

基于表面间距离度量的多视点距离 图像的对准算法*

张鸿宾** 谢 丰

(北京工业大学计算机学院, 北京 100022)

摘要 现有的距离图像对准算法, 一般是采用最近点之间的距离和作为评价函数来估计运动参数. 这种基于点对间距离的评价函数的缺点是, 不同距离图像之间的最近点不一定是物体表面同一点的采样, 而且不同采样点所代表的物体表面面积也不同. 这些问题使得基于点对间距离度量的对准算法存在不精确对应的问题, 因而对准精度不高, 对初始运动参数敏感. 提出一种基于表面间距离度量的对准算法. 通过构造三角网格来近似表示物体的表面, 采用三角网格间最近距离的均值作为评价函数来估计运动参数. 推导并简化了表面间距离计算的积分公式. 实验结果表明, 该算法有较高的对准精度, 而且收敛速度较快, 抗噪声能力较强.

关键词 距离图像 多视点 运动估计 对准

1 引言

随着距离传感器技术的发展, 从多视点的距离图像构造物体完整的三维几何模型在计算机视觉、计算机图形学、计算机辅助设计、虚拟现实和非接触测量等领域获得了越来越广泛的应用. 由于传感器只有有限的视野以及物体表面的自遮挡等原因, 单一传感器所获得的距离图像只是物体部分表面的信息, 将不同视点的距离图像进行对准和集成就成为构造三维物体完整几何模型的关键步骤之一^[1]. 所谓多视点距离图像的对准就是要求出不同视点距离数据间的位置变换即运动参数, 把多视点下的距离图像统一到一个笛卡尔坐标系下.

运动参数的估计主要是利用距离图像间重叠区域的数据来进行的. 早期的对准算法是先在不同的距离图像上找对应的特征量, 然后采用奇异值分解和单

2003-01-27 收稿, 2003-07-22 收修改稿

* 国家自然科学基金资助课题(批准号: 60275001)

** E-mail: zhb@public.bta.net.cn

SCIENCE IN CHINA Ser. E Information Science

位四元组等方法来计算运动参数^[2,3]。由于物体表面的特性、噪声以及物体的自遮挡等原因,寻找特征量以及它们之间的匹配是一件困难的工作,因此这类方法很难得到精确的对准。由于这类方法不需要初始运动参数估计,所以目前一般是利用它们的对准结果作为精对准时的初始运动参数。精对准的方法一般不需要提取特征,通过定义一个评价函数来反映重叠区域的吻合程度,对准实际上就是估计一组运动参数使得评价函数取得最小值的一个优化过程^[4]。这个优化问题可以非迭代^[5]或迭代地求解^[4,6-8]。目前应用最广泛的求解方法是由 Besl 和 Mckay^[7]以及 Chen 和 Medioni^[8]提出的迭代最近点算法 ICP(Iterative Closest Point)及其各种变形。假定 P 是数据点集,要求一个旋转矩阵 R 和平移矢量 t ,使数据点集 P 和目标点集 X 进行对准。ICP 一类的对准算法是要使下面的目标函数达到最小:

$$f_n(R_n; t_n) = \sum_{i_n} W_{in} \|x_{in} - (R_n p_{in} + t_n)\|^2, \quad (1)$$

式中 $x_{in} \in X$ 是 $p_{in} \in P$ 变换后的最近点, n 是迭代的次数, W_{in} 是赋予第 i 个最近点对的一个权值。当它的值取 0, 1 时,可以用它来反映该点对是否是偏离点(outliers)。Besl 等人用迭代的方法求解使(1)式最小的 $(R; t)$ 。当 P 是 X 的子集时,ICP 算法可以收敛到目标函数的一个局部极小点。

从(1)式中可以看出,ICP 及其变型算法的评价函数是数据点集各点到模型点集的最近距离的和。然而,如果仅仅把距离图像看成点集,就会丢失距离图像的一些信息。在一般的距离传感器所获得的距离图像中,它的像素点之间都存在邻接关系,正是这些有邻接关系的数据点才构成了物体的表面。而把距离图像看成为点集时,距离图像实际上就变成了一堆毫无关系的散乱点。另外,传统的对准算法在找对应的点对和特征时存在不精确对应(Inexact Correspondences)的问题^[1]。这是因为距离图像的像素点是对 3D 物体表面的采样,由于传感器的位置、方向和物体表面的斜率等原因,即使是重叠区域的最近点对也有可能不是对被测物体表面同一点的采样。针对这些问题,本文提出一种基于物体表面间距离度量的对准算法。该算法利用三角网格近似表示物体的表面,这样距离图像的对准问题就变成了三角网格间的对准问题,它比点集间的对准更好地反映了物体表面的对准情况。我们定义三角网格间最近距离的均值作为对准的评价函数,它比用点对间的距离更准确地反映了距离图像的对准程度。通过对三角面块的采样和点到面距离的计算,我们推导并简化了表面间平均距离计算的积分公式。在模拟和真实数据上的实验结果表明,本文的算法有较高的对准精度、较快的收敛速度和对初始参数不敏感等优点。

