

一种基于场景分割的多视频序列设置算法

郭侃侃, 陈小武, 金鑫, 蒋恺

(北京航空航天大学计算机学院虚拟现实技术国家重点实验室, 北京 100083)



摘要: 协同式增强现实系统需要在共享环境中设置多个初始视频序列, 用较少的视频序列描绘多方位的真实环境信息, 以满足多用户参与协同工作任务的需要。针对该需求, 提出了一种共享增强现实环境中基于场景分割的多视频序列设置算法, 首先将场景抽象为一个平面多边形, 并将该多边形分割为多个三角形; 然后利用分割产生的对角线选择某一个设置点, 并从多边形内去除该设置点观察到的三角形集合, 同时迭代处理剩余区域。实验证明, 本算法能减少视频序列设置的个数, 并能描绘多方位的真实环境信息。

关键词: 增强现实; 协同式增强现实; 增强现实场景; 多视频序列; 场景分割

中图分类号: TP301.6

文献标识码: A

文章编号: I07030016

A scene-partition based algorithm for multi-camera placement in shared augmented reality environment

Guo Kan-kan, Chen Xiao-wu, Jin xin, Jiang Kai

(State key laboratory of virtual reality technology, School of Computer Science and Engineering, Beihang University, Beijing, 100083)

Abstract: In Collaborative Augmented Reality (CAR) systems, it is required to initialize the camera placement to meet the demands of collaborative work: how to use a few cameras to watch the whole scene of the environment, which is always regarded as an Art Gallery Problem. A scene-partition based algorithm for camera placement in shared Augmented Reality (AR) environment is presented. It firstly divides the scene into triangles, followed by sorting the vertices with the number of diagrams joining non-neighbor vertices, the more diagrams passing through the vertex, the more possibility it is a guard. After removing the areas watched by the guard, the same method is used in the rest part of the scene. Experiment results show that this algorithm can reduce the number of initial mounted cameras, while guaranteeing all points in the scene could be watched.

Key words: Augmented Reality; Collaborative Augmented Reality; AR environment; multi-camera; scene-partition

引言

支持多用户协同操作的增强现实(Augmented Reality, 简称 AR)系统表明协同式增强现实(Collaborative Augmented Reality, 简称 CAR)已成为增强现实技术的重要组成部分^[1]。一般的, 在协同工作的某一时刻, 一个视频序列的可视范围经常是有限的, 一般不包括真实环境的全部场景信息, 这使得基于单视频序列的增强现实系统难于满足多个用户在多个方位进行协同工作的需要^[2], 为了使多个用户能够共同完成预定的协作任务, 需要建立一个共享增强现实环境, 从而需要协同式增强现实系统能够描绘多个方位的、不断变化的

真实环境。其中, 最核心的问题是研究初始多视频序列的设置方法。

多视频序列的设置方法通常被抽象为画廊问题: 给定一个画廊, 需要多少台摄像机? 应该将它们放置在什么位置? 能够确保所有摄像机观察区域的总和最大。在共享增强现实环境中, 若初始设置的视频序列较多, 会增加后续的计算时间开销和较大的经济开销; 若设置的视频序列太少, 则不能保证所获视频信息能够满足协同工作的需要。通常场景被假设为一个平面多边形。于是该问题被抽象为, 在一个平面多边形中, 需要在哪些位置设置视频序列, 使得多边形中所有的点都可以被视频序列观察到, 同时使得视频序列的个数最少。该问题被证明是一个 NP-难问题^[3], 但是本文所关注的是, 通过对具体协作任务的分析, 对于一个静态的共享增强现实环境, 如何设置较少的视频序列来描绘多方位的真实环境信息, 此时该场景中的所有点被观察到, 同时使得满足该任务的视频序列个数较少。

本文提出了一种基于场景分割的多视频序列设置算法。根据协作任务的需要, 将场景抽象为一个平面多边形, 并将其分割为多个三角形; 然后利用产生的对角线来选择某一个

收稿日期: 2007-06-15

修回日期: 2007-07-12

基金项目: 国家自然科学基金课题(60503066); 预研基金课题(51***0305HK*5); 国家 973 计划课题(2006CB303007); 国家 863 计划课题(2006AA01Z311&2006AA***324); 教育部新世纪优秀人才计划课题(NCET-05-0187)。

