



(12) 发明专利

(10) 授权公告号 CN 102831594 B

(45) 授权公告日 2015.04.01

(21) 申请号 201210129112.8

(22) 申请日 2012.04.27

(30) 优先权数据

13/097, 551 2011.04.29 US

(73) 专利权人 三菱电机株式会社

地址 日本东京都

(72) 发明人 C·O·图兹尔 S·拉姆阿里加姆

刘洛培

(74) 专利代理机构 北京三友知识产权代理有限公司

公司 11127

代理人 李辉 黄纶伟

(51) Int. Cl.

G06T 7/00(2006.01)

(56) 对比文件

US 2006/0147113 A1, 2006.07.06, 全文.

CN 101577003 A, 2009.11.11, 全文.

O. J. Morris 等. Graph theory for image analysis: an approach based on the shortest spanning tree. 《Communications, Radar and Signal Processing》. 1986, 第 133 卷 (第 2 期), 第 146-152 页.

PEDRO F. FELZENSZWALB. Efficient

Graph-Based Image Segmentation.

《International Journal of Computer

Vision》. 2004, 第 59 卷 (第 2 期), 第 167-181 页.

刘陈等. 基于图割与泛形信息的对象分割方法. 《计算机辅助设计与图形学学报》. 2009, 第 21 卷 (第 12 期), 第 1753-1760 页.

审查员 刘洋

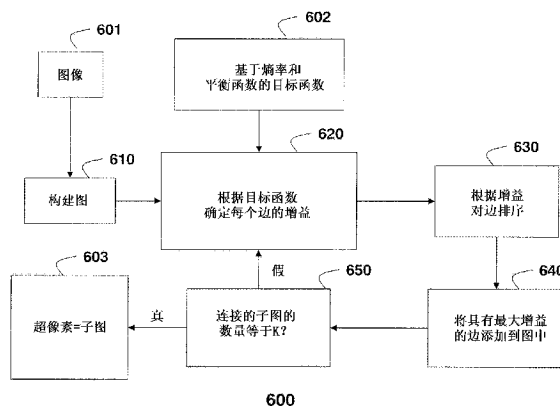
权利要求书1页 说明书8页 附图6页

(54) 发明名称

将图像分割成超像素的方法

(57) 摘要

本发明涉及将图像分割为超像素的方法。该方法通过下述将图像分割为超像素:构建具有被边连接的顶点的图,其中每个顶点对应于图像中的像素,并且每个边与指示对应的像素之间的类似度的加权相关联;选择图中的边的子集以将图分割为子图,其中所述选择基于熵率和平衡项使目标函数最大化。将具有最大增益的边添加到所述图直至子图的数量等于一些阈值为止。



1. 一种将图像分割成超像素的方法,该方法包括以下步骤:

构建具有被边连接起来的顶点的图,其中各个顶点与图像中的像素相对应,并且各个边关联有指示对应的像素之间的相似度的权值;

选择所述图中的边的子集以将所述图分割为子图,其中所述选择步骤将目标函数最大化,其中所述目标函数是子模;以及

将具有最大增益的所述边添加到所述图,直至子图的数量等于某个阈值为止,否则重复所述选择步骤和所述添加步骤;

其中,每个子图包括同质和相似大小的超像素,所述目标函数包含熵率和平衡项,其中,所述熵率用于产生同质的超像素,所述平衡项用于产生相似大小的分割块;

其中,该方法的各个步骤在处理器中执行。

2. 根据权利要求 1 所述的方法,其中,所述熵率是子模的并且单调递增。

3. 根据权利要求 1 所述的方法,其中,所述平衡项是子模的并且单调递增。

4. 根据权利要求 1 所述的方法,其中,所述图为无回路图,所述子图的数量约束是拟阵。

5. 根据权利要求 4 所述的方法,其中,使用受到所述约束的贪心处理将所述目标函数最大化。

6. 根据权利要求 1 所述的方法,其中,将最优性保证为所述目标函数的全局最小值的 $1/2$ 。

7. 根据权利要求 5 所述的方法,其中,使用堆结构实现所述贪心处理。

8. 根据权利要求 1 所述的方法,其中,以层级的方式实现所述分割。

9. 根据权利要求 8 所述的方法,其中,所述层级同时形成所述图像的多个分割。

10. 根据权利要求 1 所述的方法,其中,所述平衡项的平衡参数被自动调节。

11. 根据权利要求 1 所述的方法,其中,所述平衡项的平衡参数被用户修改以定制分割。

12. 根据权利要求 1 所述的方法,其中,所述分割在用户监督下交互地进行。

13. 根据权利要求 1 所述的方法,其中,解决了非图像域中的通用聚类问题。

将图像分割成超像素的方法

技术领域

[0001] 本发明总体上涉及像素分割,更具体地,涉及使用超像素来分割图像的方法。

背景技术

[0002] 总体上,在本领域中,超像素 (superpixel) 是数字图像中的多边形像素簇,其大于单个像素,并且可以按照相同的颜色和亮度呈现,参见美国专利 7,744,185。

[0003] 超像素分割用于对象识别、图像分割和 3D 重建的应用。使用超像素的一个主要优点在于计算效率。当与像素表示相比较时,超像素表示极大地减少了所需要的图像元的数量。

[0004] 例如,在 L- 标签 (L-label) 标签问题中,像素表示的解空间是 L^n , 其中 n 是像素的数量,通常为 10^6 。然而,超像素表示的解空间是 L^m , 其中 m 是超像素的数量,通常为 10^2 。

[0005] 通常假定超像素是来自单个对象的一组像素。这导致了对超像素分割的实际定义,将像素分割为在感觉上一致的簇。感觉一致的属性意味着超像素边界保留对象边界。

[0006] 大多数的聚类处理的特征在于超像素分割。然而,大多数现有处理针对簇的总体方面建模,并且不针对超像素分割来优化。另外,很多处理要求密集计算,并且不适用于分割。