下面第二节描述本文基于表面间距离度量的对准算法。第三节是部分实验结果。第四节是对算法的一些分析和结论。

2 基于物体表面间距离度量的距离图像对准算法

2.1 构造三角网格

如上所述, 和传统的对准方法不同, 本文利用三角网格来近似表示物体的表面, 利用三角网格间的距离度量作为目标函数来估计运动参数. 三角网格的构造方法采用了 Turk 在文献[9]中提出的方法. 它通过在距离图像的相邻两行和两列之间的四个像素点上构造零个、一个或两个边长小于一定阈值的三角形来建立三角网格.

2.2 运动参数和点到三角形网格的最短距离

下面用 M 表示三角网格, tr_i 表示一个三角面块, 即 $M = \{tr_i, i = 1, 2, \dots, n\}$, n 为面块的个数. $T = \{R, t\}$ 表示物体的运动参数, 其中 R 为 3×3 的旋转矩阵, t 为 3D 位移矢量. 用单位四元组 q 表示旋转矩阵 R : $R = R(q)^{[5]}$. 一个 3D 点 x 经过 T 变换后的点为 $T(x) = Rx + t$, 一个三角网格 M 经过 T 变换后表示为 $RM + t$. 定义 3D 点 x 到三角面块 tr_i 的距离为 x 到 tr_i 各点欧氏距离的最小值, 记为 $dis(x, tr_i)$. 而 3D 点 x 到三角网格 M 的最短距离为

$$dis(x, M) = \min_{tr_i \in M} dis(x, tr_i).$$

2.3 网格间距离的计算及化简

假设要对准的距离图像为 R_1 和 R_2 , 其三角网格表示分别为 M_1 和 M_2 . 假设它们重叠的网格部分为 M'_1 和 M'_2 . 定义对准的评价函数为三角网格间重叠部分最短距离的均值:

$$E = \frac{1}{S'_{M1}} \iint_{M'_1} dis(x, RM'_2 + t) ds = \frac{1}{S'_{M1}} \sum_{tr_i \in M'_1} \iint_{x \in tr_i} dis(x, RM'_2 + t) ds. \quad (2)$$

式中 x 为 tr_i 上的任意点, S'_{M1} 为 M'_1 的面积, ds 为面积积分元. 以下是要求出一个 $T = \{R, t\}$ 使得 E 取得最小值. 为了计算(2)式的对准误差, 可以对三角网格 M'_1 进行稠密而且按面积均匀分布的采样. 即要求这种采样满足以下要求: 假设 M_1 上任意区域 S 的面积为 S_s , 曲面 S 上的采样数为 N_s , 三角网格 M'_1 的面积为 S'_{M1} , M'_1 上的采样数为 N'_{M1} , 则

$$\frac{S_s}{S'_{M1}} = \frac{N_s}{N'_{M1}}. \quad (3)$$

记 M'_1 上的采样点集为 P'_{M1} , 其中的采样点记为 $\{p_i\}$. 此时(2)式可化为

$$E = \frac{1}{N'_{m1}} \sum_{p_i \in P'_{m1}} dis(p_i, RM'_2 + t). \quad (4)$$

通过这样的采样, (2)式表面间距离计算的积分可以近似为(4)式的求和. 对 M'_1 上

第 i 块三角形 tr_i , 记 P_{tri}^{m1} 为 tr_i 上的采样点集, N_{tri}^{m1} 为 tr_i 上的采样点数, S_{tri}^{m1} 为 tr_i 的面积. 根据(3), (4)式可以化为

$$E = \frac{1}{S'_{M1}} \sum_{tr_i \in M'_1} S_{tri}^{m1} \left(\frac{1}{N_{tri}^{m1}} \sum_{p_i \in P_{tri}^{m1}} dis(p_i, RM'_2 + t) \right). \quad (5)$$

