

海量层次信息的Focus+Context交互式可视化技术^{*}

任磊⁺, 王威信, 滕东兴, 马翠霞, 戴国忠, 王宏安

(中国科学院 软件研究所 人机交互技术与智能信息处理实验室, 北京 100190)

Focus+Context Technique for Interactive Visualization of Large Hierarchies

REN Lei⁺, WANG Wei-Xin, TENG Dong-Xing, MA Cui-Xia, DAI Guo-Zhong, WANG Hong-An

(Laboratory of Human-Computer Interaction and Intelligent Information Processing, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

+ Corresponding author: E-mail: renlei05@ios.cn

Ren L, Wang WX, Teng DX, Ma CX, Dai GZ, Wang HA. Focus+Context technique for interactive visualization of large hierarchies. *Journal of Software*, 2008,19(11):3073-3082. <http://www.jos.org.cn/1000-9825/19/3073.htm>

Abstract: A Focus+Context technique based on circle packing is presented for interactive visualization of large hierarchical data. First, a triangular mesh generating method based on externally tangent circles packing is proposed to establish context. Then a method for triangular mesh neighbors tracking is introduced. To realize the context consistency for Focus+Context technique, a layout method for the distorted brother nodes of the focus node is presented based on triangular mesh neighbors tracking, and a recursive method for the children nodes of the focus node is described. Experimental results show that this method can realize fisheye view and can achieve focus and context consistency. The method has been applied to the interactive visualization of file systems, and an interactive visualization tool is introduced.

Key words: HCI (human computer interaction); user interface; information visualization; Focus+Context; triangular mesh; context-aware; file system

摘要: 综述了海量层次信息可视化与 Focus+Context 技术的相关工作,针对海量层次信息可视化的交互问题,在嵌套圆可视化技术的基础上提出了基于上下文感知的 Focus+Context 交互式可视化技术.首先,基于外切圆排列方法提出对圆心进行三角网格剖分的方法,为变形计算建立上下文;然后,针对变形计算前后上下文一致性问题,在三角网格邻居跟踪方法的基础上,提出了用于同层兄弟节点上下文感知的外切圆变形排列方法,以及用于父子节点上下文感知的嵌套圆迭代排列方法.实验结果表明,上述方法在实现焦点突出的鱼眼视图的同时,能够有效地解决 Focus+Context 交互式可视化的上下文感知问题.上述方法应用于文件系统海量层次信息的交互式可视化问题,提供了交互式可视化工具.

^{*} Supported by the National Natural Science Foundation of China under Grant Nos.60673195, 60703078 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2007AA04Z113 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant No.2006CB303105 (国家重点基础研究发展规划(973)); the National Key Technology R&D Program of China under Grant No.2006BAF01A17 (国家科技支撑计划)

Received 2007-07-04; Accepted 2007-11-20

关键词: 人机交互;用户界面;信息可视化;Focus+Context;三角网格;上下文感知;文件系统

中图法分类号: TP391 文献标识码: A

随着计算技术、存储技术及网络技术的发展,海量信息得以存在并迅猛增长.按层次结构(通常也称这树型结构)组织数据并进行可视化表示是在有限空间内呈现海量信息的一种最佳方式.层次结构是数据存储、组织和管理的一种重要的数据结构,如 XML 数据、计算机文件系统、图书目录、物种分类等均采用此结构.层次信息可视化是信息可视化中的一个研究热点领域^[1].

层次信息可视化的方法主要分为两大类:节点连接法和区域填充法^[2].节点连接法是在欧拉空间或双曲空间内用直线或曲线段连接父子节点的方法,如 ConeTrees^[3],HyperbolicTrees^[4],DOITrees^[5],SpaceTree^[6]等. ConeTrees 的根节点位于三维空间中的顶部,子节点连接在其父节点的下方,应用于 Unix 文件系统,能够在在一个界面内显示 600 个文件夹、1 万个文件,其中有许多节点被覆盖.HyperbolicTrees 在一个圆形显示区域内用双曲线连接父子节点,在 600×600 像素的空间内能够显示 1 000 个节点.DOITrees 采用节点逐级展开的交互方式,能够在有限的界面内探索 60 万个节点信息.SpaceTree 采用动态缩放节点交互技术优化树节点的显示空间.节点连接法能够清晰地表示树的层次结构关系,但空间利用率较低,难以在有限界面内提供大量信息的整体视图^[7].区域填充法分为矩形区域填充法和径向区域填充法.Treemap^[7]及其改进方法^[7-12]是典型的矩形区域填充法.Treemap 把一个矩形显示区域划分为多个子区域表示其子节点,但容易产生狭长矩形,不利于交互操作.Bederson 等人^[10]对其加以改进,用接近正方形的矩形代替狭长矩形.Itoh 等人^[11]提出了快速排列矩形区域的算法.Balzer 等人^[12]采用 Voronoi 图描述层次信息.矩形区域填充法的空间利用率较高,能够在有限的空间内呈现大量信息的整体视图,但绝大多数空间用来显示叶节点,树的分枝及层次结构不够清晰^[2].径向区域填充法^[13-15]采用径向排列的多层圆环表示层次信息,圆环被划分为多个扇形区域,每一扇形表示一个节点,外层圆环的部分扇形是对应内层圆环扇形的子节点.径向区域填充法能够在有限的空间内提供大量信息的整体视图,层次结构关系也比较清晰,但是,随着子节点数目的增多,很多外层扇形被排列得十分狭长,不易分辨.我们基于 Treemap 提出了基于嵌套圆的可视化方法^[11],以圆表示树节点,以圆的嵌套关系表示不同层次的父子节点之间的层次结构关系,以一组相外切的圆表示同层次的一组兄弟节点.该方法能够在有限区域内对含有 100 万节点的海量层次信息进行可视化.另一方面,虽然上述方法能够在有限显示空间内呈现大量信息,但随着树节点数目和层数的增多,深层次子节点逐渐变小而难以分辨,需要通过交互技术对其详细信息进行可视化呈现.Furnas^[16]与 Card 等人^[17]指出,用户在交互探索大规模可视信息时,倾向于在探索局部焦点信息的详细视图的同时,能够保持信息整体视图的可见性,并且能够感知焦点的上下文信息、保持焦点的上下文信息的一致性.Focus+Context 交互式可视化技术是解决此类问题的主要技术^[16-18].Furnas^[16]提出了兴趣度(degree of interest,简称 DOI)函数对信息空间进行变形计算,在保持整体视图可视化的前提下,对用户交互操作选中的焦点区域(focus)放大,将焦点上下文信息(context)逐渐变小.Bederson^[19]与 Paek^[20]将其用于一维信息可视化的交互,应用于菜单与网络搜索结果列表的交互式可视化.Bederson^[21]将其用于二维信息可视化,应用于表格日历的交互式可视化.Sarkar^[22]提出了针对图结构的 Focus+Context 交互式可视化的数学模型,Darling^[23]与 Christian^[24]将其应用于节点和边组成的图形.对于焦点上下文的研究是 Focus+Context 交互式可视化技术中的一个重要问题,Gutwin^[25],Fedak^[26],Liu^[27]及 Carpendale 等人^[28]对交互过程中焦点上下文的变形对用户完成任务的影响进行了研究,通过对焦点上下文进行感知、跟踪、标注等方法,对 Focus+Context 交互式可视化技术进行改进.

