

基于 Kinect 的手部跟踪与手部分割

**Hand Tracking and
Segmentation using Kinect**

少年班学院 电子信息工程

王 宇飞 PB09000621

李厚强 教授

二〇一三年六月

中国科学技术大学
University of Science and Technology of China
本科毕业论文
A Dissertation for the Bachelor's Degree

基于 Kinect 的手部跟踪与手部分割
Hand Tracking and Segmentation using Kinect

姓 名 王 宇 飞
B.S. Candidate Yufei Wang
导 师 李 厚 强 教 授
Supervisor Prof. Houqiang Li

二〇一三年六月

June, 2013

中国科学技术大学

学士学位论文



题 目 基于 Kinect 的手部跟踪与手部分割
院 系 少年班学院 电子信息工程
姓 名 王 宇 飞
学 号 PB09000621
导 师 李厚强 教授

二〇一三年六月

University of Science and Technology of China

A Dissertation for the Bachelor's Degree



Hand Tracking and Segmentation using Kinect

B.S. Candidate Yufei Wang

Supervisor Prof.Houqiang Li

Hefei, Anhui 230026, China

June, 2013

致 谢

我要感谢许多帮助我、支持我的人。

首先感谢我的导师，李厚强老师。李老师在我研究的过程中给予我很大的指导和鼓励，在我对可以研究的方向毫无概念的时候指点迷津，在我研究遇到瓶颈的时候总能给我帮助，这对我能坚持完成这篇文章有很大意义。

还要感谢微软亚洲研究院的梅涛老师。梅涛老师作为我在微软亚洲研究院实习期间的导师，对我的学习、研究进行的指导都使我深深收益。我也要感谢梅涛老师对我各种决定的理解和支持。

还必须要感谢唐傲师兄。师兄在编程上、知识学习上都给予我了很大的帮助，没有师兄对整个系统的搭建与完善以及对项目组进度的统筹规划，我的工作不可能如此顺利和迅速的开展。

感谢整个手语组其他 7 位组员给我的帮助和支持。每周的例会上我都能学习很多同学们的新想法、新知识，有了组员之间的互相鼓励与互相促进，我们才进步的更快。虽然我们手语组的同学们一同工作的时间只有短短一年，但我们手语组 9 名成员已经融为了一个大家庭，这一年来学习、生活中的点滴都是我心中最美好的回忆。我们马上就要奔向各自的前程，在这里真心祝愿大家能有最灿烂的未来。

感谢我大学四年中各位任课老师对我的教诲，班主任倪晓玉老师一直以来对我的信任和关心，以及三位室友对我的照顾。

还要特别感谢我的男朋友，陈思同学在我科研过程中对我的支持。他不仅在生活上给予我最大的体谅和照顾，还在我遇到困难或有新点子的时候与我一同讨论，他是我坚强的后盾。

最后把感谢送给我的父母。是你们一直以来的关爱和理解成就了我，你们的健康与快乐是我最大的心愿。

目 录

致 谢	i
摘 要	v
Abstract	vii
第一章 绪论	1
第一节 研究背景及相关工作	1
第二节 设备简述	3
第三节 主要工作概述	3
第二章 手部分割算法的实现	9
第一节 肤色模型	9
一、 非参数模型	9
二、 参数模型：高斯模型	10
第二节 深度模型	12
第三节 区域生长法	13
第四节 深度图像与彩色图像的映射问题	14
第三章 手部跟踪	17
第一节 跟踪算法的实现	17
第二节 遮挡问题的解决	19
一、 椭圆物体假设	19
二、 双手互相遮挡的问题	22
三、 手与脸相碰的问题	23
第四章 特征提取	27
第一节 运动特征	27
第二节 形状特征	27

第五章 实验结果分析	29
第一节 手部跟踪与特征提取结果的分析及与其他方法的对比	29
一、 分割结果的分析及与其他分割方法的对比	29
二、 遮挡问题的处理结果以及与其他遮挡处理方法的对比	30
第二节 后期步骤——手形分类简述	33
第六章 总结和展望	37
参考文献	39

摘 要

随着计算机视觉和机器学习技术不断的发展和设备的快速更新，手语识别的研究受到了越来越多的重视，而手部的分割、跟踪与手形识别是手语识别中的一个重要而富有挑战性的问题。我们提出了一个近实时的用于手语识别的手形分类系统，而本文主要阐述了手的分割与跟踪过程，以及对遮挡问题的应对。我们利用 Kinect，将深度与彩色信息相结合进行手部的分割。这样做的优势在于系统的成功运行要求较少的条件：既不要求背景颜色是单一的或是与肤色不相近的，也不在深度信息上要求手是距离摄像头最近的物体。我们在 Kinect 的骨骼跟踪技术基础之上建立了简单有效的手部跟踪模型，使得系统满足了手语识别对实时准确跟踪手部动作的要求。另外，系统还通过建立物体假设椭圆来解决两手之间的互相碰触以及手与脸互相遮挡碰触的情况。实验结果显示我们的系统可以进行鲁棒的手部跟踪、分割和识别。

关键字： 手语识别，Kinect，肤色模型，区域生长，骨骼跟踪，遮挡处理，物体假设

Abstract

With the rapid development of computer vision and machine learning theories and renovation of human-computer interaction devices, Sign Language Recognition (SLR) has received more and more attention. Hand segmentation, hand tracking and hand shape classification are essential and challenging problems in SLR system. We build up a near real-time sign language recognition system based on Kinect, and this paper focuses on hand segmentation, tracking, occlusion handling and feature extraction. The system combines color and depth data for hand segmentation and tracking using Kinect. The system does not require uniform colored or stable background, and can handle the situation when hands are very close to other parts of the body or when hands are not the nearest object to the camera. We take advantage of the skeletal tracking system provided by Kinect SDK and build a robust and real-time hand tracking method with simple but effective tracking algorithm. The system also allows for the hands to occlude with each other or with face, establishing an object hypothesis method. The experiments demonstrate robust segmentation and tracking results, and show accuracy and robustness of our hand shape classification system

Keywords: Sign Language Recognition, Kinect, Skin Color Model, Region Growing, Skeleton Tracking, Occlusion Handling, Object Hypothesis

第一章 绪论

第一节 研究背景及相关工作

手语识别是一项十分具有社会意义的工作。如今越来越多的语音识别软件已被用户所接受，但关于手语识别的产品却非常罕见，这主要是由于手语构成的复杂性和平行性。手语者的面部表情、手形、手的位置、手的运动轨迹都是手语的重要组成，而其中手部的特征是手语识别中最重要的部分。与许多研究中的手势识别问题不同的是，手语者的手部运动迅速而且手势多变，识别对手的跟踪以及分割的准确率要求很高，因此，手的分割和跟踪是手语识别重要而基础的一步。

一些近期的研究对解决手部分割和跟踪问题做了各种尝试，它们在利用的信息种类上主要可以分为三大类：基于颜色的检测、基于深度的检测以及基于两者结合的检测。

基于颜色的手部检测主要利用的是将手部颜色和背景颜色区分开以达到分割、跟踪的目的。一个非常直接的做法就是利用肤色模型。^[1] 利用 YCbCr 空间的经验得到的阈值来进行肤色判决；^[2] 分析了大量数据得到皮肤的颜色分布直方图以表示肤色概率分布；而^[3] 结合了高斯混合模型和基于直方图的模型。但是，基于肤色的检测的缺陷在于背景中与肤色相近的物体会成为检测的干扰，而手语者的脸部也会对系统造成干扰，所以成功的手部检测依赖于背景的无肤色性。为了得到更稳定的结果，有些研究利用了颜色手套（^[4]），其颜色非常容易检测且不易与其他颜色混淆，但手套使得系统的用户体验大大降低。还有一些研究在肤色检测的基础上加入了其他减除干扰的方法，例如背景减除（^[5]，^[6]）。但这又对系统提出了新的要求：背景必须是稳定的。

面对上述基于颜色信息的检测的缺陷，有些研究将目光转向了深度信息。

Time of Flight (ToF) 深度摄像机是一个常用于获取深度信息的装置 ([7], [8], [9], [10])。而微软 2010 年推出的 Kinect 深度感应装置则提供了一个价格不高的获取深度信息的途径。许多文章利用这些设备提供的深度信息进行手部的检测 ([11], [12])。但是，大部分基于深度的检测都假设手在镜头的最前方。这个假设在手势识别的研究中是有一定道理的，但在手语识别中，手部经常在身体附近运动，还与身体的其他部分有着经常性的接触，不能满足这个要求，因此仅仅基于深度的手部检测有着很大缺陷。微软推出的 Kinect SDK[13] 中包含有骨骼跟踪系统。骨骼信息可以很好的弥补深度信息的缺陷，在本文中，我们正是利用了骨骼跟踪系统的优点，以得到可信的跟踪结果。

为了更好地解决分割问题，[12][4] 将彩色信息与深度信息结合以克服两者各自的缺陷。[4] 利用了多个 Kinect 来跟踪手部，一个高精度彩色摄像头来分割手部，所有的设备都需要被校准。[12] 同样利用了 Kinect，深度信息首先被获取，然后离镜头最近的物体被认为是手部，并进行分割。实验者需要佩戴黑色腕带。这个方法依旧存在着手部位置被限制在镜头最前方的问题。

很多方法被应用于手部跟踪问题。Kalman 滤波是最常见的方法。Kalman 滤波是一个递归的滤波器，可以对动态系统进行估计。[14] 利用 Kinect 获取深度图像，利用 Kalman 滤波进行跟踪，手的三维位置即为 Kalman 滤波中的状态，手的速度被认为是控制向量。均值漂移法（Meanshift）是一个通过迭代运算，利用梯度下降法寻找目标位置的算法。[15] 利用均值漂移法寻找合适的测量状态，利用 Kalman 滤波进行估计，在彩色视频中进行手部的跟踪。粒子滤波（Particle Filter）利用随机样本来表示概率密度函数，估计系统状态进行预测。[16] 结合了粒子滤波与均值漂移，利用均值漂移算法来优化粒子的选取，进行手部跟踪。