[0007] 一种方法使用基于图的超像素分割。图像被映射到相邻的图。该方法使用边界判定 (boundary predicate) 来顺序地切割边以构建超像素。尽管该方法速度很快,但是产生具有不规则的形状和大小的超像素。

[0008] 均值漂移 (mean-shift) 法对于局部变化是精确的,但是也受到不规则超像素问题的困扰,参见美国专利公开 20100284607。

[0009] 超像素分割的另一种方法是 NCut,参见美国专利公开 20110013837。NCut 产生具有类似大小和紧凑形状的超像素。然而,计算上也很昂贵,即使针对例如 481×321 个像素这样的中等大小的图像都需要数分钟。

[0010] TurboPixel 是实现类似规则性的有效替代方式。TurboPixel 基于从均匀布置在图像中的种子的演化曲线。该方法在曲线演化期间使用多种约束,以增强超像素规则性。

[0011] 可以使用图切割 (graph cut) 通过密集补片分配 (dense patch assignment) 技术来实现规则的超像素。在另一种方法中,使用用于限定切割成本的概率边界映射图,超像素遵循规则的网格。该处使用的目标允许图像之间的同质。

发明内容

[0012] 本发明的实施方式提供一种使用聚类目标函数将图像分割为超像素的方法。目标函数包括两个成分:随机游走的熵率;以及平衡函数。所述熵率形成紧凑和同质的簇,而平衡函数产生具有相似大小的簇。

[0013] 为了进行聚类,构建了数据点和两两相似性分别对应于顶点和边的权值。通过最大化受到拟阵约束的目标函数来分割图。

[0014] 我们用贪心处理来解目标函数,并且利用目标函数的子模性和单调性来证实恒定的近似界 (approximation bound)。

附图说明

[0015] 图 1 是根据本发明的一个实施方式的在簇形成过程中没有选择边的情况的权值的示例的示意图；

[0016] 图 2 是根据本发明的一个实施方式的随机游走的熵率的 (A)-(D) 的示意图；

[0017] 图 3 是根据本发明的一个实施方式的在获得相似大小的簇的过程中平衡函数的示例 (A)-(B) 的示意图；

[0018] 图 4 是示出根据本发明的一个实施方式的针对解的算法的示意图；

[0019] 图 5 是示出根据本发明的一个实施方式的在分割过程中产生超像素层级的聚集本质的示意图；以及

[0020] 图 6 是示出根据本发明的一个实施方式的基本步骤的示意图。

具体实施方式

[0021] 图表示

[0022] 针对表示图像中的像素的无向图,使用常规的记法 $G = (V, E)$, 其中 V 是与像素相对应的顶点的集合, E 是边的集合。 V 的第 i 个顶点表示为 v_i , 并且连接定点 v_i 和 v_j 的边表示为 $e_{i,j}$ 。加权函数 w 给出由一边连接起来的两个顶点之间的相似性。在无向图中, 边的下标是可交换的, $e_{i,j} = e_{j,i}$, 并且边的权是对称的, $w_{i,j} = w_{j,i}$ 。

[0023] 图分割

[0024] 图分割 S 是指将集合 V 分割为不相交的子集 $S = \{S_1, S_2, \dots, S_K\}$, 使得 $S_i \cap S_j = \emptyset$ ($i \neq j$), 并且 $\cup_i S_i = V$ 。图分割是子集选择问题。我们的目的是选择集合 E 中的边 A 的子集, 使得得到的图 (V, A) 具有 K 个连接的子图 (连接部分)。每个得到的子图对应于超像素。

[0025] 熵

[0026] 熵 H 测量随机变量的不确定性。具有条件概率质量函数 p_x 的离散随机变量 X 的熵可以被确定为

[0027]

$$H(X) = - \sum_{x \in \mathcal{X}} p_X(x) \log p_X(x) \quad (1)$$

[0028] 其中 x 是随机变量 X 的支撑集。只要相关的随机变量 Y 的值已知, 条件熵 $H(X|Y)$ 就量化随机变量 X 的剩余不确定性。其被定义为

$$H(X|Y) = \sum_{y \in \mathcal{Y}} p_Y(y) H(X|Y = y) = \quad (2)$$

[0030]

$$- \sum_{y \in \mathcal{Y}} p_Y(y) \sum_{x \in \mathcal{X}} p_{X|Y}(x|y) \log p_{X|Y}(x|y)$$

[0031] 其中 Y 是 Y 的支撑集, 并且 $p_{X|Y}$ 是条件概率质量函数。

[0032] 熵率

[0033] 熵率量化了随机过程 $X = \{X_t, t \in T\}$ 的不确定性, 其中 T 是指标集。针对离散随机过程, 熵率被定义为渐进测量值。

[0034]

$$\mathcal{H}(X) = \lim_{t \rightarrow \infty} H(X_t | X_{t-1}, X_{t-2}, \dots, X_1). \quad (3)$$

[0035] 在观察先前的经过轨迹之后, 熵率测量剩余随机过程的不确定性。针对平稳随机过程, 式 3 中的极限总是存在。在平稳一阶马尔可夫过程的情况下, 熵率具有以下形式

[0036]

$$\mathcal{H}(X) = \lim_{t \rightarrow \infty} H(X_t | X_{t-1}) =$$

$$\lim_{t \rightarrow \infty} H(X_2 | X_1) = H(X_2 | X_1).$$

[0038] 第一个等式是由于一阶马尔可夫属性, 第二个等式是由于平稳性的结果。我们省略了极限, 因为 $H(X_2 | X_1)$ 是独立于时间的。

[0039] 随机游走

[0040] 随机游走是图上的随机过程。令 $X = \{X_t, t \in T, X_t \in V\}$ 为图 $G = (V, E)$ 上的具有非负相似性测量值 w 的随机游走。随机游走 X 是由从顶点到顶点的连续随机跳跃构成的轨迹。我们使用传统构造。从顶点 v_i 到 v_j 的转移概率与相关联的边的权成正比, 并且被定义为

