(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102609229 A
(43)申请公布日 2012.07.25

(21)申请号 201210034661.7

(22)申请日 2012.02.16

(71)申请人 中国科学院声学研究所
地址 100190 北京市海淀区北四环西路 21 号

(72)发明人 朱小勇 姜艳 孙鹏

(74)专利代理机构 北京亿腾知识产权代理事务所 11309
代理人 陈霁

(51)Int.Cl.
G06F 3/14(2006.01)
G06F 9/38(2006.01)

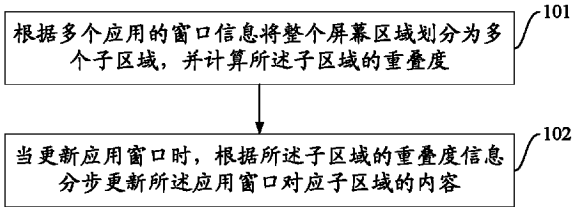
权利要求书 1 页 说明书 4 页 附图 2 页

(54)发明名称

一种嵌入式多窗口应用图形并行更新方法

(57)摘要

本发明公开了一种针对多窗口应用的图形并行更新方法，该方法包括：根据应用的窗口信息将整个屏幕区域划分为多个子区域，并计算所述子区域的重叠度；当更新应用窗口时，根据所述子区域的重叠度信息分步更新所述应用窗口对应子区域的内容。本发明减少了应用间窗口更新所需的时间，提高了系统的并行处理能力。



CN 102609229 A

1. 一种嵌入式多窗口应用图形并行更新方法，其特征在于，该方法包括：

根据应用的窗口信息将整个屏幕区域划分为多个子区域，并计算所述子区域的重叠度；

当更新应用窗口时，根据所述子区域的重叠度信息分步更新所述应用窗口对应子区域的内容。

2. 根据权利要求 1 所述的方法，其特征在于，所述重叠度是指所述子区域所覆盖的应用窗口数目。

3. 根据权利要求 1 所述的方法，其特征在于，所述窗口信息包括窗口的位置和 / 或窗口的大小。

4. 根据权利要求 1 所述的方法，其特征在于，所述根据应用的窗口信息将整个屏幕区域划分为多个子区域的步骤包括：

在所有窗口中计算两两窗口之间的重叠区域，构成第一子窗口集合；

在所述第一子窗口集合中，进一步计算子窗口两两之间的重叠区域，构成第二子窗口集合，直至所述子窗口集合的空间大小为 1 或 0；

根据所述子窗口集合划分屏幕子区域。

5. 根据权利要求 1 所述的方法，其特征在于，在所述根据所述子区域的重叠度信息分步更新所述应用窗口对应子区域的内容的步骤中优先选择空闲且重叠度高的子区域进行更新，并对更新过的子区域进行标识。

6. 根据权利要求 1 所述的方法，其特征在于，当所述子区域的重叠度为一时，则直接更新所述子区域；当所述子区域的重叠度大于一时，则通过加锁的方式来更新所述子区域。

7. 根据权利要求 1 所述的方法，其特征在于，所述子区域通过位图来表示。

8. 根据权利要求 7 所述的方法，其特征在于，该方法根据所述位图搜索需要更新的子区域。

9. 根据权利要求 7 所述的方法，其特征在于，所述子区域的重叠度按大小排序，所述重叠度大的优先更新。

10. 根据权利要求 1 所述的方法，其特征在于，当有应用加入或退出时，则重新划分并计算屏幕子区域信息。

一种嵌入式多窗口应用图形并行更新方法

技术领域

[0001] 本发明涉及嵌入式技术领域，特别涉及一种嵌入式多窗口应用图形并行更新方法。

背景技术

[0002] 随着嵌入式与多媒体技术的快速发展，电视终端正由传统的娱乐中心转向集公共广播、信息服务、文化娱乐、交流互动于一体的家庭多媒体信息平台，多应用的盛行促使电视终端图形系统必须支持多窗口并行显示技术。特别是一些如游戏、信息类应用，需要频繁地对屏幕显示进行更新操作。另外，不同于手机、PDA 等低分辨率设备，电视终端的图形分辨率较高，多为 1280x720、1920x1080。为满足多窗口并行显示、频繁刷新和高分辨率的需求，必须对嵌入式电视终端有限的图形处理性能进行优化。

