



(12) 发明专利申请

(10) 申请公布号 CN 104748746 A

(43) 申请公布日 2015. 07. 01

(21) 申请号 201410163542. 0

G06F 3/01(2006. 01)

(22) 申请日 2014. 04. 23

G06T 17/00(2006. 01)

(66) 本国优先权数据

201310736170. 1 2013. 12. 29 CN

201310736179. 2 2013. 12. 29 CN

(71) 申请人 刘进

地址 430079 湖北省武汉市洪山区珞瑜路
129 号武汉大学测绘学院

(72) 发明人 刘进 陶重芝

(74) 专利代理机构 武汉凌达知识产权事务所
(特殊普通合伙) 42221

代理人 宋国荣

(51) Int. Cl.

G01C 21/08(2006. 01)

G01C 21/10(2006. 01)

G01C 21/20(2006. 01)

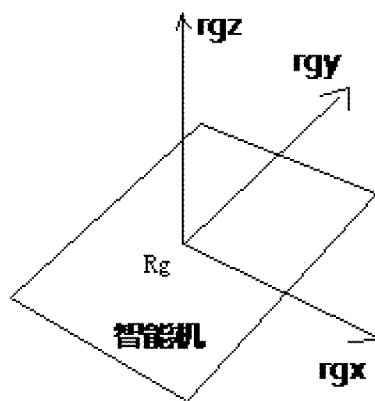
权利要求书10页 说明书17页 附图1页

(54) 发明名称

智能机姿态测定及虚拟现实漫游方法

(57) 摘要

本发明提供一种智能机姿态测定及虚拟现实漫游方法,用户在三维空间中以任意姿态摆放智能机都能在视窗内观察到与该位姿对应的周边真实或虚拟三维渲染场景。由于用户手持智能机的姿态与三维场景中渲染的三维效果完全一致,用户会体验到一种极其逼真的身临其境三维效果。利用智能机的姿态控制观察角度,手指滑动或移动控制智能机位置,用户可在三维场景中进行漫游,基于本发明可测定智能机的姿态,并实现三维数字城市漫游,商铺广告信息三维发布、虚拟现实电子商务、游戏控制、虚拟地图旅游、摄影测量测绘等。



1. 智能机姿态测定方法,包括以下步骤:

1)定义局部坐标系:局部坐标系是指原点在智能机,Z轴指向地球表面法方向,Y轴或X轴与纬线相平行的坐标系;或局部坐标系原点位于地球上任意一点,Z轴与重力方向一致,且

若智能机磁力计指示方向是北,则北代表局部坐标系 X 轴,西代表局部坐标系 Y 轴;
若智能机磁力计指示方向是东,则东代表局部坐标系 X 轴,北代表局部坐标系 Y 轴;
若智能机磁力计指示方向是南,则南代表局部坐标系 X 轴,东代表局部坐标系 Y 轴;
若智能机磁力计指示方向是西,则西代表局部坐标系 X 轴,南代表局部坐标系 Y 轴;

2)测定智能机姿态矩阵 R_g

R_g 是一个相对于局部坐标系的 3x3 单位正交矩阵,

R_g 由下法之一获得:

方法 (1) 采用重力传感器与磁力计或方向传感器

如果智能机有重力传感器,且有磁力计或方向传感器,用以下公式 (1) 计算 R_g

$$R_g = R_g \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

其中

$$R_g = \begin{bmatrix} \frac{sa_z}{L\sqrt{1-(a_y/L)^2}} & -sa_x a_y / [L^2 \sqrt{1-(a_y/L)^2}] & a_x / L \\ 0 & s\sqrt{1-(a_y/L)^2} & a_y / L \\ \frac{-sa_x}{L\sqrt{1-(a_y/L)^2}} & -sa_z a_y / [L^2 \sqrt{1-(a_y/L)^2}] & a_z / L \end{bmatrix}$$

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = R_a^T \begin{bmatrix} v_{ax} \\ v_{ay} \\ v_{az} \end{bmatrix}$$

v_{ax}, v_{ay}, v_{az} 是智能机测出来的智能机重力加速度传感器的 x, y, z 三个方向的值,或是这三个值经滤波的结果;

R_a 是重力传感器芯片相对于智能机的姿态转换矩阵;

$L = \sqrt{a_x^2 + a_y^2 + a_z^2}$ 若智能机检测到的重力加速度传感器定义为负值:当 $a_x \leq 0$ 时, $s = 1, a_x > 0$ 时, $s = -1$;若智能机检测到的重力加速度传感器定义为正值:当 $a_x \leq 0$ 时, $s = -1, a_x > 0$ 时, $s = 1$;

若采用方向传感器: $\varphi = \text{value}[0]$ 带入公式 (1) 计算 R_g ;

若采用磁力计传感器:计算

$$\begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = R_m^T \begin{bmatrix} v_{mx} \\ v_{my} \\ v_{mz} \end{bmatrix}$$

其中 R_m 是磁力传感器芯片相对于智能机的姿态转换矩阵； $\{v_{mx}, v_{my}, v_{mz}\}$ 是智能机上磁力传感器检测到的三个值或这三个值经滤波的结果；再计算

$$\begin{bmatrix} m_{0x} \\ m_{0y} \\ m_{0z} \end{bmatrix} = R_\theta^T \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}$$

利用磁力计计算方位角 φ 的三角函数为：

$$\cos \varphi = \frac{m_{0x}}{\sqrt{m_{0x}^2 + m_{0y}^2}}$$

$$\sin \varphi = \frac{m_{0y}}{\sqrt{m_{0x}^2 + m_{0y}^2}}$$

带入公式 (1) 即可算出 R_g ；

若既无磁力计，也无方向传感器， φ 角初值为 0，当模拟逆时针绕重力方向旋转时增加 φ 角，模拟顺时针绕重力方向旋转则减小 φ 角；

方法 (2) 用旋转矢量传感器

如果智能机配置有旋转矢量传感器

首先采用下述 (a) (b) (c) 中的一种方式得到 R_{g0} 矩阵

(a) 如果检测到智能机上的旋转矢量传感器只有 3 个数据 values[0], values[1], values[2]

采用以下公式得到

$$q_1 = \text{values}[0], \quad q_2 = \text{values}[1], \quad q_3 = \text{values}[2],$$

$$q_0 = \sqrt{1 - q_1^2 - q_2^2 - q_3^2}$$

则 R_{g0} 矩阵为：

$$R_{g0} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

(b) 如果检测智能机上的旋转矢量传感器有 4 个数据，

$$q_0 = \text{values}[3], \quad q_1 = \text{values}[0], \quad q_2 = \text{values}[1], \quad q_3 = \text{values}[2]$$

则 R_{g0} 矩阵为：

$$R_{g0} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix};$$

(c) 智能机上指定传感器类型为旋转矢量类型 TYPE_ROTATION_VECTOR, 实时得到旋转矢量传感器的观测值; 利用智能机上的旋转矢量矩阵函数

将传感器的观测值转换成绘图转换矩阵 R_{44} , R_{44} 矩阵的左上角 3x3 子阵是 R_{g0} ;

然后根据智能机旋转矢量传感器坐标系定义的不同, 采取以下几种情况之一得到 R_g :

若旋转矢量传感器的 X 轴指向东, Y 轴指向北, 则

$$R_g = R_{g0} \begin{bmatrix} & & & \mathbf{1} \\ & & & \\ & & & \\ -\mathbf{1} & & & \\ & & & \\ & & & \mathbf{1} \end{bmatrix}^T; \quad (2)$$

若旋转矢量传感器的 X 轴指向北, Y 轴指向西, 则

$$R_g = R_{g0}; \quad (3)$$

若旋转矢量传感器的 X 轴指向南, Y 轴指向东, 则

$$R_g = R_{g0} \begin{bmatrix} & & & -\mathbf{1} \\ & & & \\ & & & \\ & & & \\ -\mathbf{1} & & & \\ & & & \\ & & & \\ & & & \mathbf{1} \end{bmatrix}^T; \quad (4)$$

方法 (3) 采用陀螺仪、加速度计与磁力计迭代

第 1 步选用方法 (1) 或方法 (2) 计算 R_g 的初值, 将 R_g 转换成 4 元数 q_0, q_1, q_2, q_3 作为以下第 2-7 步迭代的初值

第 2 步 设定 $exInt, eyInt, ezInt$ 原始值为 0

$exInt=0, eyInt=0, ezInt=0$

第 3 步 根据接收到的磁力计矢量 $\{mx, my, mz\}$, 得到正确磁场矢量 $\{wx, wy, wz\}$

先将矢量 $\{mx, my, mz\}$ 替换成将其单位化以后的矢量 $\{mx, my, mz\} / \sqrt{m_x^2 + m_y^2 + m_z^2}$

得到局部坐标系下磁场的正确的磁场方向矢量 $\{bx, 0, bz\}$

$hx = 2 \times mx \times (0.5 - q_2 \times q_2 - q_3 \times q_3) + 2 \times my \times (q_1 \times q_2 - q_0 \times q_3) + 2 \times mz \times (q_1 \times q_3 + q_0 \times q_2);$

$hy = 2 \times mx \times (q_1 \times q_2 + q_0 \times q_3) + 2 \times my \times (0.5 - q_1 \times q_1 - q_3 \times q_3) + 2 \times mz \times (q_2 \times q_3 - q_0 \times q_1);$

$hz = 2 \times mx \times (q_1 \times q_3 - q_0 \times q_2) + 2 \times my \times (q_2 \times q_3 + q_0 \times q_1) + 2 \times mz \times (0.5 - q_1 \times q_1 - q_2 \times q_2);$

$bx = \sqrt{h_x^2 + h_y^2};$

$bz = hz$

再转换到正确磁场矢量 $\{wx, wy, wz\}$

$wx = 2 \times bx \times (0.5 - q_2 \times q_2 - q_3 \times q_3) + 2 \times bz \times (q_1 \times q_3 - q_0 \times q_2);$

$$wy = 2 \times bx \times (q1 \times q2 - q0 \times q3) + 2 \times bz \times (q0 \times q1 + q2 \times q3);$$

$$wz = 2 \times bx \times (q0 \times q2 + q1 \times q3) + 2 \times bz \times (0.5 - q1 \times q1 - q2 \times q2);$$

第 4 步 根据接收到的重力加速度计数据 ax, ay, az , 和 $\{wx, wy, wz\}$

得到误差矢量 $\{ex, ey, ez\}$ 并计算其累计值 $exInt, eyInt, ezInt$

先将矢量 $\{ax, ay, az\}$ 替换成将其单位化以后的矢量 $\{ax, ay, az\} / \sqrt{a_x^2 + a_y^2 + a_z^2}$

$$vx = 2 * (q1 * q3 - q0 * q2); \quad vy = 2 * (q0 * q1 + q2 * q3); \quad vz = q0 * q0 - q1 * q1 - q2 * q2 + q3 * q3;$$

$$ex = (ay \times vz - az \times vy) + (my \times wz - mz \times wy);$$

$$ey = (az \times vx - ax \times vz) + (mz \times wx - mx \times wz);$$

$$ez = (ax \times vy - ay \times vx) + (mx \times wy - my \times wx);$$

计算误差累计值

$exInt$ 替换为 $exInt + ex \times Ki$; $eyInt$ 替换为 $eyInt + ey \times Ki$; $ezInt$ 替换为 $ezInt + ez \times Ki$;