[0041]

$$p_{i,j} = \Pr(X_{t+1} = v_j | X_t = v_i) = \frac{w_{i,j}}{w_i} \quad (4)$$

[0042] 其中

[0043]

$$w_i = \sum_{k: e_{i,k} \in E} w_{i,k}$$

[0044] 是 v_i 的关联权值。

[0045] 平稳分布是

[0046]

$$\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_{|V|})' = \left(\frac{w_1}{w_T}, \frac{w_2}{w_T}, \dots, \frac{w_{|V|}}{w_T} \right)' \quad (5)$$

[0047] 其中

[0048]

$$w_T = \sum_{i=1}^{|V|} w_i$$

[0049] 是归一化常数。

[0050] 在特定顶点上随机游走的概率与该顶点上的总的关联权值成正比。针对断开的图, 平稳分布不是唯一的。然而, 式 5 中的 $\boldsymbol{\mu}$ 总是平稳分布。可以通过 $\boldsymbol{\mu} = P^T \boldsymbol{\mu}$ 来验证,

其中 $P = [p]_{i,j}$ 是转移矩阵。可以通过应用式 2 来确定随机游走的熵率。

[0051]

$$\mathcal{H}(X) = H(X_2|X_1) = \sum_i \mu_i H(X_2|X_1 = v_i)$$

[0052]

$$= - \sum_i \mu_i \sum_j p_{i,j} \log p_{i,j} = - \sum_i \frac{w_i}{w_T} \sum_j \frac{w_{i,j}}{w_i} \log \frac{w_{i,j}}{w_i}$$

[0053]

$$= - \sum_i \sum_j \frac{w_{i,j}}{w_T} \log \frac{w_{i,j}}{w_T} + \sum_i \frac{w_i}{w_T} \log \frac{w_i}{w_T}$$

(6)

[0054] 子模性 (submodularity)

[0055] 我们使用子模函数定义。令 E 是有限集合。如果对于全部的 $A \subseteq E, a_1, a_2 \in E$ 并且 $a_1, a_2 \notin A$,

[0056] $F(A \cup \{a_1\}) - F(A) \geq F(A \cup \{a_1, a_2\}) - F(A \cup \{a_2\})$ (7), 则集合函数 F 是子模。

[0057] 该属性具有收益递减属性的别名, 即如果在以后阶段使用, 则对模的影响较小。

[0058] 严格单调递增集合函数:

[0059] 如果对于全部的 $A_1 \subseteq A_2$,

[0060] $|F(A_1) < F(A_2)| F(A_1) \leq F(A_2)$, 则该集合函数 F 是严格单调递增。

[0061] 拟阵

[0062] 拟阵是有序对 $M = (E, \mathcal{I})$, 其包括有限集合 E 和满足以下三个条件的集合 E 的子集的集合 $\mathcal{I}$:

[0063]

$$\emptyset \in \mathcal{I}.$$

[0064] 如果 $I \in \mathcal{I}$ 并且 $I' \subseteq I$, 则 $I' \in \mathcal{I}$.

[0065] 如果 I_1 和 I_2 在 $\mathcal{I}$ 中并且 $|I_1| < |I_2|$, 则存在 $I_2 - I_1$ 的元素 e , 使得 $I_1 \cup e \in \mathcal{I}$.

[0066] 我们的目标函数单调递增的并且是子模。为了显示这些, 描述熵率和熵的等同集合函数。最大化单调递增子模函数是 NP-hard 问题。然而, 我们使用拟阵表示来阐明图分割问题并且证明简单的贪心过程产生 1/2 近似界。

[0067] 熵率聚类

[0068] 图构建

[0069] 我们从图像构建连接图 $G = (V, E)$, 其中顶点表示像素, 并且边的权值表示在相似性矩阵中给出的两两相似性。我们还假设图的每个顶点具有零权值的自环 (self-loop)。当选择边的子集以分割图时, 未被选择的边的权值被分配回得到的图中。环路被默认地选择, 并且不在集合 E 中。针对未被选择的每个边, 我们按照使得每个顶点的总关联权值保持恒定的方式来增加关联的顶点的环路的权值。

[0070] 如图 1 所示, 如果在簇形成过程中没有选择边 $e_{i,j}$, 则对应的权值 $w_{i,j}$ 被重新分

配到两个顶点的环路 102。

[0071] 该构造保持式 5 不变。这对于顺序地选择边的迭代处理很重要。在这种图构造中，针对转移概率 $p_{i,j}$ 的等价集合函数

[0072]

$$p_{i,j}(A) = \begin{cases} \frac{w_{i,j}}{w_i} & \text{如果 } i \neq j \text{ \& } e_{i,j} \in A, \\ 0 & \text{如果 } i \neq j \text{ \& } e_{i,j} \notin A, \\ 1 - \frac{\sum_{j: e_{i,j} \in A} w_{i,j}}{w_i} & \text{如果 } i = j. \end{cases} \quad (8)$$

[0073] 其中 A 是为了进行分割而选择的边的集合。因此，随机游走的熵率具有等价集合函数

[0074]

$$\mathcal{H}(A) = - \sum_i \mu_i \sum_j p_{i,j}(A) \log(p_{i,j}(A)) \quad (9)$$

[0075] 熵率

[0076] 如图 2 的 (A)-(D) 所示，我们使用相关联的图上的随机游走的熵率作为判据，以获得同质和紧凑的簇。图 2 的 (A)-(B) 对应于一个数据集，图 2 的 (C)-(D) 对应于另一个数据集。每个顶点具有环路，没有示出这些环路。图 2 的 (A) 中的紧凑簇的熵率比图 2 的 (B) 的较不紧凑的簇具有更高的目标值。图 2 的 (C) 中的紧凑簇的熵率比图 2 的 (D) 的较不同质的簇具有更高的目标值。