[0003] 在现有技术中，当应用更新窗口时，首先锁定整个屏幕区域，然后更新窗口内容，最后释放锁定。这种方法没有考虑窗口位置和大小信息，当一个应用锁定整个屏幕时，其他应用无法对屏幕进行操作。但是，在多窗口应用下，并不是所有的应用窗口都覆盖整个屏幕，有可能两个应用的窗口在屏幕上没有相交部分，没有必要分别锁定、更新。这种方法没有考虑窗口信息而盲目锁定整个屏幕，导致其他应用更新阻塞，增大了等待时间，降低了系统的并行性。

发明内容

[0004] 本发明的目的是解决上述现有技术所存在的问题。

[0005] 为实现上述目的，本发明提供了一种多窗口应用的图形并行更新方法，该方法包括：根据应用的窗口信息将整个屏幕区域划分为多个子区域，并计算所述子区域的重叠度 (degree of overlap)；当更新应用窗口时，根据所述子区域的重叠度信息分步更新所述应用窗口对应子区域的内容。

[0006] 根据本发明的方法，可以减少应用间窗口更新所需的时间，进一步提高了系统的并行处理能力。

附图说明

[0007] 图 1 是本发明实施例多窗口应用的图形并行更新方法的流程图；

[0008] 图 2 是本发明实施例多窗口应用的图形并行更新方法的屏幕划分流程图；

[0009] 图 3 是本发明实施例多窗口应用示例图；

[0010] 图 4 是本发明实施例应用窗口更新流程图。

具体实施方式

[0011] 下面通过附图和实施例，对本发明的技术方案做进一步的详细描述。

[0012] 图 1 是本发明实施例多窗口应用的图形并行更新方法的流程图。如图所示，该方

法包括步骤 101-102 ：

[0013] 在步骤 101，根据多个应用的窗口信息将整个屏幕区域划分为多个子区域，并计算所述子区域的重叠度。

[0014] 具体地，嵌入式电视终端设备在多个应用加入时，根据所有应用的窗口信息，例如窗口位置和窗口大小信息将整个显示屏幕区域划分为多个子区域，并计算上述多个子区域的重叠度信息。上述重叠度是指多个子区域中的其中一个子区域所覆盖的应用窗口数目。另外，当有新的应用加入或退出时，则重新划分并计算屏幕的子区域信息。

[0015] 以下针对子区域的重叠度进行定义：假设有 n 个应用，分别对应于 n 个窗口 w_1，w_2，……，w_n，记集合 $W^{(1)} = \{w_1, w_2, \cdots, w_n\}$。对于屏幕上任意子区间 V，集合 $W^{(1)}$ 中存在 1 个应用窗口 $w_{i1}, w_{i2}, \cdots\cdots, w_{i1}$ 覆盖区间 V，则称 1 为区间 V 的重叠度。

[0016] 以下结合图 2，针对上述屏幕区域划分的具体算法进行描述：如图 2 所示，在步骤 200，设 $W^{(K)}$ 为所有应用窗口构成的集合，其中 K = 1。在步骤 201，根据窗口集合 $W^{(K)}$ 中各窗口两两之间的关系，若两者之间存在重叠区域，则计算出任意两个区间的重叠区域，重叠区域构成的集合记为 $W^{(K+1)}$，$W^{(k+1)} = \{w_{ij} | w_{ij} \in w_i, w_{ij} \in w_j\}$。在步骤 202，判断集合 $W^{(K+1)}$ 中的窗口个数是否大小 1。如果是，则执行步骤 210，令 K = k+1，返回到步骤 201；如果不是，则执行步骤 203。以此类推，继续计算 $W^{(K+2)}$、$W^{(K+3)}$、……中的重叠区域，直至为空。在步骤 203，根据窗口集合 $W^{(K)}$、$W^{(K+1)}$，……，将整个屏幕区域进行划分。