其中 Ki 为一可调节的正系数, Ki 在 0.00001 至 0.5 中任意选取;

第 5 步 根据误差矢量 $\{ex, ey, ez\}$ 及其累计值纠正陀螺仪数据 $\{gx0, gy0, gz0\}$

假设智能机读出当前的一组陀螺仪数据为 $\{gx0, gy0, gz0\}$ $gx = gx0 + Kp \times ex + exInt$; $gy = gy0 + Kp \times ey + eyInt$; $gz = gz0 + Kp \times ez + ezInt$;

其中 Kp 为一可调节的正系数

Kp 在 0.000001 至 20.0 中任意选取;

第 6 步 根据陀螺仪数据 gx, gy, gz 修正四元数

随着不断接收到陀螺仪数据 gx, gy, gz , 对 4 元数按如下方式修正, $halfT$ 为修正周期, $halfT = 0.00001 \sim 10.0$,

q_0 替换为 $q_0 + (-q_1 \times gx - q_2 \times gy - q_3 \times gz) \times halfT$; q_1 替换为 $q_1 + (q_0 \times gx - q_3 \times gy + q_2 \times gz) \times halfT$; q_2 替换为 $q_2 + (q_3 \times gx + q_0 \times gy - q_1 \times gz) \times halfT$; q_3 替换为 $q_3 + (-q_2 \times gx + q_1 \times gy + q_0 \times gz) \times halfT$;

第 7 步 输出 R_g 矩阵和四元数

将四元数 $\{q_0, q_1, q_2, q_3\}$ 单位化成 $\{q_0, q_1, q_2, q_3\} / \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$ 输出

4 元数转 R_g 矩阵公式如下

$$\begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

第 8 步 回到第 3 步继续接收陀螺仪运动数据更新姿态 4 元数 $q_0 \sim q_3$, 在循环的过程中每次到第 7 步都能输出当前的 R_g 矩阵和四元数。

2. 智能机显示屏的姿态测定方法, 包括以下步骤:

智能机显示屏的姿态矩阵为

$$R = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

r_x 是智能机显示屏像素阵列 x 轴方向在局部坐标系中的分量；

r_y 是智能机显示屏像素阵列 y 轴方向在局部坐标系中的分量；

r_z 是垂直于智能机显示屏表面方向向智能机显示屏内部朝前看的视线方向在局部坐标系中的分量；

1) 按权利要求 1 的方法测定智能机姿态, 并得到智能机的姿态矩阵 R_g ；

2) 计算 $R = \Delta R \times R_g$ ；得到智能机显示屏的姿态矩阵；

ΔR 为智能机显示屏姿态矩阵 R 相对于智能机姿态矩阵 R_g 的一个固定的方向的转换, 智能机自身 Z 轴定义为智能机加速度计的 z 轴方向,

当显示屏观察方向和智能机自身 Z 轴方向刚好相反时 $\Delta R = \begin{bmatrix} 1 & & \\ & -1 & \\ & & -1 \end{bmatrix}$ ；

当显示屏观察方向和智能机自身 Z 轴方向完全一致时 $\Delta R = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}$ ；

否则可在智能机上选择一个与智能机显示屏姿态一致的摄像头, 即摄像头的 xyz 三轴分别与显示屏的 xyz 三轴平行且方向一致, 采用摄影测量后方交会方法测量出智能机摄像头相对于局部坐标系的姿态矩阵 R_A , 并用权利要求 1 方法测得此时智能机的姿态矩阵 R_g , 得到 $\Delta R = R_A R_g^{-1}$ 。

3. 基于智能机姿态的虚拟现实漫游方法, 包括以下步骤：

1) 采用权利要求 2 方法得到智能机显示屏的姿态矩阵 R ,

2) 移动控制漫游观察者在局部坐标系中的位置

定义漫游观察者是在局部坐标系中的一个具备自身位置、姿态且用于观察周边场景的对象；智能机显示屏所渲染的场景就是漫游观察者在局部坐标系中所应看到的场景图像；

定义漫游观察者在局部坐标系中的坐标为 X_d, Y_d, Z_d

用以下三种模式之一控制漫游观察者在虚拟场景中的移动：

(1) 模式 1: 普通二维移动

当手指按 $\{\Delta X, \Delta Y\}$ 方向滑过智能机显示屏平面的时候, 相当于局部坐标系下, X_d 增加 $V_x = \Delta X r_{11} + \Delta Y r_{21}$ ； Y_d 增加 $V_y = \Delta X r_{12} + \Delta Y r_{22}$ ； Z_d 不变；

(2) 模式 2: 智能二维移动

当手指按 $\{\Delta X, \Delta Y\}$ 方向滑过智能机显示屏平面的时候, 采用如下方法：

首先选择 R 矩阵右侧一列中的 r_{13}, r_{23}, r_{33} 中哪一个分量的绝对值最大,

如果 $|r_{33}|$ 最大, X_d 增加 $V_x = \Delta X r_{11} + \Delta Y r_{21}$ Y_d 增加 $V_y = \Delta X r_{12} + \Delta Y r_{22}$

如果 $|r_{23}|$ 最大, X_d 增加 $V_x = \Delta X r_{11} \pm \Delta Y r_{31}$ Y_d 增加 $V_y = \Delta X r_{12} \pm \Delta Y r_{32}$

$r_{23} \geq 0$ 时, \pm 取 +; $r_{23} < 0$ 时 \pm 取 - 或者 $r_{23} \geq 0$ 时, \pm 取 -; $r_{23} < 0$ 时 \pm 取 +

如果 $|r_{13}|$ 最大, X_d 增加 $V_x = \pm \Delta X r_{31} + \Delta Y r_{21}$ Y_d 增加 $V_y = \pm \Delta X r_{32} + \Delta Y r_{22}$

$r_{13} \geq 0$ 时, \pm 取 +; $r_{13} < 0$ 时 \pm 取 - 或者 $r_{13} \geq 0$ 时, \pm 取 -; $r_{13} < 0$ 时 \pm 取 +;

(3) 模式 3: 三维漫游移动

因为智能机显示屏只有二维, 单个手指滑动只能进行二维操作, 单指划过手机显示屏时 ΔX 和 ΔY 与滑动位移成正比; 此时操作智能机的用户双指滑动触碰智能机操作实现 3 维操作: 采用双指合拢相当于 $\Delta Z < 0$, 双指拉开相当于 $\Delta Z > 0$, 或者双指合拢相当于 $\Delta Z > 0$, 双指拉开相当于 $\Delta Z < 0$; 当智能机接收到手指控制信息 $\{\Delta X, \Delta Y, \Delta Z\}$ 的时候,

X_d 增加 $V_x = \Delta X r_{11} + \Delta Y r_{21} + \Delta Z r_{31}$ Y_d 增加 $V_y = \Delta X r_{12} + \Delta Y r_{22} + \Delta Z r_{32}$ Z_d 增加 $V_z = \Delta X r_{13} + \Delta Y r_{23} + \Delta Z r_{33}$;

3) 设置漫游观察者在局部坐标系里的位置和方向:

根据漫游观察者的局部坐标系坐标 X_d, Y_d, Z_d , 以及步骤 1) 测定得到的姿态矩阵 R , 利用 OpenGL, OpenES, OS X OpenGL, Direct3D 或 osg 绘图软件库中用于设定相机位姿的图形库函数, 设置漫游观察者在局部坐标系里的位置和方向:

需要设定以下几个量:

(1) 漫游观察者自身观察的垂直视场角 θ , 就是智能机显示屏上边缘中点的视线方向与下边缘中点的视线方向的夹角;

(2) 漫游观察者所在的位置 X_d, Y_d, Z_d , 由步骤 2) 漫游方法修改;

(3) 漫游者观察方向 $\{r_{31}, r_{32}, r_{33}\}$ 或者用户注视点坐标 $(X_d + r_{31}, Y_d + r_{32}, Z_d + r_{33})$

(4) 漫游者自身向上方向的矢量 $\{-r_{21}, -r_{22}, -r_{23}\}$

其中步骤 (3) 和 (4) 中的 r_{ij} 是智能机显示屏姿态 R 矩阵中的第 i 行 j 列, $i=2$ 或 3 , $j=1, 2$ 或 3 ;

4) 在局部坐标系下渲染绘制三维场景

预先在智能机的显示屏内设置一个 3 维渲染图层进行三维场景渲染;

这里被绘制的三维场景中的点、线、面和物体模型的几何坐标都要转换成局部坐标系下的坐标 $[X_i \ Y_i \ Z_i]^T$, 即与漫游者的坐标 X_d, Y_d, Z_d 在同一个局部坐标系下; 用 OpenGL, OpenES, OS X OpenGL, Direct3D 或 osg 绘图软件库实现局部坐标系下的三维绘图。

4. 根据权利要求 3 所述的方法, 其特征在于: 步骤 4) 中几何坐标转换到局部坐标系下之后, 再按如下三种方法之一渲染绘制使得局部坐标系中的一个标有广告符号、文字的纹理多边形面或广告牌的正面始终正对着漫游观察者:

方法 1:

假设广告符号、文字的纹理多边形面或广告牌的局部坐标系坐标在 $X_i \ Y_i \ Z_i$, 按下法设置模型绘制转换矩阵 M_{44} :

$$\Delta X_i = X_i - X_d \quad \Delta Y_i = Y_i - Y_d$$

M_{44} 是一个 4×4 矩阵, 其中每一元素的值如下设定

$$M_{44} = \begin{bmatrix} \Delta X_i / \Delta L & \Delta Y_i / \Delta L & 0 & 0 \\ -\Delta Y_i / \Delta L & \Delta X_i / \Delta L & 0 & 0 \\ 0 & 0 & 1 & 0 \\ X_i & Y_i & Z_i & 1 \end{bmatrix}$$

$$\Delta L = \sqrt{\Delta X_i^2 + \Delta Y_i^2}$$

将 M_{44} 设定为模型转换矩阵 ; 在 $y-z$ 平面上绘图即可在虚拟场景中让广告符号、文字的纹理多边形面或广告牌永远正对漫游观察者 ;

方法 2 :

