

使用动态规划识别自由手绘流程图*

施大鹏^{1,2}, 陈泉^{1,2}, 冯桂焕^{1,2}, 骆斌^{1,2}

¹(南京大学软件学院, 江苏南京 210093)

²(计算机软件新技术国家重点实验室, 江苏南京 210093)

Recognize Online Handwritten Flowchart with Dynamic Programming*

SHI Da-peng^{1,2}, CHEN Quan^{1,2}, FENG Gui-huan^{1,2}, LUO Bin^{1,2}

¹(Software Institute, Nanjing University, Nanjing 210093, China)

²(State key Laboratory for Novel Software Technology, Nanjing 210093, China)

Abstract: Although it is easier for people to record their inspiration with paper and pen, it is hard to store and edit these sketches digitally. Sketch recognition can be an effective solution. In this paper we present a method with which we can recognize the on-line handwritten flowchart. Our approach is based on dynamic programming and can be divided into three stages: hypotheses generation, hypotheses classification and hypotheses combination. We choose WEKA as our classification tool and Random Forest as our classification algorithm. The experiment is tested on FCinkML database, which includes 177 flowcharts as test samples. In the result, the symbol recognition precision is 63.40% and the stroke recognition precision is 76.25%. The result prove that on-line handwritten flowchart can be recognized with our approach, but the recognition precision should be improved in the future.

Key words: on-line sketch recognition, flowchart, dynamic programming

摘 要: 使用纸笔可以比使用软件更加轻松的记录下人们的创造性思维, 不过使用纸笔生成的草图却很难进行数字化存储与编辑的操作。草图识别技术可以有效解决这一问题。本文主要解决了手绘流程图的自动识别问题。我们的方法以动态规划为基础, 分为生成假设、识别假设以及组合假设三个阶段。使用了 WEKA 作为分类工具, 选择了 Random Forest 作为具体的分类算法, 没有对用户的书写顺序设置约束, 实现自由手绘。使用了 FCinkML 数据库作为测试平台, 选取了其中的 177 张流程图作为测试样本。实验结果中符号的识别率为 63.40%、笔画的识别率为 76.25%。结果证明我们的方法可以实现流程图的识别, 不过在精度上存在较大的改进空间。

关键词: 在线草图识别; 流程图; 动态规划

引言

如今各式图表被广泛应用在各个工程领域, 而针对这些图表绘制的专业软件也应运而生。用户可以利用这些软件完成自己的图表设计。但是在很多情况下, 比如头脑风暴的讨论阶段、多人合作会议中, 人们还是习惯于使用笔来草绘出自己的大致想法, 以便快速标记和修改。然后用户需要花费额外的时间使用软件依照这些草图在计算机中绘制电子图表。而作为人机交互发展新方向的草图识别技术可以为这个问题提供一个良好的解决方案[8]。用户使用电子笔绘制草图, 计算机自动将用户绘制的草图转换成标准的电子图表, 这些电子图表可以更进一步生成简单的代码框架, 并且也更加利于归档和检索。在本文中我们完成的是图表中的一种——流程图的草图识别工作。

在草图识别中主要存在两种识别策略。一种被称为即时反馈: 用户每次画完一个部分, 计算机会对这一部分进行识别。用户可以实时的得到识别结果。文献[7]中实现的系统就是基于这种策略, 但是这样的识别策略也会对用户的书写方式产生约束, 无法实现自由手绘, 比如会导致用户必须画完一个组件后, 才能开始绘制另一个组件。另一种方法被称为懒反馈: 在用户完成全部的绘制工作后, 计算机再进行识别。这样做可以使得用户没有任何绘制过程上的约束, 实现自由手绘。我们在这次实验中采用的是第二种方法。