[0077] 我们使用高斯内核将距离转换为相似度。在这些图分割的每一个中，我们得到显示为相连部分的四个不同的簇。同质和紧凑的簇产生较大的熵率。尽管对每个边的选择将熵率增加一些值，但是当我们从紧凑且同质的簇中选择边时，增加得更大。

[0078] 在所提出的图构造的情况下的随机游走的熵率是单调递增子模函数。由于边的增加提高了不确定性，所以熵率单调递增。收益递减属性来自于以下事实，即，选择边而导致的不确定性的增加在后面的阶段较小，这是因为与更多个边共享。

[0079] 因而，在我们的图构造下，图 $H: 2^E \rightarrow \Re$ 上的随机游走是单调递增子模函数。

[0080] 平衡函数

[0081] 我们对鼓励具有类似大小的簇的平衡函数进行描述。令 A 为所选择的边的集合， N_A 是图中连接的子图的数量，并且 Z_A 是簇成员的分布。对于分割 $\{S_1, S_2, \dots, S_{N_A}\}$ ，分布 Z_A 等于

[0082]

$$p_{Z_A}(i) = \frac{|S_i|}{|V|}, i = \{1, \dots, N_A\}, \quad (10)$$

[0083] 并且平衡项由下式给出

[0084]

$$B(A) \equiv H(Z_A) - N_A \quad (11)$$

[0085]

$$= - \sum_i p_{Z_A}(i) \log(p_{Z_A}(i)) - N_A. \quad (12)$$

[0086] 熵 $H(Z_A)$ 支持具有相似大小的簇；而 N_A 支持较少数量的簇。

[0087] 图 3 的 (A)-(B) 示出平衡函数在获得相似大小的簇的过程中的角色。连接的部分示出了数据集中的不同的簇。与图 3 的 (B) 中的较不平衡的平衡聚类相比，针对图 3 的 (A) 中的平衡聚类，平衡函数具有更高的目标值。

[0088] 类似于熵率，平衡函数也是单调递增子模函数。

[0089] 我们将熵率与平衡函数组合，并且解决子集选择问题

[0090]

$$\max_{A \subseteq E: N(A) \geq K} \mathcal{H}(A) + \lambda \mathcal{B}(A) \quad (13)$$

[0091] 以确定紧凑、同质和平衡的簇。式 3 中的参数 $\lambda > 0$ 是平衡加权，其用于控制对平衡聚类的优选。

[0092] 基于以上描述，目标函数

[0093]

$$\mathcal{F}(A) \equiv \mathcal{H}(A) + \lambda \mathcal{B}(A) \quad (14)$$

[0094] 是单调递增子模函数。因为簇的组合导致了平衡项中的增益，所以确保在全部的局部最佳解处恰好存在 K 个簇。

[0095] 贪心启发

[0096] 用于最大化子模集合函数的一种处理使用贪心算法。该处理开始于空集（完全断开的图， $A = \emptyset$ ），并且将边顺序地添加到该集合。在每次迭代时，该处理添加产生最大增益的边。当连接的子图的数量达到预定数量 $N_A = K$ 时，迭代停止。

[0097] 为了实现额外的加速，我们对边的集合 A 应用额外的约束，使得边的集合不能包括环。该约束立即忽略了连接的子图中的额外的边并且减少了贪心搜索中的评估的数量。这些边不改变对图的分割。尽管与原始问题相比，该约束导致了较小的解空间（仅仅树结构子图被允许），但在实践中聚类结果非常相似。

[0098] 这个无环约束与簇数约束 $N_A \geq K$ 一起导致独立的集合定义，其导出了拟阵 $M = (E, I)$ 。如果边 E 是边的集合，并且 E 的子集 A （边的集合 A 是无环的）的集合是 I ，并且构成具有大于或者等于 K 的连接部分的图分割，则对 $M = (E, I)$ 是拟阵。

[0099] 图分割块的再形成问题是

[0100]

$$\arg \max_{A \in I} \mathcal{H}(A) + \lambda \mathcal{B}(A). \quad (15)$$

[0101] 并且在图 4 中示出解的算法。

[0102] 因为贪心处理针对受到拟阵约束的非递减子模函数的最大化给出了 $1/2$ 近似，所以图 4 所示的处理示出了用于解式 15 的 $1/2$ 近似。

[0103] 有效实现

[0104] 首先，我们计算向 A 添加各个边的增益，并且构成堆。在每次迭代时，将具有最大增益的边从堆中去除并将其包括到 A 中。将堆中包含该边影响了其余边中的一些的增益。

因此,需要将堆更新。然而,子模属性允许有效地更新堆结构。关键之处在于在整个过程中,各个边的增益绝对不能增加,即,收益递减属性。因此,足以保持堆,其中更新顶部元素的增益但是不必须更新其它元素的增益。应用堆的顶部元素被更新而其它元素的值仅可以减小,所以顶部元素是最大值。

[0105] 在实践中,处理比简单实现运行得快的多。平均而言,每次迭代时针对堆执行很少的更新。在我们的实验中,对图像大小 481×321 ,提供 200 至 300 的加速系数。

[0106] 图 6 示出本发明的一个实施方式的基本步骤。输入图像 601。我们构建 610 具有被边连接起来的顶点的图,其中每个顶点与图像中的像素相对应,并且每个边与指示对应的像素之间的相似度的权值相关联。初始地,图不包括任何边。使用顺序地向图添加边的贪心处理将目标函数最大化。