(a) 假设广告符号、文字的纹理多边形面或广告牌的局部坐标系坐标在 $X_i Y_i Z_i$, 按如下方法设置模型绘制转换矩阵 M_{44} :

$$\Delta X_i = X_i - X_d \quad \Delta Y_i = Y_i - Y_d$$

M_{44} 是一个 4×4 矩阵, 其中每一元素的值如下设定

$$M_{44} = \begin{bmatrix} -\Delta Y_i / \Delta L & \Delta X_i / \Delta L & 0 & 0 \\ \Delta X_i / \Delta L & \Delta Y_i / \Delta L & 0 & 0 \\ 0 & 0 & 1 & 0 \\ X_i & Y_i & Z_i & 1 \end{bmatrix}$$

将 m 设定为模型转换矩阵 ;

(b) 在 $x-z$ 平面上绘图即可在虚拟场景中让广告牌永远正对漫游观察者 ;

方法 3 :

假设广告符号、文字的纹理多边形面或广告牌的局部坐标系坐标在 $X_i Y_i Z_i$, 按如下方法设置模型绘制转换矩阵 M_{44} :

$$\Delta X_i = X_i - X_d \quad \Delta Y_i = Y_i - Y_d$$

M_{44} 是一个 4×4 矩阵, 其中每一元素的值如下设定

$$M_{44} = \begin{bmatrix} -\Delta Y_i & \Delta X_i & 0 & 0 \\ \Delta X_i & \Delta Y_i & 0 & 0 \\ 0 & 0 & \Delta L & 0 \\ X_i & Y_i & Z_i & 1 \end{bmatrix}$$

将 m 设定为模型转换矩阵 ; 在 $x-z$ 平面上绘图即可在虚拟场景中让多边形面或广告牌永远正对漫游观察者。

5. 根据权利要求 3 所述的方法, 其特征在于 :

在所述步骤 3) 和步骤 4) 之间增加如下操作 :

(a) 获取智能机的高度 h_s , 纬度 θ_s 和经度 ϕ_s 。

θ_s 是智能机的纬度, ϕ_s 是智能机的经度, 由智能机上自带的 GPS 接受模块获取 ;

(b) 计算智能机所在经纬度的 R_s

$$R_s = \begin{bmatrix} -\sin \theta_s \cos \phi_s & -\sin \theta_s \sin \phi_s & \cos \theta_s \\ \sin \phi_s & -\cos \phi_s & 0 \\ \cos \theta_s \cos \phi_s & \cos \theta_s \sin \phi_s & \sin \theta_s \end{bmatrix}$$

(c) 计算智能机所在纬度 θ_s 经度 ϕ_s 高度 h_s 的地心坐标系坐标 T_s 。

利用大地测量学中经典的转换公式

$$\begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} = T_s = \begin{bmatrix} F_x(h_s, \theta_s, \phi_s) \\ F_y(h_s, \theta_s, \phi_s) \\ F_z(h_s, \theta_s, \phi_s) \end{bmatrix} = \begin{bmatrix} (N_s + h_s) \cos \theta_s \cos \phi_s \\ (N_s + h_s) \cos \theta_s \sin \phi_s \\ (N_s(1 - e^2) + h_s) \sin \theta_s \end{bmatrix} \text{ 得到 } T_s ;$$

其中

h_s 是智能机离地面高度

$$N_s = a / \sqrt{1 - e^2 \sin^2 \theta_s} \quad N_s \text{ 是随纬度变化的卯酉圈曲率半径；}$$

$$e = \sqrt{a^2 - b^2} / a$$

a, b 是地球椭球的长短轴, 是大地测量学中的已知常量；

e 是偏心率, 是定值；

(d) 根据地球上任何一个点的经度 ϕ_i 、纬度 θ_i 和高程 h_i 得到该点地心坐标系坐标

$$T_{0i} = [X_{0i} \ Y_{0i} \ Z_{0i}]^T$$

利用大地测量学中经典的转换公式

$$\begin{bmatrix} X_{0i} \\ Y_{0i} \\ Z_{0i} \end{bmatrix} = T_{0i} = \begin{bmatrix} F_x(h_i, \theta_i, \phi_i) \\ F_y(h_i, \theta_i, \phi_i) \\ F_z(h_i, \theta_i, \phi_i) \end{bmatrix} = \begin{bmatrix} (N_i + h_i) \cos \theta_i \cos \phi_i \\ (N_i + h_i) \cos \theta_i \sin \phi_i \\ (N_i(1 - e^2) + h_i) \sin \theta_i \end{bmatrix} \text{ 得到 } T_{0i} ;$$

其中

h_i 是点离地面高度

$$N_i = a / \sqrt{1 - e^2 \sin^2 \theta_i} \quad N_i \text{ 是随纬度变化的卯酉圈曲率半径；}$$

$$e = \sqrt{a^2 - b^2} / a$$

a, b 是地球椭球的长短轴, 是大地测量学中的已知常量；

e 是偏心率, 是定值；

如果已知一个物体所在经纬度 ϕ_i 和 θ_i , 通过本步骤计算出它的地心坐标系坐标 T_{0i} ；

(e) 地心坐标系坐标 T_{0i} 转换为局部坐标系中的坐标 $(X_i \ Y_i \ Z_i)^T$

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = R_q(T_{0i} - T_s) - \Delta T$$

ΔT 是局部坐标系中智能机显示屏的几何中心相对于智能机 GPS 接收天线的局部坐标系偏移。

6. 根据权利要求 3 所述的方法, 其特征在于：

权利要求 3 步骤 2) 中当手指滑动的时候, 根据手指滑动速度大小给予一个初速度 $\{V_x, V_y, V_z\}$, X_d, Y_d, Z_d 将不停地增加速度分量 V_x, V_y, V_z , 此过程中让速度分量 V_x, V_y, V_z 随时间推移不断衰减, 其中 V_z 在权利要求 3 步骤 2) 模式 (1) 或 (2) 情况下恒等于零；

让速度体现到局部坐标系下的漫游位移

X_d 增加 V_x

Y_d 增加 V_y

Z_d 增加 V_z

漫游观察者在虚拟场景中移动过程中速度大小不断衰减,

采用以下方法使 V_x, V_y, V_z 不停地衰减：

1) 运动过程中 V_x, V_y, V_z 自身分别不断地乘以 $k=0.1 \sim 0.9999$ 进行衰减,

运动过程中 V_x 自身不断地乘以 k , 直到 V_x 减小为 0 为止；

运动过程中 V_y 自身不断地乘以 k , 直到 V_y 减小为 0 为止;

运动过程中 V_z 自身不断地乘以 k , 直到 V_z 减小为 0 为止;

或者

2) 运动过程中 V_x , V_y , V_z 自身分别不断地进行以下增量操作

V_x 自身不断地增加 $-V_x*k$, 直到 V_x 减小为 0 为止;

V_y 自身不断地增加 $-V_y*k$, 直到 V_y 减小为 0 为止;

V_z 自身不断地增加 $-V_z*k$, 直到 V_z 减小为 0 为止;

其中 $k=0.0001\sim 0.5$

以上修改 X_d, Y_d, Z_d 及 V_x, V_y, V_z 的过程不断循环即实现局部加速度漫游效果。

7. 根据权利要求 3 所述的方法, 其特征在于:

在步骤 3) 和步骤 4) 之间或在步骤 4) 之后, 将智能机的显示屏设定为二维绘图模式并设置若干个二维标签, 针对每一个二维标签, 按如下步骤显示:

1) 首先确定二维标签的中心位置 u, v ;

如果标签用于标注三维场景中局部坐标系坐标为 (X, Y, Z) 的目标, 则根据以下公式确定目标显示的中心位置 u, v

$$Z_0 = r_{11}(X - X_d) + r_{12}(Y - Y_d) + r_{13}(Z - Z_d)$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} c_x + f \frac{r_{11}(X - X_d) + r_{12}(Y - Y_d) + r_{13}(Z - Z_d)}{Z_0} \\ c_y + f \frac{r_{21}(X - X_d) + r_{22}(Y - Y_d) + r_{23}(Z - Z_d)}{Z_0} \end{bmatrix} \quad \text{式中: } c_x, c_y \text{ 是智能机显示屏中心点}$$

的坐标, 若显示屏是矩形, $c_x = \text{屏宽} / 2$, $c_y = \text{屏高} / 2$;

$r_{11} \sim r_{33}$ 是 R 矩阵对应行列的元素;

$f = \frac{h/2}{\tan(\theta/2)}$ h 是显示屏上下边缘垂直像素长, θ 是漫游观察者自身观察的垂直视角

X_d, Y_d, Z_d 是漫游观察者在局部坐标系中的坐标;

如果满足 $Z_0 > 0$ 且 u, v 在显示屏范围内, 则 u, v 保持不变,

否则需要按下法修正 u, v

若 $Z_0 = 0$ 则将 $[u \ v]$ 替换为从点 (c_x, c_y) 出发

按矢量 $\{r_{11}(X - X_d) + r_{12}(Y - Y_d) + r_{13}(Z - Z_d), r_{21}(X - X_d) + r_{22}(Y - Y_d) + r_{23}(Z - Z_d)\}$

方向的射线与显示屏边缘相交的 2 维点; 若 $Z_0 > 0$ 则将 u, v 替换成从点 (c_x, c_y) 到点 (u, v) 的连线与显示屏边缘相交的 2 维点; 若 $Z_0 < 0$ 则将 u, v 替换成从点 (u, v) 到点 (c_x, c_y) 延长线与显示屏边缘相交的 2 维点;

如果二维标签不是用于标注目标, u, v 为显示屏范围内的一个由界面设计者设定的像素坐标, 或是用户手指触摸或鼠标点击的显示屏像素坐标;

2) 按下述方法 (1) 或 (2) 计算矢量 $\{Q_x, Q_y\}$

方法 (1) $Q_x = -r_{23}/L_a$ $Q_y = r_{13}/L_a$

其中 r_{23}, r_{13} 是权利要求 2 所述的方法中的 R 矩阵对应行列的元素,

$$L_a = \sqrt{r_{13}^2 + r_{23}^2}$$

方法 (2) 利用权利要求 1 方法 (1) 所述加速度计的前两个输出值 a_x, a_y 得到

$$Q_x = -a_y/L_a \quad Q_y = a_x/L_a, \quad L_a = \sqrt{a_x^2 + a_y^2}$$

3) 在二维平面上经过以下矩阵旋转变换以后再绘制二维标签 :

$$X' = \begin{bmatrix} u \\ v \end{bmatrix} \pm \begin{bmatrix} Q_x & Q_y \\ -Q_y & Q_x \end{bmatrix} \left(X - \begin{bmatrix} u \\ v \end{bmatrix} \right)$$

二维标签上每一点 X , 经过上式变换到坐标 X' 显示出来 ;

式中 \pm 取 + 是正放显示, 取 - 是倒放显示, 标签的水平轴始终平行于地面。

8. 根据权利要求 3 所述的方法, 其特征在于 :

所述三维场景为一个空心的三维形体, 三维形体为空心球, 空心立方体、空心柱体、空心正多面体或空心锥体, 三维形体的内壁以全景图作为纹理进行渲染 ; 所述漫游观察者在虚拟场景中位于三维形体内部的某一点, 从三维形体内部观察三维形体的内壁, 实现利用智能机显示屏姿态控制全景浏览。