现在已有一些自由手绘流程图识别方面的研究工作。Ahmad-Montaser Awal[1]等人提出了一个新的自由手绘流程图样本数据库, 并且对流程图进行了一些简单识别实验, 但是他们的实验主要集中在单个符号的识别上并且当时提出的数据库容量较小, 代表性不强。Aur' elie Lemaitre[2]等人提出了一个基于语法的流程图识别方法, 这个方法充分利用了流程图的空间结构信息, 但是没有与传统的统计学方法相结合。在 Martin Bresler[3]的方法中, 他们将流程图识别建模为一个求最大和的问题。这个方法效率高, 速度快但是在最后的生成结果中他们无法保证所有的笔画均被识别。在本篇论文中我们使用的是基于统计的动态规划方法。

* Supported by the National Natural Science Foundation of China under Grant No.61100109 (国家自然科学基金);

作者简介: 施大鹏(1991-), 男, 硕士研究生主要研究领域为人机交互, 草图识别; 陈泉(1991-), 男, 硕士研究生, 主要研究领域为人机交互, 草图识别; 冯桂焕(1980-), 女, 讲师, 博士, 主要研究领域为自然人机交互, 模式识别, 机器学习; 骆斌 (1967-), 男, 教授, 博士生导师, 主要研究领域为数据库, 操作系统, 人工智能。

流程图识别方法

问题描述

流程图是使用特定的符号加上说明来表示过程、算法或流程的一种图。图 1.1 中展示的是我们从测试数据库中抽取出来的一张流程图样本。

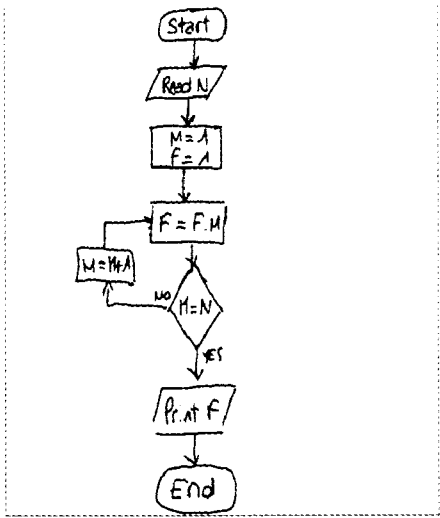


图 1.1 流程图样例

从图 1.1 中可以清楚地看出一张流程图主要是由图 1.2 中的几种符号组成。它们分别是：terminator(起止框)、arrow(箭头)、process(执行框)、data(数据框)、decision(判断框)、connection(连接点)。每一种组成元素对应特定的一种规则形状。因为本次实验主要研究的是流程图中的符号识别，所以对流程图中的文字进行了过滤。

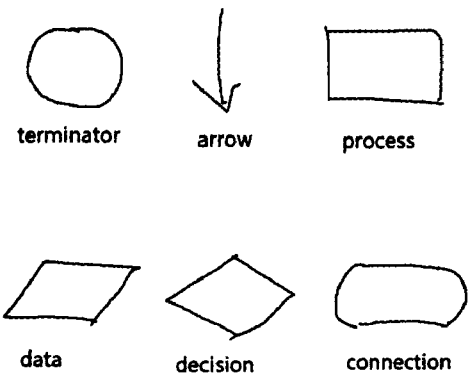


图 1.2 流程图组成符号

在线手绘流程图可以抽象为由图 1.2 所示元素的集合，即流程图 $F = \{S_1, S_2, \dots, S_n\}$ 。每一个符号由一个或多个笔画组成，可表示为符号 $S = \{s_1, s_2, \dots, s_n\}$ 。同时，每个笔画由按等时间间隔进行采样的采样点组成，即笔画 $s = \{p_1, p_2, \dots, p_m\}$ 。手绘流程图识别的目的就是在原始笔画信息的基础上，构建出组成流程图的符号集合，对 F 进行识别，最后需要得到这样的结果：

$$F = \{S_1(\text{terminator}, \{s_1 \dots s_k\}), S_2(\text{arrow}, \{s_m \dots s_n\}), \dots, S_i(\text{terminator}, \{s_x \dots s_y\})\}.$$

