

基于空间约束分层树模型的彩色图像分割

李建华, 李俊山, 陈霞, 朱秋旭

(第二炮兵工程大学, 陕西 西安 710025)

摘 要: 由于数字图像往往体现出很强的层次特征, 定义一种针对图像像素点的空间约束分层树模型, 对图像进行分割. 通过查询像素点等级, 对分层树进行概率统计, 依照置信度下的等级赋值规则进行剪枝, 得出节点的分类结果, 从而完成图像的分割. 为了加快树的生成, 增加了空间约束条件. 树的剪枝代表了图像的分割过程, 剪枝条件的设置对最后的结果有较大的影响. 实验结果表明, 基于空间约束的分层树生成速度大大提高, 图像分割结果良好.

关键词: 分层树; 聚类; 空间约束; 置信度; 等级赋值规则; 图像分割

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2012)12-0055-04

Color Image Segmentation by Hierarchical Tree Model Targeted Space Constraints

LI Jian-hua, LI Jun-shan, CHEN Xia, ZHU Qiu-xu

(The Second Artillery Engineering University, Xi'an 710025, China)

Abstract: Since digital images often display strong characteristics of hierarchical layouts, a new hierarchical tree model targeted space constraints of pixels is defined to conduct image segmentation. The whole process is accomplished by inquiring levels of pixels, making probabilistic statistics towards the hierarchical tree, and calculating the classification results of nodal points after pruning assignable rules under the condition of confidence. Space constraints are added to accelerate the generation of the Tree. Pruning of the tree represents the segmentation process and thus setup of pruning condition exerts great influence on the final result. As experiments show, hierarchical trees based on space constraints are generated more rapidly with better image segmentation results.

Key words: hierarchical tree; clustering; spatial constraints; confidence; label assignment rules; image Segmentation

1 引言

数字图像中各个颜色区域往往体现出很强的层次特征, 可以利用分类算法^[1-2]对图像进行分割或者压缩编码^[3-4]. 1963 年 Ward^[5]提出了分层聚类的方法, 其主要思想是通过计算数据集的均方差对数据元素重新分组, 以期得若干个子数据集, 使得子数据集的均方差之和最小. 然而采用 Ward 的分组聚类方式需要对整个数据集进行全搜索, 算法复杂度高. 2008 年 Gilman^[6]提出了建立二叉树对数据

集进行分类的方法, 并初步应用在了图像分割上. 2012 年 Devis^[7]提出了一种根据分层树剪枝的图像分割方法, 此方法定义了节点的等级可赋值规则. 由于在置信度区间上的取值范围较小, 出现了同一个节点可被几个像素等级赋值的问题.

针对 Ward 的分层聚类方式要对数据集进行全搜索, 算法复杂度高, 本文提出了基于空间约束条件的分层树生成方法, 降低了计算复杂度, 提高了运行效率; 同时, 弥补了 Devis 提出的置信度区间取值范围太小的不足, 扩大了置信度区间.

2 分组聚类

一个数据集可以在无监督的方式下,根据数据元素之间的相似度分成若干个子数据集.本文介绍了 Ward 提出的使用均方差对数据集进行分组聚类的方法,为分层树的定义奠定基础.

2.1 子数据集的生成

对于给定的数据集 $p = \{e_1, e_2, \dots, e_n\}$, 均方差 ESS 为:

$$ESS = \sum_{i=1}^n x_i^2 - \frac{1}{n} (\sum_{i=1}^n x_i)^2 \quad (1)$$

其中, x_i 为数据集 p 的第 i 个元素.

为了将相同的数据元素聚类到同一个子数据集,定义如下的聚类规则:

Step 1. 计算所有数据对的均方差:

$$ESS_{(e_i, e_j)} = (e_i - e_j)^2;$$

Step 2. 找出均方差最小的数据对 (e_i, e_j) ;

Step 3. 更新数据集 p :

$$p = \{e_1, e_2, \dots, \{e_i, e_j\}, \dots, e_n\};$$

Step 4. 重复 Step 1~3, 到所有子数据集的均方差都为 0 为止.