9. 控制局部坐标系中三维物体姿态的方法, 其特征在于 :

(1) 按权利要求 1 方法得到的智能机姿态矩阵 R_g ;

(2) 利用智能机姿态矩阵 R_g 控制局部坐标系中三维物体的姿态 :

在绘制三维物体之前, 将 R_g 矩阵设定为仿真软件系统中的模型转换矩阵 M_{44} 的左上角

$$3 \times 3 \text{ 矩阵 } M_{44} = \begin{bmatrix} R_g & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}$$

即实现利用智能机姿态控制局部坐标系中三维物体的姿态。

智能机姿态测定及虚拟现实漫游方法

技术领域

[0001] 本发明涉及信息技术领域,具体涉及智能机姿态测定及虚拟现实漫游方法。

背景技术

[0002] 虚拟现实技术是让用户沉浸在智能机显示屏展现的虚拟或真实三维场景中,将虚拟目标标签叠加到其中,提示交互信息给用户。传统的三维虚拟场景,如数字三维城市、游戏等应用已经非常成熟,但是在三维场景里漫游却主要通过鼠标、键盘、滚动球等传统控制设备,控制设备的方向和观察到的实际场景无法直观的对应,渲染设备庞大而笨拙,这都会给操作者带来极大不便,影响了这些三维数字技术的广泛应用。因此迫切需要一种将控制设备与显示设备合二为一的非常直观轻便的智能机定姿态与渲染技术。本发明正解决了这个问题。

[0003] 由于智能机姿态不一样,同样位置看到的现实场景不一样,智能机姿态精度影响导航精度和虚拟现实漫游仿真精度,因此智能机姿态的精确测定具有重要意义。虚拟现实应用中对智能机姿态精度提出了更高要求。传统的智能机姿态测定,主要利用方向传感器单轴航向角数据信息,系统对智能机姿态摆放有特定要求,如要求智能机平行于地面,或垂直于地面摆放,而虚拟现实应用中用户手持智能机的姿态千变万化,并不一定严格平行或者垂直于地面,这将导致导航方位不准。此外智能机姿态测定的误差将带来虚拟现实漫游误差,影响用户体验,因此迫切需要一种考虑智能机 X-Y-Z 三轴旋转的姿态测定算法。

[0004] 基于位置的服务 LBS 日益受关注,但用户更关心的不仅是位置,而是自己与目标间的方向和距离。很多老年人和方向感不强的人很难将实际看到的生活场景与地图中的东南西北联系在一起,因此智能机任意角度摆放情况下,叠加在实景视频上的目标信息及其相关服务将给用户带来所见即所指的极大便利,将成为未来信息服务的发展新趋势,我们将这种服务称为“智能机全方位虚拟现实或增强现实信息服务”。应用包括:

1. 为商铺以及供需双方发布相对于其他海量用户的方位和距离提供信息;
2. 为用户相对于指定地名,指定单位的方位与距离提供可视化信息服务;
3. 为用户之间相互联络提供方位和距离可视化信息;
4. 为车、船、人、物流提供相对于目的地的方位和距离可视化信息。

[0005] 5. 快递、送货,用于电子商务的最终递送环节。邮递员通过智能机即可知道收货人,收件人位置,收件人通过智能机知道邮递员位置和方向。

[0006] 6. 提供一种全新三维场景、数字城市、数字地球、游戏漫游模式,不需手指触碰控制方向,而采用直观的智能机姿态控制保证渲染的地图与三维场景对应。

[0007] 7. 提供了一种全新的地图浏览查询模式,不需要用手指触碰方式,而是采用直观的智能机姿态控制保证渲染出的地图与实景对应。

[0008] 传统的导航服务是地图导航服务,用户看到的是智能机姿态与周边实景不对应的地图,用户需要费力的去把实景与地图联想起来才能得到正确的方向。

[0009] 传统导航服务依据经纬度网格组成的 2 维坐标系进行导航,地球当做经纬度组成

的一个平面处理,因为相同维度差对应的距离偏移 \geq 经度差对应的距离偏移,所以不同地区导航算法需进行经纬比例调整,只适合于中低纬度地区的小区域,不适合全球,极地附近这种方法已经无法再适用,非常不便于全球化普及。而本发明采用局部坐标系转换后能把全球任何位置的三维场景很好地展现出来。

[0010] 传统的姿态检测技术主要用于飞机,汽车等大型设备,追求很高精度,用于军事、工业等领域,而且非常贵重和笨重,而本发明主要考虑用廉价轻便的智能机终端实现普通个人周边地区的视频信息叠加服务。

[0011] 本发明不仅考虑方向传感器的 value[0] 得到的航向角,在有其他传感器,如重力加速度磁力计和陀螺仪情况下,还考虑了其他几种姿态传感器捕获到的数据,使得检测到的姿态更加精确,在没有这些传感器条件下也能得到粗略姿态。本发明可以在智能机,诸如 Andriod,苹果系统上方便地实现虚拟现实漫游、地图浏览功能。

[0012] 考虑到当前智能机传统应用的各种缺陷和不足,本发明提供一种考虑智能机 3 轴旋转任意姿态测定的虚拟现实或增强现实三维渲染及漫游方法,用户在三维空间中以任意姿态摆放智能机都能在视窗内观察到与该位姿对应的周边真实或虚拟三维渲染场景。由于用户手持智能机的姿态与三维场景中渲染的三维效果完全一致,用户会体验到一种极其逼真的身临其境三维效果。

[0013] 因此基于本发明可以实现虚拟现实电子商务、虚拟现实商铺广告信息实景三维渲染发布、三维数字城市漫游,实景三维地图标注,实景三维游戏、摄影测量测绘等。同时商铺也能借助服务平台广泛发布其广告信息。

[0014] 基于本发明可实现电子商务服务,用户客户端之间,用户客户端与商铺之间可以用视频实景预览观察相互之间的位置、方位关系,同时商铺也能借助服务平台广泛发布其广告信息。

[0015] 本发明中智能机包括一切含显示显示屏和姿态传感的智能设备,如手机,平板电脑,笔记本电脑,带姿态传感和显示屏的摄像头,相机,智能显示器,智能电视机显示屏、以及各种可穿戴设备。

[0016]

发明内容

[0017] 本发明所要解决的问题是提供智能机姿态测定及虚拟现实漫游方法,该方法使用户在智能机上获得便捷、直观的可视化信息。

[0018] 本发明提供的技术方案是,智能机姿态测定及虚拟现实漫游方法,包括以下内容:

智能机姿态测量方法的输入输出定义

输入:

方法(1) 重力加速度计传感器的三个值 a_x 、 a_y 、 a_z , 要求配置最低,适用性最广

磁力计传感器的三个值 $\{m_x, m_y, m_z\}$ 或 方向传感器的航向角 value[0]

方法(2) 旋转矢量传感器的三个或四个输出。需旋转矢量传感器

方法(3) 重力加速度计,磁力计 和 陀螺仪。另需陀螺仪输出:

完整描述智能机局部坐标系姿态的矩阵 $R_g = \begin{bmatrix} r_{g11} & r_{g12} & r_{g13} \\ r_{g21} & r_{g22} & r_{g23} \\ r_{g31} & r_{g32} & r_{g33} \end{bmatrix} = \begin{bmatrix} r_{gx} \\ r_{gy} \\ r_{gz} \end{bmatrix}$

R_g 矩阵是一个 3x3 单位正交矩阵，由三行组成，每行一个三维矢量，三个矢量分别为 r_{gx}, r_{gy}, r_{gz} 。第一行 r_{gx} 是智能机自身 x 轴在局部坐标系下的单位矢量；

第二行 r_{gy} 是智能机自身 y 轴在局部坐标系下的单位矢量；

第三行 r_{gz} 是智能机自身 z 轴在局部坐标系下的单位矢量。

[0019] 如图 1 所示，智能机姿态矩阵 R_g 的三个行矢量 r_{gx}, r_{gy}, r_{gz} 是其自身三轴在局部坐标系下的三个单位矢量。

[0020] 本发明智能机姿态测定方法，包括以下步骤：

测定智能机姿态矩阵 R_g 。 R_g 是一个 3x3 单位正交矩阵，是相对于局部坐标系的，局部坐标系 Z 轴与重力方向一致，

若智能机磁力计指示方向是北，则北代表局部坐标系 X 轴，西代表局部坐标系 Y 轴；

若智能机磁力计指示方向是东，则东代表局部坐标系 X 轴，北代表局部坐标系 Y 轴；

若智能机磁力计指示方向是南，则南代表局部坐标系 X 轴，东代表局部坐标系 Y 轴；

若智能机磁力计指示方向是西，则西代表局部坐标系 X 轴，南代表局部坐标系 Y 轴；

局部坐标系的原点可以定义为位于地球上任意一点。

[0021] R_g 由下法之一获得：

方法 (1) 采用重力加速度计与磁力计

如果智能机有重力加速度计，且有磁力计或方向传感器，用以下公式 (1) 计算 R_g

$$R_g = R_\theta \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

其中

$$R_\theta = \begin{bmatrix} \frac{s\alpha_z}{L\sqrt{1-(\alpha_y/L)^2}} & -s\alpha_x\alpha_y\sqrt{L^2\sqrt{1-(\alpha_y/L)^2}} & \alpha_x/L \\ 0 & s\sqrt{1-(\alpha_y/L)^2} & \alpha_y/L \\ \frac{-s\alpha_x}{L\sqrt{1-(\alpha_y/L)^2}} & -s\alpha_x\alpha_y\sqrt{L^2\sqrt{1-(\alpha_y/L)^2}} & \alpha_z/L \end{bmatrix}$$

$$\begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix} = R_\alpha^T \begin{bmatrix} v_{ax} \\ v_{ay} \\ v_{az} \end{bmatrix}$$

v_{ax}, v_{ay}, v_{az} 是智能机测出来的智能机重力加速度传感器的 x, y, z 三个方向的值，或这三个检测值经过滤波的结果，如过程中检测到新值 $v_{ax}', v_{ay}', v_{az}'$ 可采用：

$\alpha v_{ax}' + (1-\alpha)v_{ax}$ 替换原来的 v_{ax}

$\alpha v_{ay}' + (1-\alpha)v_{ay}$ 替换原来的 v_{ay}

$\alpha v_{az}' + (1-\alpha)v_{az}$ 替换原来的 v_{az}

α 是介于 $[0, 1]$ 之间的一个值。

[0022] R_a 是重力传感器芯片相对于智能机的姿态转换矩阵。对于大部分智能手机而言，

$$R_a = \begin{bmatrix} -1 & & \\ & -1 & \\ & & -1 \end{bmatrix}$$

$$L = \sqrt{\alpha_x^2 + \alpha_y^2 + \alpha_z^2} \quad \text{当 } \alpha_x < 0 \text{ 时, } s = -1, \quad \alpha_x \geq 0 \text{ 时, } s = 1;$$