本文将 Focus+Context 交互式可视化技术用于海量层次信息可视化.基于嵌套圆可视化方法,提出焦点突出、上下文感知的 Focus+Context 交互式可视化技术.在交互过程中,当焦点圆放大时,其临近的圆的半径经过变形计算发生变化,嵌套圆重新进行外切排列,原本临近的圆可能分离得很远,导致焦点上下文不一致.其中的关键问题是焦点上下文感知.上下文感知包括两种:一种是横向上下文感知,即追踪同层节点间的关联,在各圆重排过程中要始终保持同层节点间(圆与圆之间)的临近关系;另一种是纵向上下文感知,即追踪父子节点的关联.针对横向上下文感知问题,我们在外切圆排列方法的基础上提出对圆心进行三角网格剖分的方法,为焦点圆建

立上下文.根据三角网格的邻居关系,可以确定圆和圆之间的邻居关系.基于三角网格邻居跟踪的方法,我们提出基于三角网格邻居跟踪的外切圆排列方法,在重新排列圆的过程中,根据邻居关系优先排列临近的圆,保持圆和圆之间的相对稳定.重排后更新三角形的邻居关系,实现对焦点圆同层兄弟节点圆的上下文一致性的计算.针对纵向上下文感知问题,在横向上下文感知的基础上,我们提出了父子节点的嵌套圆变形迭代排列方法.实验结果表明,上述方法能够有效地解决上下文感知问题.文件系统的应用实例表明上述方法是实用、高效的.

1 基于外切圆排列的圆心三角网格剖分方法

我们在文献[1]的外切圆排列方法的基础上,对圆心进行三角网格剖分.将同层兄弟节点对应的一组圆记为 $\{(C_i, R_i) | i=1, 2, \dots, N\}$, 其中, C_i 表示节点对应圆的圆心, R_i 表示节点对应圆的半径, N 表示节点个数.首先将 3 个圆(简记为圆 C_1 、圆 C_2 、圆 C_3)排列起来,使其两两相互外切.连接 3 个圆心得到封闭的边界线 L ,我们使用双向链表作为 L 的数据结构,初始 3 个外切圆的边界线 L 可表示为双向链表 $L=\{C_1 \leftrightarrow C_2 \leftrightarrow C_3 \leftrightarrow C_1\}$.取 $\Delta C_1 C_2 C_3$ 的内心作为原点 $O(0,0)$.每当有新的圆 C_i 加入时,令圆 C_i 沿着 L 滚动至距离原点 O 最近的较凹的位置,使圆 C_i 与此位置的两个圆相外切,然后更新边界线 L ,将 C_i 加入边界线 L 中.如此下去,初始圆组不断向外扩展,直到所有圆排列完毕.这里要求新加的圆总是加在距中心点尽可能近的地方,以防止圆组沿某一方向不断扩展而形成细长的形状.在外切圆排列的同时,我们对圆心进行三角网格剖分,为 Focus+Context 交互式可视化的实现建立上下文.与经典的 Delaunay^[29]三角网格剖分方法不同,我们在新加入外切圆的过程中,逐步生成三角网格.在 $\Delta C_1 C_2 C_3$ 基础上,每当有新的圆 C_i 加入时,设与圆 C_i 相外切的两个圆为圆 C_m 和圆 C_n ,边界线 L 位于圆 C_m 和圆 C_n 之间的部分设为 $L_{mn}=\{C_m \leftrightarrow C_{m+1} \leftrightarrow \dots \leftrightarrow C_n\}$, 其中, $(n-m) \geq 1$.如图 1 所示,依次连接圆心 C_i 与 L_{mn} 中的各个圆心,生成新的三角形网格.图 1 中当加入圆 C_4, C_5 与 C_6 时, L_{mn} 中包含 2 个圆心,新生成的三角形为 1 个;当加入圆 C_7 时, L_{mn} 中包含 3 个圆心,新生成的三角形为 2 个.