下面我们以 tr_i 为例, 说明它的采样点集 P_{tri}^{m1} 和三角网格 $(RM'_2 + t)$ 的距离是如何计算的. 假设三角形 tr_i 的三个顶点为 p_1, p_2 和 p_3 , 它们到三角网格 $(RM'_2 + t)$ 上的最近点分别为 p'_1, p'_2 和 p'_3 . 首先以向量 $p_1 p'_1, p_2 p'_2$ 和 $p_3 p'_3$ 的均值 pp' 作为三角形 tr_i 向三角网格 $(RM'_2 + t)$ 的投影方向, 然后沿着方向 pp' 将 tr_i 上各采样点投影到 M'_2 上, 求出它们的对应点. 由于 pp' 是三个顶点到它们最近点的方向的均值, 所以可以近似地认为这个投影方向是所有投影方向中, 使三角形 tr_i 上各采样点到它们对应点的平均距离最小的方向. 这种求对应点的方法与 ICP 等算法所用的求最近点的方法相比, 其优点可以用图 1 来说明. 图 1(a)和(b)说明了当距离图像中存在较大偏差的像素点时, 两种求对应点的方法所得到的对应点对情况. 图 1(c)和(d)说明了在物体表面变化比较激烈的部分, 两种求对应点方法所得到的对应点对情况. 可以明显看出, 本文求对应点的方法比传统的最近点方法有较少的虚假匹配.

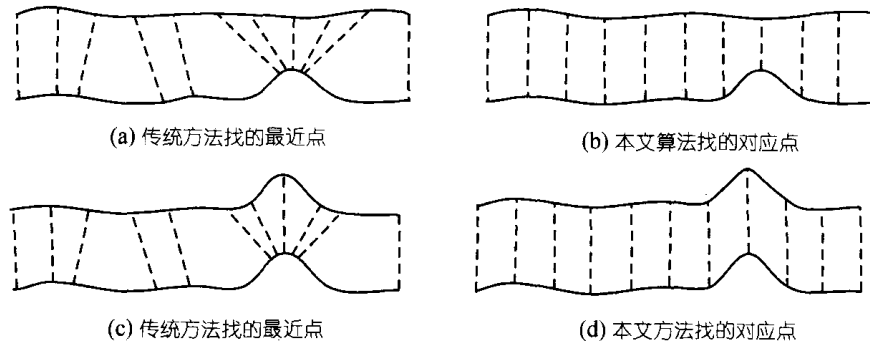


图 1 本文求对应点方法和传统最近点方法的比较

求出 tr_i 在网格 $(RM'_2 + t)$ 上的对应点后, 根据 tr_i 上各采样点的对应点在 $RM'_2 + t$ 的不同三角形上, 可以把三角形 tr_i 划分为一些多边形, 然后把这些多边形进一步划分为一些三角形的集合. 我们把三角形 tr_i 根据上面方法划分成的三角形集合记为 M_{tri} , 则(5)式可化为

$$E = \frac{1}{S'_{M1}} \sum_{tr_i \in M'_1} \sum_{tr_j \in M_{tri}} S_{tr_j}^{tri} \left(\frac{1}{N_{tr_j}^{tri}} \sum_{p_i \in P_{tr_j}^{tri}} dis(p_i, RM'_2 + t) \right), \quad (6)$$

其中 tr_j 为 M_{tri} 上的第 j 块三角形, $S_{tr_j}^{tri}$ 为 tr_j 的面积, $N_{tr_j}^{tri}$ 为 tr_j 上的采样点数, $P_{tr_j}^{tri}$ 为 tr_j 上的采样点集. 如果我们用对应点代替最近点, 并且利用三角面块的平面性

质, 则(6)式中最右面一项的平均点-面距离可以近似为

$$\frac{1}{N_{trj}^{tri}} \sum_{p_i \in P_{trj}^{tri}} dis(p_i, RM'_2 + t) \approx \frac{\|p_{trj1} p'_{trj1}\| + \|p_{trj2} p'_{trj2}\| + \|p_{trj3} p'_{trj3}\|}{3}, \quad (7)$$

式中 p'_{trj1} , p'_{trj2} 和 p'_{trj3} 分别为 p_{trj1} , p_{trj2} 和 p_{trj3} 在 $RM'_2 + t$ 上的对应点. 根据(7)式, (6)式可以化为

$$E \approx \frac{1}{S'_{M1}} \sum_{tri \in M'_1} \sum_{trj \in M_{tri}} S_{trj}^{tri} \left(\frac{\|p_{trj1} p'_{trj1}\| + \|p_{trj2} p'_{trj2}\| + \|p_{trj3} p'_{trj3}\|}{3} \right). \quad (8)$$

下面将直接使用(8)式作为对准的评价函数. (8)式的求解可以利用传统 ICP 算法的迭代框架.