若采用方向传感器： $\varphi = \text{value}[0]$ 带入公式 (1) 计算 R_g ；

若采用磁力计传感器： $\{v_{mx}, v_{my}, v_{mz}\}$ 是智能机上磁力计传感器检测到的三个值或这三个值经滤波的结果，如过程中检测到新值 $v_{mx}', v_{my}', v_{mz}'$ 可采用：

$$\alpha_m v_{mx}' + (1 - \alpha_m) v_{mx} \text{ 替换原来的 } v_{mx}$$

$$\alpha_m v_{my}' + (1 - \alpha_m) v_{my} \text{ 替换原来的 } v_{my}$$

$$\alpha_m v_{mz}' + (1 - \alpha_m) v_{mz} \text{ 替换原来的 } v_{mz}$$

α_m 是介于 $[0, 1]$ 之间的一个值。

[0023] 计算

$$\begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = R_m^T \begin{bmatrix} v_{mx} \\ v_{my} \\ v_{mz} \end{bmatrix}$$

其中 R_m 是磁力计传感器芯片相对于智能机的姿态转换矩阵。对于大部分智能手机而言，

$$R_m = \begin{bmatrix} -1 & & \\ & -1 & \\ & & 1 \end{bmatrix}$$

再计算

$$\begin{bmatrix} m_{0x} \\ m_{0y} \\ m_{0z} \end{bmatrix} = R_g^T \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}$$

利用磁力计计算方位角 φ 的三角函数为：

$$\cos \varphi = \frac{m_{0x}}{\sqrt{m_{0x}^2 + m_{0y}^2}}$$

$$\sin \varphi = \frac{m_{0y}}{\sqrt{m_{0x}^2 + m_{0y}^2}}$$

带入公式 (1) 计算 R_g

反复利用上述公式 (1) 计算 R_g 。

若既无磁力计，也无方向传感器，将 φ 角初值定为 0，并当智能机需要模拟逆时针绕重力方向旋转时增加 φ 角，需要模拟顺时针绕重力方向旋转则减小 φ 角。

[0024] 方法 (2) 用旋转矢量传感器

如果智能机配置有旋转矢量传感器

首先采用下述 (a) (b) (c) 中的一种方式得到 R_{g_0} 矩阵

(a) 如果检测到智能机上的旋转矢量传感器只有 3 个数据 values[0], values[1], values[2]

采用以下公式得到

$$q_1 = \text{values}[0], \quad q_2 = \text{values}[1], \quad q_3 = \text{values}[2],$$

$$q_0 = \sqrt{1 - q_1^2 - q_2^2 - q_3^2}$$

则 R_{g_0} 矩阵为 :

$$R_{g_0} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

(b) 如果检测智能机上的旋转矢量传感器有 4 个数据,

$$q_0 = \text{values}[3], \quad q_1 = \text{values}[0], \quad q_2 = \text{values}[1], \quad q_3 = \text{values}[2]$$

则 R_{g_0} 矩阵为 :

$$R_{g_0} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}; \quad \text{有的版本的}$$

android 系统需要先将 {values[3], values[0], values[1], values[2]} 单位化以后才能作为 {q0, q1, q2, q3} 使用。

[0025] (c) 智能机上指定传感器类型为旋转矢量类型, 实时得到旋转矢量传感器的观测值 利用智能机上的旋转矢量转矩阵函数

将传感器的观测值转换成模型转换矩阵 R_{44} , R_{44} 矩阵的左上角 3x3 子阵是 R_{g_0}

然后根据智能机旋转矢量传感器坐标系定义的不同, 采取以下几种情况之一得到 R_g :

若旋转矢量传感器的 X 轴指向东, Y 轴指向北, 则

$$R_g = R_{g_0} \begin{bmatrix} & & \mathbf{1} \\ -\mathbf{1} & & \\ & & \mathbf{1} \end{bmatrix}^T; \quad (2)$$

若旋转矢量传感器的 X 轴指向北, Y 轴指向西, 则

$$R_g = R_{g_0}; \quad (3)$$

若旋转矢量传感器的 X 轴指南, Y 轴指向东, 则

$$R_g = R_{g_0} \begin{bmatrix} -\mathbf{1} & & \\ & -\mathbf{1} & \\ & & \mathbf{1} \end{bmatrix}^T; \quad (4)$$

方法 (3) 采用陀螺仪迭代方法计算智能机姿态矩阵 R_g

若智能机有陀螺仪,可采用以下方法迭代得到 R_g 矩阵:

第 1 步计算 q_0, q_1, q_2, q_3 的初值

选用上述公式 (1) 或 (2) (3) (4) 计算 R_g 的初值,然后利用姿态矩阵转 4 元素成熟技术将 R_g 矩阵转换成 4 元素 q_0, q_1, q_2, q_3

将 4 元数 q_0, q_1, q_2, q_3 作为以下迭代计算的初值

第 2 步设定 $exInt, eyInt, ezInt$ 值为 0

$exInt=0, eyInt=0, ezInt=0$

第 3 步根据接收到的磁力计矢量 $\{m_x, m_y, m_z\}$, 得到正确磁场矢量 $\{w_x, w_y, w_z\}$:

先将矢量 $\{m_x, m_y, m_z\}$ 替换成将其单位化以后的矢量 $\{m_x, m_y, m_z\} / \sqrt{m_x^2 + m_y^2 + m_z^2}$

得到局部坐标系下磁场的正确的磁场方向矢量 $\{b_x, 0, b_z\}$

$h_x = 2 \times m_x \times (0.5 - q_2 \times q_2 - q_3 \times q_3) + 2 \times m_y \times (q_1 \times q_2 - q_0 \times q_3) + 2 \times m_z \times (q_1 \times q_3 + q_0 \times q_2);$

$h_y = 2 \times m_x \times (q_1 \times q_2 + q_0 \times q_3) + 2 \times m_y \times (0.5 - q_1 \times q_1 - q_3 \times q_3) + 2 \times m_z \times (q_2 \times q_3 - q_0 \times q_1);$

$h_z = 2 \times m_x \times (q_1 \times q_3 - q_0 \times q_2) + 2 \times m_y \times (q_2 \times q_3 + q_0 \times q_1) + 2 \times m_z \times (0.5 - q_1 \times q_1 - q_2 \times q_2);$

$b_x = \sqrt{h_x^2 + h_y^2};$

$b_z = h_z$

再转换到正确磁场矢量 $\{w_x, w_y, w_z\}$

$w_x = 2 \times b_x \times (0.5 - q_2 \times q_2 - q_3 \times q_3) + 2 \times b_z \times (q_1 \times q_3 - q_0 \times q_2);$

$w_y = 2 \times b_x \times (q_1 \times q_2 - q_0 \times q_3) + 2 \times b_z \times (q_0 \times q_1 + q_2 \times q_3);$

$w_z = 2 \times b_x \times (q_0 \times q_2 + q_1 \times q_3) + 2 \times b_z \times (0.5 - q_1 \times q_1 - q_2 \times q_2);$

第 4 步根据接收到的重力加速度计数据 a_x, a_y, a_z , 和第 3 步得到的正确磁场矢量 $\{w_x, w_y, w_z\}$ 得到误差矢量 $\{ex, ey, ez\}$ 并修改 $exInt, eyInt, ezInt$ 的值

先将矢量 $\{a_x, a_y, a_z\}$ 替换成将其单位化以后的矢量 $\{a_x, a_y, a_z\} / \sqrt{a_x^2 + a_y^2 + a_z^2}$

$vx = 2 \times (q_1 \times q_3 - q_0 \times q_2); vy = 2 \times (q_0 \times q_1 + q_2 \times q_3); vz = q_0 \times q_0 - q_1 \times q_1 - q_2 \times q_2 + q_3 \times q_3;$

$ex = (ay \times vz - az \times vy) + (my \times wz - mz \times wy);$

$ey = (az \times vx - ax \times vz) + (mz \times wx - mx \times wz);$

$ez = (ax \times vy - ay \times vx) + (mx \times wy - my \times wx);$

计算误差累计值

$exInt$ 替换为 $exInt+ex \times Ki$; $eyInt$ 替换为 $eyInt+ey \times Ki$; $ezInt$ 替换为 $ezInt+ez \times Ki$;

其中 Ki 为一可调节的正系数, Ki 在 0.00001 至 0.5 中任意选取;

第 5 步根据误差矢量 $\{ex, ey, ez\}$ 及其累计值将陀螺仪数据 $\{gx_0, gy_0, gz_0\}$ 纠正成 $\{gx, gy, gz\}$

假设智能机读出当前的一组陀螺仪数据为 $\{gx_0, gy_0, gz_0\}$ $gx = gx_0 + Kp \times ex +$

$exInt; gy = gy0 + Kp \times ey + eyInt; gz = gz0 + Kp \times ez + ezInt;$

其中 Kp 为一可调节的正系数

Kp 在 0.000001 至 20.0 中任意选取;

第 6 步根据陀螺仪数据 gx, gy, gz 修正四元数

随着不断接收到陀螺仪数据 gx, gy, gz , 对 4 元数按如下方式修正, $halfT$ 为修正周期, $halfT = 0.00001 \sim 10.0$,

q_0 替换为 $q_0 + (-q_1 \times gx - q_2 \times gy - q_3 \times gz) \times halfT$; q_1 替换为 $q_1 + (q_0 \times gx - q_3 \times gy + q_2 \times gz) \times halfT$; q_2 替换为 $q_2 + (q_3 \times gx + q_0 \times gy - q_1 \times gz) \times halfT$; q_3 替换为 $q_3 + (-q_2 \times gx + q_1 \times gy + q_0 \times gz) \times halfT$;

第 7 步输出 4 元数和 R_g 矩阵

将四元数 $\{q_0, q_1, q_2, q_3\}$ 单位化成 $\{q_0, q_1, q_2, q_3\} / \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$

4 元数转 R_g 矩阵公式如下

$$R_g = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

第 8 步 回到第 3 步继续接收陀螺仪运动数据更新姿态 4 元数 $q_0 \sim q_3$, 在循环的过程中每次到第 7 步都能输出当前的 R_g 矩阵;

本发明测定智能机上显示屏姿态的方法:

定义智能机上显示屏的姿态矩阵为

$$R = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

r_x 是显示屏水平方向在局部坐标系中的分量;

r_y 是显示屏上下方向在局部坐标系中的分量;

r_z 是垂直于显示屏表面方向向显示屏内部看的视线方向在局部坐标系中的分量。

[0026] 包括以下步骤:

1) 按本发明测定智能机姿态的方法, 得到智能机的姿态矩阵 R_g ;

2) 计算 $R = \Delta R \times R_g$; 得到显示屏的姿态 R 矩阵

ΔR 为智能机显示屏姿态矩阵 R 相对于智能机姿态矩阵 R_g 的一个固定的方向的转换,

当显示屏观察方向和智能机自身 Z 轴方向刚好相反时 $\Delta R = \begin{bmatrix} 1 & & \\ & -1 & \\ & & -1 \end{bmatrix}$, 当显示屏观察方