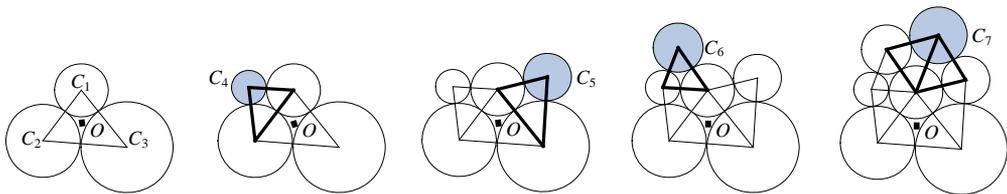


Fig.1 Generate triangular mesh after a new externally tangent circle is added
图 1 加入新外切圆时的三角网格剖分

设圆 C_m 的初始值为 L 中与原点 O 距离最近的圆,圆 C_n 的初始值是 L 中与圆 C_m 相邻的圆, L_{mn} 为 L 中位于圆 C_m 和 C_n 之间的部分, L_{mn} 的初始值为 $\{C_m \leftrightarrow C_n\}$, 如图 2(a)所示.设圆 C_i 是待加入圆组的圆,基于外切圆排列的三角网格剖分算法(algorithm of circle packing and triangular mesh generation, CPTMG 算法)如下:

- (1) 根据圆 C_i 的半径 R_i , 与圆 C_i 相外切的圆 C_m 和圆 C_n 的圆心与半径计算圆 C_i 的圆心坐标.
- (2) 在 L 中搜索与圆 C_i 相交的圆 $C_j(j \neq m$ 且 $j \neq n)$.
- (3) 如果圆 C_i 未与 L 中的任何圆相交,则
 - (i) 生成三角形网格:依次连接圆心 C_i 与 L_{mn} 中的各个圆心,如图 2(c)所示;
 - (ii) 更新边界线 L :删除 L 中的 L_{mn} 部分.将 C_i 加入 L ,将 $\{\dots C_m \leftrightarrow C_n \dots\}$ 更新为 $\{\dots C_m \leftrightarrow C_i \leftrightarrow C_n \dots\}$,如图 2(b)所示.

将圆 C_i 加入圆组的排列过程与生成新三角网格的剖分过程结束.

(4) 如果圆 C_i 与 C_j 相交,且 C_j 在边界线 L 中位于 C_n 的后方,如图 2(d)所示,则将 L 中从 C_m 至 C_j 的部分赋予 L_{mn} .将 C_n 值置为 C_j ,转向(1).更新的边界线与新建的三角网格如图 2(e)和图 2(f)所示.

(5) 如果圆 C_i 与 C_j 相交,且 C_j 在边界线 L 中位于 C_m 的前方,如图 2(g)所示,则将 L 中从 C_j 至 C_n 的部分赋予 L_{mn} .将 C_m 值置为 C_j ,转向(1).更新的边界线与新建的三角网格如图 2(h)和图 2(i)所示.

For $i=4$ to N ,重复以上过程.

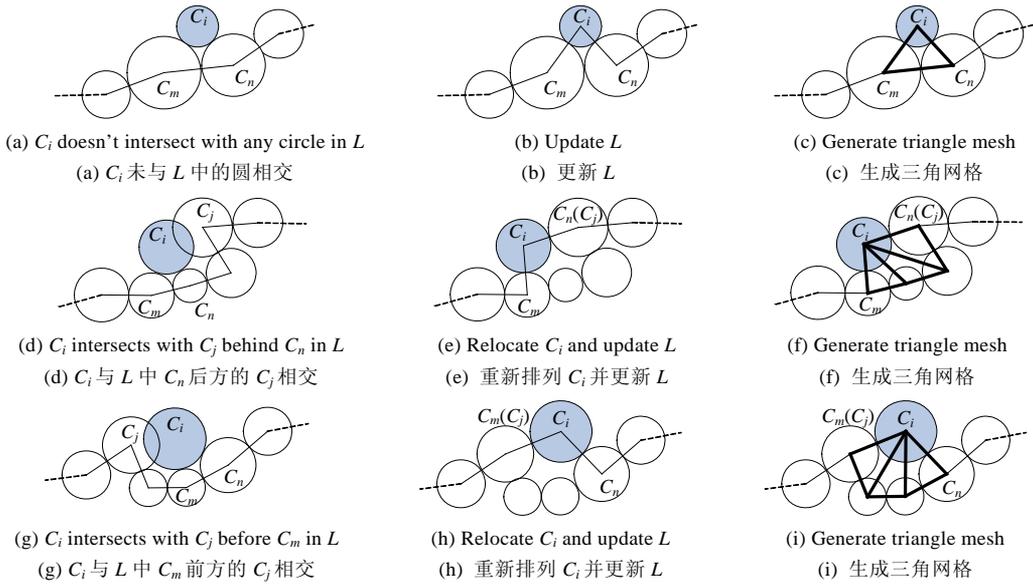


Fig.2 Algorithm of circle packing and triangle mesh generation

图 2 基于外切圆排列的三角网格剖分算法

对于每个新加入的圆,只需要沿着边界线从最接近圆组中心的位置起滚动一定的次数,且滚动次数的均值为常数,即可完成排列.因此,新加入圆的排列过程为线性时间复杂性.对于 N 个圆的排列过程,算法时间复杂性为 $O(N)$.我们将其在 CPU2.41GHZ, RAM512M 及 VC++6.0 编译器的 PC 机环境中实现,对 1 000 个圆(半径随机取值范围为 1~1 000)进行实验.如图 3 所示,图 3(a)是对 1 000 个圆进行外切排列的结果以及三角形网格剖分结果,由图可见,整体排列结果接近圆形,空间利用率较高.图 3(b)~图 3(d)是图 3(a)中黑框标出部分的放大视图.图 3(c)显示了不同半径外切圆的紧密排列结果.图 3(d)显示了三角形网格的剖分结果.CPTMG 算法排列上述 1 000 个圆所花费的 CPU 时间仅为 0.015s,边界线中圆的个数为 132 个,所剖分的三角形为 1 564 个.

