

基于立体视觉系统的手势分割算法研究

房 稳, 孙新柱, 范县成, 刘梦悦

(安徽工程大学电气工程学院, 安徽 芜湖 241000)

摘 要:手势分割是保障人机交互过程得以准确高效实现的重要步骤之一,但在分割的过程中,常常会受到类肤色背景等因素的外在干扰,使得分割操作出现欠分割的情况。针对此问题,采用 Opti-Track™DPC640/30 立体视觉系统,引入深度图像来指导分割,并针对原有的三维 OTSU 算法在进行分割时存有大量冗杂的运算,计算量过大的问题,简化其主函数的运算,即对其主变量 w_0 和 w_1 进行递推处理,使得每次计算离散度矩阵的迹时无需从起始位置开始,只需从上一次递推初始处开始即可,有效地减少了三重循环的次数。实验结果表明,该算法比基于 YCrCb 颜色空间分割和 Grabcut 分割法在处理有类肤色背景干扰时更为有效,且与原有的 OTSU 算法相比,算法执行效率得以提高。

关键词:手势;彩色图像分割;深度图像分割;三维 OTSU 算法

中图分类号:TP391.41

文献标志码:A

引 言

日常生活中,手势往往扮演着重要的角色,它不仅表达丰富的信息量,而且还具有自然、高效、直观等优点,是实现人机交互至关重要的途径之一。近年来,手势已成为人机交互领域的研究热点^[1]。随着科技的不断进步和发展,手势的输入和输出设备也在快速地更新换代,从开始的数据手套、肌电信号传感器、惯性传感器,一直发展到如今的计算机视觉,越来越便捷地为人们提供了研究手势的新思路和新途径。

为了在手势研究中兼顾效率和效果,对于手势目标区域的提取和分割工作就变得尤为重要,精准的分割将会为后续研究工作打下坚实的基础,可以避免大量不必要的运算和处理,极大地提高研究效率。但手势分割也存在难点,一是我们日常所处环境的复杂多样性,手势分割会受到光照、颜色、遮挡等无关信息的干扰;二是手势姿态的灵活性,这会造成手的外部形状特征多变。这些问题都会给分割工作带来困难^[2]。

尽管国内外的众多研究人员对相关内容进行了大量研究,但仍然存在尚需继续完善和解决的问题。例如,曹建秋等人^[3]详细对比了映射到 YCrCb 和 HSV 颜色空间的优缺点,但未能提出具体的分割改进措施;Boykov 等人^[4]首次将图割理论(Graphcut)应用于分割实践领域,该算法基于马尔科夫随机场(MRF)最小化能量模型,综合各类理论知识对全局进行最优求解,展现出独特的优越性,可以说是开创了图像分割的一种新思路,但因其使用一次性的单一交互式分割,所得到的最终分割准确性不太理想;徐秋平等^[5]使用了分水岭和图割联合算法,分割效果比较稳定,但是导致边界过于粗糙;Rother 等人^[6]在 Graphcut 算法的基础上进行了改进和扩展,提出了 Grabcut 算法,使用高斯混合模型(GMM)取代灰度直方图,同时使用迭代算法取代一次最小化估计来完成最小化能量的计算,最后通过非完全编号极大地降低了交互的复杂度。这些改进使得 Grabcut 较前者能更好地提高分割精度和效率,是图割理论应用于分割实践领域的成熟代表方法之一,但 Grabcut

收稿日期:2016-04-27

基金项目:安徽高校省级自然科学研究重大项目(KJ2014ZD04)

作者简介:房 稳(1992-),男,安徽芜湖人,硕士生,主要从事手势识别方面的研究,(E-mail)1508427763@qq.com

仍存有不足,即对颜色背景变化较为敏感。虽然有研究者不断地对其进行研究改进,比如伊聪聪^[7]等人提出了一种基于前景形态的联合分水岭(shed)和Grabcut多尺度图割算法,可以大幅度提高迭代效率,周良芬等人^[8]引入了熵惩罚因子,使得能量分割函数得以大幅优化,有效地抑制了目标信息的损失,但这些改进仍未能解决其核心问题,即对于手势分割而言,无法消除有类肤色背景干扰时的误分割,在处理此类问题时略显乏力,容易出现分割不到位,即欠分割的情况。因此,有必要结合场景深度信息来指导图像分割,以便更好达到人们的分割预期。赵志平等人^[9]尝试通过计算混合概率密度求出最优阈值,使用其在深度图像中进行分割,但只成功分离出了人体和背景;王艳等人^[10]采用双阈值法进行深度图像分割,但分割之后手掌边缘锯齿状明显,不便于后续应用;徐鹏飞等人^[11]对类间方差法进行了研究,并对其应用于图像分割领域中的可行性进行了探索。