向和智能机自身 Z 轴方向完全一致时 $\Delta R = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}$, 否则可在智能机上选择一个与显示屏

xyz 三轴姿态一致的摄像头, 采用成熟摄影测量后方交汇法通过摄像头精确标定 ΔR 。方法

如下：

1. 建立控制场坐标系,即原点位于智能机的 GPS 接收点或北斗定位点的局部坐标系；
2. 用智能机摄像头拍摄控制场坐标系中的点集;按权利要求 1 方法测定智能机拍摄点集时的 R_g, R_g 也是相对于控制场坐标系定义的；
3. 根据控制场点集中各已知点转换到控制场坐标系的 3 维坐标与拍摄图像中点集像坐标的对应关系,采用摄影测量后方交会方法测量出智能机摄像头相对于控制场坐标系的姿态矩阵 R_A 和摄像头光心相对于控制场坐标系的坐标 ΔT ；
4. 按下式得到 $\Delta R=R_A R_g^{-1}$ 。

[0027] ΔT 同时也是摄像头光心,近似为显示屏中心,相对于智能机 GPS 或北斗定位点的局部坐标系偏移；

通过测定该摄像头的姿态 R 得到虚拟现实智能机显示屏观察姿态矩阵 R 。

[0028] 本发明还提供了基于智能机姿态的虚拟现实漫游方法：

基于前述方法得到智能机显示屏姿态矩阵 R 矩阵,采用以下步骤：

1) 采用本发明方法得到与智能机显示屏姿态一致的摄像头姿态 R 即为智能机显示屏姿态矩阵 R

若无此摄像头,则当智能机自身姿态 Z 轴方向和智能机显示屏观察姿态方向相反时

$$R = \begin{bmatrix} 1 & & \\ & -1 & \\ & & -1 \end{bmatrix} R_g, \text{ 当智能机自身姿态 } Z \text{ 轴方向和智能机显示屏观察姿态方向一致时 } R=R_g.$$

2) 移动控制漫游观察者在局部坐标系中的位置

定义漫游观察者是在局部坐标系中的一个具备自身位置、姿态、用于观察周边场景的对象。智能机显示屏所渲染的场景就是漫游观察者在局部坐标系中所应看到的场景图像；

定义漫游观察者在局部坐标系中的坐标为 X_d, Y_d, Z_d

用以下三种方法之一控制漫游观察者在虚拟场景中的移动：

(1) 模式 1: 普通 2 维移动

当用户手指按 $\{\Delta X, \Delta Y\}$ 方向滑过智能机显示屏平面的时候,相当于局部坐标系下, X_d 增加 $V_x = \Delta X r_{11} + \Delta Y r_{21}$; Y_d 增加 $V_y = \Delta X r_{12} + \Delta Y r_{22}$; Z_d 不变

此方法主要适合于智能机显示屏接近水平时的情况。

[0029] (2) 模式 2: 智能 2 维移动

当手指按 $\{\Delta X, \Delta Y\}$ 方向滑过智能机显示屏平面的时候,采用如下方法：

首先选择 R 矩阵右侧一列中的 r_{13}, r_{23}, r_{33} 中哪一个分量的绝对值最大,

如果 $|r_{33}|$ 最大, X_d 增加 $V_x = \Delta X r_{11} + \Delta Y r_{21}$ Y_d 增加 $V_y = \Delta X r_{12} + \Delta Y r_{22}$

如果 $|r_{23}|$ 最大, X_d 增加 $V_x = \Delta X r_{11} + \Delta Y r_{31}$ Y_d 增加 $V_y = \Delta X r_{12} + \Delta Y r_{32}$

如果 $|r_{13}|$ 最大, X_d 增加 $V_x = \Delta X r_{31} + \Delta Y r_{21}$ Y_d 增加 $V_y = \Delta X r_{32} + \Delta Y r_{22}$

(3) 模式 3: 3 维漫游移动

因为手机显示屏只有 2 维,单个手指滑动只能进行 2 维操作,单指划过手机显示屏时 $\Delta X, \Delta Y$ 与滑动位移成正比,那么可采用用户双指滑动触碰智能机操作实现 3 维操作,采用双指合拢相当于 $\Delta Z < 0$, 双指拉开相当于 $\Delta Z > 0$ (或者双指合拢相当于 $\Delta Z > 0$, 双指拉开相当于 $\Delta Z < 0$)。当智能机接收到手指控制信息 $\{\Delta X, \Delta Y, \Delta Z\}$ 的时候,

X_d 增加 $V_x = \Delta X_{r_{11}} + \Delta Y_{r_{21}} + \Delta Z_{r_{31}}$ Y_d 增加 $V_y = \Delta X_{r_{12}} + \Delta Y_{r_{22}} + \Delta Z_{r_{32}}$ Z_d 增加 $V_z = \Delta X_{r_{13}} + \Delta Y_{r_{23}} + \Delta Z_{r_{33}}$

3) 设置漫游观察者在局部坐标系里的位置和方向：

根据漫游观察者的局部坐标系坐标 X_d, Y_d, Z_d , 以及步骤 1 测定得到的姿态矩阵 R , 利用 OpenGL, OpenES, OS X OpenGL, Direct3D 或 osg 绘图软件库中用于设定虚拟观察相机位姿的图形库函数, 设置漫游观察者在局部坐标系里的位置和方向：

需要设定以下几个量：

1. 漫游观察者自身观察的垂直视场角 θ , 就是智能机显示屏上边缘中点的视线方向与下边缘中点的视线方向的夹角。通常可选取 $\theta = 45$ 度。

2 漫游观察者所在的位置 X_d, Y_d, Z_d , 由步骤 2 漫游算法修改；

3. 漫游者观察方向 $\{r_{31}, r_{32}, r_{33}\}$ 或者用户注视点坐标 $(X_d + r_{31}, Y_d + r_{32}, Z_d + r_{33})$

4. 漫游者自身向上方向的矢量 $\{-r_{21}, -r_{22}, -r_{23}\}$

其中 r_{ij} 是智能机显示屏姿态 R 矩阵中的第 i 行 j 列

比如在 Android OpenES 中采用

`GLU.gluLookAt(gl, (float)Xd, Yd, Zd,`

`Xd + r31, Yd + r32, Zd + r33,`

`- r21, - r22, - r23);`

其中 r_{ij} 是 R 矩阵第 i 行 j 列的元素值

编程实现设置漫游观察者位置和姿态

4) 在局部坐标系下渲染绘制三维场景

预先在智能机的可视显示屏内设置一个 3 维渲染图层进行三维场景渲染, 可视窗口内预先叠加视频或地图图层；

这里被绘制的三维场景中的点、线、面和物体模型的几何坐标都要转换成局部坐标系下的坐标 $[X_i \ Y_i \ Z_i]^T$, 即与漫游者的坐标 X_d, Y_d, Z_d 在同一个局部坐标系下；用 OpenGL, OpenES, OS X OpenGL, Direct3D 或 osg 绘图软件库实现局部坐标系下的三维绘图。

[0030] 在上述步骤 4) 中, 在局部坐标系中绘制一个标有广告符号、文字的纹理多边形面或模型定义为广告牌。将广告牌的实际地理坐标转换到局部坐标系, 再按如下三种方法渲染绘制都可使得广告牌的正面始终正对着漫游观察者。

[0031] 方法 1

(a) 假设商铺广告牌的局部坐标系坐标在 $X_i \ Y_i \ Z_i$, 按如下方法设置模型绘制转换矩阵：

$$\Delta X_i = X_i - X_d \quad \Delta Y_i = Y_i - Y_d$$

$$m[16] =$$

$$\{ \Delta X_i / L \quad \Delta Y_i / \Delta L \quad 0 \quad 0 \}$$

$$\{ -\Delta Y_i / L \quad \Delta X_i / \Delta L \quad 0 \quad 0 \}$$

$$\{ 0 \quad 0 \quad 1 \quad 0 \}$$

$$\{ X_i \quad Y_i \quad Z_i \quad 1 \}$$

$$\Delta L = \sqrt{\Delta X_i^2 + \Delta Y_i^2}$$

将 m 设定为模型转换矩阵。

[0032] (b) 在 $y-z$ 平面上绘图即可让广告牌永远正对智能机方向
方法 2

(a) 假设商铺广告牌的局部坐标系坐标在 $X_i Y_i Z_i$, 按如下方法设置模型绘制转换矩阵:

$$\Delta X_i = X_i - X_d \quad \Delta Y_i = Y_i - Y_d$$

$$m[16] =$$

$$\{-\Delta Y_i / \Delta L \quad \Delta X_i / \Delta L \quad 0 \quad 0\}$$

$$\{\Delta X_i / \Delta L \quad \Delta Y_i / \Delta L \quad 0 \quad 0\}$$

$$\{0 \quad 0 \quad 1 \quad 0\}$$

$$\{X_i \quad Y_i \quad Z_i \quad 1\}$$

将 m 设定为模型转换矩阵。

[0033] (b) 在 $x-z$ 平面上绘图即可让广告牌永远正对智能机方向方法 (1) (2) 绘制出来的广告牌都具有近大远小的立体效果。

[0034] 方法 3

(a) 假设商铺广告牌的局部坐标系坐标在 $X_i Y_i Z_i$, 按如下方法设置模型绘制转换矩阵:

$$\Delta X_i = X_i - X_d \quad \Delta Y_i = Y_i - Y_d$$

$$m[16] =$$

$$\{-\Delta Y_i \quad \Delta X_i \quad 0 \quad 0\}$$

$$\{\Delta X_i \quad \Delta Y_i \quad 0 \quad 0\}$$

$$\{0 \quad 0 \quad \Delta L \quad 0\}$$

$$\{X_i \quad Y_i \quad Z_i \quad 1\}$$

将 m 设定为模型转换矩阵。

[0035] (b) 在 $x-z$ 平面上绘图即可让广告牌永远正对智能机方向

这样绘制出来的广告牌将不具有近大远小的效果, 远近一样大, 方便用户观察。

[0036] 上述渲染的商铺信息内容包括名称、符号属性、类别、与智能机的距离、角度和 / 或高度。

[0037] 根据智能机的真实地理坐标进行虚拟现实漫游:

本发明设定局部坐标系原点是智能机显示屏中心, 即 X_d, Y_d, Z_d 是相对于智能机显示屏中心的局部偏移坐标, 步骤 4) 之前增加如下操作:

(a) 获取智能机的高度 h_s , 纬度 θ_s 和经度 ϕ_s

θ_s 是智能机的纬度, ϕ_s 是智能机的经度, 由智能机上自带的 GPS 接受模块获取;