为了进行手形的分类，很多种手形特征被尝试。它们主要可以被分为低层次的特征和高层次的特征。低层次的特征寻找手部形状某一方面的特点，例如手的主轴方向、手部拟合椭圆的离心率等等；而高层次的特征试图寻找一种描述子来描述手的整个形状，例如不变矩、边界信息、傅里叶描述子等等。[11] 中提取出两种特征：空间占用率特征和轮廓特征，来进行手形的分割。而 [1] 中则除了轮廓特征外，提取了（用脸部面积归一化后的）手的面积特征、以及不变矩特征。这些特征都是人工提取的特征，它们的优点是十分直观，但缺点是特征的表述力不足。我们的方法中，手部图像作为特征直接被提取。我们试

图利用神经网络的方法，避免了人工提取特征表述力不足的问题。

第二节 设备简述

在我们的系统中，Kinect（1.1）被用作获取原始数据流和骨骼信息的装置。Kinect 是微软在 2010 年推出的用于人的肢体动作捕捉的感应器。它有三个镜头，其中中间的 RGB 彩色摄影机，左右两边的摄像头分别为一个红外线发射器和一个红外线 CMOS 摄影机，它们构成了 3D 结构光深度感应器。

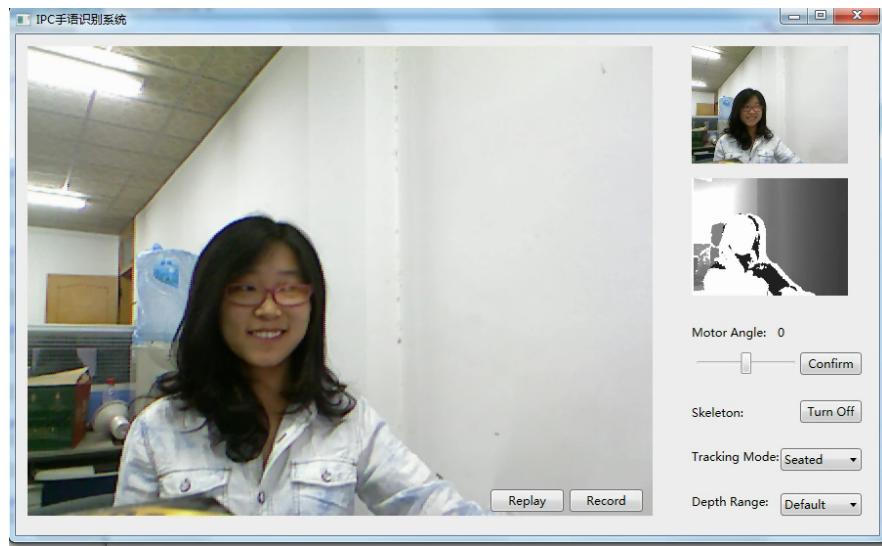


图 1.1 Kinect 装置

基于 Kinect 最常用的两个开发包是微软官方提供的 Kinect SDK [13] 和 OpenNi [17]。我们在这里使用的是微软官方提供的开发包。它提供了每秒 30 帧的低精度颜色和深度信息（图1.2），我们使用的精度为其提供的最高精度 640×480 。同时 Kinect 还提供了骨骼跟踪结果。骨骼跟踪系统有两种模式：“就座模式”和普通模式（图1.3），“就座模式”只追踪上半身 10 个关节（肩膀，肘部，腕部，手臂以及头部），由于手语识别只有上半身的肢体提供有用信息，所以我们选择“就座模式”。

第三节 主要工作概述

本文的工作主要包括以下几个方面：



(a) 彩色图像



(b) 深度图像

图 1.2 Kinect 获取的数据流

1. 手部的分割:

利用区域生长法，结合深度与彩色信息，建立深度模型和肤色模型作为判决依据，进行手部的分割。非参数肤色模型和高斯肤色模型共两种肤色模型被建立并比较，最终非参数模型被选取。

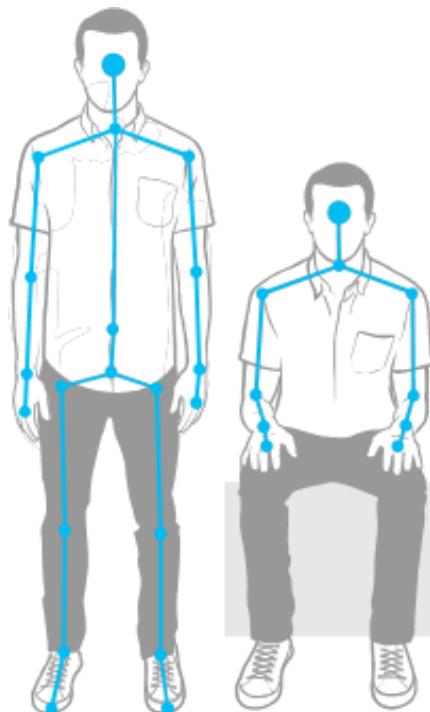


图 1.3 两种跟踪模式 左：普通模式 右：就座模式

2. 手部的跟踪：

利用 Kinect SDK 中的骨骼跟踪算法，将手部跟踪算法与骨骼跟踪结果相结合，得到更加稳定可靠的跟踪结果。针对手语中经常出现的遮挡情况进行处理，建立物体假设椭圆，利用将物体假设和检测区域对应起来的方法，处理手部互相触碰以及手部与脸部相接触、遮挡的情况，得到在遮挡情况下的手部位置信息和形状信息。

3. 手部特征的提取：

手部的运动信息以及形状特征被记录。当遮挡情况发生时，对于手部互相遮挡情况，双手区域作为一个整体被记录为手形特征，而对于手与脸互相遮挡的情况，手与脸则被分开，手部区域单独作为手形特征被记录。

系统的主要流程如图1.4。

本文的组织结构如下：

- 第二章介绍手形分割算法的实现，主要介绍了肤色模型的建立、深度模

型的建立以及利用区域生长法进行分割的过程。

- 第三章主要介绍了手部跟踪算法的实现以及遮挡情况的处理。遮挡主要分为两种情况：两手互相遮挡触碰的情况，以及手与脸部互相遮挡的情况。
- 第四章简要介绍了系统提取的手部特征。
- 第五章分析了我们的实验结果，将我们的方法与其他文章中所用的方法进行了对比，并简要展示了后续工作——手形分类的结果。
- 第六章进行了简单的总结，并对以后工作进行了展望。

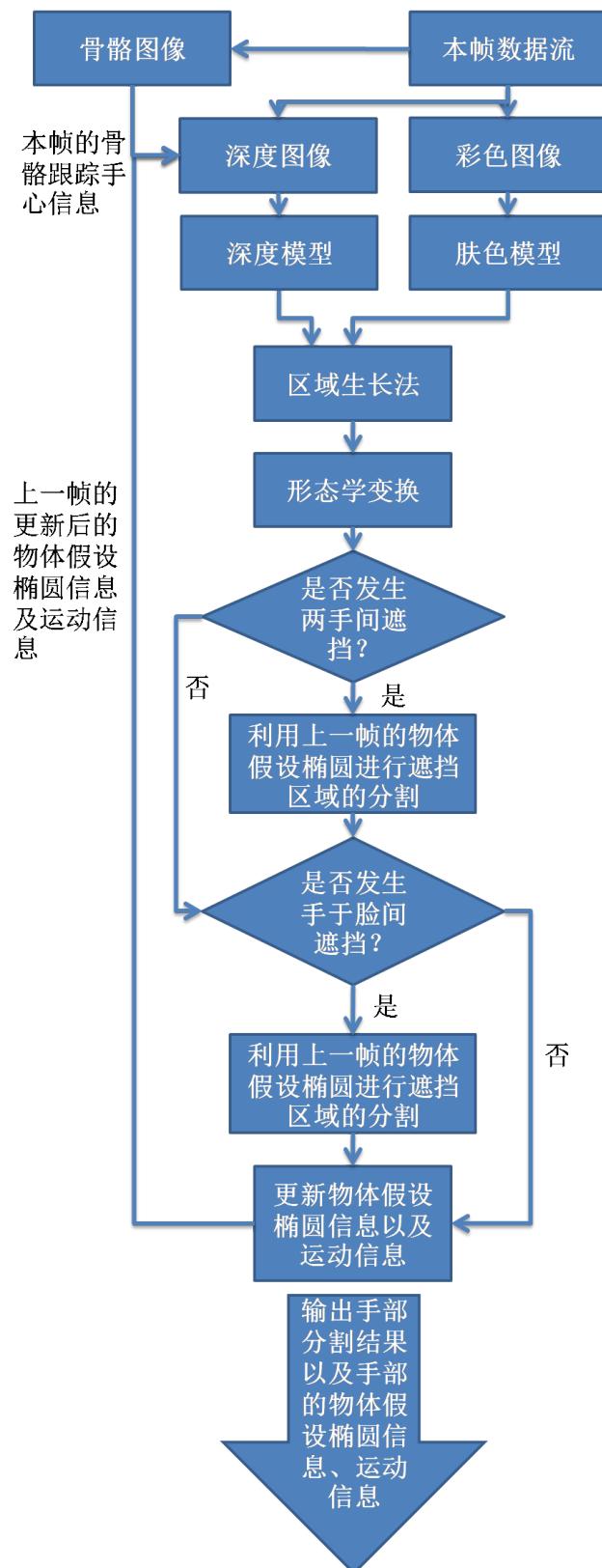


图 1.4 系统流程

第二章 手部分割算法的实现

本章主要叙述了手部分割的算法。我们建立了肤色模型和深度模型，利用区域生长法进行手部分割。

第一节 肤色模型

采用肤色模型是一个合理的选择，因为人的肤色在颜色空间上有着可预测的分布。我们在系统里假设实验者身着长袖衣服，衣服的颜色与肤色可以区分开。

首先，我们将色彩图像从 RGB 颜色空间转化到 YC_bC_r 空间。 Y 分量表征了一个点的亮度， C_b 和 C_r 表征了这个点的色度，它们分别表示蓝色色度分量和红色色度分量。在 YC_bC_r 空间操作的优点是我们只对 C_b 和 C_r 这二维颜色参数进行建模，这样使得建立的肤色模型可以在一定程度上抵御亮度的变化。

RGB 转换至 C_r 的公式如下：

$$\begin{cases} Y = 0.299 \times R + 0.587 \times G + 0.114 \times B \\ C_r = 0.500 \times R - 0.419 \times G - 0.081 \times B \\ C_b = 0.169 \times R - 0.331 \times G - 0.500 \times B \end{cases} \quad (2-1)$$