本文针对立体视觉系统的特性,分别对其获取的色彩和深度图像进行手势分割研究,并综合加以对比,在原有三维OTSU阈值手势分割算法的基础上加以改进。改进后的算法不仅可以克服类肤色的干扰,而且该算法引入了递推思想,使其主变量 w_0 和 w_1 的每次求解运算,不必从起始位置开始,只需从上一次的终止位置开始继续进行迭代即可,大幅度地减少了计算量,有效地提高了算法的运算效率。

1 深度图像分割

无论是基于YCrCb颜色空间的分割算法,还是在Grabcut基础上加以改进的Grabcut算法,都是围绕彩图分割所展开的。对于彩图,从其获取原理上可知,是显示在二维平面上的投影,注定会丢失深度信息,而人们对周围的感知是建立在三维世界的基础上,所以联系实际,在研究图像分割时,引入图像的深度信息有着极为重要的意义和价值。

1.1 深度图像获取原理

DPC640/30系列是通过景深来获取三维图像信息的,基本原理为:通过红外相机接收由红外发射器所发出的红外光照射到物体表面上的散斑点数据,从而在底板上记录相应的成像点,获取深度图像。原理示意图如图1所示。

由三角相似性可知:

$$\frac{D}{b} = \frac{Z_0 - Z_k}{Z_0}, \frac{d}{f} = \frac{D}{Z_k}$$

联立求解可得:

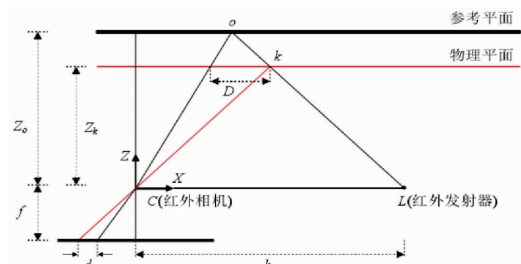


图1 三角测距原理

$$Z_k = \frac{Z_0}{1 + \frac{Z_0}{fb}d}$$

其中: Z_k 即为深度相机与目标物体间的距离; Z_0 表示参考平面上障碍物与相机间的距离; f 表示相机焦距; b 表示红外发射器与相机间的距离; d 表示照射散斑与参考平面上散斑的成像距离。

1.2 深度图像预处理

OptiTrack™ DPC640/30立体视觉系统,具有采集方便,便于应用等特点,但是在采集的过程中,由于设备本身特性、实验条件以及物体特性等诸多方面的原因,图像中会含有很多黑洞和大量的噪声,这对后期的图像分割极为不利,所以首先需要对深度图像进行修复操作。本文采用的是一种基于形态学与中值滤波的算法对图像进行修复。主要步骤为:首先通过形态学滤波中的开闭运算,选取合适的圆盘形(disk)结构元素,对深度图像进行第一次处理,可以修复绝大部分黑洞;其次再使用中值滤波,对图像中仍存有的孤立黑洞和边缘信息模糊等问题进行再一次地修复,最终达到完全填充黑洞、平滑图像边缘以及去除噪声的目的。

在形态学中,膨胀和腐蚀是其最基本的运算,如图2所示。除此之外,还有建立在两者基础上的二次运算。根据运算的先后顺序不同,将其分为开运算和闭运算^[12]。两者具有诸如对偶性、扩展性、单调性、平移不变性和等幂性等极为重要的代数性质。设图像 S 对图像 X 做运算,先进行腐蚀再进行膨胀的称为开运算,记为 $X \circ S$,定义为 $X \circ S = (X \ominus S) \oplus S$;先进行膨胀再进行腐蚀的称为闭运算,记为 $X \bullet S$,定义为 $X \bullet S = (X \oplus S) \ominus S$,具体过程如图2~图4所示。