(b) 计算智能机所在经纬度的 R_v

$$R_v = \begin{bmatrix} -\sin \theta_s \cos \phi_s & -\sin \theta_s \sin \phi_s & \cos \theta_s \\ \sin \phi_s & -\cos \phi_s & 0 \\ \cos \theta_s \cos \phi_s & \cos \theta_s \sin \phi_s & \sin \theta_s \end{bmatrix}$$

(c) 计算智能机所在纬度 θ_s 经度 ϕ_s 高度 h_s 的地心坐标系坐标 T_s

利用大地测量学中经典的转换公式。

$$\begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} = T_s = \begin{bmatrix} F_x(h_s, \theta_s, \phi_s) \\ F_y(h_s, \theta_s, \phi_s) \\ F_z(h_s, \theta_s, \phi_s) \end{bmatrix} = \begin{bmatrix} (N_s + h_s) \cos \theta_s \cos \phi_s \\ (N_s + h_s) \cos \theta_s \sin \phi_s \\ (N_s(1 - e^2) + h_s) \sin \theta_s \end{bmatrix}$$

[0038] 其中

h_s 是智能机离地面高度

$$N_s = a / \sqrt{1 - e^2 \sin^2 \theta_s} \quad N_s \text{ 是随纬度变化的卯西圈曲率半径。}$$

$$e = \sqrt{a^2 - b^2} / a$$

[0039] a, b 是地球椭球的长短轴, 是大地测量学中的已知常量;

e 是偏心率, 是定值。

[0040] (d) 根据地球上任何一个点的经度 ϕ_i 、纬度 θ_i 和高度 h_i 得到该点地心坐标系坐标 $T_{0i} = [X_{0i} \ Y_{0i} \ Z_{0i}]^T$

利用大地测量学中经典的转换公式

$$\begin{bmatrix} X_{0i} \\ Y_{0i} \\ Z_{0i} \end{bmatrix} = T_{0i} = \begin{bmatrix} F_x(h_i, \theta_i, \phi_i) \\ F_y(h_i, \theta_i, \phi_i) \\ F_z(h_i, \theta_i, \phi_i) \end{bmatrix} = \begin{bmatrix} (N_i + h_i) \cos \theta_i \cos \phi_i \\ (N_i + h_i) \cos \theta_i \sin \phi_i \\ (N_i(1 - e^2) + h_i) \sin \theta_i \end{bmatrix}$$

其中

h_i 是点离地面高度

$$N_i = a / \sqrt{1 - e^2 \sin^2 \theta_i} \quad N \text{ 是随纬度变化的卯西圈曲率半径。}$$

$$e = \sqrt{a^2 - b^2} / a$$

[0041] a, b 是地球椭球的长短轴, 是大地测量学中的已知常量;

e 是偏心率, 是定值。

[0042] 如果已知一个物体所在经纬度 ϕ_i 和 θ_i , 通过本步骤计算出它的地心坐标系坐标 T_{0i}

(e) 地心坐标系坐标 T_{0i} 转换为局部坐标系中的坐标 $(X_i \ Y_i \ Z_i)^T$

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = R_i(T_{0i} - T_i) - \Delta T$$

ΔT 是智能机显示屏中心相对于智能机 GPS 接收天线的局部坐标系偏移。这样保证了权利要求 3 中的 $(X_i \ Y_i \ Z_i)^T$ 和 $(X_d \ Y_d \ Z_d)^T$ 都是相对于智能机显示屏中心的偏移坐标。可近似取 $\Delta T = [0 \ 0 \ 0]^T$

权利要求 3 中步骤 4) 中绘制三维场景须用这种转换为局部坐标系中的坐标 $(X_i \ Y_i \ Z_i)^T$ 。

[0043] 在基于智能机姿态的虚拟现实漫游方法步骤 2) 中设置加速度模拟漫游效果:

步骤 2) 中如果需要模拟加速度漫游移动特效, 当手指滑动的时候, 根据手指滑动速度大小给予一个初速度 $\{V_x, V_y, V_z\}$, X_d, Y_d, Z_d 将不停地增加速度分量 V_x, V_y, V_z , 然后让速度分量 V_x, V_y, V_z 随时间推移不断衰减, 其中 V_z 在权利要求 3 步骤 2) 模式 1, 2 情况下恒等于

零。

[0044] 让速度体现到局部坐标系下的漫游位移

X_d 增加 V_x

Y_d 增加 V_y

Z_d 增加 V_z 仅用于权利要求 3 步骤 2) 模式 3

最后要在不停的移动过程中速度大小不断衰减,

V_x, V_y, V_z 不停地衰减;可采用以下两种方法:

1) 运动过程中 V_x, V_y, V_z 自身分别不断地乘以 $k=0.1 \sim 0.9999$ 进行衰减,

运动过程中 V_x 自身不断地乘以 k , 直到 V_x 减小为 0 为止;

运动过程中 V_y 自身不断地乘以 k , 直到 V_y 减小为 0 为止;

运动过程中 V_z 自身不断地乘以 k , 直到 V_z 减小为 0 为止;

或者

2) 运动过程中 V_x, V_y, V_z 自身分别不断地进行以下增量操作

V_x 自身不断地增加 $-V_x * k$, 直到 V_x 减小为 0 为止;

V_y 自身不断地增加 $-V_y * k$, 直到 V_y 减小为 0 为止;

V_z 自身不断地增加 $-V_z * k$, 直到 V_z 减小为 0 为止;

其中 $k=0.0001 \sim 0.5$

以上过程不断循环即可实现局部加速度漫游效果。

[0045] 本发明还涉及虚拟现实场景中的 2 维标签的恒水平于地面的显示方法:

2 维标签是指 2 维文字、图符、图片、标记、编码、对话框、编辑框、列表框等一切在智能机显示屏中以 2 维形式显示的对象。

[0046] 因用户双眼是水平于地面的, 前述基于智能机姿态的虚拟现实漫游方法只能保证 3 维物体能正确显示, 2 维标签的显示姿态会随智能机旋转而旋转, 用户看到的标签就不是正放的。故设计本方法使得标签始终正放显示。方法不局限于矩形显示屏, 对常用矩形显示屏而言, 采用以下方法。

[0047] 在前述基于智能机姿态的虚拟现实漫游方法步骤 3 和步骤 4) 之间或步骤 4) 结束之后, 将显示屏设定为 2 维绘图模式并设置若干个 2 维标签, 针对每一个 2 维标签, 按如下步骤显示:

步骤 1) 首先确定虚拟现实场景中 2 维标签的中心位置 u, v ;

如果标签用于标注三维场景中局部坐标系坐标为 (X, Y, Z) 的目标, 则根据以下公式确定目标显示的中心位置 u, v

$$Z_0 = r_{11}(X - X_d) + r_{12}(Y - Y_d) + r_{13}(Z - Z_d)$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{w}{2} + f \frac{r_{11}(X - X_d) + r_{12}(Y - Y_d) + r_{13}(Z - Z_d)}{Z_0} \\ \frac{h}{2} + f \frac{r_{21}(X - X_d) + r_{22}(Y - Y_d) + r_{23}(Z - Z_d)}{Z_0} \end{bmatrix} \quad \text{式中: } w \text{ 是智能机显示屏宽度, } h \text{ 是智能机显示屏高度。}$$

能机显示屏高度。

[0048] $r_{11} \sim r_{33}$ 是权利要求 2 所述的方法中的 R 矩阵对应行列的元素。

$$[0049] \quad f = \frac{h/2}{\tan(\theta/2)} \quad \theta \text{ 是垂直视场角}$$

X_d, Y_d, Z_d 是漫游观察者在局部坐标系中的坐标。

[0050] 如果满足 $Z_0 > 0$ 且 $0 < u < w$ 且 $0 < v < h$ 则 u, v 保持不变, 否则需要按下法修正 u, v

若 $Z_0 = 0$ 则将 $[u \ v]$ 替换为从点 $(w/2, h/2)$ 出发

按矢量 $\{r_{11}(X-X_d)+r_{12}(Y-Y_d)+r_{13}(Z-Z_d), r_{21}(X-X_d)+r_{22}(Y-Y_d)+r_{23}(Z-Z_d)\}$

方向的射线与显示屏边缘相交的 2 维点。若 $Z_0 > 0$ 则将 $[u \ v]$ 替换为点 $(w/2, h/2)$ 到 (u, v) 的连线与显示屏边缘相交的 2 维点。

[0051] 若 $Z_0 < 0$ 则将 $[u \ v]$ 替换为点 (u, v) 到 $(w/2, h/2)$ 的延长线与显示屏边缘相交的 2 维点。

[0052] 如果标签不是用于标注目标, u, v 为界面设计者设定的显示屏坐标, 或是用户手指或鼠标点击的显示屏坐标。

[0053] 步骤 2) 计算矢量 $\{Q_x, Q_y\}$

方法 (1) $Q_x = -r_{23}/L_a \quad Q_y = r_{13}/L_a$

其中 r_{23}, r_{13} 是权利要求 2 所述的方法中的 R 矩阵对应行列的元素,

$$L_a = \sqrt{r_{13}^2 + r_{23}^2}$$

方法 (2) 利用权利要求 1 方法 (1) 所述加速度计的前两个输出值 a_x, a_y 得到

$$Q_x = -a_y/L_a \quad Q_y = a_x/L_a, \quad L_a = \sqrt{a_x^2 + a_y^2}$$

步骤 3) 在 2 维平面上经过以下矩阵旋转变换以后再绘制 2 维标签:

$$X' = \begin{bmatrix} u \\ v \end{bmatrix} \pm \begin{bmatrix} Q_x & Q_y \\ -Q_y & Q_x \end{bmatrix} \left(X - \begin{bmatrix} u \\ v \end{bmatrix} \right)$$

2 维标签上每一点 X , 经过上式变换到坐标 X' 显示出来。

[0054] \pm 取 + 是正放显示, 取 - 是倒放显示, 标签的水平轴始终平行于地面。

[0055] 就会看到显示屏上该显示对象自身的水平轴始终平行于地平面, 标签始终处于正放状态。

[0056] 在基于智能机姿态的虚拟现实漫游方法的步骤 4) 中所述三维场景为一个空心的三维形体, 三维形体可以是球, 立方体、柱体、正多面体、锥体, 三维形体的内壁以全景图作为纹理进行渲染。所述漫游观察者在虚拟场景中位于三维形体内部的某一固定点 $[X_d, Y_d, Z_d] = [0, 0, 0]$, 一般位于三维形体中心, 从三维形体内部观察三维形体的内壁纹理, 实现利用智能机显示屏姿态 R 矩阵控制全景浏览。

[0057] 本发明进一步提供了控制局部坐标系中三维物体姿态的方法:

将 R_g 矩阵设定为仿真软件系统中的模型转换矩阵 M 的左上角 3x3 矩阵 $M_{44} = \begin{bmatrix} R_g & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$