在这里，有两种肤色模型被试验并比较。

一、非参数模型

大量实践表明，肤色的色度分量一般分布在如下区域内：

$$\begin{cases} 135 < C_b < 180 \\ 85 < C_r < 135 \end{cases} \quad (2-2)$$

因此，一个简单的非参数阈值设定即可将肤色与其他颜色分离出来。利用非参数模型进行皮肤识别的结果如图2.1(a)。在图中，整幅彩色图像被扫描，被判别为肤色的点的颜色保持不变，被判别为非肤色的点的颜色设为 0（黑色），这样，得到了如图2.1(a)的结果。我们的系统使用非参数模型得到的手部分割结果如图2.1(b) 所示。图中，背景为黑色，人体骨骼由 Kinect 骨骼跟踪系统识别，蓝色椭圆为脸部肤色区域，两只手的分割结果用两种方式描绘：在原位置上，左手轮廓用蓝色的轮廓线描绘，右手轮廓用紫色的轮廓线描绘，而两手的分割图像被分别描绘在了图的左上角处。另外，为了更直观的表现系统对手部进行跟踪的过程，我们将右手拟合的椭圆用灰色椭圆表示，右手的预测椭圆用白色椭圆表示。左手周围的白色方框为特征提取时的矩形感兴趣区域。若有两手遮挡情况的发生，双手区域将用拟合的紫色椭圆来表示。这些内容的具体实现在第三章有详细的描述。

二、参数模型：高斯模型

与 [18] 中所用方法类似，为了使肤色模型适应于不同人的肤色和实验的光照条件，我们利用线上训练的方式训练肤色的高斯模型。人脸的信息被利用，因为人脸的检测稳定可靠，且每个人的手部肤色与其脸部肤色相近。

首先，人的脸部肤色点被检测出来。检测方法为利用 Kinect 骨骼跟踪得到的脸部位置进行区域扩散的方法。

接下来脸部肤色点被用来建立二维高斯模型。

$$f(\mathbf{C}) = \exp\left(-\frac{1}{2}(\mathbf{C} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{C} - \boldsymbol{\mu})\right) \quad (2-3)$$

其中， $\mathbf{C} = (C_b, C_r)$ ， $\boldsymbol{\mu} = (\mu_b, \mu_r)$ 为肤色的色度分量的样本均值 $\boldsymbol{\mu} = E(\mathbf{C})$ ， Σ 为高斯模型的协方差矩阵 $\Sigma = E[(\mathbf{C} - \boldsymbol{\mu})(\mathbf{C} - \boldsymbol{\mu})^T]$ 。当某点的高斯函数 $f(\mathbf{C})$ 大于指定的阈值时，我们认为这个点的颜色为肤色。为了训练出更可信稳定的模型，我们每五帧一次地将脸部数据加入训练数据库。高斯模型的阈值选取十分重要：当阈值较高时，手部不能完整被识别；而当阈值较低时，容易产生误识别。经过实验，我们找到合适的阈值 0.036。结果如图2.2所示。如图2.2(a) 所示，仅对整幅彩色图像进行肤色判断，得到了与2.1(a) 相差不多的结果。系统利用建立的高斯模型分割的手部结果如图2.2(b) 所示。

可以看出，高斯模型与非参数模型都可以达到不错的手部分割效果。对比图2.1(a)和图2.2(a)可以发现两种模型对颜色的分析稍有不同，非参数模型对冷色调的颜色的剔除效果更好，而高斯模型可以更好的分开红色等暖色调的颜色与肤色的差别。总的来说，高斯模型的建立并没有取得明显的检测结果上的优化，图2.1(b)与图2.2(b)的效果接近。但另一方面，线上高斯建模的方式的运算代价比简单的非参数模型大得多。因此，我们在这里选择非参数模型作为系统的肤色模型。

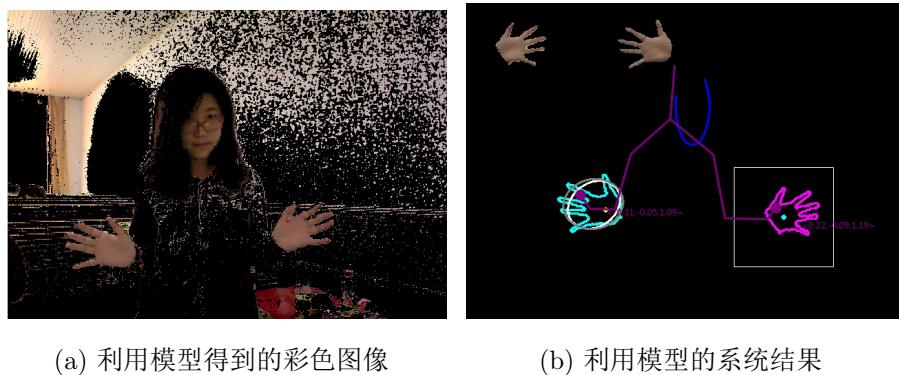
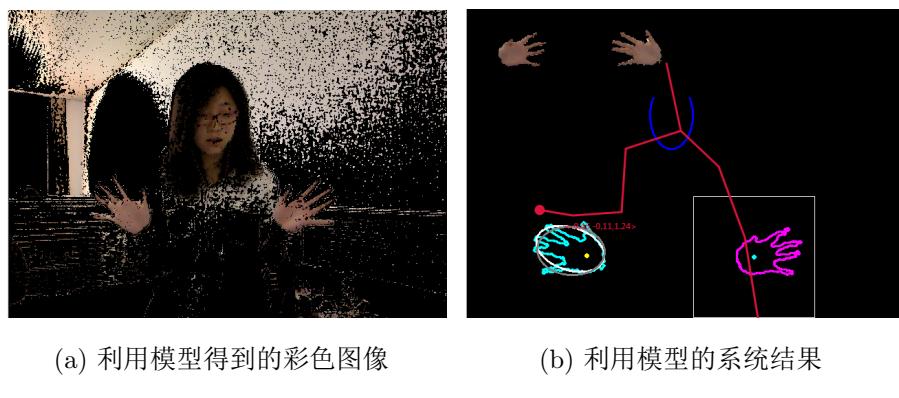


图 2.1 非参数肤色模型结果

图 2.2 高斯肤色模型结果，参数 $f = 0.036$

第二节 深度模型

为了提高检测结果的准确性，我们利用了深度信息。我们假设一个属于手部的点的深度与手心的深度差值在某个阈值以内。

首先，深度图像被映射到了彩色图像空间，这使得每一个 *RGB* 图像中的点都有了其对应的深度值。

然后，手心的深度信息 *palmDepth* 被预测。手心位置的预测是由骨骼跟踪结果和手部跟踪结果共同决定的，这部分在第三章会有详细阐述。

当预测的手心深度被决定后，手部区域可以通过设定阈值来找到：

$$palmDepth - \Delta d_1 < z(x, y) < palmDepth + \Delta d_2 \quad (2-4)$$

其中, Δd_1 和 Δd_2 是预先确定的常量， $z(x, y)$ 为点 (x, y) 的深度。利用深度信息进行分割得到的结果如图2.3。

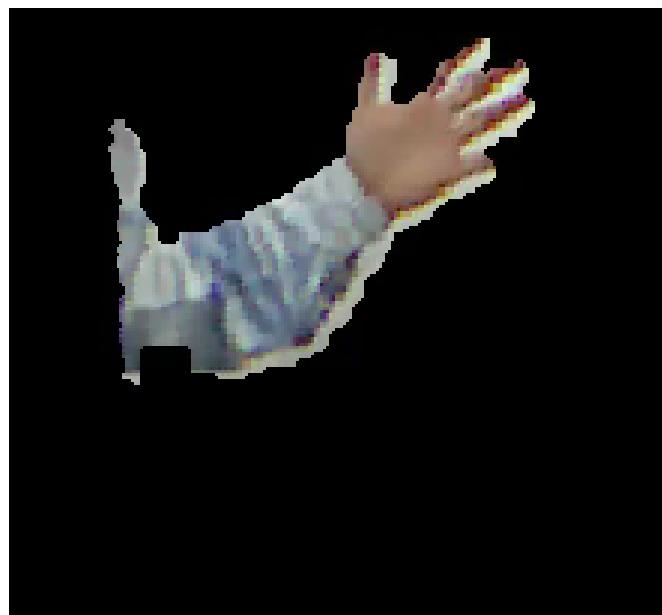


图 2.3 用深度模型进行手部分割结果

第三节 区域生长法

为了得到闭合的手部区域，且满足系统对运算速度的要求，我们使用区域生长法来进行对图像进行分割。它的优点在于在没有指定感兴趣区域的前提下，分割不需要遍历整幅图像，而且分割结果正是我们所需要的闭合的区域，省去了繁琐的后期处理步骤。

区域生长法的第一步要确定种子点，由手部跟踪结果和骨骼跟踪结果共同确定。当骨骼跟踪的手心点被采用为预测手心点时，种子点是预测的手心点以及其 5×5 邻域内的满足要求的点；当对手部检测的跟踪结果被采用为预测手部位置时，种子点是预测区域内所有符合要求的点。跟踪算法在第三章有详细的介绍。

接下来，区域生长的生长准则由深度与彩色信息同时决定。在生长过程中，我们选择 8 连通的连接区域，以得到更理想的区域边缘。区域生长法的算法流程如以下伪码所示：

算法：手部的区域生长

- 01 输入：手部预测信息（骨骼跟踪手心点 P_{hand} ，手部区域的预测区域：栈 $predictPoints$ ）。彩色图像与深度图像。
- 02 输出： $skinPixelData$ 数组表征的检测区域。值为 255 的点为检测区域，值为 0 的点为背景。
- 03 初始化 $skinPixelData$ 数组，以存入手部区域信息；初始化栈 $pointStack$ 。

第一部分：选取种子点

- 04 如果 骨骼跟踪的手心点 $P_{hand} = (p_x, p_y)$ 被采用 那么
- 05 $PalmDepth := P_{hand}.Depth$
- 06 对于 -每个 $p \in P_{hand}$ 的周围 5×5 邻域内 做
- 07 如果 p 满足肤色模型以及深度模型 那么
- 08 $skinPixelData[p.pixelIndex] := 255$
- 09 p 的坐标入栈 $pointStack$
- 10 结束 -如果
- 11 结束 -循环

```

12    结束 -如果
13    如果 由之前帧的手部检测结果得到的预测被采用 那么
14        当 栈 predictPoints 不空 做
15            tempPoint := predictPoints.Pop
16            如果 tempPoint 满足肤色模型 那么
17                skinPixelData[tempPoint.pixelIndex] := 255
18                p 的坐标入栈 pointStack
19                p.Depth 被用作计算 PalmDepth
20            结束 -循环
21        结束 -循环
22    结束 -如果