流程图识别方法

本文方法如图 2.1 所示，分为三个阶段，分别是：聚合笔画生成假设、识别符号假设、组合假设生成结果。

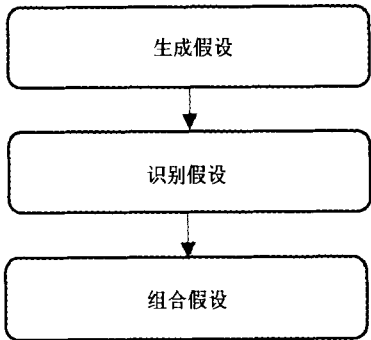


图 2.1 流程图识别过程

其中：

- 阶段 1——生成假设：
这是一个自顶向下的过程。需要根据流程图的自身特性设置约束条件实现笔画的聚合。其中使用动态规划算法作为实现方法。
 - 阶段 2——识别假设：
在这一步中使用预先训练好的分类器对得到的假设进行分类，得到假设的类别和置信度，并根据类别和置信度对假设进行过滤。
 - 阶段 3——组合假设：
在本阶段需要对已经完成分类的假设进行组合生成候选结果。然后根据给定的条件从候选结果中选出最优解。
- 以下给出各步骤的详细解释说明。

生成假设

要将流程图正确的识别出来，第一步需要知道流程图有多少符号，每一个符号由那些笔画组成。在实验中我们将聚合笔画的过程称为流程图的分割。将分割过程中生成的笔画组合称为假设。

一张流程图中的笔画会达到几十甚至上百笔，分割一张流程图产生的笔画组合数目会十分庞大。这会极大地降低整个识别流程图的效率。因此需要设置约束条件，对笔画组合进行过滤，选取其中合理的笔画组合作为假设。约束条件可以从以下三个方面考虑：1) 一个符号是由有限的笔画组成的；2) 同一个符号中的两个笔画之间的距离不会相距过远；3) 同一个符号中的笔画具有一定的顺序性，但是不需要每一笔都是连续的。基于这三个方面我们得到了以下三个具体的约束条件：

- 1) 对于 \forall 符号 $S \in \text{流程图 } F$ ， S 的笔画数量不大于 5
首先由流程图的组成元素中可以看出，符号最多由四边形组成。但是在对样本进行检查时，可以清楚地发现用户在使用电子笔绘制符号时会出现冗余笔画的现象。用户在绘制一条边时会画出数个笔画。这是人们在使用笔进行绘制时经常会出现的情况。所以我们在设置一个符号中最大笔画数量时，不能简单的将其设置为 4，需要考虑笔画冗余的情况。通过对样本进行分析，我们的到了以一个符号中的笔画数量为横坐标，以符号数量为纵坐标的统计图，如图 2.2 所示。从图中可以发现由 5 个以上笔画组成的符号很少，所以选择 5 作为一个符号中最大的笔画数量。
- 2) 对于 \forall 笔画 $s_1, s_2 \in \text{符号 } S$ ， s_1, s_2 之间的距离不大于阈值 T_1

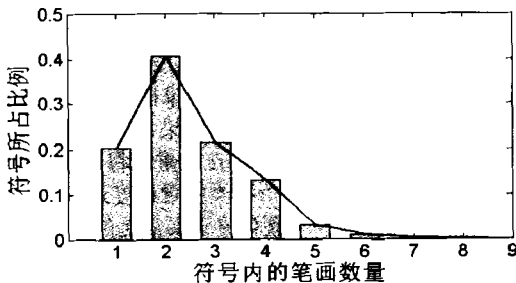


图 2.2 符号内笔画数量统计

根据流程图中符号的特点，可以发现空间距离上相聚太远的两个笔画不可能出现在一个符号中。所以实验中设置了阈值 T_1 ，要求同一个符号中笔画之间的距离均要需要不大于 T_1 。为使得方法中和距离相关的常量能够独立于流程图的大小，本文对于所有和距离相关的常量计算公式如公式 (1)，其中， $\text{Median}_{\text{length}}$ 是流程图所有笔画长度的中位数， Factor 是通过实验得到的参数。对于 T_1 ，本文选取的 Factor 取值为 0.3。