这实现了利用智能机姿态控制三维场景中某一物体的姿态。

附图说明

[0058] 图 1 为智能机姿态矩阵 R 的三个行矢量 r_{gx}, r_{gy}, r_{gz} 在局部坐标系中的所指的矢量方向。

具体实施方式

[0059] 本发明针对 3 种方法需要智能机的 3 种配置类型的传感器输入

方法 (1) 用以下公式 $R_g = R_g \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$ 计算 R_g

其中

$$R_g = \begin{bmatrix} \frac{s a_x}{L \sqrt{1-(a_y/L)^2}} & -s a_x a_y \sqrt{L^2 \sqrt{1-(a_y/L)^2}} & a_x/L \\ 0 & s \sqrt{1-(a_y/L)^2} & a_y/L \\ \frac{-s a_x}{L \sqrt{1-(a_y/L)^2}} & -s a_x a_y \sqrt{L^2 \sqrt{1-(a_y/L)^2}} & a_x/L \end{bmatrix}$$

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = R_g^T \begin{bmatrix} v_{ax} \\ v_{ay} \\ v_{az} \end{bmatrix}$$

$$\begin{bmatrix} m_{0x} \\ m_{0y} \\ m_{0z} \end{bmatrix} = R_g^T \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = R_g^T \begin{bmatrix} v_{mx} \\ v_{my} \\ v_{mz} \end{bmatrix}$$

利用磁力计计算方位角 φ 的三角函数为：

$$\cos \varphi = \frac{m_{0x}}{\sqrt{m_{0x}^2 + m_{0y}^2}}$$

$$\sin \varphi = \frac{m_{0y}}{\sqrt{m_{0x}^2 + m_{0y}^2}}$$

在实际操作中,若出现地磁偏角 $\Delta \varphi$ 导致磁力方向不是指着正北,可通过下式计算 R_g

$$\cos \varphi' = \cos \varphi \cos \Delta \varphi - \sin \varphi \sin \Delta \varphi$$

$$\sin \varphi' = \sin \varphi \cos \Delta \varphi + \cos \varphi \sin \Delta \varphi$$

$$R_g = R_g \begin{bmatrix} \cos \varphi' & -\sin \varphi' & 0 \\ \sin \varphi' & \cos \varphi' & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

重力加速度计传感器的三个值 $\{v_{ax}, v_{ay}, v_{az}\}$ 要求配置最低,适用性最广；

磁力计传感器的三个值 $\{v_{mx}, v_{my}, v_{mz}\}$ 或 方向传感器的航向角 value[0]

若既无磁力计,也无方向传感器, φ 角可通过用户界面滑块控制增减 φ 角。往地心看

绕重力方向,模拟逆时针旋转时 φ 角增加,模拟顺时针旋转时 φ 角减小。

[0060] 方法 (2) 旋转矢量传感器的三个或四个输出,需旋转矢量传感器。

[0061] 方法 (3) 重力加速度计,磁力计 和 陀螺仪,另需陀螺仪。

[0062] 计算过程中,很多变量如 L 和 $\sqrt{1-(a_y/L)^2}$ 可以预存下来,避免重复计算,以提高速度。

[0063] 本发明输出矩阵 R_g 的几何意义如下

$$\text{完整描述智能机局部坐标系姿态的矩阵 } R_g = \begin{bmatrix} r_{g11} & r_{g12} & r_{g13} \\ r_{g21} & r_{g22} & r_{g23} \\ r_{g31} & r_{g32} & r_{g33} \end{bmatrix} = \begin{bmatrix} r_{gx} \\ r_{gy} \\ r_{gz} \end{bmatrix}$$

R_g 矩阵是一个 3×3 单位正交矩阵,由三行组成,每行一个三维矢量,三个矢量分别为 r_{gx}, r_{gy}, r_{gz} 。第一行 r_{gx} 是智能机自身 x 轴在局部坐标系下的单位矢量;

第二行 r_{gy} 是智能机自身 y 轴在局部坐标系下的单位矢量;

第三行 r_{gz} 是智能机自身 z 轴在局部坐标系下的单位矢量。

[0064] 如图 1 所示,智能机姿态矩阵 R_g 的三个行矢量 r_{gx}, r_{gy}, r_{gz} 是其自身三轴在局部坐标系下的三个单位矢量。

[0065] (1) 智能机显示屏的姿态测定方法中,

显示屏观察方向和智能机自身 Z 轴方向刚好相反时 $\Delta R = \begin{bmatrix} 1 & & \\ & -1 & \\ & & -1 \end{bmatrix}$, 显示屏观察方

向和智能机自身 Z 轴方向完全一致时 $\Delta R = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}$, 否则可在智能机上选择一个与显示屏

观察方向一致的摄像头,该摄像头的拍摄方向一定与观察显示屏的方向一致,通过摄影测量后方交汇法测定该摄像头的局部坐标系姿态 R_A 以及对应的智能机姿态矩阵 R_g , 采用 $\Delta R = R_A R_g^{-1}$ 精确标定摄像头。

[0066] 因 ΔR 描述了显示屏观察姿态和智能机自身姿态之间的固定转换关系,一旦精确测定以后,就是一个恒定值。任意给定 R_g 就可得到对应的显示屏姿态 $R = \Delta R R_g$ 。

[0067] 基于智能机姿态的虚拟现实漫游描述的方法可用于与地理位置无关的虚拟现实漫游领域,如非实景地形游戏,体感游戏、三维虚拟数据仿真,虚拟全景等领域;

用户可采用实际旋转智能机的方式操控智能机姿态,所对准的方向和姿态在虚拟现实场景中观察的内容与其显示屏上显示的是对应的,采用手指滑动方式操纵控制在虚拟现实场景中平移。

[0068] 基于智能机姿态的虚拟现实漫游步骤 3 的具体实施方式如下,

将显示屏设定为 3 维绘图模式可采用 android 系统中以下 OpenES 语句编程实现:

```
gl.glMatrixMode(GL10.GL_PROJECTION); // 开始设置投影矩阵
gl.glLoadIdentity(); // 下面转换成二维平面信息绘制模式
```

GLU.gluPerspective(gl, θ , 屏宽 / 屏高, 0.01, (float) 1000000); 可选取 θ =45 度

在 Android OpenES 中采用

```
GLU.gluLookAt(gl, Xd, Yd, Zd,
Xd + r31, Yd + r32, Zd+ r33,
- r21, - r22, - r23);
```

在 OpenGL 里可采用以下语句实现

```
gluLookAt(Xd, Yd, Zd,
Xd +r31, Yd + r32, Zd + r33,
- r21, -r22, -r23)
```

其中 r_{ij} 是 R 矩阵第 i 行 j 列的元素值

编程实现设置漫游观察者位置和姿态

广告牌渲染效果通过如下方法实现：

$$\Delta X_i = X_i - X_d \quad \Delta Y_i = Y_i - Y_d$$

以上方法可采用 OpenGL 中的

```
m[16] = {-ΔYi, ΔXi, 0, 0, ΔXi, ΔYi, 0, 0, 0, 0, L, 0, Xi, Yi, Zi, 1}
glMultMatrixf(m, 0); 语句编程实现
```

或

OpenES 中的

```
m[16] = {-ΔYi, ΔXi, 0, 0, ΔXi, ΔYi, 0, 0, 0, 0, L, 0, Xi, Yi, Zi, 1}
gl.glMultMatrixf(m, 0); 语句编程实现
```

根据智能机的真实地理坐标进行虚拟现实漫游中描述的方法可用于与地理位置有关的应用,如地图漫游,数字城市、飞行模拟三维漫游。在没有高程数据的情况下可以直接将图片绑定成纹理绘制在 X, Y 平面上,有高程信息条件下,可结合高程信息在虚拟现实 XYZ 空间中绘制出三维网格地形,实现智能机姿态结合手指滑动控制地图漫游。

[0069] 漫游加速度模拟中可在软件中将 V_x, V_y, V_z 的衰减计算过程放在渲染线程中计算,也可以单独用一个线程处理。

[0070] 智能机显示屏恒水平显示是指不论怎样绕 X, Y, Z 三轴任意转动智能机,智能机显示屏上 2 维标签自身的水平轴始终平行于地平面。

[0071] 可将要显示的 2 维标签中的文字、标签,对话框等作为纹理绑定到内存。渲染的时候将此纹理渲染在经过

$$m = \begin{bmatrix} Q_x & Q_y & & \\ -Q_y & Q_x & & \\ & & 1 & \\ \mathbf{x} & \mathbf{v} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

变换的多边形上。

[0072] 因用户的双眼是平行于地平面的,所以不论智能显示器怎么旋转,其上显示的 2 维标签总能给观众一个正确的姿态显示效果。

[0073] 可通过 $glMultMatrix(m)$; 实现

$$\text{其中 } m = \begin{bmatrix} q_x & q_y & & \\ -q_y & q_x & & \\ & & 1 & \\ u & v & 0 & 1 \end{bmatrix}$$

实现；

也可先 `Translatef(u, v, 0)`，然后 `glMultMatrix(m)`；实现

$$\text{其中 } m = \begin{bmatrix} q_x & q_y & & \\ -q_y & q_x & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

接着智能机姿态的虚拟现实漫游方法步骤 4) 之后，

将显示屏设定为 2 维绘图模式可采用 android 系统中的以下 OpenES 语句编程实现：

```
gl.glLoadIdentity(); // 下面转换成二维平面信息绘制模式
gl.glViewport(0, 0, A_Main.screenWidth, A_Main.screenHeight);
gl.glMatrixMode(GL10.GL_PROJECTION); // 开始设置投影矩阵
gl.glLoadIdentity(); // 下面转换成二维平面信息绘制模式
gl.glOrthof(0, A_Main.screenWidth, A_Main.screenHeight, 0, -100, 100);
gl.glMatrixMode(GL10.GL_MODELVIEW); // 开始设置模型绘制矩阵
```

全景浏览方法中的三维形体一般采用球，立方体，正多面体这样的对称形状，漫游观察者一般位于对称体内部中心点。可采用 ALPHA 检测技术对球面纹理进行渲染。

[0074] 控制局部坐标系中三维物体姿态的方法中可以用智能机姿态或者智能机显示屏姿态操纵物体，如果是智能机姿态操纵，采用矩阵 R_g 计算；如果是智能机显示屏姿态操纵，采用矩阵 R 计算。

[0075] 应当理解的是，对本领域普通技术人员来说，可以根据上述说明加以改进或变换，而所有这些改进和变换都应属于本发明所附权利要求的保护范围。

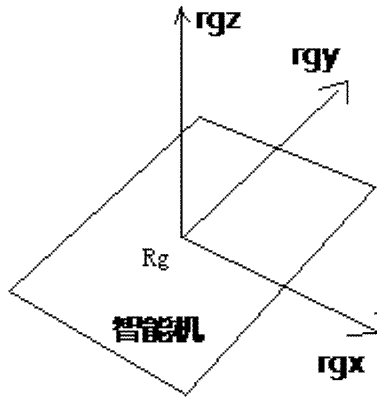


图 1