作者简介: 郭侃侃(1983-), 男, 湖北宜昌人, 土家族, 硕士研究生, 研究方向为增强现实; 陈小武(1972-), 男, 湖南安仁人, 汉族, 副教授, 研究方向为增强现实、网络技术; 金鑫(1983-), 男, 安徽安庆人, 汉族, 博士研究生, 研究方向为增强现实; 蒋恺(1984-), 女, 河南固始人, 回族, 博士研究生, 研究方向为增强现实。

设置点, 并从多边形内去除该设置点观察到的三角形集合, 同时迭代处理剩余区域。实验结论证明, 本算法能减少视频序列设置的个数, 并能够描述多方位的真实环境信息。

1 国内外研究现状

1973 年, Victor Klee 提出著名的画廊问题: 如果给定一个含有 n 面墙的画廊, 多少个守卫是必需且有效的, 可以用于监视画廊中所有的艺术作品。该问题首先被 Chvatal 指出, $\lfloor n/3 \rfloor$ 是守卫数目的上限^[4], 对于一个含有 n 个顶点的多边形, 该多边形中所有的点总可以被观察到。Fisk 证明了该理论^[5]。而在有些情况下, 多边形往往还存在着空洞。O' Rourke 在 1987 年证明, 对于任意含有 n 个顶点和 h 个空洞的多边形, 总可以用 $\lfloor (n+2h)/3 \rfloor$ 个顶点守卫来保证多边形内的所有点被观察到^[6]。不久 Shermer 推论得出, 对含有空洞的情形, $\lfloor (n+h)/3 \rfloor$ 个守卫即可保证多边形内的所有点被观察到^[7]。

在现有增强现实系统中, 德国克雷斯勒技术部门为了表现真实环境信息, 在 4 个任意方位设置摄像头, 使得用户观察同一个共享增强现实场景^[8]。美国卡内基梅隆大学的 Kanade 等人提出一个基于多视频图像序列的真实环境表示方法, 它使用 51 个摄像机散布在场景的各个区域, 获取真实环境的视频图像序列^[9]。微软亚洲研究院的 Lou 为了使得用户能够观察到不同方位下视频的平滑切换, 其将摄像头呈弧状摆放, 为用户提供了高度交互的多方位视频服务^[10]。墨尔本大学的 Fraser 等人发现了一些设置多视频序列需要关注的因素, 相对于平行设置, 将多个视频序列朝向会聚于一个区域, 会使得系统的精确度提高; 若一个度量点在多个视频序列的捕获图像都会出现, 也会提高系统准确度^[11]。

通过对国内外研究现状的分析可以发现, 当前大多数增强现实系统根据其应用, 都假定设置的视频序列能够满足具体任务的需要, 并没有给出多视频序列设置的理论基础。因此, 本论文提出了一种共享增强现实环境中基于场景分割的多视频序列设置算法, 为协同工作任务提供理论前提。

2 基于场景分割的多视频序列设置算法

本论文提出了一种基于场景分割的多视频序列设置方法, 首先将场景抽象为一个平面多边形, 并将其分割为多个三角形; 然后利用产生的对角线来选择某一个设置点, 并从多边形内去除该设置点观察到的三角形集合, 同时迭代处理剩余区域。

2.1 算法概述

共享增强现实场景被抽象为平面上的一個简单多边形 P , 该多边形有 n 个顶点, 且任意不相邻的两条边互不相交。若某条闭线段 e 经过某个点 q , 记为 $q \in e$ 。若线段 e 完全在多边形 P 内或者为 P 的一条边, 则记为 $e \subseteq P$ 。对于所有的

视频序列, 假设其观察角度为 2π (即使视频序列观察角度有限, 也可以通过自身旋转获得 2π 的观察角度), 观察的最大距离为 d , 对于多边形内的一个点 p , 假设能够观察到的最大区域为 $V(p)$, 若多边形内一点 q 使得 pq 的连线完全在多边形内且 $\|p - q\| \leq d$, 则称 p 可以观察到 q , 记为 $q \in V(p)$ 。对于 P 内部中的一个多边形区域 Q , 若对于 Q 的任意顶点 q , 都有 $q \in V(p)$, 则可记为 $Q \subseteq V(p)$ 。

对于设置点类型的选择, 目前主要存在任意点设置、边设置以及顶点设置的情形。任意点设置可以将视频序列散布在多边形内任意的位置, 但是会带来设置点之间的视野遮挡。边设置是把视频序列放置在多边形的边上。对于边设置来讲, 即使视频序列可以旋转, 也只能获得最大为 π 的观察角度。顶点设置是把视频序列放置在多边形的顶点位置, 这样不仅能够保证设置点获得大于 π 的观察角度, 而且不会产生设置点之间的视野遮挡。因此在共享增强现实场景中, 将视频序列设置在多边形的顶点位置。