$$T = \text{Factor} * \text{Median}_{\text{length}} \tag{1}$$

3) 对于 \forall 符号 $S \in$ 流程图 F , S 中的跳跃数量不大于 1

流程图中的笔画并不总是按照一定的顺序连续绘制出来的。如图 2.3 中 (a) 所示, 流程图是按照理想的顺序绘制得到的, 但是 (b) 中所示箭头符号 A 由 $\{s1, s5\}$ 组成, 这说明用户在绘制时先绘制出箭杆 $\{s1\}$, 再绘制出平行四边形 $\{s2, s3, s4\}$, 最后再绘制箭头 $\{s5\}$ 。对于这种绘制顺序中出现的间断点, 我们将之称为笔画跳跃。但是这种笔画跳跃在一个符号中也是有限的。通过对样本库的统计, 发现一个符号中笔画跳跃的数量最大为 1。所以在这里将 1 设置为一个假设中所允许的最大笔画跳跃数量。

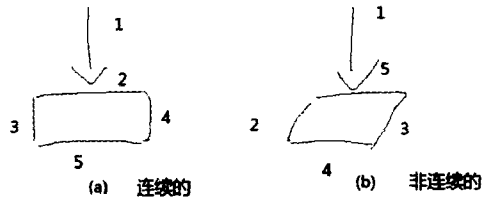


图 2.3 笔画跳跃样例

在生成假设的具体实现上, 可以将它视作一个求子集合的问题。这个子集合有元素数量和“跳跃”数量两个维度。因此我们采用的是二维动态规划方法, 这样可以高效的获得所有满足三个约束的假设。

识别假设

在实验中使用了 WEKA 作为假设分类工具。WEKA[4]最早由新西兰的 Waikato 大学计算机系提供, 现在它已经成为世界上最大的开源的机器学习平台。WEKA 有着良好的易用性, 它不需要研究人员掌握过多的机器学习知识就可以进行自己的实验。

在分类中使用的特征集是以[6]中的特征集为基础。总共可以分为 4 个大类, 分别为: 1 曲率、2 密度、3 方向、4 交点。每个大类中包括数个具体小类, 最后总计使用了 23 个特征。

在本次实验中使用的分类算法是随机森林 (Random Forest)。随机森林[5]是一个由多个随机决策树组成的分类器, 最后输出的类别是由单个决策树输出的类别的众数决定的。在提供类别的同时, 还可以根据该类别决策树的数量来确定该类别的置信度。

本文定义了如下 7 种类别, 分别是: 1) arrow, 2) terminator, 3) connector, 4) data, 5) process, 6) decision 以及 7) junk。前六种是对应图 1.2 中的流程图组成元素。同时由于假设生成过程中缺少足够的指导依据, 不可避免会将不属于同一个元素的笔画组合在一起, 从而生成非真实符号的假设, 本文使用 junk 对这类错误聚合的假设进行标注。图 2.4 给出了几种常见的 junk 假设。图 2.4Type1 是由于缺失了部分笔画使得这个假设被分类为 junk, Type2、Type3 是由于多出了部分笔画导致 junk 类型假设的出现。

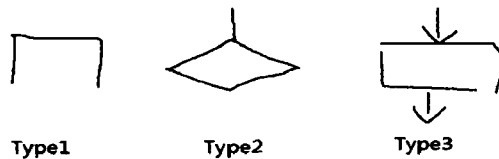


图 2.4 “Junk” 样例

分类器识别精度对整个的识别有着非常重要的影响。如果它把错误的分成正确的了, 会使第三阶段中用于组合的假设数量增多, 降低生成结果的效率; 如果它把正确的分成错误的了, 又会从根本上降低流程图的识别精度。所以对分类器的识别率进行了测试, 结果在第三节中有详细的描述。

