

# Fast Reconstruction of 3-D Object Based on Color Image Segmentation Stereo Matching and Point Cloud Reduction

Hexi Li, Juanjuan Zhang, Lingyun Sun

School of Computer Science, Wuyi University, Jiangmen  
Email: [jmlihexi@163.com](mailto:jmlihexi@163.com)

Received: Nov. 6<sup>th</sup>, 2014; revised: Nov. 22<sup>nd</sup>, 2014; accepted: Nov. 29<sup>th</sup>, 2014

Copyright © 2014 by authors and Hans Publishers Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY).  
<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

Color image segmentation stereo matching and point cloud reduction method is used for fast reconstruction of 3-dimensional (3-D) object in this paper. For two captured images of a 3-D object, color image segmentation is first carried out using mean shift algorithm and initial disparity is computed using fast region-based stereo matching, and then the accurate disparity and point cloud of the 3-D object are obtained using belief propagation method to optimize global disparity. The 3-D object is reconstructed using Delaunay triangulation algorithm and point cloud reduction processing based on a surface curvature criterion. The experimental results show that the combination of color image segmentation with belief propagation method can improve stereo matching efficiency and ensure matching quality, and the point cloud reduction technique can rise 3D reconstruction speed and obtain satisfactory 3-D reconstruction result.

## Keywords

Mean Shift, Image Segmentation, Stereo Matching, Point Cloud, 3D Reconstruction

---

## 采用彩色分割立体匹配与简化点云的三维目标快速重建

李鹤喜, 张娟娟, 孙玲云

五邑大学计算机学院, 江门

Email: [jmlihexi@163.com](mailto:jmlihexi@163.com)

收稿日期: 2014年11月6日; 修回日期: 2014年11月22日; 录用日期: 2014年11月29日

## 摘要

本文将彩色分割立体匹配和简化点云技术用于三维目标的快速重建。对于采集的左右两幅三维目标图像, 首先采用均值漂移算法进行彩色图像分割, 然后按区域匹配算法进行快速立体匹配, 获得初始视差, 再应用置信传播法进行全局视差优化, 从而得到精确的视差图与空间点云。应用空间表面曲率准则对获取的密集点云进行简化, 并采用Delaunay三角剖分算法进行三维重建。实验结果表明: 采用彩色图像分割与置信传播相结合, 能够改善立体匹配效率并保证了匹配质量; 基于曲率的点云简化, 既提高了三维重建速度也得到了满意的重建效果。

## 关键词

均值漂移, 图像分割, 立体匹配, 点云, 三维重建

## 1. 引言

在无标识的双目立体视觉系统中, 立体匹配是最重要也是最困难的一部分, 其目的是为了得到 3D 物体的空间坐标; 一般立体匹配算法可分为局部和全局匹配算法, 局部匹配算法匹配速度相对快, 实时性好, 但对光照强度和对比度的变化很敏感, 同时匹配窗口的选取也是一个难点, 当图像存在纹理特征重复、平滑区和遮挡现象比较严重的情况下, 会引起匹配混淆, 错误匹配概率较高。全局的匹配算法, 如图割算法[1] [2]、置信传播[3] [4]和动态规划[5]等算法能够对整个图像进行有效的约束, 匹配结果也较局部匹配算法精确, 但是实时性不好, 匹配时间过长。另外, 立体视觉中对得到的空间点云如何精简以便能够准确描述物体空间形状特征也是加快 3D 物体三维重建速度的关键技术点。

文中针对局部匹配算法和全局匹配算法的缺点和不足, 采用一种区域匹配与全局匹配相结合的算法进行立体匹配。首先, 利用均值漂移算法[6]对左右两幅图像进行彩色图像分割, 再使用相似度测量计算相关度, 得到初始匹配视差图; 其次, 采用置信传播算法, 针对大的遮挡区域和低纹理区域中置信度低的区域, 取邻域相关系数最大的视差值; 此外, 对边缘像素进行修正, 并对整个视差图进行滤波, 从而得到效果精确的视差图。最后, 采用边界特征点提取获得简化的三维点云, 采用 Delaunay 算法进行 3D 目标的三维立体快速重建。

