng，Guo Jun.Inertial and big drop fall algorithm[J].International Journal of Information Technology，2006，12(4)：39-48.

[11] 常丹华，何耘娴，苗丹.中英混排文档图像粘连字符分割方法的研究[J].激光与红外，2010，40(12)：1369-1373.

[12] Ostu N.A threshold selection method from gray-level histograms[J].IEEE Transactions on Systems，Man and Cybernetics，1979，9(1)：62-66.

[13] 殷光.基于SVM的验证码识别算法研究[D].合肥：安徽大学，2010.

[14] 许明.验证码的识别与反识别[D].南京：南京理工大学，2007.

[15] Yan J，El Ahmad A S.A low-cost attack on a Microsoft CAPTCHA[C]//Proceedings of the 15th ACM Conference on Computer and Communications Security，2008：543-554.