2.2 子数据集的程序定义

为了便于仿真验证,给出了一种新数据元素的定义方式.当数据集的大小由 n 变为 $n-1$ 时,数据对 (p_{n-1}, q_{n-1}) 生成的新数据元素定义为:

$$S(p_{n-1}, n-1) = [S(p_{n-1}, n) \cup [S(q_{n-1}, n)]] \quad (2)$$

其中, p_{n-1} 是原数据集的数据对中编号较小的数据元素,用来定义新的数据元素; q_{n-1} 是原数据集的数据对中编号较大的数据元素,这个数据元素将不再使用.数据对 (p_{n-1}, q_{n-1}) 的联合值为:

$$Z[p_{n-1}, q_{n-1}, n-1]$$

其中,最右边的参数 $n-1$ 代表了数据集中剩余数据元素的个数.

根据上面的定义,整个数据集的聚类循环可定义为: $Z[i, j, k-1]$ 为数据对 $[S(i, k) \cup [S(j, k)]]$ 的联合值,其中

$$i = 1, 2, \dots, n-1$$

$$i \neq q_{n-1}, q_{n-2}, \dots, q_k$$

$$j = i+1, i+2, \dots, n$$

$$j \neq q_{n-1}, q_{n-2}, \dots, q_k$$

数据集中找出的均方差最小的数据对生成的新数据元素定义为:

$$S(p_{k-1}, k-1) = [S(p_{k-1}, k) \cup [S(q_{k-1}, k)]] \quad (3)$$

联合值为: $Z[p_{k-1}, q_{k-1}, k-1] (p_{k-1} < q_{k-1})$

3 图像分层树的生成

图像可以由各个像素点的像素值或者索引值组成一个数据集 p . 像素值之间的均方差体现了它们的相似度.一般来说,像素值或者索引值均方差较小的像素点相似度较高,可以将它们划分到同一个区域.

在本文算法中对联合值进行了修改,不再使用均方差作为联合值,而是使用第一个数据元素的值作为联合值.因为在图像中使用均方差作为联合值会使很多区域中点杂乱地聚合到一起.仿真验证时曾尝试使用两个数据的均值作为联合值,但这样会使数据集的均方差发生较大变化,导致聚类结果不理想.

3.1 图像像素值的聚合

根据前述定义的分组聚类规则,对于图像像素值组成的数据集 $p = \{e_1, e_2, \dots, e_n\}$ (其中 m 为最大编号),定义如下分层树生成规则:

Step 1. 寻找数组中均方差最小的数据对 $(e_i, e_j) i \neq j$;

Step 2. 为均方差最小的数据对添加父节点,节点编号为 $m+1$;

Step 3. 更新数据集 p , 即删除 e_j , 将 e_i 的值做为新的数据元素值,编号改为 $m+1$;

Step 4. 重复 Step 1~3 直到数组中只有一个数据元素为止.

3.2 分层树的生成

树是由一系列节点 v 组成的.图 1 中给出了一棵分层树的例子.每个节点含一包定数量的叶节点,例如 $n_{13} = 7, n_{12} = 3$. 由于生成的是二叉树,所以节点数量为 $n_v = 2n - 1$ (n 为叶节点数量).

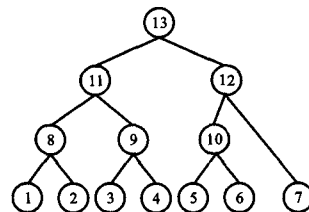


图 1 分层树

3.3 空间约束条件

对于图像中的某个像素点,在整幅图像中可能有若干个像素点与其相似,若要全部找出来,必然要进行全搜索,增加了算法复杂度.为了加快树的生成和降低算法复杂度,对于 p 中元素的两两组合增加空间约束条件,即只寻找当前像素的相邻像素点,计

算它们的相似度.随着算法的进行,像素点的数量在不断减少,原来与该像素点不相邻的相似节点会成为相邻节点,这样不会影响算法的精度,并且提高了算法的效率.