本论文只考虑在协同工作任务中所有真实物体静止的情况下, 将场景区域抽象为一个简单多边形 P 之后, 需要在 P 的哪些顶点设置视频序列用于描绘多方位的真实环境信息, 以确保 P 内部的所有点都被观察到, 同时使得视频序列的设置个数较少。根据协同工作任务的需要, 需要将整个场景 P 划分为一个个子多边形: $\{P_1, P_2, P_3, \dots, P_m\}$, 使得每个子多边形 P_i ($1 \leq i \leq m$) 的点都能刚好被一个视频序列观察到。然后根据多边形每个顶点 p 能观察的子多边形区域, 确定每个顶点的 $V(p)$, 其中 $V(p)$ 较大的顶点被确定为设置点。显然, 对于多边形的每个顶点 p , 都有一个对应的 $V(p)$, 则其可以构成一个集合 $U = \{V(p_1), V(p_2), \dots, V(p_n)\}$, 因此, 需要找出 U 的一个子集 $I \subseteq U$, 并满足 $I = P$ 。在划分场景 P 的过程中, 产生了大量对角线来连接不相邻的顶点, 若在某一个顶点处有多条对角线通过, 则该顶点被多个子多边形所拥有。可以认为, 若经过某顶点 q 的对角线越多, 该顶点 q 就可以观察到越多的子多边形, 其对应的 $V(q)$ 就越大。这样在 q 点设置一个视频序列, 就能保证 $V(q)$ 内的所有区域被观察到。选择 q 作为设置点之后, 计算 q 点的观察区域 $V(q)$ 。对于剩下的区域 $P - V(q)$, 采用同样的方法选择设置点, 一直到 P 为 0 为止。算法的主要步骤如下所示:

1) 将多边形分割为多个子多边形 $\{P_1, P_2, P_3, \dots, P_m\}$, 此时产生 1 条对角线, 对角线构成一个集合:

$$D = \{d_i | i = 1, 2, \dots, 1\} \quad (1)$$

2) 计算经过每个顶点 $p_i (i = 1, 2, \dots, n)$ 的对角线条数 $S(p_i)$, 选择最大值 $S(pk)$ 对应的顶点 pk 作为设置点。并计算 pk 点能观察到的子多边形集合 $R(pk)$ 以及 $R(pk)$ 内的对角线 $D(pk)$ 。

3) $P = P - R(pk)$, $D = D - D(pk)$, 其表示更新剩余的多边形及对角线。若 $P = 0$, 则结束。否则转 2。

```
graph LR; Input[输入] --> A[场景分割]; A --> B[确定设置点]; B --> C[计算观察区域]; C --> D[更新场景]; D --> End[结束]; D --> B;
```

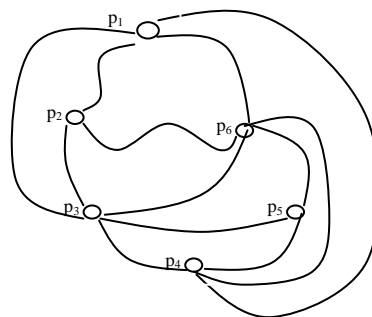
2.2 场景区域的分割

2.3 多视频序列的设置

$$R(G_k) = \{T_{ti}\} \quad (1 \leq ti \leq m, \text{ 且 } T_{ti} \subseteq V(G_k)) \quad (2)$$
$$\sum_{k=0}^i R(G_k) = \sum_{i=0}^m T_i = P \quad (3)$$

在 多 边 形 分 解 为 三 角 形 的 过 程 中 ， 会 产 生 连 线 来 连 接 多

确定设置点之后,需要计算该设置点的观察范围。自此可以构造图 $G(V, E)$, 其中 V 为多边形 P 的顶点集合, E 是 $G(V, E)$ 的边的集合。对于多边形 P 内任意两个顶点 p, q , 若 $q \in V(p)$ 且 $p \in V(q)$, 则可以构造 $\{p, q\}$ 构成 E 的一个元素。图 3 为图 2 中 P 对应的 $G(V, E)$ 。对于顶点 p_1 , 若其它任意顶点 q , 在 $G(V, E)$ 有对应的边 $\{p_1, q\}$ 或 $\{q, p_1\}$ 存在, 则 q 被 p_1 观察到。对于一个三角形 T_i , 若其三个顶点都被 p_j 观察到, 则 $T_i \subseteq V(p_j)$ 。图 3 中, $p_2 p_3 p_6$ 构成的三角形 $T_{p_2 p_3 p_6}$ 虽然没有边经过 p_1 , 但是仍然有 $T_{p_2 p_3 p_6} \subseteq V(p_1)$ 。