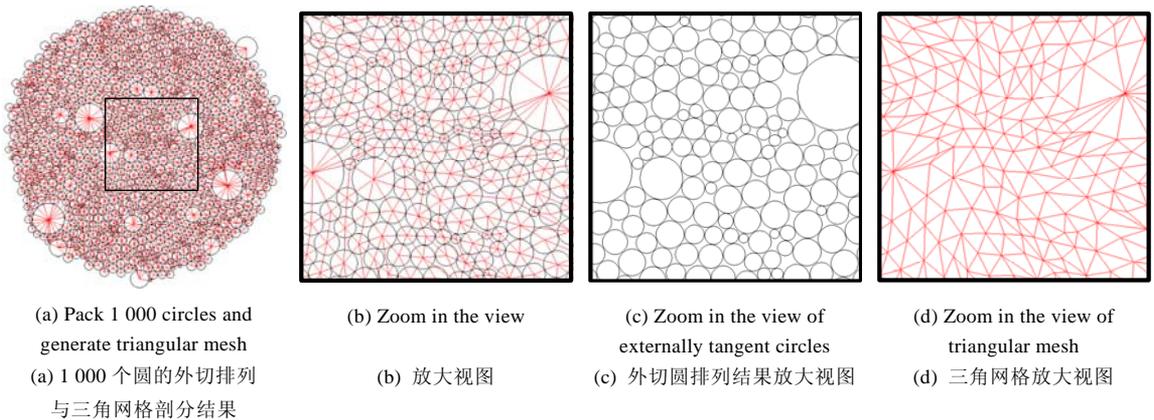


Fig.3 Pack 1 000 circles and generate triangular mesh by algorithm CPTMG

图 3 CPTMG 算法对 1 000 个圆的外切排列与三角网格剖分

2 基于三角网格邻居跟踪的Focus+Cotext实现方法

2.1 同层兄弟节点圆的DOI函数

Focus+Context 交互式可视化技术通过 DOI 函数或变形函数^[22,28]对信息空间进行变形计算,实现焦点信息

放大、焦点周围信息逐渐变小的变形效果.我们针对焦点圆同层次的各个兄弟节点圆,对其半径的变化建立 DOI 函数,即对于焦点圆 C_f 的任意一个同层兄弟节点圆 $C_i(1 \leq i \leq N)$,设圆 C_i 与焦点圆 C_f 的距离为 l_i ,设圆 C_i 的半径为 r_i ,建立 r_i 随 l_i 的增大而减小的函数 $r_i=f(l_i)$.设圆 C_p 为焦点圆 C_f 的父节点对应的圆,圆心 C_f 与圆 C_p 圆周上的点的最大距离为 l_{\max} ,则 l_i 的定义域为 $0 \leq l_i \leq l_{\max}$.其中, l_{\max} 为线段 $C_f C_p$ 的长度与圆 C_p 半径之和.设焦点圆 C_f 半径 r_f 放大后的半径为 r_0 ,放大率为 k ,则 $r_0=k \cdot r_f$.我们根据 Gaussian 辅助函数^[28]构造 DOI 函数式(1),其图像如图 4 所示.

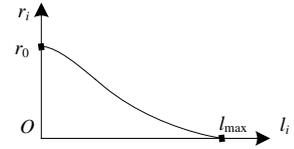


Fig.4 DOI function

图 4 DOI 函数

$$r_i = f(l_i) = k \cdot r_f \cdot \left(1 - 0.2 \sin \left(\frac{l_i}{l_{\max}} \cdot \frac{\pi}{2} \right)^{0.8} \right) \tag{1}$$

2.2 基于三角网格的邻居跟踪算法

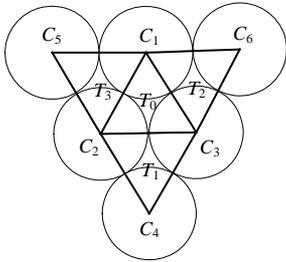


Fig.5 Triangular mesh neighbors

图 5 三角邻居

对于 CPTMG 算法建立的三角网格,三角形的顶点由各个圆的圆心组成,如图 5 所示.本文对文献[30]中的三角网格遍历算法进行改进,将具有公共边 e 的两个三角形 T_i 与 T_j 定义为关于此公共边 e 的三角邻居.如图 5 中,三角形 T_0 与 T_1 是关于公共边 C_2C_3 的三角邻居,三角形 T_0 的邻居集合为 $\{T_1, T_2, T_3\}$.为了实现变形计算中的上下文感知,需要在重新排列圆的过程中能够保持圆和圆之间相对稳定的临近关系,首先需要解决的问题是对于三角邻居进行跟踪,为每个三角形建立邻居集合,从而可以根据三角形邻居追踪临近的圆.如图 5 所示,首先对于三角网格中的任意一个三角形 T_0 ,搜索其邻居得到邻居集合 $\{T_1, T_2, T_3\}$,然后对于 T_1, T_2, T_3 搜索其邻居集合.依次类推,建立所有三

角形的邻居集合.

我们使用二叉树来表示三角网格集合,将三角形 T_k 的数据结构定义如下:

- T_k //三角网格集合中第 k 个三角形
- {
- C_1^k // T_k 的第 1 个顶点,对应圆 C_1^k
- C_2^k // T_k 的第 2 个顶点,对应圆 C_2^k
- C_3^k // T_k 的第 3 个顶点,对应圆 C_3^k
- P_1^k // T_k 关于边 $C_2^k C_3^k$ 的三角邻居
- P_2^k // T_k 关于边 $C_1^k C_3^k$ 的三角邻居
- P_3^k // T_k 关于边 $C_1^k C_2^k$ 的三角邻居
- }

设 N_t 为三角网格集合中三角形的个数.设 N_c 为三角形顶点(圆心)个数.设 $S[i]$ 是与三角形顶点 $i(1 \leq i \leq N_c)$ 相连的所有三角形的集合.基于三角网格的邻居跟踪算法(algorithm of triangular mesh neighbors tracking,简称 TMNT)描述如下:

- (1) 将 $S[i]$ 初始化为空.
- (2) 对于三角网格集合中的每个三角形 $T_k(1 \leq k \leq N_t)$,将 T_k 分别加入 $S[C_1^k], S[C_2^k], S[C_3^k]$,其中 C_1^k, C_2^k, C_3^k 为 T_k 的 3 个顶点.
- (3) 对三角网格集合中每个三角形 $T_k(1 \leq k \leq N_t)$,搜索并记录下 T_k 的邻居.搜索过程为:搜索同时出现在 $S[C_1^k]$ 和 $S[C_2^k]$ 集合中的三角形 T_j ,如果 $T_j \neq T_k$,则 T_j 是 T_k 关于边 $C_1^k C_2^k$ 的邻居,故将 T_j 赋值于 P_3^k .同理,搜索关于边

$C_2^k, C_3^k, C_1^k, C_3^k$ 的邻居并给 P_1^k, P_2^k 赋值.