```

第二部分：区域生长

```

23    当 栈 pointStack 不空 做
24        tempPoint := pointStack.Pop
25        对于 -每个 p ∈ tempPoint 的周围 8 邻域 做
26            如果 p 满足两个模型且 skinPixelData[p.pixelIndex] = 0 那么
27                p 的坐标入栈 pointStack
28                skinPixelData[p.pixelIndex] := 255
29            结束 -如果
30        结束 -循环
31    结束 -循环

```

之后，我们对分割出的区域进行了形态学优化。为了使得区域边界更加平滑，闭运算被使用。当我们找到了一个区域 r ，且它的面积比某个已设定的阈值 T_{area} 大的时候，我们认为它是一个检测到的手部区域 (r_{hand})。图2.4显示了一个被检测到的手部区域。

第四节 深度图像与彩色图像的映射问题

在分割算法中，深度图像到彩色图像的映射是基础的一步，而映射结果的准确性则是影响检测结果的一个重要因素。不幸的是，Kinect SDK 提供的

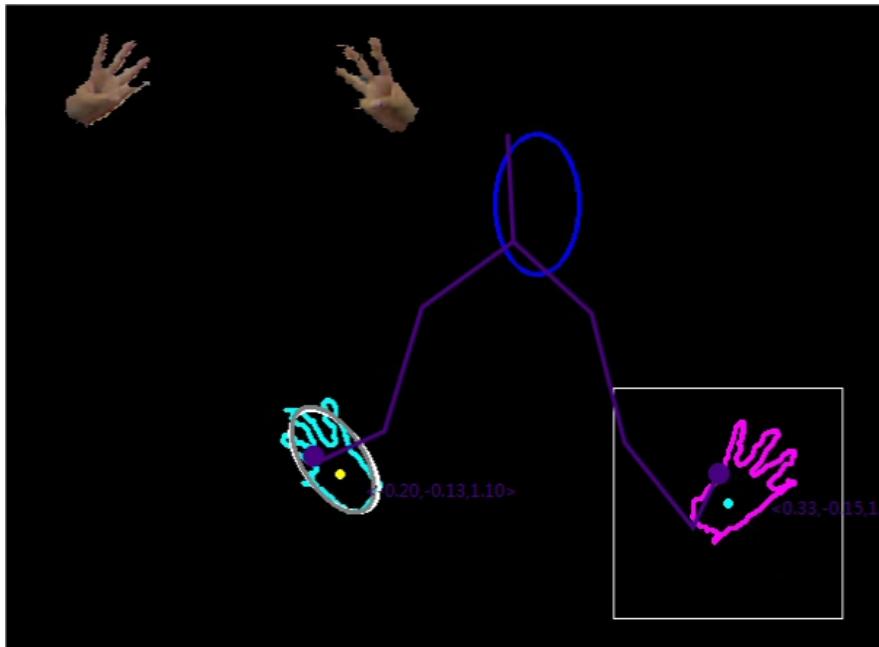


图 2.4 区域生长法得到的手部区域

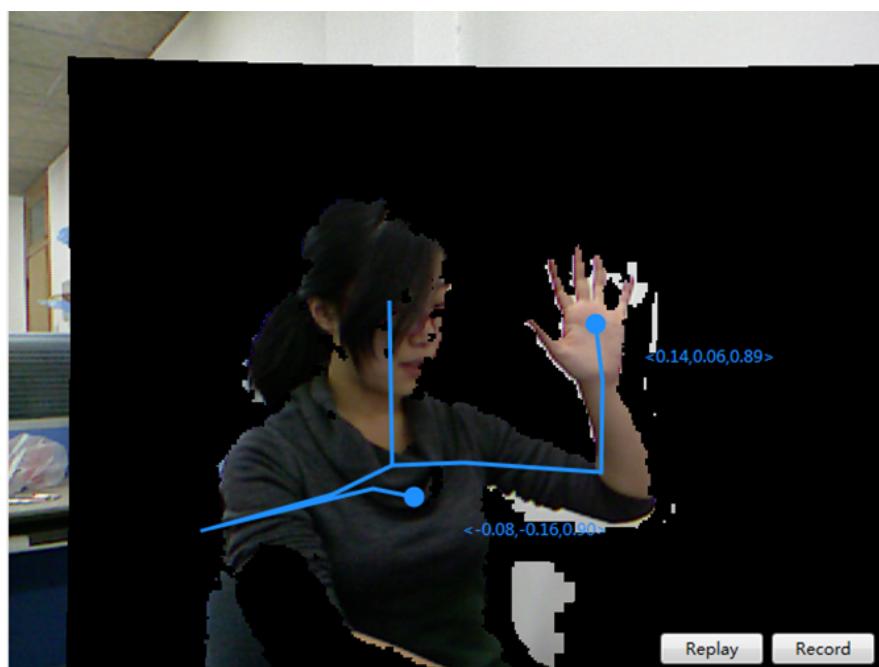
映射结果不能完全令人满意，彩色像素点与深度像素点不能准确的一一对应。图2.5给出了这个问题的形象表示。图2.5(a)为将利用深度信息找到的手部边界映射到彩色图像中后，可以看出边界并没有精确的与彩色图像的边界契合。图2.5(b)为将深度图像中识别的人体映射到彩色图像上，并在彩图中抠出时，可以明显看到映射误差的存在。因此，我们在进行区域生长时将深度判断标准进行了一定程度的放宽，即公式2-4 变为：

$$palmDepth - \Delta d_1 < z(x', y') < palmDepth + \Delta d_2 \quad (2-5)$$

其中 (x', y') 为当前点 (x, y) 的 3×3 邻域内的点。



(a) 将利用深度信息找到的手部边界映射到彩色图像



(b) 将深度图像中识别的人体映射到彩色图像上，并在彩图中抠出

图 2.5 深度图像与彩色图像的映射问题

第三章 手部跟踪

本章介绍了手部跟踪过程的实现，以及对于遮挡问题的处理。遮挡问题指的是两手互相接触遮挡以及手与脸部互相遮挡的情况。

第一节 跟踪算法的实现

与第一章中所提到的各种跟踪算法不同的是，我们试图利用一个较可信的观测结果，在其基础之上进行更可靠的跟踪。

Kinect SDK 提供的骨骼跟踪提供了两只手心的三维位置。这个结果多数情况可以被采用。但是，骨骼跟踪系统并不能一直很好的跟踪手部的运动，尤其当两只手互相接触或是它们之间的距离较近的时候（图3.1），骨骼跟踪系统对手部的跟踪经常会有跳跃性的跟丢。而两只手互相接触的动作是手语中常见的动作，所以骨骼跟踪系统对手部的跟丢情况时常发生。

用于手部分割的区域生长法的准确性很大程度上依赖于种子点是否选取得当。仅靠骨骼跟踪的方法不能得到准确的种子点，因此我们结合了骨骼跟踪和手部跟踪的结果，以得到每帧手部的预测位置，作为本帧检测的种子点。

$$S(t) = F(S_{predict}(t), S_{skeletal}(t)) \quad (3-1)$$

其中， $S(t) = \{(x, y) | (x, y) \in \vec{p}_{seed}\}$ 表示了时刻 t 的种子点点集， $S_{predict}(t)$ 表示由之前帧的手部分割结果预测的种子点点集， $S_{skeletal}(t)$ 表示骨骼跟踪得到的种子点点集。而 F 是决策函数，选择更可靠的手心预测结果作为种子点点集。

$S_{predict}(t)$ 是通过一个简单的规则预测的。它基于 t 时刻前三帧的观测结果。我们假设手的运动是匀加速运动，所以我们对每个属于手部区域的点的预



图 3.1 当两手接近时，骨骼跟踪时常会出现错误

测 $\hat{p}(t)$ 满足：

$$\hat{p}(t) = \vec{p}(t-1) + \vec{v}(t-1) + \frac{1}{2} \vec{a}(t-1) \quad (3-2)$$

其中 $\vec{p}(t-1)$ 是前一帧的手部点的位置。前一帧此手部点的移动速度由手心 $\vec{C}(t-1)$ 的移动速度表示， $\vec{v}(t-1) = \vec{C}(t-1) - \vec{C}(t-2)$ ，而加速度 $\vec{a}(t-1) = \vec{v}(t-1) - \vec{v}(t-2)$ 。每一个手部区域的预测点组成了本帧 t 的手部区域预测点点集，这些点中满足第二章所述的肤色模型的点组成了点集 $S_{predict}(t)$ 。

而 $S_{skeletal}(t)$ 为骨骼跟踪结果所示的手心位置 $\vec{p}_{hand}(t)$ 以及其周围 5×5 区域内满足肤色模型和深度模型的点集。

F 是一个启发式的判决标准。当前一帧的显示表明两只手距离较近时， $S_{predict}(t)$ 被认为是更加可靠的种子点。否则， $S_{skeletal}(t)$ 是首先被选作种子点。当根据选定好的种子点检测到的区域面积不够大 ($Area(R) < T_{area}$) 时，另一种方法预测的点集被选作种子点。

当一个手部区域 r_{hand} 被检测到后，观测手心的二维位置 $\vec{C}(t)$ 被计算出来。观测手心位置由 r_{hand} 决定，是到 r_{hand} 边界距离最大的 r_{hand} 区域内部的点。观测手心位置 $\vec{C}(t)$ 被记录，用于下一帧的手心预测。

跟踪结果如图3.2所示。从左至右，从上到下为按时间顺序截取的视频图像。可以看出，当双手逐渐靠近时，骨骼跟踪的手心点变得不可靠，而我们的跟踪算法保证了双手的正确跟踪。图中右手的预测位置用白色椭圆表示，可以看出，通过预测我们可以在骨骼跟踪结果错误的情况下较可靠地进行手部的跟踪。预测椭圆的参数设置在下一节中会有详细描述。