## 2. 基于均值漂移算法的彩色图像分割

均值漂移算法是根据图像像素点的颜色信息和周围空间的分布特性之间的关系, 沿着平均梯度方向找出每个像素点的相似颜色收敛点, 根据收敛点的不同而划分不同的区域, 从而实现图像分割。对于一幅彩色图像统一考虑图像的空间信息和彩色信息, 特征空间可由 2 维的位置空间  $\mathbf{s}$  和 3 维色度空间组成  $\mathbf{r}$ , 图像像素转换成 5 维空间的采样点  $\{x_s, x_r\} = \{x, y, L, u, v\}$ , 则均值漂移向量可以表达为:

$$M_h(x) = \frac{\sum_{i=1}^n G_{sr}(x_i - x) w(x_i) (x_i - x)}{\sum_{i=1}^n G_{sr}(x_i - x) w(x_i)} \quad (1)$$

其中

$$G_{sr}(x_i - x) = G_s \left( \frac{x_{si} - x_s}{h_s} \right) G_r \left( \frac{x_{ri} - x_r}{h_r} \right) \quad (2)$$

这里  $G_{sr}(x_i - x)$  为不同样本点  $x_i$  到中心样本点  $x$  的不同距离对偏移向量的不同贡献而引入的核函数权值，它决定了采样点  $x_i$  与核中心  $x$  之间的相似性度量， $w(x_i)$  为体现样本点  $x_i$  重要性的权重系数， $h_s$  和  $h_r$  分别为坐标与颜色窗口宽度，文中采用单位高斯核函数，即  $G(x) = \exp(-\|x\|)$ ，则变换(1)式为

$$M_h(x) = m_h(x) - x \quad (3)$$

其中：

$$m_h(x) = \frac{\sum_{i=1}^n G_{sr}(x_i - x) w(x_i) x_i}{\sum_{i=1}^n G_{sr}(x_i - x) w(x_i)} \quad (4)$$

$M_h(x)$  表示加权平均偏移向量，指向概率密度梯度方向，则  $m_h(x)$  可以看成是样本点  $x$  沿着概率密度梯度方向偏移后的样本点。在已知  $G(x)$  和  $w(x_i)$  的情况下，根据式(4)进行迭代会收敛到待漂移样本点附近的概率分布密度峰值处。迭代步骤如下：

- 1) 在特征空间中任意选择初始搜索区域圆，设置其搜索半径和误差阈值  $\varepsilon$ ；
- 2) 根据式(8)计算圆中采样点的均值  $m_h(x)$ ，判断是否满足  $\|m_h(x) - x\| < \varepsilon$ ，若满足，则停止循环，否则进入第三步；
- 3) 令  $x = m_h(x)$ ，继续返回第一步开始循环。

### 3. 初始视差图的获取

对彩色图像进行分割之后，采用区域匹配算法求得初始视差图，并且该区域匹配是在分割块中进行的匹配，进一步提高了匹配效率。区域匹配中代价函数是用来计算图像间像素的相似性的测量函数，文中采用 NCC (normalized cross correlation) 归一互相关作为区域匹配的测度函数。由于匹配窗口的大小难以选择，所以文中采用自适应窗算法，该算法步骤如下：

- 1) 设定初始匹配窗口的大小为  $3 \times 3$ ；
- 2) 将窗口的沿上、下、左、右四个方向分别向外扩展一个像素点的大小，判断此时窗口内包含的像素点是否超过规定阈值，若超过，则停止，否则继续执行下一步；
- 3) 记录此时窗口的大小，以左图的该像素区域窗口作为模板，在右图中进行匹配，记录此时得到的视差值为  $d_1$ ，之后再以右图像的该像素区域窗口作为模板，在左图像中进行匹配，记录此时得到的视差值为  $d_2$ ；
- 4) 比较  $d_1$  和  $d_2$  的大小，若  $d_1 = -d_2$ ，则记录此像素点视差值为  $d_1 = d_2$ ，否则令  $d = 0$ 。