因为对于任意一个三角形顶点来说,与此顶点相连的三角形数目的平均值为常数,所以搜索每一个三角形邻居的过程的时间复杂性为线性.因此,对于 N 个三角形,该算法的时间复杂性为 $O(N)$.通过该算法,所有三角形的邻居都被记录下来,为变形计算后的圆在重排过程中确定圆与圆之间的临近位置关系建立了基础.

2.3 基于三角网格邻居跟踪的同层兄弟节点的变形排列算法

对于焦点圆同层兄弟节点圆进行变形计算后,可以根据 TMNT 算法建立的三角邻居集合对每个三角形的邻居进行跟踪,由于每个三角形的 3 个顶点对应着 3 个圆,因此可以根据三角形邻居关系确定对应的圆在变形计算前相互之间的临近关系,按照圆与圆之间的临近关系对焦点圆同层兄弟节点圆进行重新排列.如图 6 所示,首先根据焦点圆心 C_1 所在的 $\triangle C_1C_2C_3$ 的另外两个顶点 C_2 与 C_3 ,确定圆 C_1 的两个外切圆 C_2 和 C_3 ,根据 TMNT 算法确定的 $\triangle C_1C_2C_3$ 的邻居集合 $\{\triangle C_1C_2C_4, \triangle C_1C_3C_5, \triangle C_2C_3C_6\}$,将三角邻居集合中的每个三角邻居确定的外切圆依次加入相应的位置进行排列.如图 6 所示,对于 $\triangle C_1C_2C_4$ 对应的 3 个圆 C_4, C_1 和 C_2 ,将圆 C_4 加入圆组并排列在与圆 C_1 和 C_2 相外切的位置.同理,对 $\triangle C_1C_3C_5$ 和 $\triangle C_2C_3C_6$ 对应的外切圆加入圆组进行外切排列.然后对 $\triangle C_1C_2C_4, \triangle C_1C_3C_5, \triangle C_2C_3C_6$ 重复上述跟踪邻居与加入外切圆的过程,直到邻居集合为空.

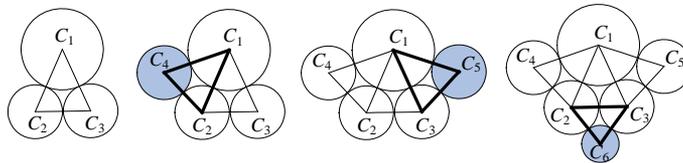


Fig.6 Packing distorted circles by tracking triangular neighbors

图 6 基于三角网格邻居跟踪的外切圆变形排列过程

设焦点圆为 C_f ,我们使用先进先出队列 *Queue* 作为存储三角形邻居集合的数据结构,基于三角邻居跟踪的同层兄弟节点圆的变形排列算法(algorithm of packing distorted circles by tracking triangular neighbors,简称 PDCT 算法)如下:

- (1) 为 C_f 放大率 k 赋初值,计算变形后的半径 R_0 ,对 C_f 的所有同层次兄弟节点圆 C_i ,根据式(1)计算 C_i 的半径.
- (2) 根据 C_f 所在的一个三角形 T_k ,将 T_k 的 3 个顶点对应的圆进行外切排列.
- (3) 根据 TMNT 算法得到的 T_k 的三角形邻居集合 $\{P_1^k, P_2^k, P_3^k\}$.对于 $P_j^k(j=1,2,3)$,如果 P_j^k 指向的圆未被标记过(尚未被排列),则将 P_j^k 加入 *Queue* 尾部,转(4).
- (4) 从 *Queue* 头部取出第 1 个三角形 P_j^k ,在 P_j^k 指向的 3 个圆中,设 P_j^k 所指向的未被标记过的圆为 C_i, P_j^k 所指向的已被标记过的两个圆为 C_m 和 C_n .使用 CPTMG 算法将 C_i 排列在相应的位置.对 C_i 做标记表示已被排列过.转(5).
- (5) 若 *Queue* 不为空,则将 P_i^k 的值赋予 T_k ,将 P_i^k 从 *Queue* 中删除,转(3);若 *Queue* 为空,则排列过程结束.

对于包含 N 个三角形的网格,每个三角形邻居最多有 3 个,每个三角形最多被遍历 1 次.并且,由于 CPTMG 算法排列一个圆的时间复杂性为线性,因此,PDCT 算法时间复杂性为 $O(N)$.如图 7 所示,我们对 1 000 个圆应用 PDCT 算法进行了实验.图 7(a)中黑色粗体线标出的圆为焦点圆 A .图 7(b)是图 7(a)中黑色粗体线框标出的焦点圆及周围临近的圆的放大视图.图 7(c)是应用 PDCT 算法对圆 A 及同层兄弟节点圆进行变形计算并重新排列后的结果.可见,焦点圆 A 放大并显示于中央位置,周围的圆逐渐缩小,呈现鱼眼视图效果.同时,与圆 A 临近的圆 B, C, D, E, F, G 仍然排列在 A 的周围,保持与焦点圆的临近关系.