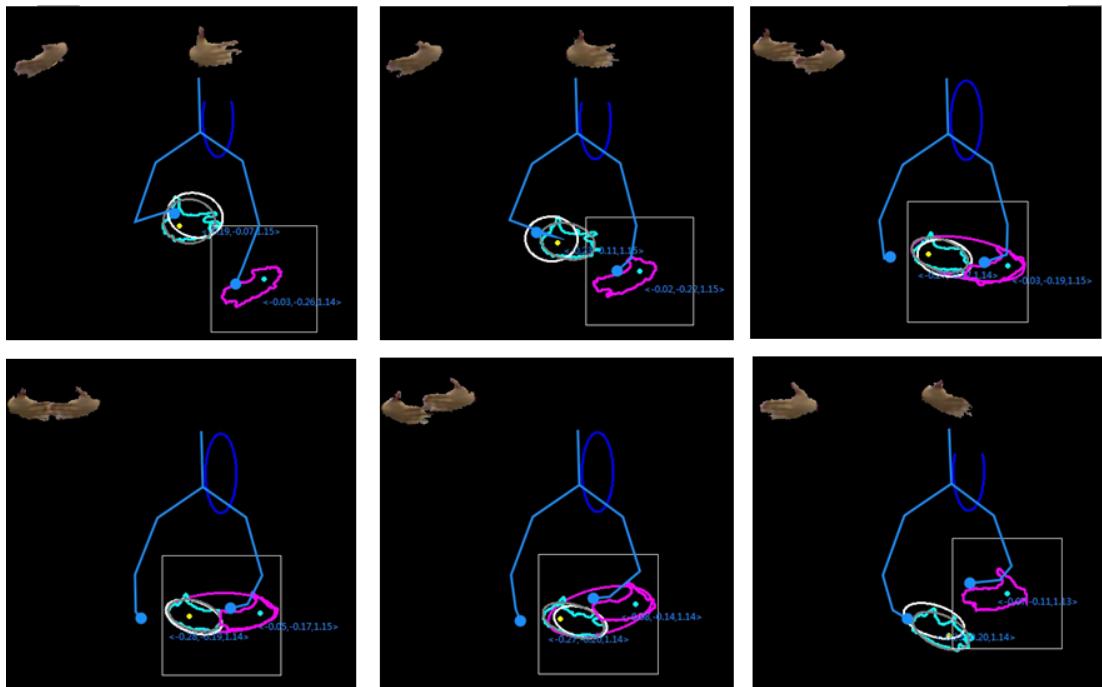


图 3.2 系统进行手部跟踪的结果

第二节 遮挡问题的解决

遮挡问题有几种类型，我们主要处理两种遮挡情况：两手互遮挡相碰；以及手部遮挡了脸部，与脸部相接触。我们建立了椭圆物体假设以解决这两类问题。

一、椭圆物体假设

我们参考了 [2] 中的思想，建立了适应 Kinect 跟踪的椭圆物体假设以处理遮挡问题。

在时间 t , 我们利用第二章叙述的分割方法找到了 M 个区域。每个区域 r_j $1 \leq j \leq M$, 都对应一组连通的手部观测点。由于两手相接触的情况的存在, 这些区域 r 与物体并不是一一对应的。例如, 两只互相交叉的手是两个物体, 但它们都对应了同一个区域, 因为存在着两手间的互相遮挡。我们假设物体可以对应一个区域或是一个区域的一部分, 也就是说, 一个区域可以对应一个或者多个物体。

另外一个重要的假设就是, 物体的形状可以用椭圆来拟合。这个假设对于手和脸来说都是合适的。令 N 为时刻 t 画面中存在的物体个数, $O_i, 1 \leq i \leq N$ 是组成第 i 个物体的点集。我们用 $h_i = h_i(c_{x_i}, c_{y_i}, \alpha_i, \beta_i, \theta_i)$ 来表示物体拟合出的椭圆。其中, c_{x_i}, c_{y_i} 是椭圆的中心, α_i 为椭圆长轴, β_i 为椭圆短轴, θ_i 表示椭圆的方向。然后, 我们用 $R = \bigcup_{j=1}^M r_j$, $O = \bigcup_{i=1}^N o_i$, $H = \bigcup_{i=1}^N h_i$ 来分别表示区域的集合、物体的集合和椭圆的集合。我们定义距离 $D(p, h)$ 为点 $p(x, y)$ 到椭圆 $h_i = h_i(c_{x_i}, c_{y_i}, \alpha_i, \beta_i, \theta_i)$ 的距离:

$$D(p, h) = \sqrt{\vec{v} \cdot \vec{v}} \quad (3-3)$$

其中

$$\vec{v} = \begin{bmatrix} \frac{\cos(\theta)}{\alpha} & -\frac{\sin(\theta)}{\alpha} \\ \frac{\sin(\theta)}{\beta} & \frac{\cos(\theta)}{\beta} \end{bmatrix} \cdot \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix}$$

从 $D(p, h)$ 的定义上可以看出, 当一个点在椭圆外部时, $D(p, h) > 1$, 当一个点在椭圆内部时, $D(p, h) < 1$, 当一个点在椭圆之上时, $D(p, h) = 1$ 。

这样, 遮挡问题就转化为了决定物体假设 h_i 和观测区域 r_j 关系的问题。

这个问题有三个情况: 1. 物体假设的建立; 2. 物体假设的跟踪; 3. 物体假设的去除。下面分别来叙述这三种情况的解决办法。

物体假设的建立 当一个物体第一次出现在画面中时, 物体假设需要被建立。图3.3中的区域 r_3 就是这种情况。

物体假设椭圆的建立非常直观。我们利用第一节的跟踪方法检测到左右两手的区域后, 物体假设椭圆便被建立。建立的椭圆参数用最小二乘法确定。

物体假设的跟踪 当之前检测到的物体移动时，需要进行物体假设的跟踪。遮挡问题也是在这种情况下处理的。

跟踪的目标是将物体假设和观测区域对应起来。在此过程中，主要遵循了两条原则：

- 原则 1：如果一个点，属于某个区域，且其位于一个物体假设的椭圆内部，则这个点被认为是属于这个物体假设的。
- 原则 2：如果一个点，属于某个区域，但不在任何物体假设椭圆的内部则它被分配给离它最近的物体假设椭圆。距离由公式3-2决定。

也就是说，被分配给物体假设 h 的点集可以表示成 $o = R_1 \cup R_2$ ，其中 $R_1 = \{p \in B | D(p, h) < 1.0\}$, $R_2 = \{p \in B | D(p, h) = \min_{k \in H} \{D(p, k)\}\}$

当没有遮挡发生时，如图3.3中的区域 r_2 和物体假设椭圆 h_1 的情况，将物体假设与观测区域对应起来比较容易。我们利用第一节的跟踪方法检测到左右两手的区域后，左右两手的物体假设便自然与这两个区域对应了起来。

当有遮挡发生时，如图3.3中的区域 r_3 和物体假设椭圆 h_2 和 h_3 的情况。两个不同的物体假设椭圆 (h_2 和 h_3) 对同一个观测区域 r_3 进行“竞争”。根据上述的原则 1，所有在 h_2 以内的观测点被分配给 h_2 ，根据同样的原则，所有在 h_3 以内的观测点被分配给 h_3 。根据原则 2，在 r_2 内的那些没有在任何物体假设椭圆内的点，被分配给离它最近的物体假设椭圆。

当所有观测区域内的点都被分配给相应的物体假设椭圆之后，物体假设椭圆的参数被重新确定，方法仍是最小二乘法。

t 时刻物体假设的跟踪是基于前一帧 $t - 1$ 时刻根据观测结果更新以后的物体假设进行的。这之间的时间间隔需要被考虑。所以，在每帧 t 开始之前，需要根据上一帧 $t - 1$ 更新后的物体假设预测本帧的物体假设。预测方法与第一节所描述的跟踪方法类似，假定物体假设椭圆的中心进行的是与物体运动参数相同的匀加速运动。物体运动参数 $\vec{a}(t - 1), \vec{v}(t - 1)$ 确定后（公式3-2），预测的椭圆物体假设如下：

$$\widehat{h_i(t)} = h_i(\widehat{c_{x_i}(t)}, \widehat{c_{y_i}(t)}, \alpha_i(t - 1), \beta_i(t - 1), \theta_i(t - 1)) \quad (3-4)$$

预测的物体假设椭圆 $\widehat{h_i(t)}$ 用于本帧的物体假设椭圆与观测区域的对应工作。

物体假设的去除 当物体移出镜头或是被其他物体完全遮挡无法检测时，物体假设椭圆需要被去除。当我们利用上一节的跟踪方法无法检测到某只手后，物体假设被去除。

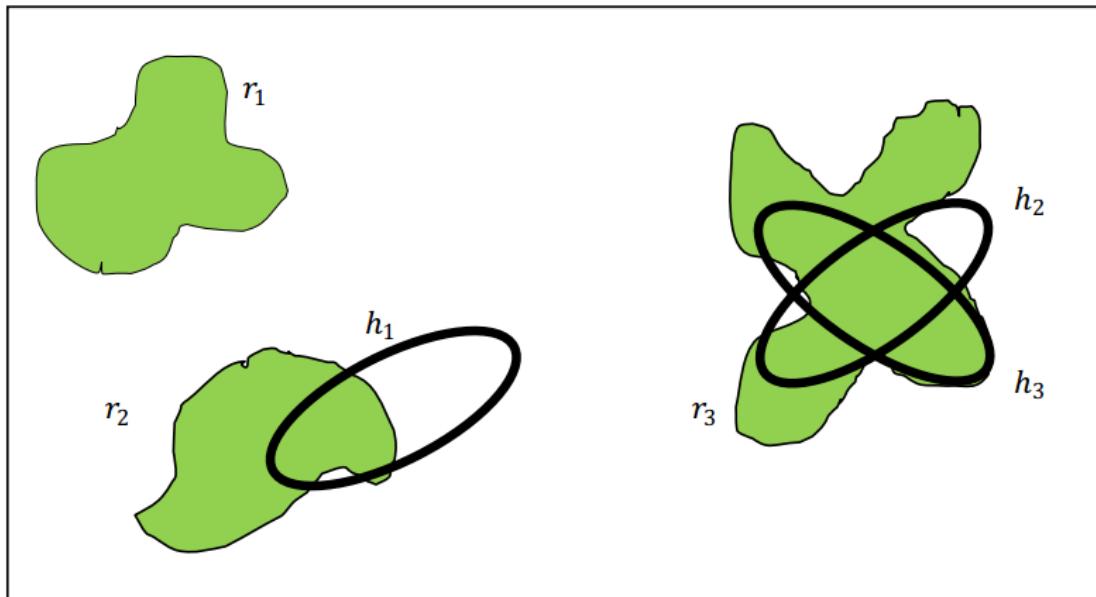


图 3.3 物体假设椭圆与观测区域之间的不同关系

二、双手互相遮挡的问题

1 遮挡情况的判定

利用物体假设椭圆的方法在处理两手遮挡问题时，首先要判定是否发生了两手之间的遮挡。若本帧被判定为遮挡发生，则遮挡处理步骤开始进行。所以，遮挡情况的判断是基础的一步。