在上面的推导中, 通过对三角面块的采样, 将表面间距离计算的积分(2)式化为求和计算的(4)式, 进一步又利用三角面块的平面性质, 将对所有采样点进行的点-面距离计算的(6)式简化为计算三角形的三个顶点和它们对应点间距离的均值与三角形面积的积((8)式). 这极大地简化了(2)式目标函数的计算量. (8)式实际上反映了两个表面间间隙的近似体积.

ICP 算法的框架中需要计算两个点集的互协方差矩阵 Σ (cross-covariance matrix). 和上面的推导类似, Σ 的计算可以化为

$$\begin{aligned} \Sigma = & \left(\frac{1}{S'_{M1}} \sum_{tri \in M'_1} \sum_{trj \in M_{tri}} \frac{S_{trj}^{tri}}{N_{trj}^{tri}} \sum_{i=1}^{N_{trj}^{tri}} p_i p_i^T \right) \\ & - \frac{1}{(S'_{M1})^2} \left[\sum_{tri \in M'_1} \sum_{trj \in M_{tri}} \frac{S_{trj}^{tri}}{N_{trj}^{tri}} \sum_{i=1}^{N_{trj}^{tri}} p_i \right] \left[\sum_{tri \in M'_1} \sum_{trj \in M_{tri}} \frac{S_{trj}^{tri}}{N_{trj}^{tri}} \sum_{i=1}^{N_{trj}^{tri}} p'_i \right]^T. \end{aligned} \quad (9)$$

假设三角形 tr_j 和它对应三角形的质心坐标分别为 p_{trj} 和 p'_{trj} , 则 $p_i = p_{trj} + \alpha_i$, $p'_i = p'_{trj} + \beta_i$, 式中 α_i 和 β_i 分别为采样点 p_i 和它的对应点 p'_i 到它们所在三角形质心的偏移量. 则有

$$\begin{aligned} \frac{1}{N_{trj}^{tri}} \sum_{i=1}^{N_{trj}^{tri}} p_i p_i^T &= \frac{1}{N_{trj}^{tri}} \sum_{i=1}^{N_{trj}^{tri}} (p_{trj} + \alpha_i)(p'_{trj} + \beta_i)^T \\ &= \frac{1}{N_{trj}^{tri}} \sum_{i=1}^{N_{trj}^{tri}} (p_{trj} p_{trj}^T + \alpha_i p_{trj}^T + p_{trj} \beta_i^T + \alpha_i \beta_i^T). \end{aligned}$$

由于 p_i ($p_i \in P_{trj}^{tri}$) 和它的对应点 p'_i 所在的三角形面积都非常小, 偏移量 α_i 和 β_i 相对于 p_{trj} 和 p'_{trj} 可以忽略, 所以上式可以近似为

$$\frac{1}{N_{trj}^{tri}} \sum_{i=1}^{N_{trj}^{tri}} p_i p_i^T \approx p_{trj} p_{trj}^T.$$

则

$$\Sigma \approx \frac{1}{S'_{M1}} \sum_{tri \in M'_1} \sum_{trj \in M_{tri}} S^{tri}_{trj} \mathbf{p}_{trj} \mathbf{p}_{trj}^T - \frac{1}{(S'_{M1})^2} \left(\sum_{tri \in M'_1} \sum_{trj \in M_{tri}} S^{tri}_{trj} \mathbf{p}_{trj} \right) \left(\sum_{tri \in M'_1} \sum_{trj \in M_{tri}} S^{tri}_{trj} \mathbf{p}_{trj} \right)^T. \quad (10)$$

由(10)和(8)式可以看出, 只要能求出各个 tr_j 的三个顶点及其对应点就能得到 ICP 算法所需要的互协方差矩阵和评价函数.

距离图像是对真实物体表面的采样结果, 它的每一个像素点实际上代表了物体表面上的一块区域. 由于传感器的位置、方向和物体表面的斜率等原因, 这样的采样是不均匀的, 每一个 3D 采样点所代表的区域大小是不相同的. 如图 2 所示, b 点附近的采样点由于代表的区域较大, 应该比 a 点附近的采样点有更高的权值. 采用点对间最近距离的度量以及它们的互协方差矩阵没有反映出 3D 采样点之间的这种差异. 本文(8)式中的对准误差 E 实际上是用三角形面积对匹配三角形顶点之间的平均距离进行了加权. (10)式也可以看作是用三角形面积对互协方差矩阵中的 3D 采样点进行了加权. 本文通过三角网格之间的距离度量来评价和计算运动参数, 较好地反映了距离图像中 3D 数据点之间的这种差异.