假设多边形 P 有 n 个顶点，分割产生了 $n-2$ 个三角形。设置多视频序列算法的主要描述如下：

$$S(pi)=F(D, pi), (i=1, 2...n) \quad (4)$$

2) 计算 $\{S(pi)(i=1, 2, \dots, n)\}$ 中的最大值:

$$S(pk) = \max(S(p1), S(p2) \dots S(pn)) \quad (5)$$

其中 $S(pk)$ 对应的顶点 pk 作为设置点。

3) 计算 pk 可以观察到的三角形的集合 $T(pk)$ 和 $T(pk)$ 内部的对角线集合 $D(pk)$ 。

$$T(pk)=\{Ti|Ti \subset V(pk), i=1, 2, \dots, n-2\} \quad (6)$$

$$\begin{aligned} D(pk) = \{di | di \subseteq Tw, \quad di \subseteq Tr, \quad Tw \subseteq T(pk), \\ Tr \subseteq T(pk), \text{ 且 } di \in D\} \end{aligned} \quad (7)$$

4) $P=P-T(pk)$, $D=D-D(pk)$ 。其表示为将 pk 观察到的三角形区域从 P 中去除, 同时更新剩余区域的对角线。

5) 如果 $P=0$, 则结束, 否则转 1.

2.4 “空洞”处理

在某些情况下, 场景 P 内存在真实物体, 真实物体内部的点并不需要被观察到, 此时将场景中的真实物体抽象表示为 P 的一个内部多边形 P' , 即 P 内部的一个“空洞”。真实物体的部分表面点可被某个视频序列观察到, 而其它表面点由于物体自身的遮挡, 并不能被观察到。对于此问题, 本算法在初始工作中利用对角线分割, 将 P' 的区域从 P 中去除, 此时只需要保证 $P-P'$ 内的所有点被观察即可。

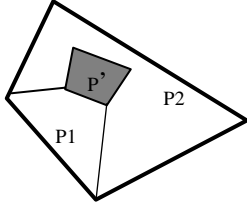


图4 “空洞”处理

如图4所示, 对于某多边形 P , 内部的灰色区域 P' 表示一个“空洞”, 通过引入对角线, 可将多边形 P 分解为两个不含“空洞”的多边形 $P1$ 和 $P2$ 。对于 $P1$ 、 $P2$, 按照节2.1中的方法确定视频序列的设置。

2.5 时间复杂度分析

在将多边形分解为三角形的过程中, 采用扫描线算法^[13], 时间复杂度为 $O(n \log n)$ 。

为了确定一个视频序列的设置点, 需要遍历多边形内的所有对角线, 若存在 $n-3$ 条对角线, 则所花费的时间为 $\Theta(n)$ 。当确定一个设置点之后, 需要计算该设置点能观察到的所有三角形, 此时对所有的三角形进行遍历, 若存在 $n-2$ 个三角形, 则时间开销为 $\Theta(n-2)$ 。则确定一个设置点及计算该设置点的观察区域, 时间开销为 $\Theta(n)$ 。

在设置视频序列后, 对于含有 n 个顶点的多边形, 分解会产生 $n-2$ 个三角区域, 假设第一个视频序列可以观察到 $k1$ 个区域, 第二个视频序列可以观察到 $k2$ 个区域, 第 m 个视频序列可以观察到 km 个区域, 此时 $k1+k2+\dots+km=n-2$ 。显然确定第一个设置点及计算该点的观察区域的总处理时间为 $\Theta(n-2)$, 第二个的处理时间为 $\Theta(n-2-k1)$, 第三个的处理时间为 $\Theta(n-2-k1-k2)$, 以此类推, 则总共的处理时间为:

$$T = \Theta(n-2) + \Theta(n-2-k1) + \Theta(n-2-k1-k2) + \dots \quad (8)$$

在最坏的情况下, 每个视频序列只能观察到一个三角形区域, 则此时的时间开销为:

$$T = \Theta(n-2) + \Theta(n-3) + \dots = \Theta(n^2). \quad (9)$$

在最好的情况下, 第一个视频序列就能观察到所有的区域, 此时的时间开销为

$$T = \Theta(n-2) = \Theta(n) \quad (10)$$

在一般的情况下, 由于在每次遍历中, 都可以看作将三角区域的集合分为两部分, 故可以构造一棵二叉树, 二叉树的层次最大为 $\log n$, 而对于每层的操作, 最大的时间开销为 $\Theta(n)$, 所以设置多视频序列的复杂度为 $O(n \log n)$ 。