[0017] 在实际应用情况下，由于多个应用的窗口之间的重叠度不会太大，因此递推次数有限。假设最终的重叠度最大值为 1，即通过以上的方法共得到 1 个集合 $\{W^{(1)}, W^{(2)}, \cdots W^{(1)}\}$。不难看出，$W^{(1)}$ 中的所有区域的任意子区间的重叠度均为 1。由于集合 $W^{(1-1)}$ 中包含了 $W^{(1)}$，因此其任意子区间的重叠度为 1 或 1-1，但 $W^{(1-1)}-W^{(1)}$ 的子区间的重叠度为 1-1。假设集合 $U^{(i)}$ 中的区间个数为 C_i，屏幕被划分的区域包括：

[0018] 重叠度为 1 的集合 $U^{(1)}$ 中区间个数为 C_1，$U^{(1)} = W^{(1)}$；

[0019] 重叠度为 m 的集合 $U^{(m)}$ 中区间个数为 C_m，$U^{(m)}$ 计算如下：

[0020] $U^{(m)} = \{u_i | u_i = w_i^{(m)} - \sum_j w_{ij}^{(m+1)} + \sum_k w_{ijk}^{(m+2)} - \cdots\}$，其中 $w_i^{(m)} \in W^{(m)}$，$w_{ij}^{(m+1)} \in w_i^{(m)}$，$w_{ijk}^{(m+2)} \in w_{ij}^{(m+1)}$，…；

[0021] 重叠度为 1 的集合 $U^{(1)}$ 中区间个数为 C_1，$U^{(1)}$ 计算如下：

[0022] $U^{(1)} = \{u_i | u_i = w_i - \sum_j w_{ij}^{(2)} + \sum_k w_{ijk}^{(3)} - \cdots\}$，其中 $w_i \in W^{(1)}$，$w_{ij}^{(2)} \in w_i$，$w_{ijk}^{(3)} \in w_{ij}^{(2)} \cdots$；

最终整个屏幕将被划分为 $k = \sum_{m=1}^{l} C_m$ 个区间，其集合形式可表示为 $U = \{\sum_{m=1}^{l} U^{(m)}\}$。

[0023] 在一个例子中，以一个四元级 (x_1, y_1, x_2, y_2) 表征一个窗口参数，其中 (x_1, y_1) 为窗口左上角坐标，(x_2, y_2) 为窗口右下角坐标，选取屏幕左上角为坐标原点。任意两个窗口 $(x_{i1}, y_{i1}, x_{i2}, y_{i2})$ 和 $(x_{j1}, y_{j1}, x_{j2}, y_{j2})$，当窗口坐标满足关系式 $\max(x_{i1}, x_{j1}) < \min(x_{i2}, x_{j2})$ 且 $\max(y_{i1}, y_{j1}) < \min(y_{i2}, y_{j2})$ 时，存在重叠部分，重叠窗口的坐标为 $(\max(x_{i1}, x_{j1}), \max(y_{i1}, y_{j1}), \min(x_{i2}, x_{j2}), \min(y_{i2}, y_{j2}))$。

[0024] 以下结合图 3，阐述多窗口应用的具体实例。如图 3 所示，共有 4 个窗口应用，其整个屏幕的图形分辨率为 1280x720。其中，w_1 为全屏的应用，窗口坐标为 (0,0,1280,720)，即

包括区域 U_1、U_2、U_3 和 U_4 ;w_2 应用占据屏幕的左半边，坐标参数为 (0,0,640,720)，即包括区域 U_1、U_2 和 U_3 ;w_3 应用占据 w_2 的下半部分，坐标参数为 (0,360,640,720)，即包括区域 U_1 和 U_2 ;w_4 应用占据 w_3 的右半部分，坐标参数为 (320,360,640,720)，即区域 U_1。