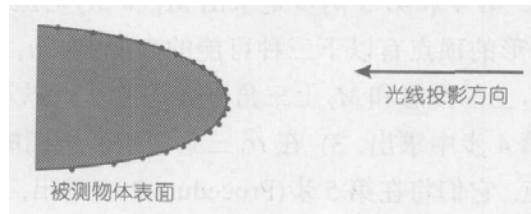


图2 采样点所代表的物体表面面积的不同

2.4 基于面间距离度量的对准算法的主要步骤

下面是本文对准算法 MainProgram 的主要步骤:

MainProgram:

- 1) 构造距离图像 R_1 和 R_2 的三角网格 M_1 和 M_2 ;
令迭代次数 $NI = 0$, 初始对准误差 $E_0 = +\infty$, $T_0 =$ 初始运动参数;
- 2) 求对应三角形集 TR_1 和 TR_2 (过程 Procedure1, 详细描述见下面);
- 3) 将 TR_1 和 TR_2 代入(10)式, 根据求出的 Σ 计算新的运动参数 $T_{M1} = \{R_{M1}, t_{M1}\}$, 并更新三角网格 M_1 为 $R_{M1} M_1 + t_{M1}$;
- 4) 根据(8)式求出新的对应三角形的平均距离 E_{NI+1} ; $NI = NI + 1$;
- 5) 若(对应三角形的平均距离差 $E_{M1-1} - E_{M1}$ 大于阈值 τ), 转 2; 否则返回运动参数 T_{M1} .

其中第 2)步求对应三角形集 TR_1 和 TR_2 的过程 Procedure 1 如下:

Procedure 1:

- 1) 计算并存储三角网格 M_1 的顶点集 P_1 中每一点到三角形网格 M_2 的最近距离和最近点;
- 2) for(M_1 上每一个三角形 tr_i) {
- 3) if(三角形 tr_i 的三个顶点 p_1, p_2 和 p_3 到 M_2 的最近距离都小于阈值 Dis) {
- 4) 求投影方向 pp' , 计算并存储 tr_i 的三条边在 M_2 上的投影以及和 M_2 上三角形各边的交点集 S_1 , 计算 tr_i 的三个顶点的对应点集 S_2 , 得到点集 $Q = S_1 \cup S_2$, 以及 $Q = \{q_i\}$ 在三角形 tr_i 上相对应的点集 $Q' = \{q'_i\}$;
- 5) 求出 $Q_j = \{q_j \mid q_j \in P_2 \text{ 且 } q_j \text{ 在 } tr_i \text{ 三条边投影所围成的区域内}\}$ 和 Q_j 在三角形 tr_i 上的对应点集 $Q'_j = \{q'_j\}$; 并把点集 Q_j 和 Q'_j 并入 Q 和 Q' (过程 Procedure 2, 详细描述见后面);
- 6) 根据 Q, Q' 生成三角形 tr_i 在 M_2 上细分的对应三角形集 $\{tr_i^2\}$, 并求出 $\{tr_i^2\}$ 在三角形 tr_i 上细分的对应三角形集 $\{tr_i^1\}$, 将三角形集 $\{tr_i^1\}$ 和 $\{tr_i^2\}$ 分别并入 TR_1 和 TR_2 . }

上述 Procedure 1 中的第 3 步是通过点对的距离判断它们是否属于重叠区域, 以便排除偏离点(outliers). 第 4 和第 5 两步是求出 M_1 和 M_2 对应三角形集 $\{tr_1\}$ 和 $\{tr_2\}$ 的顶点集. 对应三角形的顶点有以下三种可能的来源: 1) tr_i 的顶点和它们的对应点. 2) tr_i 三条边在 M_2 上的投影和 M_2 上三角形各边的交点以及它们的对应点. 以上两种类型的顶点在第 4 步中求出. 3) 在 tr_i 三条边投影所围成的区域内的 M_2 的顶点以及它们的对应点. 它们将在第 5 步(Procedure 2)中求出, 具体过程如下:

Procedure 2:

- 1) 首先把 $Q = \{q_i\}$ 中每一个点所在的三角形的三个顶点都放入堆栈 $vertex_stock$.
- 2) while($vertex_stock$ 不为空) {
- 3) $x \leftarrow pop(vertex_stock)$;
- 4) if ($In_Tri(x, tr_i, pp')$ 且 x 不属于 Q_j)
- 把 x 加入点集 Q_j , 并把它相邻的而且没有在 $vertex_stock$ 和 Q_j 的顶点加入 $vertex_stock$.