综上所述, 本算法在最坏情况下, 时间开销为 $\Theta(n^2)$, 在最好情况下为 $\Theta(n)$, 一般的情形下为 $O(n \log n)$ 。

3 实验分析

所有实验均在普通 PC 机上进行(PC 机配置为 Pentium4 3.0GHz CPU, 512M 内存)。

实验一对于不同类型的多边形, 测试了本算法产生的视频序列设置(此时多边形内无空洞), 并且将本算法生成的设置结果和三染色算法^[5]进行了比较。实验效果如图5所示, 其中, 假设视频序列的观察范围 d 为无限远, 且观察角度为 2π 。圆圈点表示视频序列的位置, 细线为三角形分割产生的对角线。

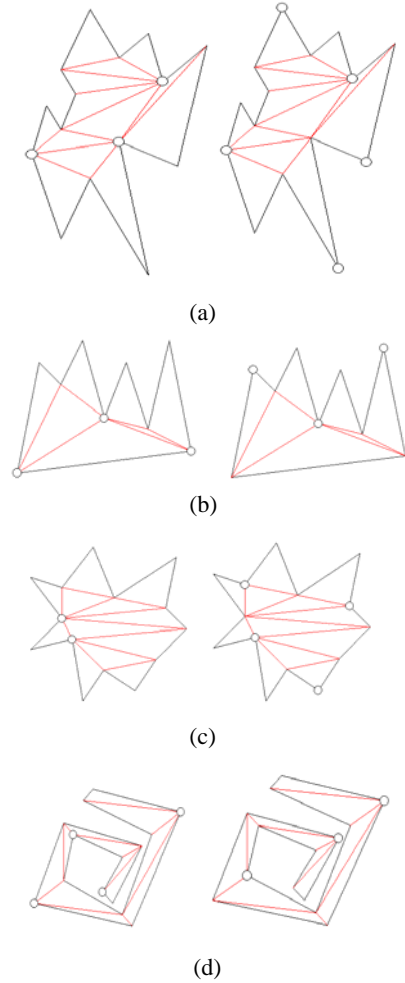


图5 (a) (b) (c) (d) 左侧为采用本算法获得的视频序列分布, 右侧为采用三染色算法(3-color)获得的视频序列分布。

结果表明, 对于多种多边形, 本算法均可以保证多边形内的所有点被生成的视频序列观察到。通过与三染色算法结果比较发现, 除了(b)中的梳状多边形, 本算法都只用较少的视频序列, 就可以观察到场景内所有的点。

实验二在多边形顶点个数不同的情况下(此时多边形内无空洞), 将本算法视频序列的设置个数和三染色算法进行

了对比。

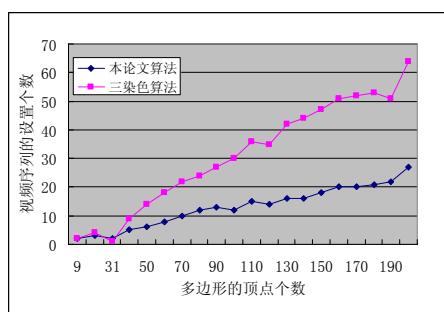


图6 多边形内无空洞时, 本论文算法和三染色算法的比较

如图6所示, 实验随机产生多个多边形, 多边形的顶点数目从9到200不等。随着多边形顶点数目的增多, 由于本论文算法充分考虑了视频序列所处位置点能够观察到的区域, 因此能够减少视频序列的设置个数。

实验三对于多边形内存在空洞的情况进行测试, 在图6的(a)(b)(c)(d)内各加入一个空洞, 此时本算法生成的视频序列也能够观察到多边形内部的所有点。如图7所示。