4 树的剪枝过程

根据前面定义的树结构,相似叶节点会聚类到同一个上层节点,因此只要将整棵树剪枝,找到这些节点,就可以完成图像的分割.

4.1 叶节点的标记

为了能够确定当前剪枝节点所处的像素等级(可以是先验的亦可以由用户自定义),要对整棵树进行叶节点标记.

标记过程定义如下:

Step 1 随机选择一个叶节点,设为标记节点,并查询其像素等级;

Step 2 根据树结构,自下而上寻找其父节点,直到寻找到根节点;

Step 3 对于此线路上的所有节点,其叶节点标记数和相应像素等级数加 1.

图 2 给出了一个具体示例:假定选择的随机点为第 2 个数据元素,则向上寻找其父节点.

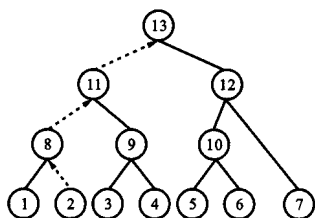


图 2 叶节点标记

4.2 剪枝算法

如何度量一个节点是否可以分割是剪枝的重点.假设一个节点 v 已经有相当数量的节点被标记.当前节点的每个像素等级的相关比例为 $p_{v,l}$ ($l=1, 2, \dots, c$).节点应该被赋予比例最大的像素等级的分割误差为:

$$\epsilon_v = 1 - \max_l \{p_{v,l}\} \quad (4)$$

由于访问节点数量的限制,节点被赋予的像素等级的可信度很低,因此为了更加准确对树进行分割,定义置信度和可赋值准则.

4.3 置信度

依据已得出的相关比例对节点进行像素等级赋值可能会导致误分割,尤其是在标记的像素点的数量比较少的情况下.因此,使用置信度区间来计算当前节点的像素等级的相关比例.置信度上限为 $p_{v,l}^{UB}$,

置信度下限为 $p_{v,l}^{LB}$.

$$p_{v,l}^{UB} = \min(p_{v,l} + \Delta_{v,l}, 1) \quad (5)$$

$$p_{v,l}^{LB} = \max(p_{v,l} - \Delta_{v,l}, 0) \quad (6)$$

其中

$$\Delta_{v,l} \approx (1 - n_v^l/n_v)/n_v + \sqrt{(1 - n_v^l/n_v)p_{v,l}(1 - p_{v,l})/n_v}$$

这里, n_v^l 为当前节点下被标记的叶节点数量.

4.4 可赋值准则

在进行点的抽样的时,可能会出现当前节点有几个可标记像素等级,尤其是抽样点的数量比较少的时候.为了能够准确对树进行剪枝,对当前节点的所有像素等级都要进行计算.当一个像素等级满足式(7)时

$$A_{v,l}(t) = \text{true} \Leftrightarrow 1 - p_{v,l}^{LB} < 2(1 - p_{v,l}^{UB}), \quad \forall l' \neq l \quad (7)$$

则当前节点的像素等级可标记为 l .也就是说,当前节点标记为等级 l 的最大分割误差小于其他任何等级最小分割误差的 2 倍时,当前节点可以赋值为等级 l .

在式(5)和式(6)中,由于 n_v 是一个很大的数值,所以 $\Delta_{v,l}$ 对于置信度区间的影响不是很大,使得当所有 $p_{v,l}^{UB} < 0.5$ 时式(7)恒成立,即节点可以被所有的像素等级赋值.为此,本文修改了 $\Delta_{v,l}$ 的定义公式:

$$\Delta_{v,l} \approx \sqrt{n_v^l/n_v p_{v,l}(1 - p_{v,l})} \quad (8)$$

修改后的置信度区间增大,减少了节点可以被所有的像素等级赋值出现的概率,在查询值较低的情况下出现该问题的概率也很低,基本上可以分割出正确的结果.