其中 $In_Tri(x, tr_i, pp')$ 当 x 沿着方向 pp' 能够投影到三角形 tr_i 内时, 返回为真, 否则为伪. 根据对应三角形的顶点集 Q 和 Q' , 我们可以容易地求出对应三角形集 $\{tr_i^1\}$ 和 $\{tr_i^2\}$.

3 实验结果

利用模拟和实际的距离图像, 我们对上面的对准算法进行了实验, 下面是部

分实验结果.

实验 1 实验对象是模拟距离图像, 由一个椭球面和一个单叶双曲面所组成的一个看似花瓶状的 3D 物体, 物体表面的数学描述如下:

$$\begin{cases} \frac{x^2}{100^2} + \frac{y^2}{60^2} + \frac{z^2}{80^2} = 1, & -100 \leq x < 25\sqrt{6}, \\ -\frac{x^2}{50^2} + \frac{y^2}{30^2} + \frac{z^2}{40^2} = 1, & 25\sqrt{6} \leq x < 100. \end{cases}$$

把物体正交投影到 xy 平面上, 得到模拟的距离图像(图 3(a)), 将它作为本文算法的第一幅距离图像. 然后将这个 3D 模型分别旋转 15° (图 3(b)), 30° (图 3(c)) 和 45° (图 3(d)), 将生成的距离图像分别作为本文算法的第二幅距离图像.

对于上面的距离图像, 我们还分别对它们的噪声图像进行了实验. 每幅距离图像中加入范围为 -100 到 100 的均匀分布的随机噪声(这种噪声在距离图像中比高斯噪声更难处理). 图 3(e)和(f)分别为加入 10%的随机噪声以及加噪声后再旋转 15° 时得到的 3D 物体. 图 3(g)和(h)分别为加入 20%的随机噪声以及加噪声后再旋转 30° 时得到的 3D 物体.

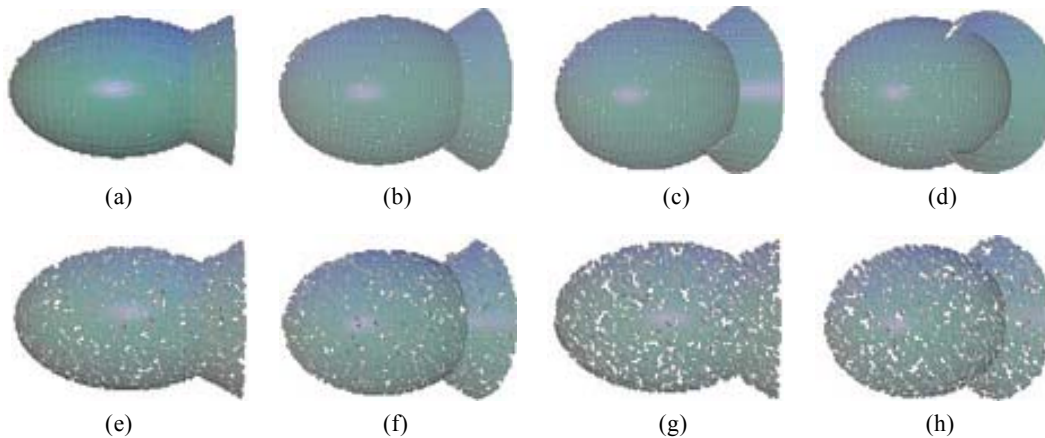


图3 实验 1 中的部分模拟距离图像

(a) 初始位置; (b) 旋转 15° ; (c) 旋转 30° ; (d) 旋转 45° ; (e) 加入 10%噪声; (f) 旋转 15° , 加入 10%噪声; (g) 加入 20%噪声; (h) 旋转 30° , 加入 20%噪声

实验 1 的结果列于表 1. 可以看出, 在没有噪声和具有 10%的噪声情况下, 算法都得到了较好的运动参数估计. 为了更好地检验本文算法的性能, 我们也实现了文献[9]的算法, 两种算法的比较结果见表 2. 由于是模拟距离图像, 不存在获得距离图像的传感器的光源, 为便于比较, 我们将文献[9]中算法的 w_i 设为 1(w_i 为匹配点对可信度的乘积, 而点的可信度通过它的法线方向与点和传感器光源连线的内积求得, 详见文献[9]). 从表 2 可以看出, 本文对准算法的相对误差比文献[9]的要小. 当噪声较大时, 本文算法的优点更加明显. 这主要是因为: 1. 在构

造原始的三角形网格时,对三角形边长的限制具有有效地去除图像噪声的功能,见图 3(e)~(g)和(h); 2. 如 2.5 节所述,本文基于面间距离的对准方法比基于最近点间距离的对准方法有更强的鲁棒性.