判断遮挡情况主要遵循着以下两条原则：

- **原则 1：**如果被检测到的两手区域（ $r_{rightHand}$ 和 $r_{leftHand}$ ）互相重合，并且重合的像素点数目大于某特定阈值，则判断遮挡情况发生，并且

$$r_{bothHands} = \begin{cases} r_{rightHand}, & Area(r_{rightHand}) > Area(r_{leftHand}) \\ r_{leftHand}, & otherwise \end{cases}$$

其中 $r_{bothHands}$ 为系统判定的两手组成的共同区域，它将被用于与物体假设椭圆做对应（如第二节第一部分中所述）。

- **原则 2：**当两只手距离足够近（由之前帧的观测决定）时，如果两只手的物体假设椭圆的大部分都落到了同一个区域内（ $r_{rightHand}$ 或 $r_{leftHand}$ ），那么判定这个区域为 $r_{bothHands}$ 。

2 遮挡情况下手部的分割

当遮挡情况被判定发生后，利用第二节第一部分中的方法，我们将两个物体假设椭圆 $h_{rightHand}$ 和 $h_{leftHand}$ 与观测区域 $r_{bothHands}$ 对应起来，且两个物体假设椭圆的参数被重新确定。

但我们并不在记录手形特征的步骤中分别记录两只手各自的区域（即 $r_{bothHands}$ 中分别与 $h_{rightHand}$ 和 $h_{leftHand}$ 对应的区域），而是将 $r_{bothhands}$ 记录为遮挡情况下的手部图像。这样做的原因在于，虽然我们利用物体假设椭圆的方法将 $r_{bothhands}$ 分割成了左手区域、右手区域两个部分，但是这个分割仅是一个近似的分割，并不能很好的反应手指的精细边缘。由于两只手的颜色、深度差异在这种情况下都很小，要求在两手之间得到精细的分割也是不现实的。因此，若利用两只手的近似分割结果作为手部图像，手形分类结果将十分不理想。相反的，我们将 $r_{bothhands}$ 作为一个整体记录，不仅得到了精细的分割，而且还将手形分类的范围缩小到了遮挡手形之中，提高了手形分类的准确率。

图3.4给出了一个两手遮挡情况的例子。如前文所说，当两手遮挡情况发生时，用紫色的椭圆来拟合双手区域，并将双手区域作为一个整体记录为手形特征，如图中左上角的分割结果所示。图3.2给出了另一个例子。如图中的三、四、五三张图所示，当遮挡被判定发生时，紫色椭圆拟合了两手区域，而且双手手形作为一个整体被记录。

三、手与脸相碰的问题

手语中有很多手与脸互相遮挡的情况出现（图5.5(b)）。当这种情况出现时，可能会产生将脸部区域误认为成手部区域的一部分的错误（如图3.5(a) 所示）。

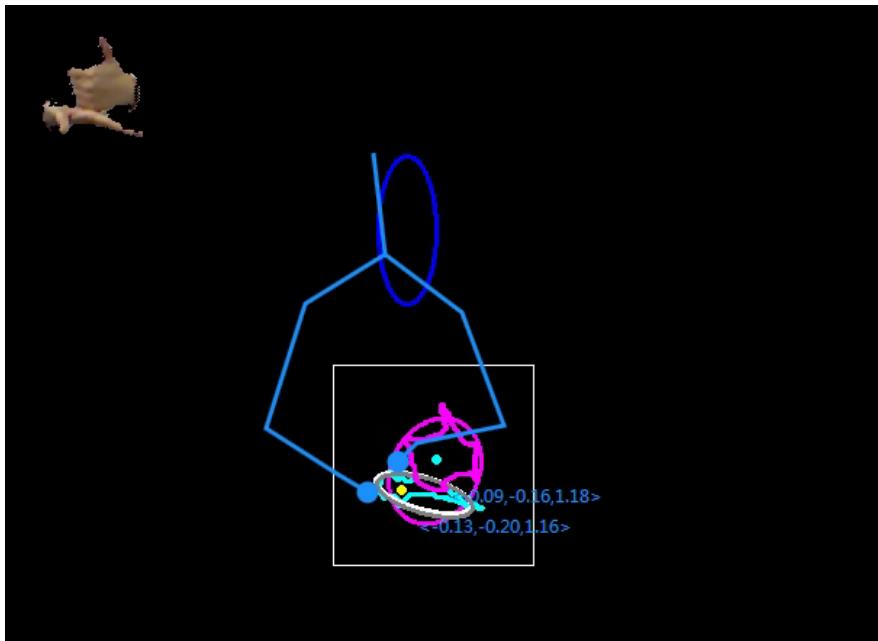


图 3.4 两手互相遮挡时的处理结果

为了解决这种问题，我们采用了如下方法。

首先，我们利用 Kinect SDK 提供的骨骼跟踪提供的脸部三维位置寻找脸部区域，并且用拟合的椭圆记录脸部信息： $h_{head} = h_{head}(c_{x_i}, c_{y_i}, \alpha_t, \beta_t, \theta_t)$ 。椭圆拟合的方法为最小二乘法。

1 遮挡情况的判定

判断手与脸遮挡的情况是否发生主要遵循着以下两条原则：

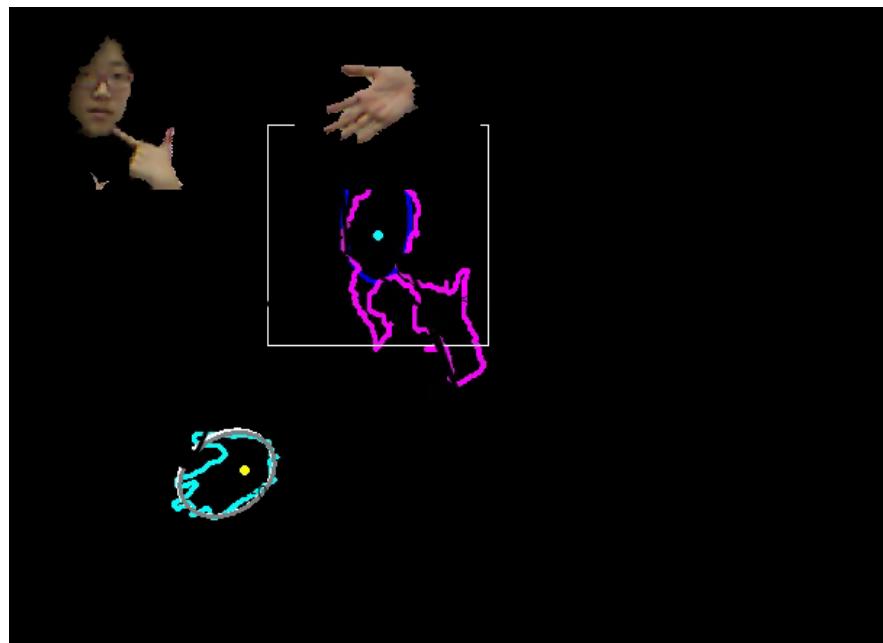
- **原则 1：**如果脸部对应的椭圆中心在被检测到的两手区域 ($r_{rightHand}$ 或 $r_{leftHand}$) 之内，则判断遮挡情况发生，且这个区域为 $r_{HeadOcclusion}$ 。
- **原则 2：**如果检测到的手心位置在脸部对应的椭圆内部，则判断遮挡情况发生，且这个区域为 $r_{HeadOcclusion}$ 。

2 遮挡情况下手部的分割

利用第二节第一部分中的方法，我们要做的是将手的物体假设椭圆 h_{hand} 和脸的物体假设椭圆 h_{head} 分别与检测到的遮挡区域 $r_{HeadOcclusion}$ 对应起来。

当所有观测区域 $r_{HeadOcclusion}$ 内的点都被分配给相应的物体假设椭圆之后，物体假设椭圆的参数被重新确定。

图3.5给出了一个手部与脸部互相遮挡情况的处理结果示例。未进行遮挡处理时，会产生如图3.5(a)的效果，而进行遮挡处理以后，效果如图3.5(b)。



(a) 手与脸部相接触时易产生的错误



(b) 手与脸部相接触时的遮挡处理结果

图 3.5 手与脸部相接触时遮挡处理的前后对比

第四章 特征提取

当手部被跟踪后，需要提取手部特征作为手语识别以及手形识别的输入。在我们的系统中，主要有两种特征被提取。

第一节 运动特征

手的三维位置 $\vec{C}(t) = (c_x(t), c_y(t), z(c_x(t), c_y(t)))$ ，三维速度 $\vec{v}(t) = (v_x(t), v_y(t), v_z(t))$ ，以及加速度 $\vec{a}(t) = (a_x(t), a_y(t), a_z(t))$ 被记录。