[0025] 根据我们的屏幕划分方法，可分为以下几个步骤 :步骤一，在应用窗口集合 $w^{(1)}$ 中找出两两窗口重叠部分，构成集合 $w^{(2)}$，共有 6 个窗口，包括 $w_{1,2}$(0,0,640,720)，$w_{1,3}$(0,360,640,720)，$w_{1,4}$(320,360,640,720)，$w_{2,3}$(0,360,640,720)，$w_{2,4}$(320,360,640,720) 和 $w_{3,4}$(320,360,640,720)，其中 $w_{i,j}$ 指第 i 个窗口和第 j 个窗口的重叠区域 ;步骤二，继续在应用窗口集合 $w^{(2)}$ 中找出两两窗口重叠部分，构成集合 $w^{(3)}$，包括 3 个窗口 $w_{1,2,3}$(0,360,360,720)，$w_{1,2,4}$(320,360,640,720)，$w_{1,3,4}$(320,360,640,720) ;步骤三，继续在应用窗口集合 $w^{(3)}$ 中找出两两窗口重叠部分，构成集合 $w^{(4)}$，包括 1 个窗口 $w_{1,2,3,4}$(320,360,640,720)。因此得到划分屏幕的子区域集合，该子区域集合包括 :一个重叠度为 4 的窗口 u_1，坐标为 (320,360,640,720) ;一个重叠度为 3 的窗口 u_2，坐标为 (0,360,320,720) ;一个重叠度为 2 的窗口 u_3，坐标为 (0,0,640,360) ;一个重叠度为 1 的窗口 u_4，坐标为 (640,0,1280,720)。

[0026] 在步骤 102，当更新应用窗口时，根据所述子区域的重叠度信息分步更新所述应用窗口对应子区域的内容。当需要被更新的子区域的重叠度为一时，表明该子区域只被一个应用所使用，可直接更新该子区域 ;当子区域的重叠度大于一时，则通过加锁的方式来更新该子区域，以保证该子区域间多应用的互斥访问。

[0027] 本发明实施例子区域分步更新的方法包括以下步骤 :1、首先确定窗口需要更新的所有屏幕子区域 ;2、在所需要更新的区域中优先选择空闲且重叠度高的子区域进行更新，并对更新过的子区域进行标识 ;3、继续按步骤 2 的方法更新未更新的子区域，直到所有子区域更新完毕。

[0028] 在一个实施例中，每个应用通过使用一个位图来表示各自窗口需要更新的子区域，置位表示该子区域包含在该应用窗口中。例如，应用 1 的位图为 (1,1,1,1)，表明该应用 1 的窗口覆盖了 u_1、u_2、u_3、u_4。依次类推应用 2 的位图为 (1,1,1,0)，应用 3 的位图为 (1,1,0,0)，应用 4 的位图为 (1,0,0,0)。

[0029] 下面结合图 4，以更新应用 1 窗口为例说明分布更新的方法，如图 4 所示，该方法包括步骤 401-406 :

[0030] 在步骤 401，获取应用 1 窗口对应的位图。图 4 所示的应用 1 窗口的位图为 (1,1,1,1)，表明该应用 1 的窗口覆盖了 u_1、u_2、u_3、u_4。

[0031] 在步骤 402，根据子区域位图搜索可更新的子区域。

[0032] 具体地，根据应用 1 的位图 (1,1,1,1) 搜索需要更新且未被其他应用占用的子区域。子区域的重叠度按大小排序，位图顺序对应的子区域的重叠度从大到小依次排列。由于重叠度越大的子区域越容易发生更新冲突，因此根据应用位图从头搜索可以优先保证重叠度大的且空闲的子区域被更新，进一步降低冲突的可能性。

[0033] 在步骤 403，根据步骤 402 搜索结果更新应用 1 对应的子区域。

[0034] 在步骤 404，将更新后的子区域在位图中的对应位清零。

[0035] 在一例子中，假设应用 4 所对应的子区域 U4 正被其他应用所占用，则只更新应用 1 所覆盖的子区域 U1、U2 和 U3，更新后的应用 1 位图为 (0,0,0,1)，即将原位图 (1,1,1,1) 中子区域 U1、U2 和 U3 对应的二进制“1”清零。