2.4 父子节点的嵌套圆变形迭代排列算法

PDCT 算法解决了变形过程中焦点圆同层兄弟节点圆的上下文感知问题.在 PDCT 算法对焦点圆及同层兄

弟节点圆进行排列后,需要解决父子节点的上下文感知问题.即随着焦点圆及同层兄弟节点圆的变形,其子节点圆随之自动变形,并且保持父子节点圆的嵌套关系.我们采用迭代方法,将子节点按照父节点的缩放率进行同比例缩放,再经 CPTMG 算法排列起来并移动到父节点圆中.对于焦点圆及同层兄弟节点圆的子节点的嵌套圆迭代排列算法描述如下:

对焦点圆及同层兄弟节点圆 $C_i(0 \leq i \leq N)$,执行以下过程:

- (1) 置 C_i 为当前节点.如果 C_i 为叶节点则返回,如果 C_i 为分支节点,则计算 C_i 变形计算前后的缩放率 Z_i ,转(2).
- (2) 对于 C_i 的所有子节点,按照 Z_i 进行缩放.
- (3) 以 C_i 的圆心为原点,使用 CPTMG 算法将 C_i 所有子节点圆排列起来并建立三角网格.
- (4) 将排列好的子节点圆组经适当平移与缩放,使其恰好容纳在父节点圆 C_i 中.
- (5) 对 C_i 的所有子节点,将子节点值赋予 C_i ,迭代执行(1)~(4).

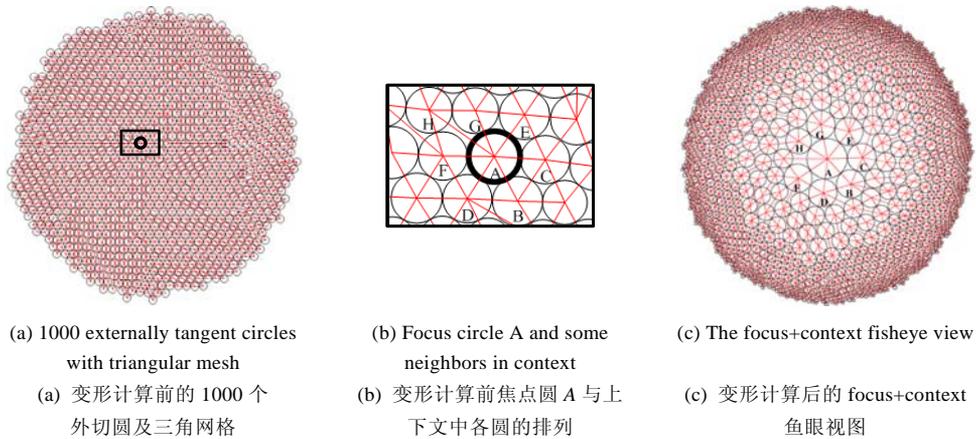


Fig.7 Pack 1 000 distorted circles by algorithm PDCT

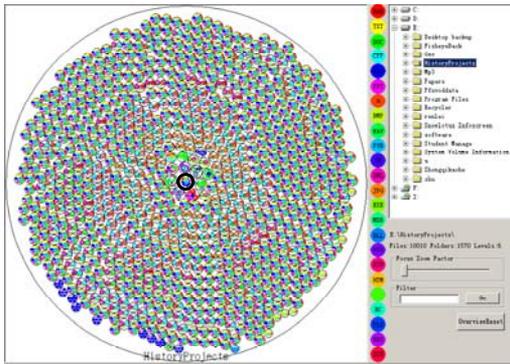
图 7 PDCT 算法对 1 000 个圆进行变形排列

3 应用实例

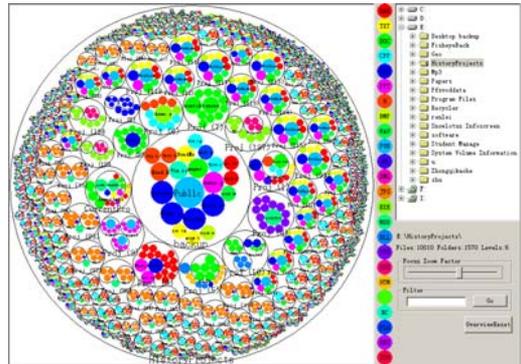
随着计算机存储空间的增大,计算机文件系统的层次信息规模越来越大,给用户维护与检索信息带来了困难.常用的 Windows 资源管理器能够清楚地表示文件夹之间的层次关系,但是,在同一屏内仅能显示有限个文件或文件夹的数目(PC 机上最多有几百个),绝大多数子文件是不能直接看到的,只有通过逐层展开才能看到,而且文件夹按行排列,如果展开节点较多,则需不断地滚动查看,即使是自己熟悉的文件,一旦忘记了文件的存放位置,由于缺乏直观的整体视图,查找起来仍然很麻烦,即使采用自动搜索也很费时.

我们将上述方法应用于文件系统的交互式可视化,使用 VC++6.0 开发了 Focus+Context 交互式可视化工具(Focus+Context visualization tool,简称 FCVT).FCVT 的用户界面如图 8(a)所示,界面右上方窗口显示 Windows 文件系统树结构,当用户点击树结构中某个节点时,以该节点作为根节点的层次信息的嵌套圆可视化整体视图显示在界面左侧主窗口内.图 8(a)中显示的是 E:\HistoryProjects 文件夹的嵌套圆可视化整体视图,其中共包含 10 010 个文件及 1 570 个文件夹.其中使用无填充色的圆表示树分支节点(文件夹),分支节点的名字显示在圆内的底端位置;使用具有灰度的圆表示叶节点(文件),文件的名字显示在圆内的中央位置.当圆的半径小于一定值时,文件夹或文件的名字自动隐藏.灰度深浅不同的圆代表不同的文件类型,界面主窗口右侧的一列灰度深浅不同的小圆表示不同的文件类型,图中共列出 24 种文件类型,文件后缀名显示在小圆内中央位置.界面右下方的 Focus Zoom Factor 滑动条用于设置 Focus+Context 交互时焦点信息的放大率.当用户设置了焦点信息放大率后,可以通过鼠标与嵌套圆整体视图进行交互.如图 8(b)所示,当鼠标指向用户感兴趣的焦点圆