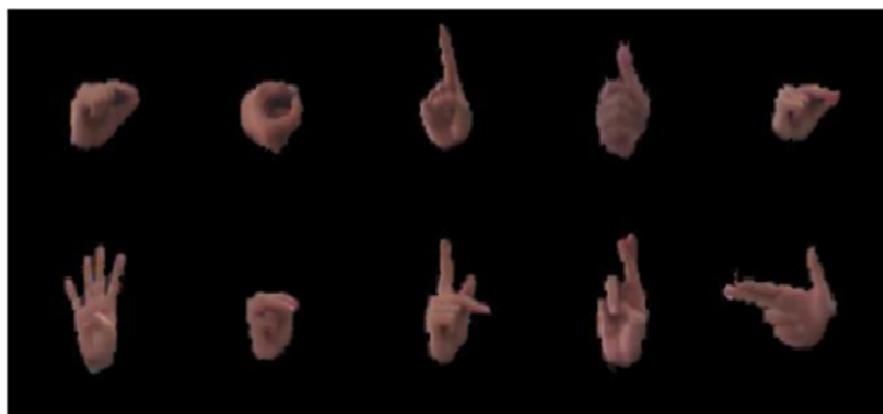
第二节 形状特征

手形特征是主要的手部特征。

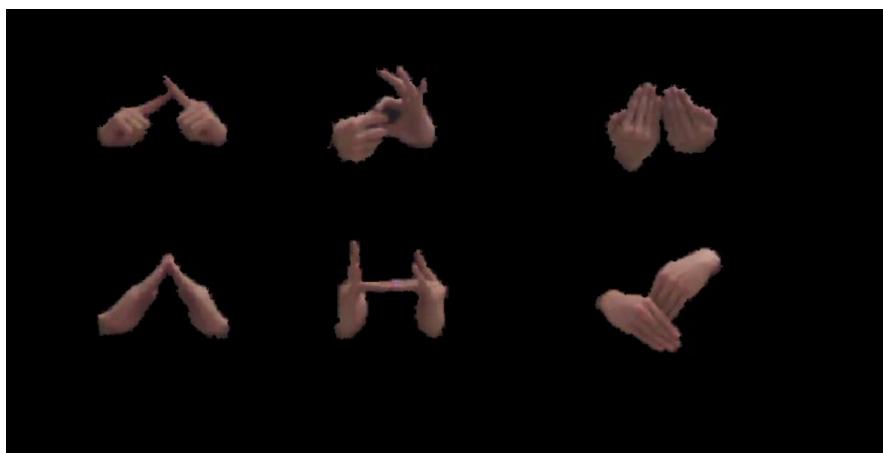
首先，通过手心深度信息 $z(c_x(t), c_y(t))$ ，我们确定提取手部图像的矩形区域大小。这一步骤使得在我们获取的手部图像中手部区域占据的大小不受手距摄像头的远近影响。这使得下一步的识别步骤更加准确。

接下来，我们将手部区域标准化成 128×128 的图像，然后将背景设置为黑色，手部取 RGB 值作为手形特征。图4.1(a)为部分（单手）手形图像。

像第三章第二节所提到的一样，在两手互相遮挡触碰的问题发生时，我们双手作为一个区域记录下来。其他步骤与上述步骤相同。图4.1(b)为部分双手手形图像。



(a) 单手手形图像



(b) 双手手形图像

图 4.1 记录手形图像为手形特征

第五章 实验结果分析

本文使用了 C# 进行编程实践，在处理器为 3.5GHz i7-3770K 的 CPU，内存为 32GB 的计算机上，我们进行了每秒 10 帧的实验，达到了近实时的效果。我们将我们所用方法与其他文章中所用方法进行了对比。

第一节 手部跟踪与特征提取结果的分析及与其他方法的对比

一、 分割结果的分析及与其他分割方法的对比

我们的方法结合了深度信息与彩色信息的优点，并充分利用了 Kinect 骨骼跟踪的结果，与其他使用单一信息进行手部分割的方法相比，优势十分明显。

[11] 中利用 Kinect 提供的深度信息使用最大类间方差法分割手部。文章假设摄像头前仅有一名实验者，且占据了摄像机摄像范围内的主要空间。更重要的是，手部区域距离摄像头相比身体其他区域距离摄像头明显更近。分割结果如图5.1所示。



图 5.1 文章 [11] 分割结果。从左到右依次为：原始深度图像；人体分割图像；初始手部区域；修正后的手部区域

这种方法只有在手明显在身体其他部分之前时才有好的识别结果，而这个条件在许多手语动作中是不成立的。



图 5.2 文章 [1] 分割结果。左图为原始彩色图像。右图为检测到的肤色区域

[1] 中通过建立肤色模型的方法提取手部轮廓。分割结果如图5.2所示。

这种方法明显的局限性在于它对背景颜色要求较高，需要背景中没有颜色与肤色相近的物体，[1] 则直接采用了单一颜色的背景作为实验背景，这在实际手语识别操作中显然是极不方便的。

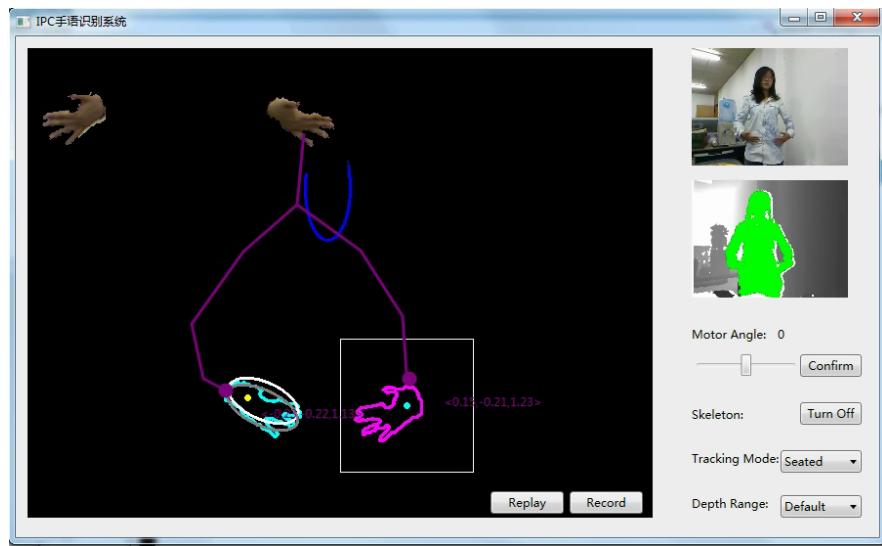
在我们的分割系统中，上述问题被有效的避免。

当手部与身体其他部分有相近的深度数值时，颜色信息的优势充分体现出来，手部分割结果不受影响。图5.3(a)中手部紧贴实验者的衣服，但系统依旧可以将手部正确分割。而当背景颜色复杂且有颜色与肤色相近的物体作为干扰时，深度信息的利用保证了手的正确分割，如图5.3(b)，有人脸作为背景时，手部区域被正确的分割（彩色图像可由右上方小窗口看出，人脸在手的正后方）。

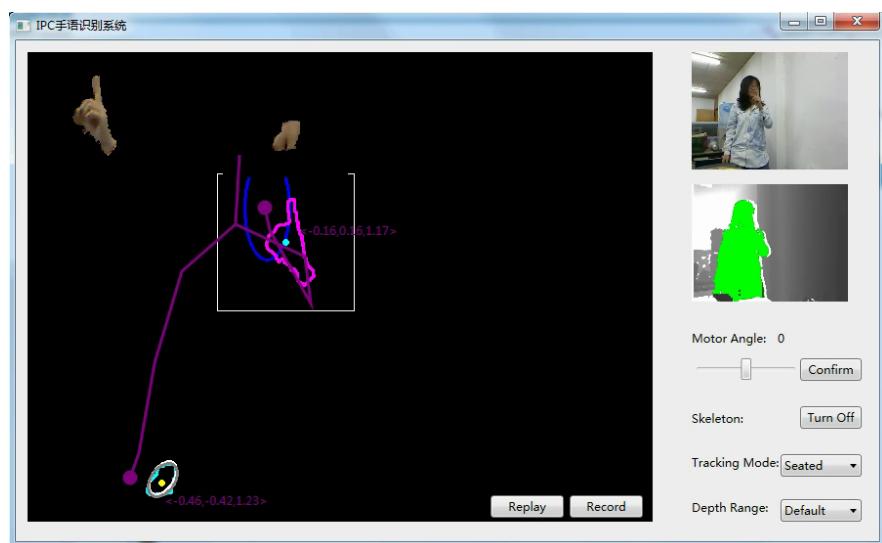
二、 遮挡问题的处理结果以及与其他遮挡处理方法的对比

我们采用建立物体假设椭圆的方式，进行遮挡处理，并在双手遮挡情况发生时保存双手手形作为手形特征。这种处理方法既获得了相对准确的双手运动信息，又保证了在遮挡发生时双手手形被准确记录。与其他遮挡处理方法对比，有明显的优势。

文章 [19] 利用形态学方法进行遮挡问题的处理。处理方法如图5.4所示。



(a) 手部与身体其他部分有相近的深度数值时



(b) 背景中有颜色与肤色相近的物体作为干扰时

图 5.3 系统结合深度、彩色信息进行分割的优势

三个区域（脸，左手，右手）被检测。如图5.4(b)、5.4(c)中所示，首先，对原始图像进行肤色检测，得到初始的肤色掩膜 S_0 ，并利用形态学方法对 S_0 进行改善，填充空洞并平滑边缘，得到 S_2 。由于 S_2 中包含小于 3 个的连通区域，判断遮挡情况发生。文中假设当遮挡情况发生时，遮挡的不同区域由一个



图 5.4 文章 [19] 中遮挡处理方法

细“桥”连接，如图5.4(c)所示，实验者的右手和脸组成了一个遮挡区域，他们之间有一个较细的“桥”。所以，文章利用腐蚀算法，得到了图5.4(d)所示的结果，即得到了3个连通区域。然后再将图S2中剩余的点根据距离分配到三个连通区域中。最终得到的三个区域即为分割的结果。

这种方法的缺陷在于手语中很多双手遮挡手形并不满足“桥”的假设。例如图5.5中所示，图5.5(a)中遮挡区域中不存在细“桥”，此时文中方法不能成功分割出两个区域；图5.5(b)中右手与脸部遮挡的情况，虽然“桥”存在，可以分割出右手于脸部的区域，但由于右手食指在脸部范围内，使得分割出的右手手形不包括食指，极度失真，给手形分割过程造成了很大困难。

文章 [2] 利用了物体假设椭圆的方法进行遮挡处理。当遮挡发生时，遮挡区域被分割开。结果如图5.6所示。这种方法的缺陷在于无法做到两手之间得到精细的分割，给识别过程造成困难。

在我们的处理方法中，上述问题得到了解决。

如图5.7所示，两手遮挡存在时，即使没有“桥”的存在，两手也被有效地近似分割。图5.8展示了当实验者做“刷牙”动作，发生了手部与脸部发生遮挡时的成功分割。图5.9则提供了一个两手遮挡发生时记录的准确手形，左上方经过标准化的区域中记录了双手手形。