[0107] 我们基于包括熵率和平衡项的目标函数 602 来确定 620 向图添加各个边的增益。

[0108] 我们根据增益对边进行排序 630,并且将具有最大增益的边添加 640 到图。

[0109] 我们确定连接的子图的数量是否等于某个阈值 K 。如果不成立,则重复步骤 620、630 和 640。如果成立,则我们具有期望的超像素 603。

[0110] 方法 600 的步骤可以在与本领域中已知的存储器和输入输出接口相连接的处理器上执行。

[0111] 自动参数调节

[0112] 我们描述自动调整平衡参数 λ 的方法。给定初始的用户规定值 λ' ,基于以下调整最终平衡参数 λ :(1) 超像素的数量和(2)从输入的图像计算得出的数据相关动态参数 β 。当要求大量的超像素时,簇数 K 被引入以更加强调平衡项。数据相关项由在将单个边包括在图中时最大熵率增加与最大平衡项增加的比

[0113]

$$\beta = \frac{\max_{e_{i,j}} \mathcal{H}(e_{i,j}) - \mathcal{H}(\emptyset)}{\max_{e_{i,j}} \mathcal{B}(e_{i,j}) - \mathcal{B}(\emptyset)}$$

[0114] 给出,并且补偿目标函数中两个项之间的大小差异。最终平衡参数由 $\lambda = \beta K \lambda'$ 给出。

[0115] 超像素层级

[0116] 分割处理开始于将各个像素作为单个簇,并且逐渐地将簇组合以构建更大的超像素。该分割层级同时产生了图像的多个分割。基于该应用,可以选择分割中的正确的超像素数量。图 5 示出在分割过程中产生超像素层级的聚集本质。

[0117] 这种层级对于很多视觉应用是有用的,诸如交互式编辑或利用来自多个超像素分割的信息的算法。在完成初始分割后,用户可以选择特定的超像素并且基于层级进一步合并或分裂这些超像素。这对于诸如其中超像素可能包含多个器官并且超像素的进一步组合/分割产生了器官的分离的医学分割这样的交互式分析很重要。

[0118] 通用聚类

[0119] 尽管针对图像分割描述了本发明,但是该方法适用于任何聚类问题。在给出了图像的一组点而非像素的情况下,可以基于距离度量来定义这些点之间的相似度。相同的图构造接着应用于此一般点的集合,其中点是图的顶点,并且边将点连接到该点的 L 个最近

的相邻点。所述算法产生此点集合的聚类。

[0120] 通用聚类的领域包括但是不限于数据挖掘、金融、生物信息学、医学、神经科学等。

[0121] 发明效果

[0122] 本发明提供了一种用于产生超像素的新颖的聚类目标。该聚类目标是基于对图上的随机游走的熵率和平衡函数的组合的最大化。我们使用目标函数的子模性及拟阵表示来解决使用贪心函数将目标函数最大化的问题。