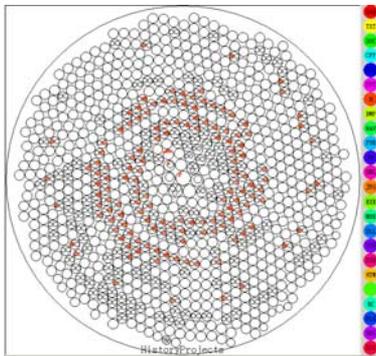
(E:\HistoryProjects\backup 文件夹)时,焦点圆放大并显示于父节点的中央位置,与焦点圆同层的兄弟节点圆经过变形计算,在焦点圆周围沿径向逐渐缩小,并保持变形前后上下文的一致性.焦点圆的子节点圆随着焦点圆作相应变形计算,保持父子节点上下文的一致性.经过变形计算,形成焦点突出、焦点圆同层兄弟节点圆上下文一致、焦点圆父子节点圆上下文一致的鱼眼视图效果,用户可在查看焦点信息的同时根据上下文进行交互导航,并且可以随时调整焦点信息放大率滑动条以达到满意的清晰度.界面右下方的 Filter 文本框用于对文件进行过滤,缩小用户搜索信息的区域.如图 8(c)所示,当用户设置了文件名关键词的过滤条件后,嵌套圆可视化整体视图中将隐藏不符合过滤条件的文件对应的圆,而只显示符合过滤条件的文件对应的圆.如图 8(d)所示,在缩小文件分布区域的基础上,用户可以进一步使用 Focus+Context 交互对文件分布区域进行查看,以达到更高的文件搜索效率.界面右下方的 OverviewReset 按钮用于对嵌套圆可视化整体视图的重置,返回图 8(a)的状态.我们对 FCVT 初步进行了可用性评估,获得了较高的用户满意度.



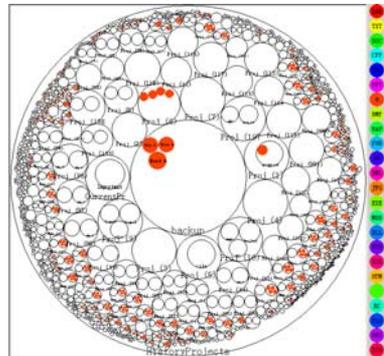
(a) Overview of "E:\HistoryProjects"
(a) E:\HistoryProjects 文件夹整体视图



(b) Context-Aware fisheye view in Focus+Context interaction
(b) Focus+Context 交互式可视化中的鱼眼视图



(c) Filter overview
(c) 过滤后的整体视图



(d) Focus+Context interaction in the filter overview
(d) 在过滤视图中的 Focus+Context 交互式可视化

Fig.8 Focus+Context interactive visualization tool

图 8 Focus+Context 交互式可视化工具

4 结束语

本文针对海量层次信息可视化的交互问题,在层次信息的嵌套圆可视化技术基础上提出了基于上下文感知的 Focus+Context 交互式可视化技术.首先基于外切圆排列方法提出对圆心进行三角网格剖分的方法,为变形计算建立上下文,根据三角网格上下文能够对焦点圆与临近兄弟圆上下文进行跟踪.为了解决变形计算前后上下文一致性问题,在三角网格邻居跟踪方法的基础上,提出了用于同层兄弟节点上下文感知的外切圆变形排列

方法,以及用于父子节点上下文感知的嵌套圆迭代排列方法.实验结果表明,上述方法一方面能够实现焦点突出的鱼眼视图,同时能够有效地解决 Focus+Context 交互式可视化的上下文感知与上下文一致性问题.上述方法应用于计算机文件系统海量层次信息的交互式可视化问题,并提供了交互式可视化工具 FCVT,给出了应用实例.未来的工作是在对 FCVT 进行可用性评估及分析的基础上,研究基于语义距离的 DOI 函数以及三维半球面中的 Focus+Context 交互式可视化技术.

References:

- [1] Wang WX, Wang H, Dai GZ, Wang H. Visualization of large hierarchical data by circle packing. In: Grinter R, Roden T, Aoki P, Cutrell E, Jeffries R, Olson G, eds. Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. New York: ACM Press, 2006. 517–520.
- [2] Wang WX, Dai GZ. Information visualization. *Advances in Information Science*, 2006,6:80–118 (in Chinese with English abstract).
- [3] Robertson GG, Mackinlay JD, Card SK. Cone trees: Animated 3D visualizations of hierarchical information. In: Robertson SP, Olson GM, Olson JS, eds. Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. New York: ACM Press, 1991. 189–194.
- [4] Lamping J, Rao R, Pirolli P. A Focus+Context technique based on hyperbolic geometry for visualizing large hierarchies. In: Katz LR, Mack R, Marks L, Rosson MB, Nielson J, eds. Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. New York: ACM Press, 1995. 401–408.
- [5] Heer J, Card SK. DOI Trees revisited: Scalable, space-constrained visualization of hierarchical data. In: Costabile MF, ed. Proc. of the Working Conf. on Advanced Visual Interfaces. New York: ACM Press, 2004. 421–424.
- [6] Plaisant C, Grosjean J, Bederson BB. SpaceTree: Supporting exploration in large node link tree, design evolution and empirical evaluation. In: Wong PC, Andrews K, eds. Proc. of the IEEE Symp. on Information Visualization. Los Alamitos: IEEE Computer Society Press, 2002. 57–64.
- [7] Johnson B, Shneiderman B. Tree-Maps: A space-filling approach to the visualization of hierarchical information. In: Nielson GM, Rosenblum L, eds. Proc. of the 2nd IEEE Visualization Conf. Los Alamitos: IEEE Computer Society Press, 1991. 284–291.
- [8] Van HF, Van WJ. Beamtrees: Compact visualization of large hierarchies. In: Wong PC, Andrews K, eds. Proc. of the IEEE Symp. on Information Visualization. Los Alamitos: IEEE Computer Society Press, 2002. 31–39.
- [9] Van WJ, Van WH. Cushion Treemaps: Visualization of Hierarchical Information. In: Bryson S, Rhyne TM, eds. Proc. of the IEEE Symp. on Information Visualization. Los Alamitos: IEEE Computer Society Press, 1999. 73–78.
- [10] Bederson BB, Shneiderman B. Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Trans. on Graphics*, 2002,21(4):833–854.
- [11] Itoh T, Yamaguchi Y, Ikehata Y, Kajinaga Y. Hierarchical data visualization using a fast rectangle-packing algorithm. *IEEE Trans. on Visualization and Computer Graphics*, 2004,10(3):302–313.
- [12] Balzer M, Deussen O. Voronoi Treemaps for the visualization of software metrics. In: Naps T, Pauw WD, eds. Proc. of the 2005 ACM Symp. on Software visualization. New York: ACM Press, 2005. 165–172.
- [13] Chi EH, Pitkow J, Mackinlay J, Pirolli P, Gossweiler R, Card SK. Visualizing the evolution of Web ecologies. In: Karat CM, Lund A, Coutaz J, Karat J, eds. Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. New York: ACM Press, 1998. 400–407.
- [14] Stasko J, Zhang E. Focus+Context display and navigation techniques for enhancing radial, space-filling hierarchy visualization. In: Mackinlay J, Roth S, Keim DA, eds. Proc. of the IEEE Symp. on Information Visualization. Los Alamitos: IEEE Computer Society Press, 2000. 57–65.
- [15] Feng YD. Research on key issues for information visualization [Ph.D. Thesis]. Beijing: Peking University, 2001 (in Chinese with English abstract).
- [16] Furnas GW. Generalized fisheye views. In: Mantei M, Orbeton P, eds. Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. New York: ACM Press, 1986. 16–23.
- [17] Card SK, Mackinlay JD, Shneiderman B. *Readings in Information Visualization: Using Vision to Think*. San Francisco: Morgan Kaufmann Publishers, 1999.
- [18] Gutwin C, Skopik A. Fisheye views are good for large steering tasks. In: Cockton G, Korhonen P, eds. Proc. of the SIGCHI Conf.

- on Human Factors in Computing Systems. New York: ACM Press, 2003. 267–274.
- [19] Bederson BB. Fisheye menus. In: Ackerman M, Edwards K, eds. Proc. of the 13th Annual ACM Symp. on User Interface Software and Technology. New York: ACM Press, 2000. 217–225.
- [20] Paek T, Dumais S, Logan R. WaveLens: A new view onto Internet search results. In: Elizabeth DE, Manfred T, eds. Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. New York: ACM Press, 2004. 727–734.
- [21] Bederson BB, Clamage A, Czerwinski MP, Robertson GG. DateLens: A fisheye calendar interface for PDAs. ACM Trans. on Computer-Human Interaction, 2004,11(1):90–119.
- [22] Sarkar M, Brown MH. Graphical fisheye views of graphs. In: Bauersfeld P, Bennett J, Lynch G, eds. Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. New York: ACM Press, 1992. 83–91.
- [23] Darling E, Recktenwald K, Kalghatgi N, Burgman A. Effects of fisheye on visualizing connections between nodes. In: Kellogg W, Zhai S, eds. Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. New York: ACM Press, 2005. 1328–1330.
- [24] Christian T, James A, Frank H, Heidrun S. Fisheye tree views and lenses for graph visualization. In: Ikeuchi K, Suda Y, eds. Proc. of the 10th Int'l Conf. on Information Visualization. Washington: IEEE Computer Society Press, 2006. 17–24.
- [25] Gutwin C. Improving focus targeting in interactive fisheye views. In: Wixon D, ed. Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. New York: ACM Press, 2002. 267–274.
- [26] Fedak C, Gutwin C. Improving revisitation in fisheye views with visit wear. In: Kellogg W, Zhai S, eds. Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. New York: ACM Press, 2005. 771–780.
- [27] Liu F, Gleicher M. Automatic image retargeting with fisheye-view warping. In: Baudisch P, Czerwinski M, eds. Proc. of the 18th Annual ACM Symp. on User Interface Software and Technology. New York: ACM Press, 2005. 153–162.
- [28] Carpendale S, Light J, Pattison E. Achieving higher magnification in context. In: Feiner SK, Landy JA, eds. Proc. of the 17th Annual ACM Symp. on User Interface Software and Technology. New York: ACM Press, 2004. 71–80.
- [29] Shewchuk JR. Delaunay refinement algorithms for triangular mesh generation. Computational Geometry: Theory and Applications, 2002,22(1):21–74.
- [30] Lo SH, Wang WX. A fast robust algorithm for the intersection of triangulated surfaces. Engineering with Computers, 2004,20(1): 11–21.

附中文参考文献:

- [2] 王威信,戴国忠.信息可视化.情报学进展,2006,6:80–111.
- [15] 冯艺东.信息可视化若干问题研究[博士学位论文].北京:北京大学,2001.



任磊(1979—),男,山东济南人,博士生,主要研究领域为人机交互技术,信息可视化.



王威信(1970—),男,博士,助理研究员,CCF高级会员,主要研究领域为人机交互技术,信息可视化,CAD&CG.



滕东兴(1973—),男,博士,副研究员,CCF高级会员,主要研究领域为人机交互技术,虚拟现实技术,计算机辅助设计.



马翠霞(1975—),女,博士,助理研究员,CCF高级会员,主要研究领域为人机交互技术,草图计算,上下文感知交互技术.



戴国忠(1944—),男,研究员,博士生导师,CCF高级会员,主要研究领域为人机交互技术,计算机图形学.



王宏安(1963—),男,博士,研究员,博士生导师,CCF高级会员,主要研究领域为人机交互技术,实时数据库,实时调度.