(a) 中国手语中的“抽屉”



(b) 中国手语中的“刷牙”

图 5.5 文章 [19] 无法处理的情况

第二节 后期步骤——手形分类简述

手部形状特征被提取后，我们进行了手形分类。在这里，本节简单介绍了我们系统中手形分类的过程。

我们利用深度神经网络进行手形分类，利用了目前流行的深度神经网络结构：卷积神经网络（CNN）和深度置信神经网络（DBN）。

与过去依靠人工提取特征的方法不同，深度神经网络模拟人的视觉系统，能够直接处理图像像素信息，网络自动提取层次特征，从低层到高层的特征表示（representation）抽象度逐步提高，不同类型的数据在特征空间的距离也越明显，有利于进一步降噪和分类。实验证明 DBN 和 CNN 在手形分类上均取得了较高的分类效果。

实验数据采集自 9 个人录制的共 549 段视频转成的图片，包含 61 种基本手形。每只手共有约 10 万张图片。原始图片如图 4.1。手形分类过程选取的图



图 5.6 文章 [2] 遮挡处理结果

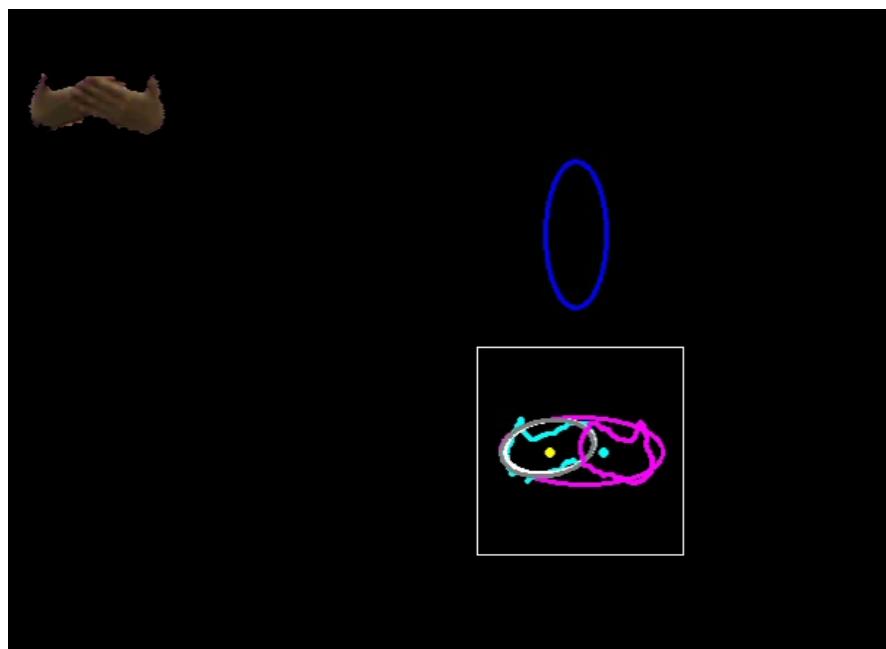


图 5.7 没有“桥”存在时的两手遮挡处理结果



图 5.8 手与脸互相遮挡时的处理结果，手语者做“刷牙”动作

像格式为 32×32 的灰度图。考虑到即使是同一个人不同时间做同样的手形也会存在角度的差异，为增加分类效果的鲁棒性，研究者对数据进行了扩充，将录制的数据进行了 5 次正负 60 度之间随机旋转，最终得到约 50 万张图片。其中我们随机抽取了 60% 作为训练集，剩余的 40% 作为测试集。

利用 CNN 网络模型进行手形分类，最终的识别率高达 98.9%，而利用 DBN 进行分类，识别率亦达到 95.5%。

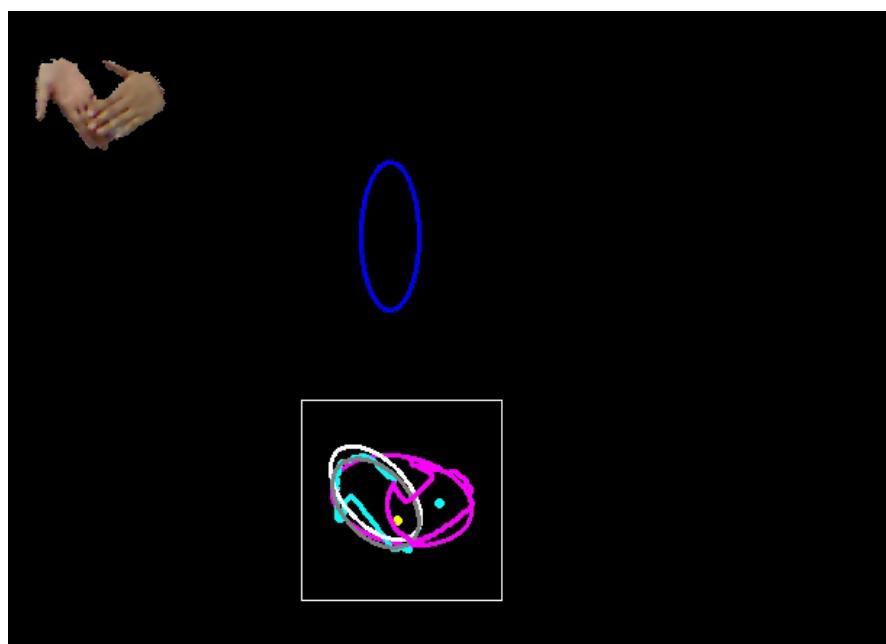


图 5.9 两手发生遮挡时记录的准确手形

第六章 总结和展望

手语识别是一个困难重重的工程问题。我们试图利用新的技术产品 Kinect 的优势，搭建一个手语识别系统。我所做的是其中的前端工作：手部的分割和跟踪。

文章结合了 Kinect 提供的深度信息、彩色信息与骨骼信息，建立深度模型与肤色模型进行手部的分割，建立简单有效地跟踪模型进行手部的跟踪，利用物体假设椭圆进行遮挡问题的处理，最终达到了很好的分割、跟踪、遮挡处理的效果，为下一步的手形分类以及手语识别提供了可靠、鲁棒的特征。

文中所述系统的优点在于：

1. 不对实验背景做任何颜色上的要求；
2. 不要求实验者站在镜头最前端，亦不要求手部区域在身体的前端；
3. 在手语过程中，可以准确的跟踪手部；
4. 可以良好的处理两手互相接触遮挡、手与脸互相遮挡的情况，当以上情况发生时，一方面可以获得手部的准确运动信息，另一方面可以获得手部的准确形状信息。

但是，我的工作还有很多需要改进的地方，我也将在以后继续这些工作：

1. 建立更鲁棒的肤色模型；
2. 加入并行运算，提高系统的运算速度，使得系统达到真正的实时处理；
3. 寻找更有效地边缘优化方法，得到更准确的手形分割。

通过两个学期的学习与实践，我在科研技巧、编程技巧上有了很大的提高，在理论知识的掌握上有了长足的进步。希望手语组可以继续努力，最终将我们的手语识别系统搭建成一个真正可以投入实际应用的产品。

参考文献

- [1] Stefano Squartini Francesco Piazza Marco Fagiani, Emanuele Principi. A new system for automatic recognition of italian sign language. *Smart Innovation, Systems and Technologies*, 19:60–79, 2013.
- [2] Antonis A. Argyros and Manolis I.A. Lourakis. Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *ECCV*, 2004.
- [3] Michael Van den Bergh and Luc Van Gool. Combining rgb and tof cameras for real-time 3d hand gesture interaction. In *WACV*, 2011.
- [4] Benjamin Dums Georg Umlauf Manuel Caputo, Klaus Denker. 3d hand gesture recognition based on sensor fusion of commodity hardware. In *CDDU*, 2012.
- [5] M.Krishnaveni V.Radha. Threshold based segmentation using median filter for sign language recognition system. In *NaBIC*, 2009.
- [6] A. Argyroszy X. Zabulis, H. Baltzakis. Vision-based hand gesture recognition for human-computer interaction. In *ITI*, 2010.
- [7] Ray Jarvis Zhi Li. Real time hand gesture recognition using a range camera. In *ACRA*, 2009.
- [8] Hervé Lahamy and Derek Litchi. Real-time gesture recognition using range cameras. In *CGC*, 2010.

- [9] Michael Van den Bergh Luc Van Gool Dominique Uebersax, Juergen Gall. Real-time sign language letter and word recognition from depth data. In *ICCV*, 2011.
- [10] Luc Van Gool Michael Van den Bergh. Combining rgb and tof cameras for real-time 3d hand gesture interaction. In *WACV*, 2011.
- [11] Z. Liu A. Kurakin, Z. Zhang. Real time system for dynamic hand gesture recognition with a depth sensor. In *EUSIPCO*, 2012.
- [12] Zhengyou Zhang Zhou Ren, Junsong Yuan. Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In *MultiMedia*, 2011.
- [13] Microsoft Corp. Kinect sdk. <http://www.microsoft.com/en-us/kinectforwindows/>.
- [14] J. Kim S. Kim S. Park, S. Yu and S. Lee. 3d hand tracking using kalman filter in depth space. *EURASIP Journal on Advances in Signal Processing*, 2012:36, 2012.
- [15] Robert Niese Mahmoud Elmezain, Ayoub Al-Hamadi and Bernd Michaelis. A robust method for hand tracking using mean-shift algorithm and kalman filter in stereo color image sequences. *International Journal of Information and Communication Engineering*, 6:3, 2010.
- [16] Tieniu Tan Frédéric Ojardias Caifeng Shan, Yucheng Wei. Real time hand tracking by combinin g particle filtering and mean shift. In *AFGR*, 2004.
- [17] OpenNI. Open kinect sdk. <http://www.openni.org/>.
- [18] Rodrigo Verschae Hardy Francke, Javier Ruiz-del-Solar. Real-time hand gesture detection and recognition using boosted classifiers and active learning. *Advances in Image and Video Technology*, 4872:533–547, 2007.

- [19] Vassilis Pitsikalis Anastasios Roussos, Stavros Theodorakis and Petros Maragos. Hand tracking and affine shape-appearance handshape sub-units in continuous sign language recognition. In *ECCV*, 2010.