在分类器识别假设之后, 可以得到每一个假设的分类结果, 以及每一个结果相对应的置信度。在实验中, junk 类型的假设在后续阶段中没有被使用到, 所以在这里就将所有该类型的假设全部过滤掉。接着为了使程序运行的更有效率, 我们还需要去掉其他六个种类中置信度过低的假设。因为置信度越低, 这个假设被分类正确的可能性也就越低。我们设置了一个置信度需要满足的最小阈值, 取值为 0.3。综上, 在经过分类过程后, 我们得到的是置信度大于等于 0.3 的 junk 类型的假设。

组合假设

在这阶段中我们组合假设, 生成识别结果。在组合之前, 首先使用流程图的空间结构信息优化假设分类结果。

在流程图中, 箭头扮演着一个十分重要的角色。它如同脊柱一样支撑着整个流程图的框架。同时流程图的走势也是由箭头决定的。以此为基础, 可以使用空间结构条件对非箭头的假设进行一次筛选。这些条件如下:

- 1) 箭头的两端与距其最近的两个非箭头且没有笔画重合的假设之间的距离 D 需要不大于阈值 $T2$ 。
- 2) 这两个非箭头假设之间的覆盖率 R 需要不大于阈值 $T3$ 。

其中假设之间的距离 D 是假设之中最近的两个笔画之间的距离。T2 按照前面的公式 (1) 计算得到, Factor 取值为 0.5。R 的计算公式如下

$$R=S(H_a\cap H_b)/S(Max(H_a,H_b)) \tag{2}$$

T3 是通过实验得到的经验值, 为 0.5。
在这一步结束之后, 可以得到三种类型的结果: 1) 两端分别与非箭头假设相连的箭头假设, 2) 至少有一端没有与非箭头假设相连的箭头假设, 3) 没有与任何箭头假设相连的非箭头假设。在 1) 中箭头假设与两个非箭头假设相连, 符合流程图的空间结构要求, 所以保留这三个假设, 2) 中的箭头假设与 3) 中的非箭头假设均不符合流程图的空间结构要求, 所以需要将在 2), 3) 中出现的假设从假设集合中剔除掉。在经过进一步优化的假设集合中, 使用以下两个约束来生成候选结果:

- 1) 对于两个符号 S_1, S_2 , 使得 $S_1\cap S_2=\varnothing$ 。
 - 2) 符号 S_1, S_2, \dots, S_n , 使得 $S_1\cup S_2\cup \dots \cup S_n=\text{流程图 } F$ 。
- 这个过程可以抽象为以下问题: 已知集合 Set, 和它的部分子集合 $\{Subset_1, Subset_2, \dots, Subset_n\}$, 要求出可以由这一些子集合组成的所有的集合 Set 的划分。在实验中使用了动态规划算法来解决这一问题。递归公式如下

$$G(S, H)=G(S, H')\bigcup G(S', H')+H_k \tag{3}$$

S 表示的是需要被覆盖到的笔画的集合。H 表示的是用来进行组合的假设的集合。 H_k 表示的是第 K 个假设。 H' 表示的是去掉了元素 H_k 的假设集合。 S' 表示的是去掉了 H_k 中所有笔画的笔画集合。图 2.5 表示的是使用动态规划算法寻找候选结果的具体过程。

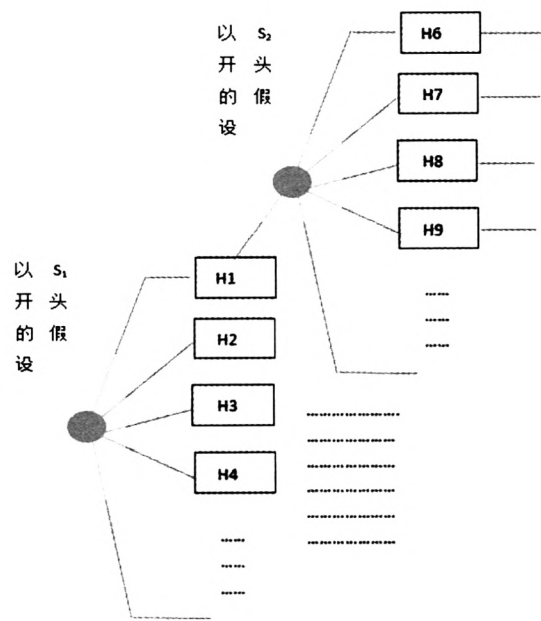


图 2.5 组合过程

最后在所有候选结果中, 寻找到符号平均置信度最大的作为最优解。

实验以及结果

在实验中, 我们使用 FCinkML 数据库作为我们的测试数据库。这个测试数据库中的流程图样本超过 150 个。样本是由 25 为测试用户绘制得到的。表 3.1 中所示是测试数据库的具体统计结果。