分割结果可以用可赋值对 (v, l) 表示,分割误差 $\epsilon_{v,l}$ 为:

$$\epsilon_{v,l} = \begin{cases} 1 - p_{v,l} & \text{if } (v, l) \in A(t) \\ 1 & \text{if } (v, l) \notin A(t) \end{cases} \quad (9)$$

4.5 剪枝

剪枝采用自上而下的方法,即从树的根节点开始.若当前节点不满足式(7)则将该节点删除,将其孩子节点作为当前剪枝,直到没有可剪枝的节点时结束.按照节点所处的像素等级,对节点下的叶节点进行相应像素等级赋值.对于图 1 来说,若根节点 13 的所有像素等级的统计概率较为相近,即 13 是一个像素等级混杂程度较高的节点,应将该节点分割成其子孙节点 $\{10, 12\}$,结果如图 3 所示,继续对节点 10 和 12 进行概率统计.

对于图 1,可能有几种不同的剪枝结果,例如可

以将树分割为 $\{n_{11}, n_{12}\}$ 两个节点的组合,也可以分割为 $\{n_{11}, n_{10}, n_7\}$ 三个节点的组合.但是无论分割结果如何,最后的节点组合都要覆盖所有的叶节点.

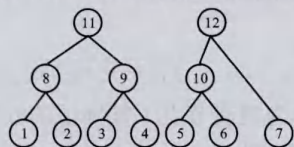


图3 分层树剪枝

4.6 计算复杂度

采用分层树对图像进行分割需要两个步骤:第一是分层树的生成;第二是树的剪枝.本文分别计算了两个步骤的时间复杂度.第一个步骤的时间复杂度与所采用的算法有很大的关系,本文采用的是空间约束的分层聚类方法,未加空间约束的分层聚类方法最高时间复杂度为 $O(n^2 \log n)^{[8]}$,采用空间约束后时间复杂度大大降低,仅为 $O(n \log n)$.而且第一部分的计算只需要一次.第二个步骤的时间复杂度与叶节点的标记和节点像素等级的计数有关.而且随着分割的不断进行,节点数在不断减少,因此第二部分的最高时间复杂度为 $O(2n(\gamma))$,其中 γ 为查询次数.

4.7 实验步骤和结果分析

本实验利用 <http://www.eecs.berkeley.edu/image> 网站的3幅分割测试图像,进行了仿真实验.图像格式为BMP索引图像,每幅图像大小为 200×200 ,共40000个像素点.实验使用的编程环境为Visual Studio 2005 C++,处理器为Pentium(R) Dual-Core Cpu E5200 @ 2.50GHz,内存2GB,64位操作系统.实验的分割结果如图4所示.分层树生成的程序运行时间比较如表1所示.

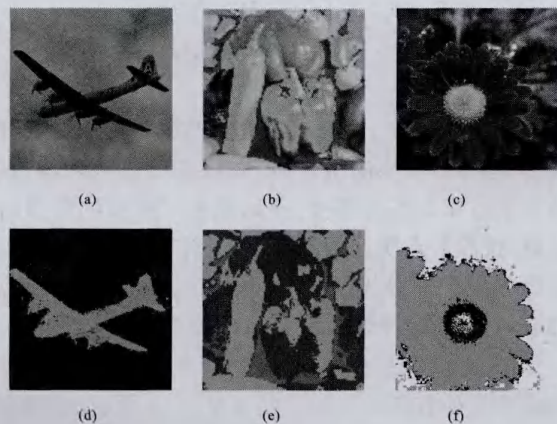


图4 分割结果

表1 分层树生成程序运行时间比较

图片	改进算法/s	原算法/s
(a)	80.5	6002.4
(b)	85.2	6500.5
(c)	87.1	6115.2

由表1可得,生成等级树程序运行时间平均为84 s,比原算法运行速度提高了约80倍.

分割的过程中,改进的置信度区间降低了节点可以被多个像素等级赋值的概率,提高了分割的效率.最终的结果中一般会分割出3~7个节点,分割的效率非常高,时间在毫秒级.从实验结果来看,分割的结果能够较好地描述颜色区域,而且小块的颜色区域不会丢失.