表 1 实验 1 的对准结果

	旋转角度(°)	旋转轴			位移量			图例
		X	Y	Z	X	Y	Z	
真实值	15.00	0.00	1.00	0.00	-1.13	0.00	-0.15	(a) (b)
没有噪声	14.99	0.00	1.00	0.00	-1.13	0.00	-0.15	
具有 10%的噪声	15.04	0.00	1.00	0.00	-1.12	0.00	-0.13	
具有 20%的噪声	14.67	0.02	1.00	0.00	-1.02	0.00	-0.12	
具有 30%的噪声	14.21	0.10	0.99	0.03	-0.90	0.01	-0.09	
真实值	30.00	0.00	1.00	0.00	-1.10	0.00	-0.16	(a) (c)
没有噪声	29.98	0.00	1.00	0.00	-1.10	0.00	-0.16	
具有 10%的噪声	30.14	0.00	1.00	0.00	-1.03	0.00	-0.15	
具有 20%的噪声	29.03	0.03	1.00	0.01	-0.90	0.00	-0.11	
具有 30%的噪声	26.62	0.11	0.99	0.01	-0.72	-0.02	-0.11	
真实值	45.00	0.00	1.00	0.00	-1.07	0.00	-0.31	(a) (d)
没有噪声	44.97	0.00	1.00	0.00	-1.07	0.00	-0.30	
具有 10%的噪声	45.22	0.00	1.00	0.00	-1.09	0.00	-0.34	
具有 20%的噪声	42.65	0.07	1.00	0.01	-0.87	0.00	-0.23	
具有 30%的噪声	39.91	0.15	0.98	-0.12	-0.65	0.04	-0.17	

表 2 旋转角估计中两种算法相对误差的比较

噪声	算法	旋转 15°	旋转 30°	旋转 45°
10%	文献[9]	1.20%	1.83%	2.26%
	本文	0.27%	0.47%	0.49%
20%	文献[9]	2.38%	6.95%	7.41%
	本文	2.20%	3.23%	5.22%
30%	文献[9]	46.41%	27.42%	24.46%
	本文算法	5.27%	11.27%	11.31%

实验 2 实验数据是利用我们自己研制的距离传感设备对石膏头像扫描获得的. 距离图像的点集如图 4(a), (c)和 (e)所示. 由 2.1 节的算法所得到的相应的三角形网格如图 4(b), (d)和(f)所示.

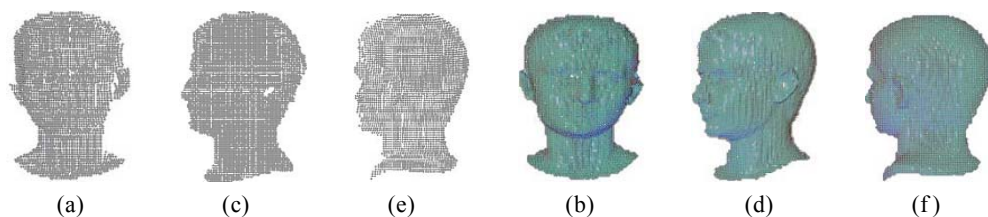


图 4 距离图像及构造的三角形网格

图 5(b)为图 4(b)和 (d)对准的结果. 它们的初始位置如图 5(a)所示, 两幅距离图像大约有 50° 的角度偏转. 图 5(c)为对图 5(b)的对准结果进行集成后的结果(具体的集成算法这里略). 图 6(b)为图 5(c)和图 4(f)的对准和集成结果, 它们的初始位置如图 6(a)所示, 两幅图像也大约有 50° 的角度偏转. 两次对准的图像都有较大的初始位置偏差, 本文算法仍然取得了较好的对准结果.

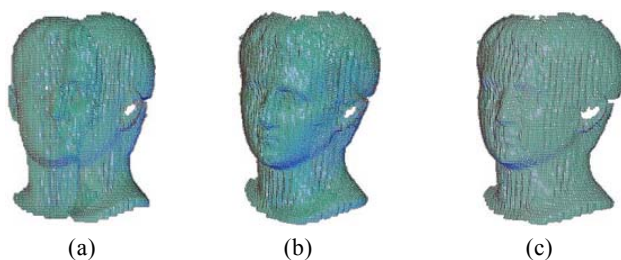


图 5 对准结果(1)