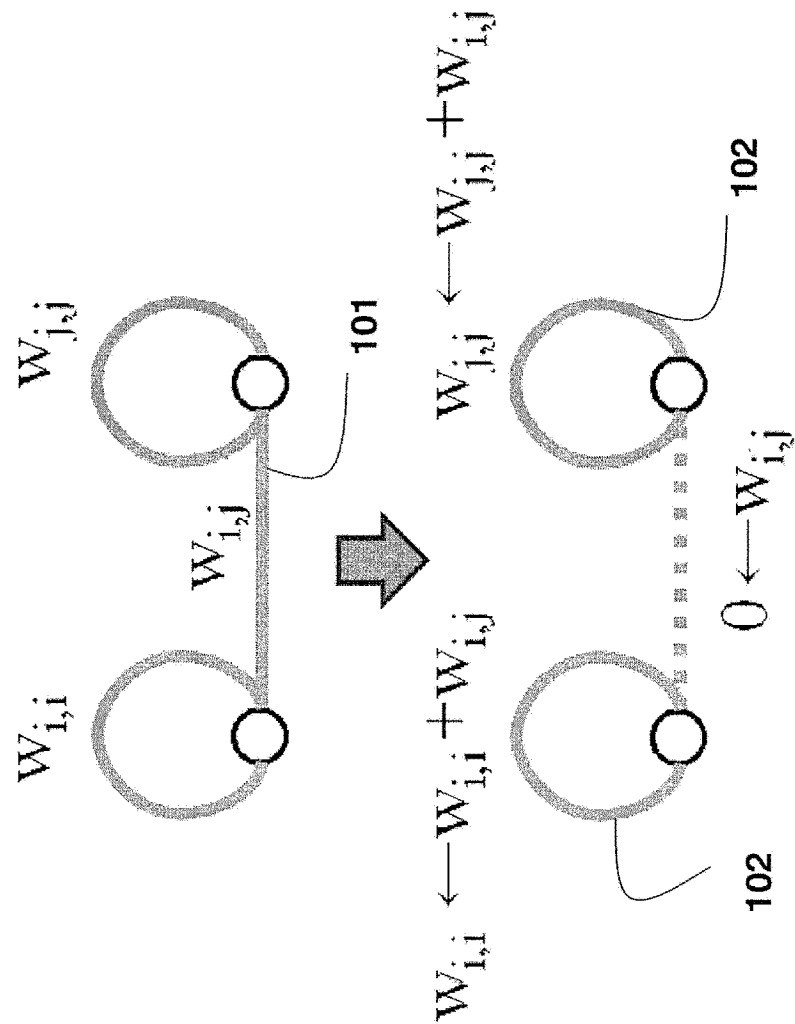


图 1

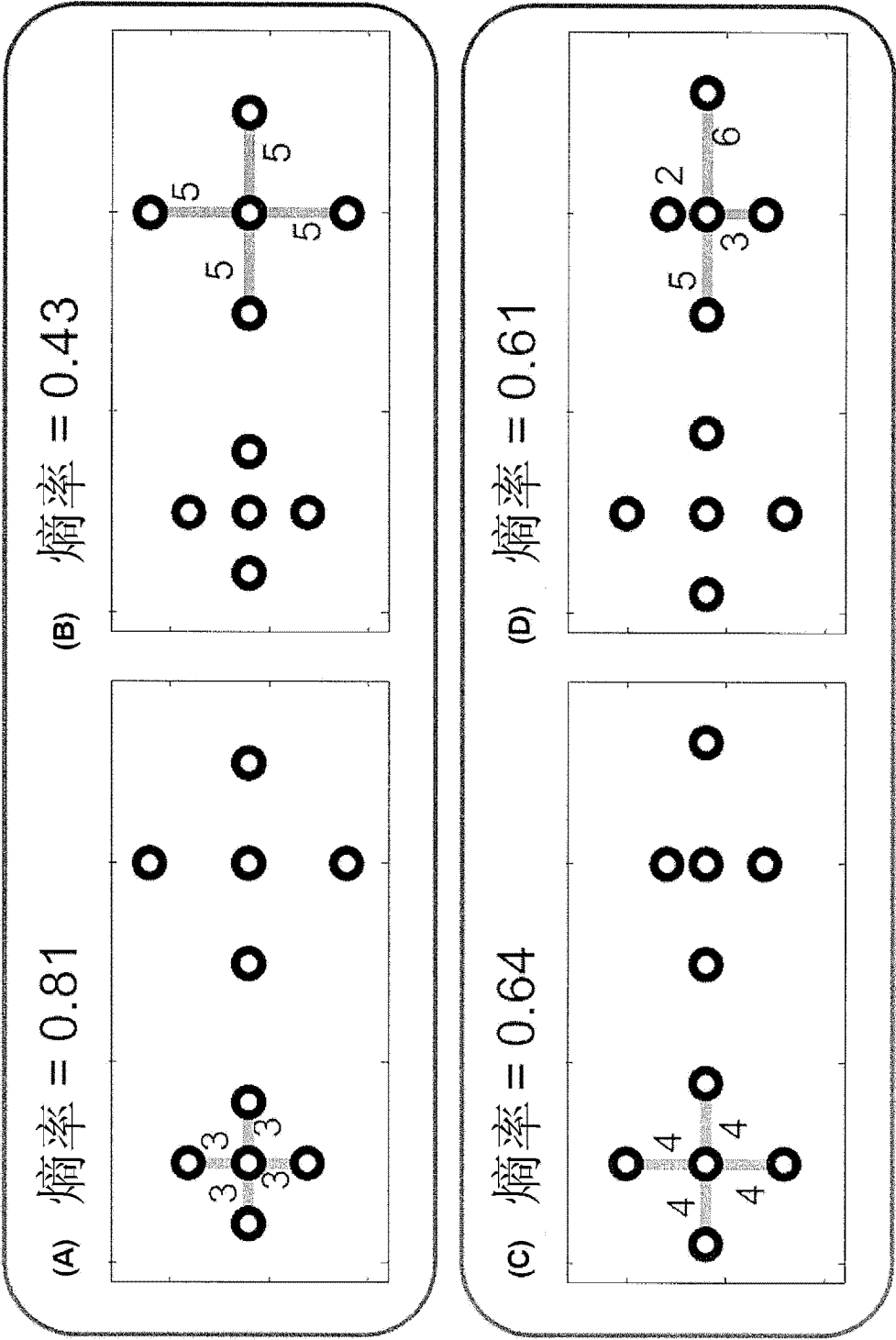


图 2