表 3.1 测试样本统计

	流程图	符号	笔画
数量	177	2369	5562

在测试数据库的基础上, 我们对分类器进行了测试, 结果如表 3.2 所示
因为箭头在流程图识别中占据了一个比较重要的位置, 所以在训练分类器时, 我们尽可能提高了箭头的识别率, 这导致了其他类型符号的识别率有些偏低。

表 3.2 各符号的识别率

符号	识别率
arrow	84.47%
connector	83.87%
data	67.68%
decision	56.46%
process	77.86%
terminator	53.96%
junk	73.62%

最后的流程图识别结果是从以下两个方面进行的评估：

- 1) 笔画的识别准确度
- 2) 符号的识别准确度

至于流程图空间结构的识别准确度，我们很难给出一个精确的统计结果。因为测试样本中没有包含流程图的空间结构信息。

表 3.3 中显示的是我们从实验中得到的识别结果。我们总共正确识别了 1502 个符号，符号的识别准确度为 63.40%，总共正确识别了 4241 个笔画，笔画的识别准确度为 76.25%。

表 3.3 结果统计

符号	笔画
1502	4241
63.40%	76.25%

符号识别准确率低于笔画识别准确率，是因为只要符号中的任意一个笔画识别错误，我们就认为这个符号识别错误。

结果表明本文方法对于流程图识别来说具有可行性，但是识别的精度却依然不高，因此还不具备较强的实用价值，后续需要继续对该方法进行改善，以提高最终的识别准确度。

总结

本文使用动态规划方法实现流程图的识别。整个识别过程可以分解为三个阶段。在第一阶段，我们将笔画按照一定的约束进行聚合生成符号假设。在第二个阶段，我们使用 Random Forest 分类器来识别这些符号假设。最后一个阶段，我们组合这些假设来得到符合预期的最优解。

最后的结果表明我们的方法可以实现对流程图的识别，但是识别精度依然有很大的提升空间。可以从两方面来提高识别的精度：（1）提高分类器的识别精确度，（2）更加充分的利用流程图的结构信息，而不仅仅用来过滤掉一些不合理的符号假设。同时本次实验中过滤掉了文字笔画，在以后的研究中也需将这些笔画加入到识别样本中。

References:

[1] Awal A M, Feng G, Mouchere H, et al. First experiments on a new online handwritten flowchart database[C]//IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics, 2011: 78740A-78740A-10.

[2] Lemaitre A, Mouchere H, Camillerapp J, et al. Interest of syntactic knowledge for on-line flowchart recognition[M]//Graphics Recognition. New Trends and Challenges. Springer Berlin Heidelberg, 2013: 89-98.

[3] Bresler M, Prua D, Hlavac V. Modeling flowchart structure recognition as a max-sum problem[C]//Document Analysis and Recognition (ICDAR), 2013 12th International Conference on. IEEE, 2013: 1215-1219.

[4] Holmes G, Donkin A, Witten I H. Weka: A machine learning workbench[C]//Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on. IEEE, 1994: 357-361.

[5] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[6] Blagojevic R V. Using data mining for digital ink recognition[D]. ResearchSpace@ Auckland, 2011.

[7] 诸彬, 何骅, 徐从富. 基于草图的程序流程图自动识别翻译系统[J]. 计算机应用与软件, 2010, 27(5): 55-56.

[8] 孙正兴, 冯桂焕, 周若鸿. 基于草图的人机交互技术研究进展[J]. 计算机辅助设计与图形学学报, 2005, 17(9): 1889-189