5 结束语

本文的实验结果表明,算法对于颜色区域的分割比较理想,并且降低了分层树生成算法的复杂度,提高了生成效率.值得注意的是,算法对区域较小的颜色比较敏感,这主要是因为改进了可赋值规则, Δ 值比较大,所以会将小的区域分割出来.

进一步研究的问题:(1)如何进一步加快分层树的生成.空间约束的条件的设置是否可以更加合理或者用更为合理的方式对图像的各个点进行聚类.(2)分割主要依赖于对颜色等级的查询,对于材质相同而颜色不同的区域是否可以改进算法,使其聚类到同一个区域.

参考文献:

- [1] Ursani A, Kpalma K. Fusion of textural and spectral information for tree crop and other agricultural cover mapping with very-high resolution satellite images[J]. Applied Earth Observations and Remote Sensing, 2012, 5(1):225-235.
- [2] Panagiotakis C, Grinias I, Tziritas G. Natural image segmentation based on tree equipartition[J]. Bayesian Flooding and Region Merging, 2011, 20(8): 2276-2287.
- [3] 刘国英,王爱民,陈荣元,等.基于小波域TS-MRF模型的监督图像分割方法[J].红外与毫米波学报,2011, 30(1):091-096.
- [4] 陈俊丽,卿定湖,等,万旺根.基于多小波的彩色图像分层树集合分裂算法[J].上海大学学报:自然科学版, 2011, 17(1):39-43.

(下转第63页)

5 结束语

为了提高从单元接口的可复用性,缩短 SoC 的研发周期,本文实现了面向 AHB 总线从单元的可复用接口模块设计,包括协议转换单元、寄存器单元、接收和发送 FIFO、特殊处理单元、同步单元等。各个模块在寄存器单元的控制下协调工作,支持各种数据传输方式,具有很高的灵活性。通过配置该接口可以适应多数 AHB 从单元的要求,真正实现可复用。本文给出了一个面向 AHB 总线从单元 UART 的综合结果,并对其验证环境进行了说明。

本文提出的可复用接口模块已经应用在多个 SoC 芯片中,实现了 CAN、UART、三角函数、I²C 等从单元到 AHB 的集成,均流片成功。实践证明,通过修改编译选项参数及软件配置,该接口模块具有较高的可复用性及结构上的一致性,极大地缩短了 SoC 开发周期。

参考文献:

- [1] 张嶂,权进国,乔飞,等. 面向 SOC 的可配置 AHB 接口组件[J]. 电子与信息报,2008,30(8).
- [2] ARM Inc. AMBA Specification Rev2.0[S]. 1999:3-5.
- [3] Zhangzhen, Zhang Hui. Parameterized bus interface design based on verilog language [C]//International Conference on Computer Application and System Modeling. China Taiyuan, 2010:185-189.
- [4] Duoli Zhang, Gaoming DU. Study on universal bus interface based on sequence configuration [C]//International Conference on Computational Intelligence and Software Engineering. China Wuhan, 2009.

作者简介:

陈庆宇 男,(1988—). 研究方向为 IC 设计.

盛廷义 男,(1974—). 研究员. 研究方向为 SoC 设计、验证.

吴龙胜 男,(1970—). 总工程师、研究员. 研究方向为 IC 设计、SoC 设计.

(上接第 58 页)

- [5] Joe H Ward. Hierarchical grouping to optimize an objective function[J]. Journal of the American Statistical Association, 1963,58(301):236-244.
- [6] Gilman Drive, La Jolla. Hierarchical sampling for active learning[C]//Proceedings of the 25th International Conference on Machine Learning. Finland: Helsinki, 2008,10(9):257-264.
- [7] Devis Tuia, Jordi Munoz-Mari. Remote sensing image segmentation by active queries[J]. Pattern Recogni-

tion, 2012,45(10):2180-2192.

- [8] Aichholzer O, Aurenhammer F. Classifying hyperplanes in hypercubes[J]. SIAM Journal on Discrete Mathematics, 1996,9(2):225-232.

作者简介:

李建华 男,(1988—),硕士研究生. 研究方向为图像处理.