[0036] 在步骤 405，判断位图是否有标志位被置位，即位图中子区域所对应的二进制是“1”。若有，则说明仍有未被更新的子区域，转向步骤 402，否则转向步骤 406。

[0037] 在步骤 406，更新完成。

[0038] 本发明实施例将整个屏幕根据重叠度进行分割，使得多窗口应用能够尽可能并行更新窗口，提高了系统的并行性。另外，由于在更新应用窗口时根据重叠度的高低进行子区域的更新，进一步地降低应用间更新窗口冲突的可能性，减少了窗口更新所需的时间。

[0039] 以上所述的具体实施方式，对本发明的目的、技术方案和有益效果进行了进一步详细说明，所应理解的是，以上所述仅为本发明的具体实施方式而已，并不用于限定本发明的保护范围，凡在本发明的精神和原则之内，所做的任何修改、等同替换、改进等，均应包含在本发明的保护范围之内。

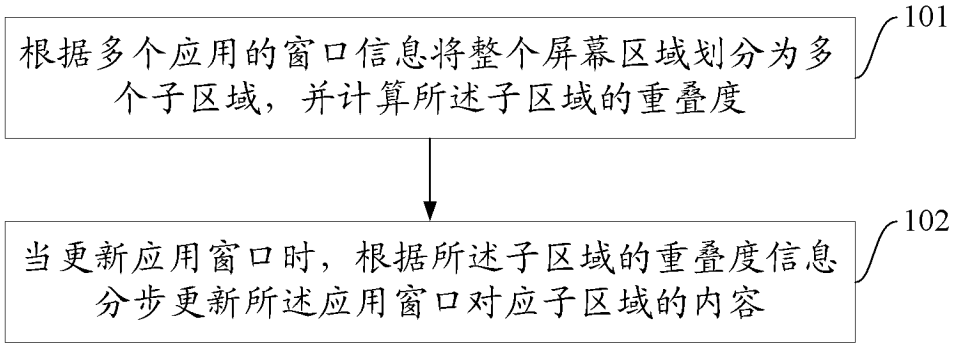

根据多个应用的窗口信息将整个屏幕区域划分为多个子区域，并计算所述子区域的重叠度
101
当更新应用窗口时，根据所述子区域的重叠度信息分步更新所述应用窗口对应子区域的内容
102

图 1

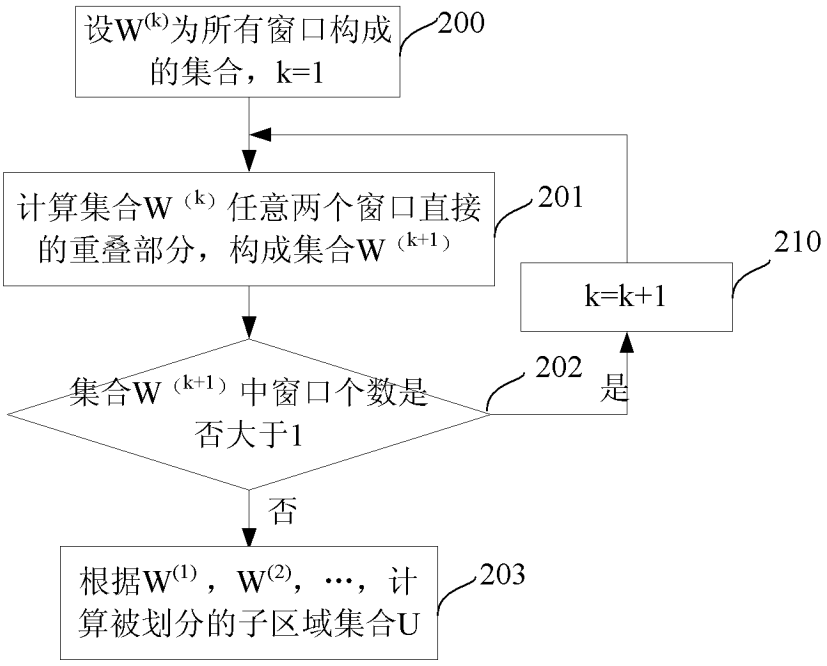

设W(k)为所有窗口构成的集合，k=1
200
计算集合W(k)任意两个窗口直接的重叠部分，构成集合W(k+1)
201
k=k+1
210
集合W(k+1)中窗口个数是否大于1
202
是
否
根据W(1)，W(2)，…，计算被划分的子区域集合U
203