### 4. 置信传播算法进行模板视差最优分配

通过区域匹配得到的视差平面模板不够精确，只考虑了模板内像素点之间的影响，而没有考虑到区域块间的相互影响。文中采用全局匹配算法—置信传播算法对初始视差进行全局优化，从而得到更加精确稠密的视差图。置信传播算法主要是利用消息传输和置信度传输机制来实现全局能量函数的最小化的。

#### 4.1. 置信传播算法原理

定义全局能量函数为：

$$E(d) = \sum_{p,q \in N} V(d_p, d_q) + \sum_{p \in P} D_p(d_p) \quad (5)$$

其中  $d$  表示整幅图像的视差分配,  $N$  表示图像中所有像素的四邻域点集,  $d_p$  表示点  $p$  所分配的视差值, 平滑项  $V(d_p, d_q)$  表示两相邻像素点  $p$  和点  $q$  分配视差  $d_p$  和  $d_q$  时的视差不连续惩罚量,  $P$  表示图像中像素点的集合, 数据项  $D_p(d_p)$  表示  $p$  点视差为  $d_p$  时非相似性测度。定义置信传播算法的消息迭代传输:

$$m_{p \rightarrow q}^t(d_q) = \min_{d_p \in \Omega} \left\{ D_p(d_p) + V(d_p, d_q) + \sum_{r \in N(p) \setminus q} m_{r \rightarrow p}^{t-1}(d_p) \right\} \quad (6)$$

$m_{p \rightarrow q}^t(d_q)$  表示第  $t$  次迭代时点  $p$  传输给视差值为  $d_p$  的邻域点  $q$  的消息,  $\Omega$  表示视差搜索空间范围,  $N(p)$  表示点  $p$  的四邻域集,  $N(p) \setminus q$  表示点  $p$  的三邻域集, 它不包括接收消息的点  $q$ 。

经过  $T$  次迭代后, 消息传输趋于稳定, 此时可通过置信度传输计算图像中各个像素点的置信度, 计算点  $p$  置信度的公式为:

$$b_p(d_p) = D_p(d_p) + \sum_{r \in N(p)} m_{r \rightarrow p}^T(d_p) \quad (7)$$

图中各个像素点的最佳视差可以通过最小化置信度获得, 计算点  $p$  的最佳视差  $d_p^*$  的公式为:

$$d_p^* = \arg \min_{d_p \in \Omega} b_p(d_p) \quad (8)$$

## 4.2. 置信传播算法的改进

置信传播算法是在像素点之间的置信度传播, 图像中像素点多, 数量庞大, 也大大增加了算法计算量。而且图像中可能存在单个像素点畸变, 如果使用这些畸变像素点进行消息传输, 会大大降低算法的精度。文中引入彩色图像分割的思想, 采用基于分割区域之间的置信度传播, 用每个分割区域代替单个像素点, 可显著提高立体匹配的速度。

## 5. 三维重建

### 5.1. 空间点云的形成

本实验采用 Bumblebee2 双目摄像机获取三维目标的左右图像, 并通过前面的立体匹配计算获取目标表面的空间三维坐标。对空间物体及周边结构的三维数据采用区域取值选择法进行如下处理:

1) 设置 ROI (Region of Interest) 区, 通过背景差分去掉背景, 获取目标范围, 并以目标为核心截取获取图像 ROI 像素尺寸为  $320 \times 240$  大小, 从而减少空间其他无关点云, 得到较为准确的三维坐标空间点云;