(a) 要对准的两幅距离图像的初始位置; (b) 本文算法的对准结果; (c) 集成的结果

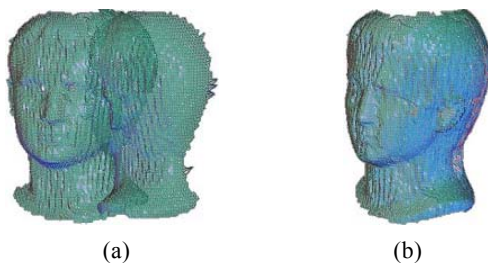


图 6 对准结果(2)

(a) 要对准的两幅距离图像的初始位置; (b) 本文算法的对准结果

我们和文献[9]的算法还进行了收敛速度的对比. 初始运动参数和判断点是否过分偏离的阈值(Procedure1 中第 3 步的 Dis)都取相同的数值. 图 7 为对准图 5(c)和图 4(f)时, 两种算法的迭代次数和对应点对的平均距离之间的关系图. 由图 7 可以看出, 本文算法有更快的收敛速度. 相对于文献[9]的算法, 虽然本文算法的每一次迭代有更多的计算(文献[9]的每一次叠代的平均时间为 5.3s, 本文算法的平均时间为 5.9s, 实验环境为 P II 微机, 主频 333M, 内存 96M), 但由于它的迭代次数较少, 所以综合起来本文算法的收敛速度仍优于文献[9]的算法.

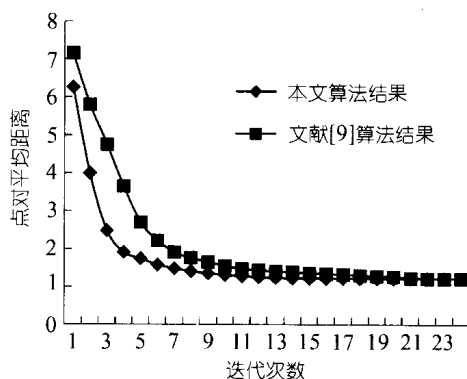


图 7 本文算法和文献[9]的对准算法的迭代次数比较

4 结论

本文提出了一种基于物体表面间距离度量的距离图像的对准算法. 本文的主要贡献是: 1) 提出了一种新的评价距离图像对准误差的目标函数, 它比经典的基于最近点对间距离和的方法更准确地反映了对准的程度. 本文的方法是一种基于表面对准的算法. 和基于点对对准的方法相比, 本文方法有较少的不准确和虚假对应, 因而有更高的对准精度和更快的收敛速度. 2) 通过对三角面块的均匀采样以及利用三角面块的平面性质, 本文将表面对准误差的积分计算简化为计算三角形的三个顶点和它们对应点间距离的均值与三角形面积的积. 这极大地减少了目标函数的计算量, 同时又保证了必要的对准精度.

实验表明, 本文算法对初始运动参数不敏感, 抗噪声能力较强, 同时又有较高的对准精度和较快的收敛速度.

参 考 文 献

- 1 Rodrigues M, Fisher R, Liu Y. Special issue on registration and fusion of range images. *Computer Vision and Image Understanding*, 2002, 87: 1~7
- 2 Horn B K P. Closed-form solution of absolute orientation using unit quaternions. *J Opt Soc Am A*, 1987, 4(4): 629~642
- 3 Chen C S, Hung Y P, Cheng J B. RANSAC-Based DARCES: A new approach to fast automatic registration of partially overlapping range images. *IEEE Trans Pattern Anal Mach Intell*, 1999, 21(11): 1229~1234
- 4 Arun K S, Huang T S, Blostein S D. Least-squares fitting of two 3-D point sets. *IEEE Trans Pattern Anal Mach Intell*, 1987, 9(9): 698~700
- 5 Faugeras O, Herbert M. A 3D recognition and positioning algorithm using geometrical matching between primitive surfaces. In: *Proc of the International Joint Conference on Artificial Intelligence*, Karlsruhe, August 1983, 996~1002
- 6 Huang T, Blostein S D, Margerum E A. Least-squares estimation of motion parameters from 3D point correspondences. In: *Proc IEEE Conf Comput Vision and Pattern Recog*, Miami Beach, FL, June 1986, 198~200
- 7 Besl P J, McKay N D. A method for registration of 3-D shapes. *IEEE Trans Pattern Anal Mach Intell*, 1992, 14(2): 239~256
- 8 Chen Y, Medioni G. Object modeling by registration of multiple range images. *Image Vision Comput*, 1992, 10: 145~155
- 9 Turk G, Levoy M. Zippered polygon meshes from range images. *Computer Graphics*, 28 (Annual Conference Series), 1994, 28: 311~318