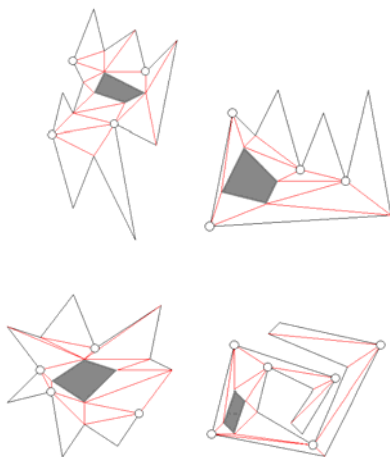


图7 图中的灰色区域为多边形中的空洞, 空洞中的点不需要被观察到。

实验四和实验二类似, 在多边形内存在空洞的情况下, 对于不同顶点个数的多边形, 测试本算法的视频序列设置结果, 并将本算法产生的结果和 Shermer^[7]推论值进行了比较。实验结果表明, 随着多边形顶点的增加, 相比较 Shermer 推论值, 本算法的设置个数较少。如图8所示。

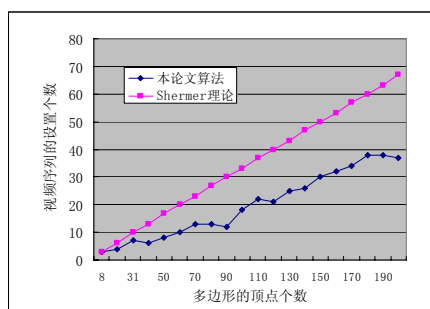


图8 多边形内存在空洞时, 本论文算法和 Shermer 推论值的比较

通过以上实验可以得, 本论文采用的算法可以减少视频序列的设置个数, 并且描绘多方位的真实环境信息。

4 结论

为了支持共享增强现实场景的协同工作任务, 本文提出了一种基于场景分割的视频序列设置算法。将场景抽象为一个平面多边形, 并将其分割为多个三角形; 然后利用产生的对角线来选择某一个设置点, 并从多边形内去除该设置点观察到的三角形集合, 同时迭代处理剩余区域。实验结果显示, 本方法只需设置较少的视频序列, 就能够描绘多方位的真实环境信息, 本算法能够满足协同工作任务的需要。

本算法已经用于共享增强现实环境的一个原型系统中, 用于指导初始视频序列的设置。在未来的工作中, 需要研究在一个动态环境下, 如何较好的设置初始视频序列, 并结合后续进入的视频序列, 为协同工作提供帮助。

参考文献:

- [1] Azuma R, Baillet Y, Behringer R, Feiner S, Julier S, MacIntyre B. Recent Advances in Augmented Reality[J]. Computer Graphics and Applications(0272-1716), IEEE, Volume: 21, Issue: 6, 2001,Nov.-Dec. Pages: 34 – 47.
- [2] 陈小武, 赵沁平. 协同式增强现实概述[J]. 计算机研究与发展, 2005, Volume 42.
- [3] D.Lee and A.Lin. Computational complexity of art gallery problems[J]. IEEE Trans. Inform. Theory(0018-9448). 1986, 32: 2760-282
- [4] V.Chvatal. A combinatorial theorem in plane geometry[J]. Journal of Combinatorial Theory(0095-8956) Ser B. 1975, 18: pp 39-41.
- [5] S. Fisk. A short proof of Chvatal's watchman theorem[J]. Journal of Combinatorial Theory(0095-8956) Ser B. 1978, 24: p. 374.
- [6] J O'Rourke. Art Gallery Theorems and Algorithms[M]. New York: Oxford University Press, 1987
- [7] T.Shermer. Recent results in Art Galleries[J]. Proceedings of the IEEE(0018-9219), 1992, September : pp 1384-1399, .
- [8] Holger T. Regenbrecht, Michael T. Wagner. Interaction in a Collaborative Augmented Reality Environment[C]. CHI 2002 Conference on Human Factors in Computing Systems, Minneapolis, USA. 2002
- [9] T.Kanade et al. Virtualized reality: digitizing a 3D time-varying event as is and in real time[C]. mixed reality: merging real and virtual worlds, Springer-Verlag. Yuichi Ohta and Hideyuki Tamura . 1999
- [10] JianGuang Lou, Hua Cai, and Jiang Li. A Real-Time Interactive MultiView Video System[C]. ACM International Conference on Multimedia (ACM MM 05), Singapore, Nov 6-11, 2005
- [11] Fraser, Network design. In Close Range Photogrammetry and Machine Vision[M]. Caithness, Scotland, U.K: Whittles Publishing, K.B Atkinson, 2001.
- [12] F.R.Garey, D.S.Johnson, F.P.Preparata, and R.E.Tarjan. Triangulating a simple polygon[J]. Inform Process Lett(0020-0190). 1978, 7: 175-179.
- [13] D.T.Lee and F.P.Preparata. Location of a point in a planar subdivision and its applications[J]. SIAM J. Comput(0097-5397). 1977, 6: 594-606.