2) 空间物体边界点自动提取处理, 曲面的基本特性主要由曲面法向量和曲率表示。要求得曲面上某一点的法向量, 要先求取该点的切平面。文中引入二维图像的梯度边缘检测来求取三维曲面的法向量和曲率。对于得到的三维空间点云, 看作是一幅深度图像。任意一点  $(x, y)$  的深度值为  $f(x, y)$ , 则二次曲面  $P(x, y)$  可表示为如下:

$$P(x, y) = [x, y, f(x, y)]^T \quad (9)$$

$x$  和  $y$  方向的方向导数分别为  $P_x$  和  $P_y$ , 表示如下:

$$\begin{cases} P_x = [1 & 0 & f_x]^T \\ P_y = [1 & 0 & f_y]^T \end{cases} \quad (10)$$

则单位法向量为:

$$N = \frac{P_x \times P_y}{|P_x \times P_y|} = \frac{1}{\sqrt{1+f_x^2+f_y^2}} \begin{bmatrix} -f_x & -f_y & 1 \end{bmatrix}^T \quad (11)$$

文中使用 Prewitt 算子得到  $f_x$  和  $f_y$ , Prewitt 算子在  $x$  方向和  $y$  方向的掩膜  $G_x$  和  $G_y$  如下:

$$G_x = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad (12)$$

由式(11)可知, 可以求出曲面上任意一点的法向量。曲率为法向量沿着某一给定方向的变化率。在二维曲面上, 假设一点  $p$  点的法向量为  $N$ , 则  $p$  指向  $q$  点的方向矢量记为  $(q-p)$ , 则  $p$  沿该方向的曲率为:

$$k(p, q) = \frac{|N_q - N_p|}{|q-p|} \sigma(p, q) \quad (13)$$

其中  $\sigma(p, q)$  表示如下:

$$\sigma(p, q) = \begin{cases} 1 & (N_p \times N_q) \cdot |N_p \times (q-p)| > 0 \\ -1 & \text{others} \end{cases} \quad (14)$$

由式(13)可求出曲面上任一点四个方向基本曲率表示。假设点  $p$  的  $3 \times 3$  邻域为  $\Omega(p)$ , 则其最大最小曲率及它们的方向表示如下:

$$\begin{cases} k_{\min}(p) = k(p, q_1) = \min_{q \in \Omega(p)} k(p, q), & \text{曲率方向 } M_1 = q_1 - p \\ k_{\max}(p) = k(p, q_2) = \max_{q \in \Omega(p)} k(p, q) & \text{曲率方向 } M_2 = q_2 - p \end{cases} \quad (15)$$

平均曲率为

$$k_{\text{mean}}(p) = [k_{\min}(p, q) + k_{\max}(p, q)] / 2 \quad (16)$$

利用上述曲率值可以从散乱空间点云中提取出曲面的边界。对于初始离散空间点云, 首先, 应该求出候选边界点点集。设定一个阈值大小为  $R$ , 计算散乱空间点云中每个点的平均曲率大小  $k_{\text{mean}}$ , 若  $k_{\text{mean}} > R$ , 记为边界候选点。

3) 非边界点处理, 对于非边界点, 采用固定区域块大小提取保留点, 每  $5 \times 5$  点云数据块中选取一点作为三维重建平面点集。

## 5.2. Delaunay 三角剖分算法

Delaunay 三角剖分主要是将散点集剖分成不均匀的三角形网格。假设  $V$  是二维实数域上的有限点集, 边  $e$  是由点集中的点作为端点构成的封闭线段,  $E$  为  $e$  的集合。那么该点集  $V$  的一个三角剖分  $T = (V, E)$  是一个平面图  $G$ 。该平面图中除了端点, 平面图中的边不包含点集中的任何点。并且平面图中没有相交边, 所有的面都是三角面, 所有三角面的合集是散点集  $V$  的凸包。Delaunay 三角剖分算法的步骤如下:

- 1) 构造一个超级三角形, 包含所有散点, 放入三角形链表;
- 2) 将点集中的散点依次插入, 在三角形链表中找出其外接圆包含插入点的三角形(称为该点的影响三角形), 删除影响三角形的公共边, 将插入点和影响三角形的全部顶点连接起来, 从而完成一个点在 Delaunay 三角形链表中的插入, 如图 1 所示;
- 3) 根据优化准则对局部新形成的三角形进行优化。将形成的三角形放入 Delaunay 三角形链表;
- 4) 循环执行第 2 步, 直到所有散点插入完毕。

## 6. 实验结果

文中实验结果所采用的立体图像对来自 Bumblebee2 双目摄像机，主要截取的 ROI 图大小为  $320 \times 240$ 。计算机硬件条件为：双核 CPU，主频为 2.5 GHz，内存为 2 G。软件编译环境为 VC2008 和 Matlab 7.0。文中匹配算法结合了区域匹配以及全局匹配算法的优点，在精度和速度之间进行权衡，得出了更加精确的视差图，匹配速度也比传统的匹配算法有了很大的提高。图 2 为文中算法所得出的彩色图像分割效果图以及视差图，图 3 为圆柱体和曲面柱体在 Matlab 下利用 Delaunay 算法对目标物体进行快速三维立体重建效果图。

从图 4 看，从重建效果看，采用快速立体匹配和简化点云，尽管这种简化处理形状上有些失真，但

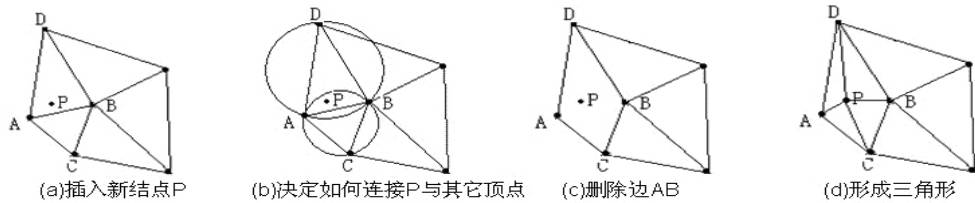


Figure 1. Formation of Delaunay triangulation  
图 1. Delaunay 三角形形成过程

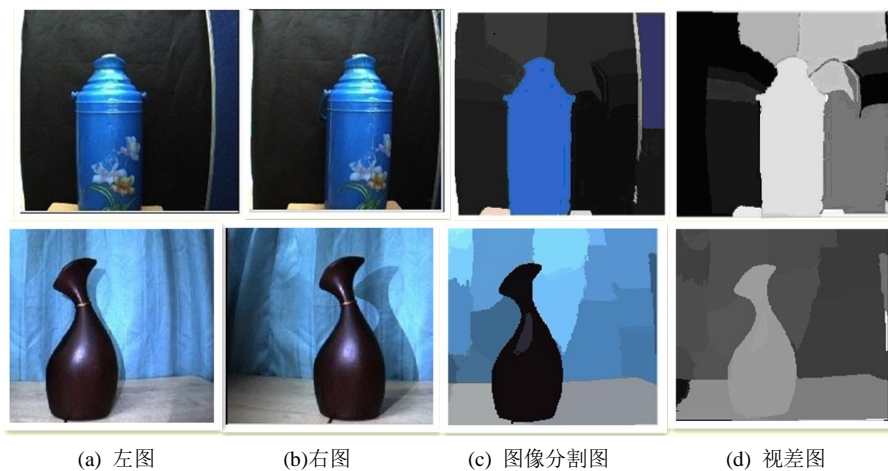


Figure 2. The color segmentation and disparity graph of two 3D-objects  
图 2. 两个 3D 目标的彩色图像分割与最终视差图