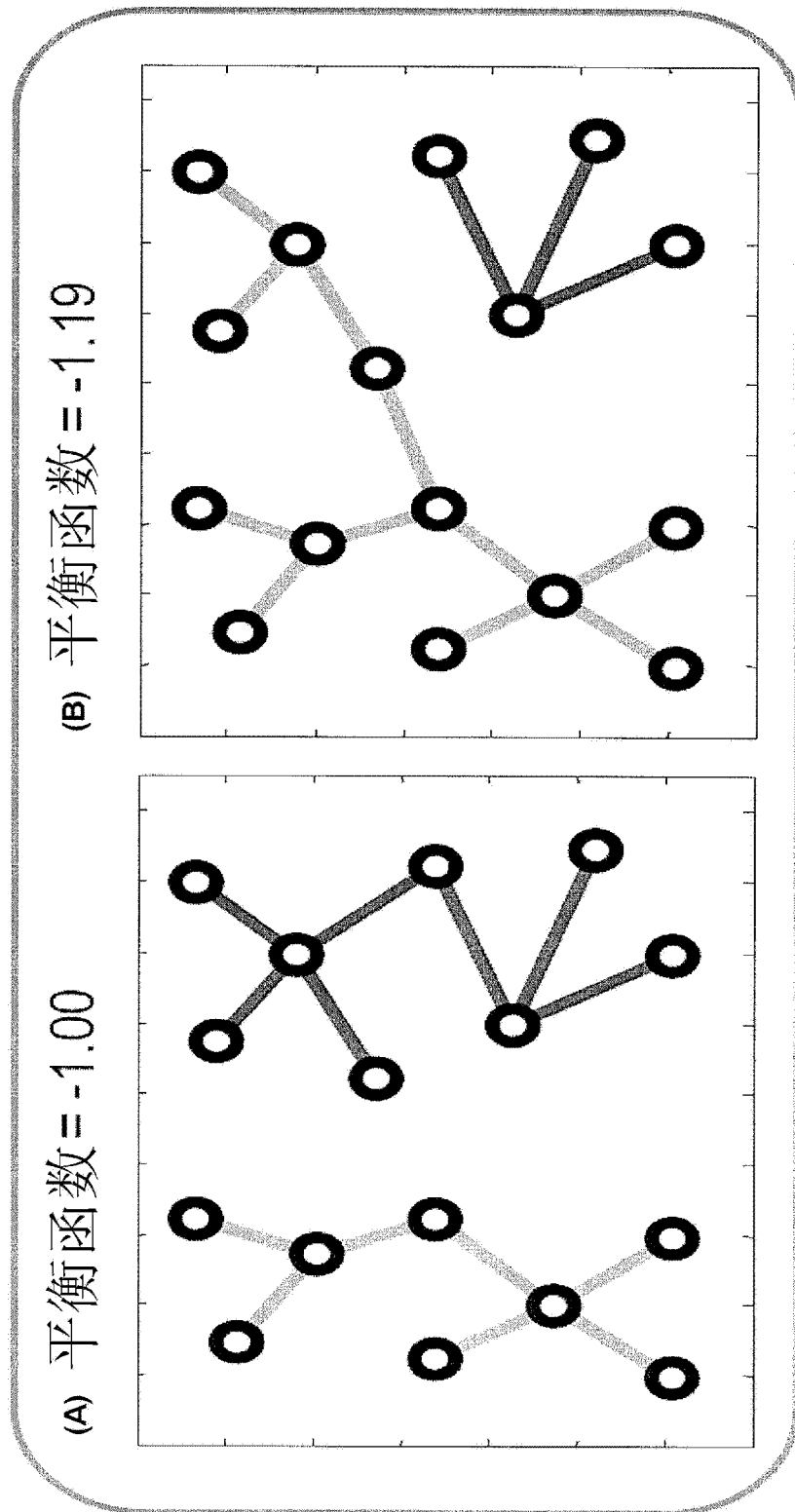


图 3

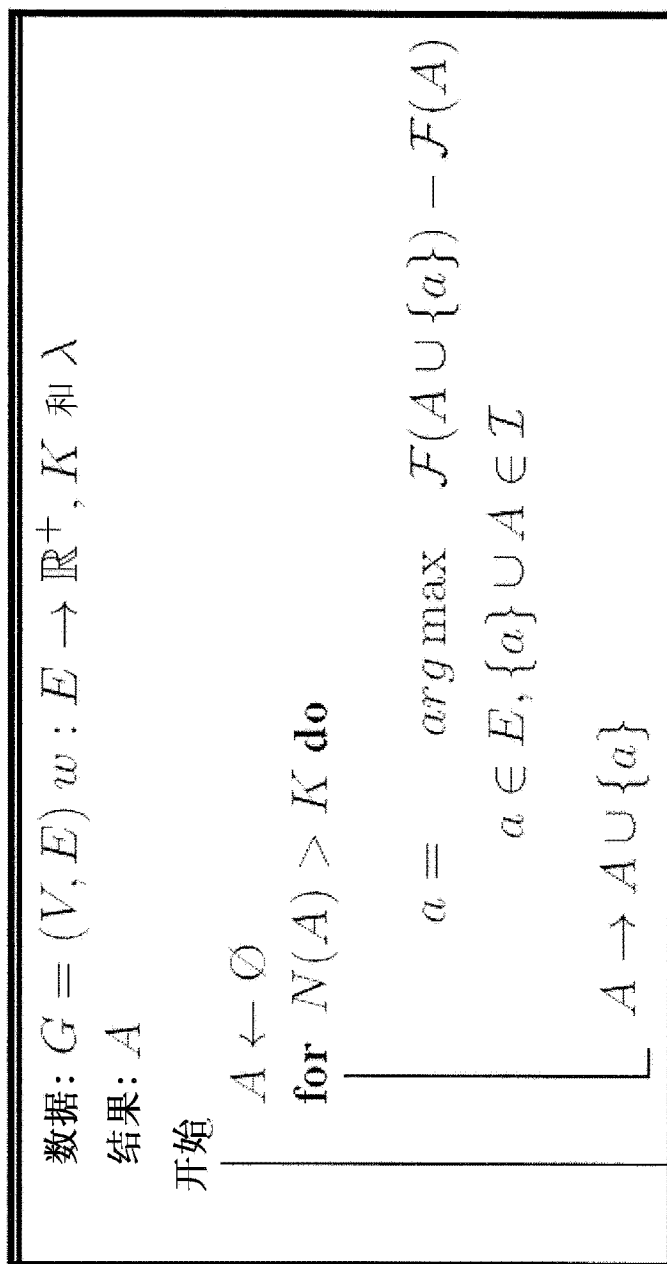


图 4