图 2

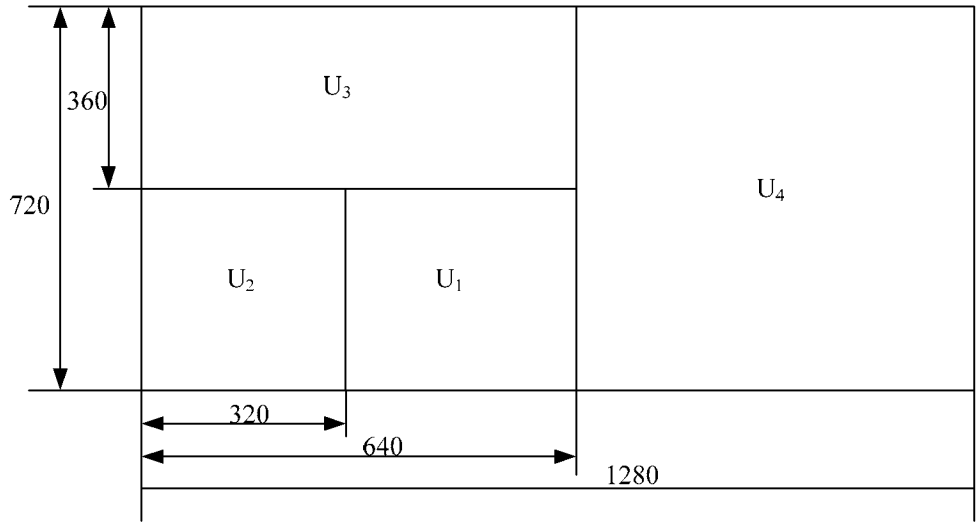

360
720
U3
U4
U2
U1
320
640
1280

图 3

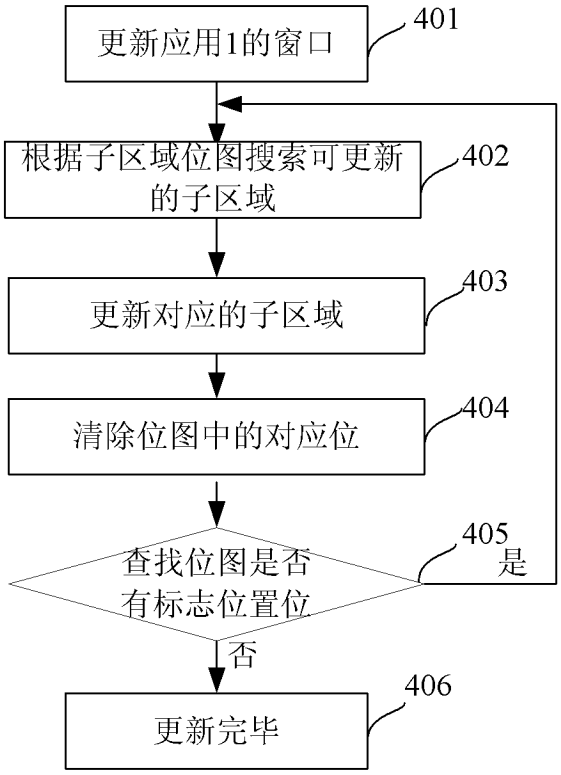

更新应用1的窗口
401
根据子区域位图搜索可更新的子区域
402
更新对应的子区域
403
清除位图中的对应位
404
查找位图是否有标志位置位
405
是
否
更新完毕
406

图 4