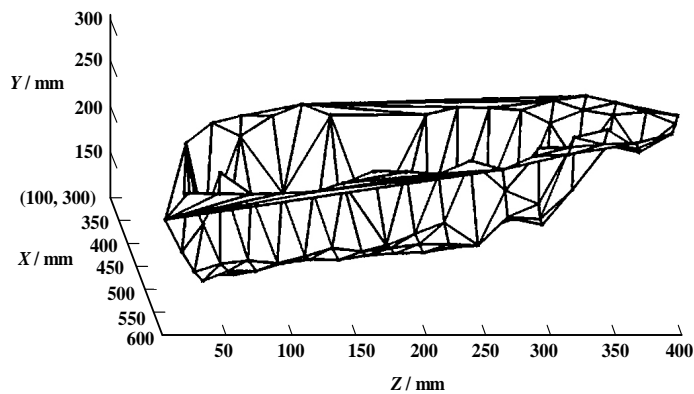


Figure 3. 3D-reconstruction of a cylinder-shaped object from front-view images  
图 3. 圆柱体正面三维重建效果图

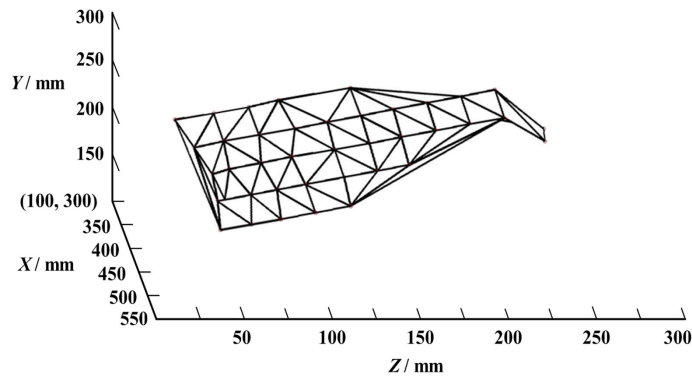


Figure 4. 3D-reconstruction of a curved surface object from front-view images

图 4. 曲面柱体正面重建效果图

空间 3D 目标的整体轮廓得到了重现，并且检测与重建时间大幅度降低，可以满足机器人导航、避障和环境空间立体视觉检测的实时性需求。

## 7. 结论

利用均值漂移彩色图像分割和置信传播相结合的立体匹配算法求得视差图，避免了单独采用区域匹配算法存在的一些问题，如匹配窗口的大小难以选择、左右两幅图像存在重复结构的纹理特征或者遮挡现象引起的匹配混淆等。采用分割的区域块代替原始的像素点进行置信传播计算，减少了匹配所用的时间，同时也可以得到较精确的视差图。最后，按曲率准则对空间点云进行简化，并利用 Delaunay 三角剖分算法进行三维目标重建，速度有了很大提高，同时重建效果图良好，基本上达到了快速三维立体重建的目的。

## 基金项目

广东省自然科学基金资助项目(S2012010010265)。

## 参考文献 (References)

- [1] Kolmogorov, V. and Zabih, R. (2001) Computing visual correspondence with occlusions using graph cuts. *Eighth International Conference on Computer Vision*, 2, 508-515.
- [2] Felzenszwalb, P.F. and Zabih, R. (2011) Dynamic programming and graph cut algorithms in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33, 721-740.
- [3] Yang, Q., Wang, L., Yang, R., et al. (2008) Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31, 492-504.
- [4] Klaus, A., Sormann, M. and Karner, K. (2006) Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. *18th International Conference on Pattern Recognition*, 3, 15-18.
- [5] Deng, Y. and Lin, X. (2006) A fast line segment based dense stereo algorithm using tree dynamic programming. *Proceedings of 9th European Conference on Computer Vision*, Graz, 7-13 May 2006, 201-212.
- [6] Comaniciu, D. and Meer, P. (1999) Mean shift analysis and applications. *International Conference on Computer Vision*, 2, 1197-1203.