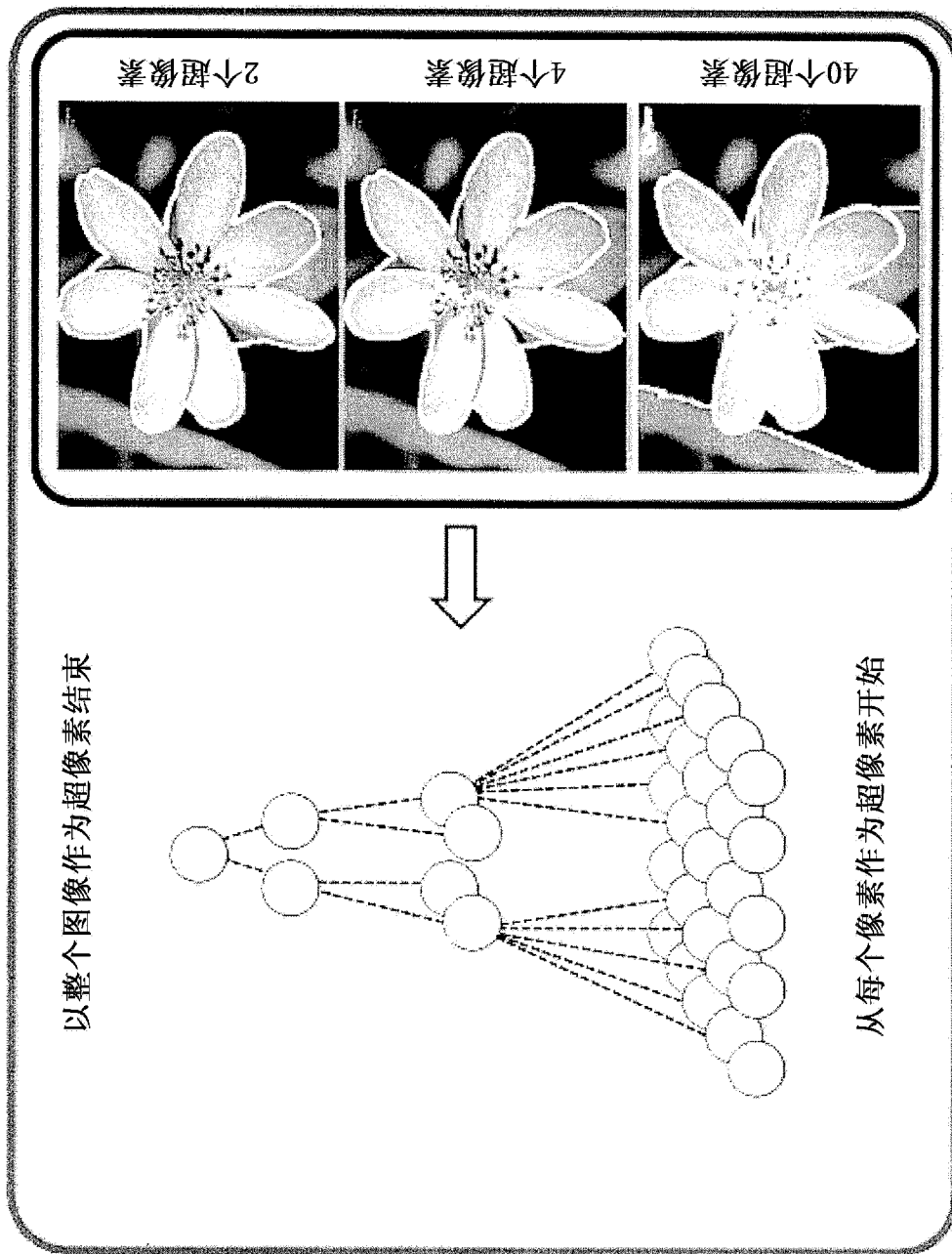


图 5

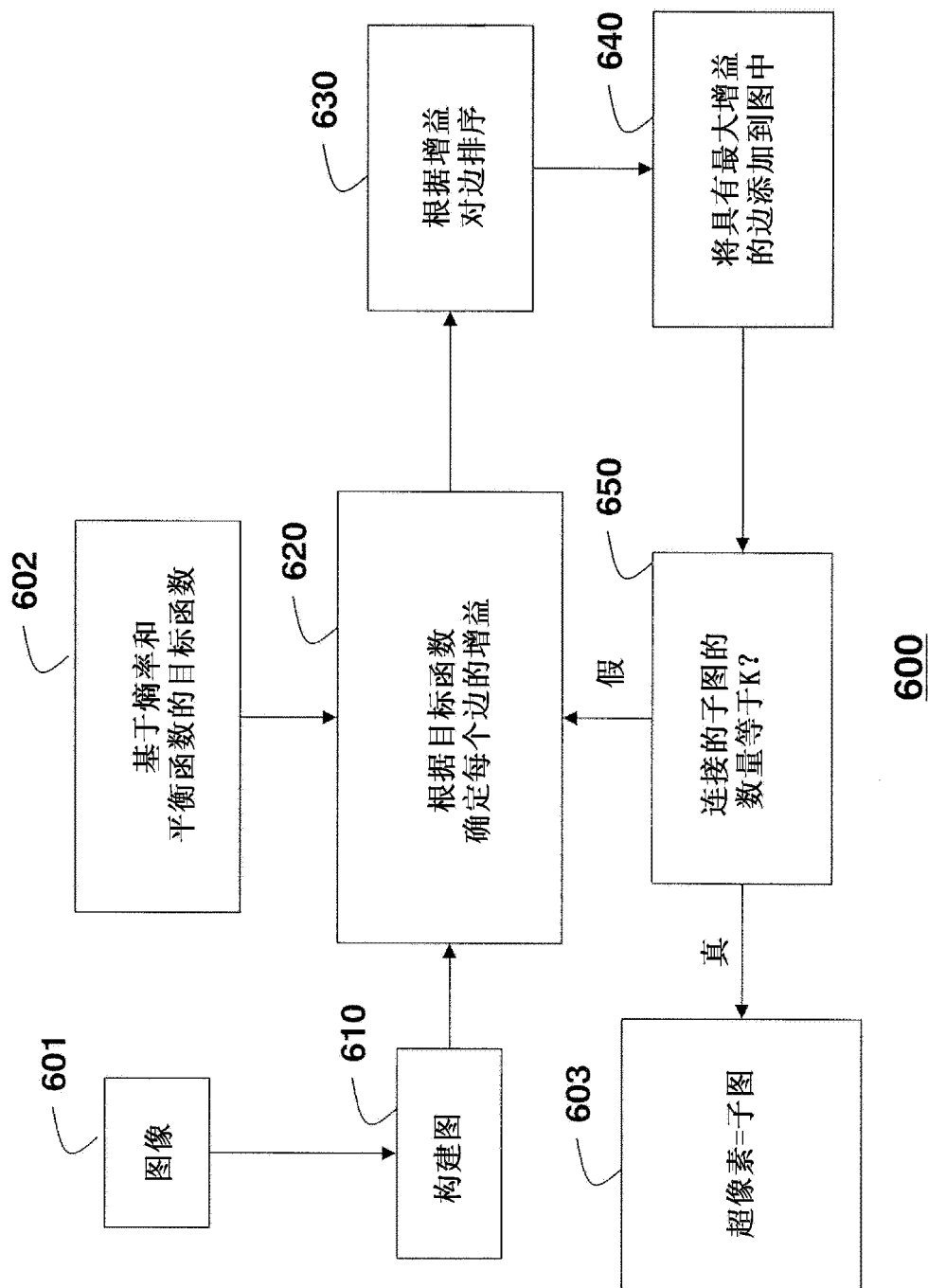


图 6