图2 膨胀和腐蚀

开闭运算都是单边算子,若单独对图像进行处理,

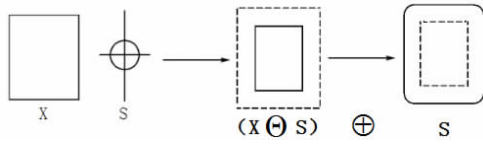


图 3 开运算

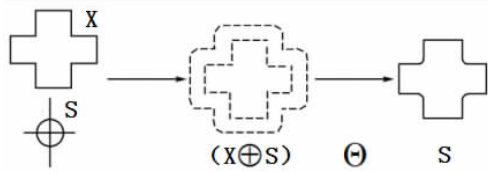


图 4 闭运算

则会由于各自独有的性质,从而影响图像的输出结果。开运算对于信号上方的噪声有很好的效果,但是对下方噪声尖峰则无能为力,反之,闭运算也与之类似,对上方尖峰处理效果不佳。一般情况下,常将两者联合起来共同构成形态学滤波器,使用迭代运算能够有效地减弱这些影响,常见的有 $(X \circ S) \cdot S$ 或 $(X \cdot S) \circ S$ 等。本文是将两者算法进行均分后使用。具体可表示为:

$$F = \frac{(X \circ S) \cdot S + (X \cdot S) \circ S}{2}$$

中值滤波的算法原理为:先将选定好大小的模板放入图像中,使其在图中遍寻,当发现目标像素点时,使其中心与像素点重合,然后开始计算各对应像素点的灰度值并且从小到大进行排列,最后取其中间的灰度值代替原有像素点邻域的灰度值。

中值滤波的这一特性,使其较形态学滤波能够更好地保护图像细节和边缘,更适宜修复图中孤立的噪声。而且在与上述形态学滤波算法进行联合时,定义中值滤波仅对于待修复区域采用,这样可以极大地提高整体滤波的修复速度,同时也能够最大限度地保持原有深度图像像素点的深度值。

1.3 改进的三维 OTSU 阈值法

深度图像的分割算法有很多种,比如有基于神经网络、小波变换、模糊理论、遗传理论、粒度计算论、免疫理论等的分割算法。其中阈值法最为经典,常见的阈值法有:人工确定法、自适应阈值法和分水岭算法。自适应阈值法可细分为:迭代阈值法、OTSU 法和最小误差法。由于 OTSU 法具有简单、高效等优点,本文结合获取景深数据的深度相机,选取 OTSU 法对深度图像进行分割操作。首先假设手的深度数据最小,这一假设符合人们的客观认知,因为在实际中,手通常处于背景或人体的前方,那么只要选取合适的阈值,就足以把前景目标(即

手)与后置背景分开,从而绕过彩图分割中所受到的类肤色干扰,达到精确分割手势的目的。

OTSU 阈值法又称为最大类间方差法或大津法,是由日本学者大津展之所提出的一种自适应阈值选取法,其基本原理是建立在最小二乘的基础上,计算前景和背景的方差,当前后方差达到最大时,即以这个值作为阈值进行分割。由于深度图像引入了景深数据,所以需要将 OTSU 引入三维空间中使用。

三维 OTSU 基本理论为:定义图像景深值 $f(x, y)$, 则领域均值 $g(x, y)$ 为

$$g(x, y) = \frac{1}{k^2} \sum_{i=-k/2}^{k/2} \sum_{j=-k/2}^{k/2} f(x+i, y+j)$$

领域加权中值 $h(x, y)$ 为

$$h(x, y) = \text{Med}\{f(x+i, y+j) \mid -k/2 \leq i \leq k/2, -k/2 \leq j \leq k/2\}$$

故某点的三维向量为 $(f(x, y), g(x, y), h(x, y))$ 。背景 $C_0 = \{(i, j, k) \mid i \leq s, j \leq t, k \leq q\}$, 其类内概率和类内均值表示如下:

$$\begin{aligned} w_0 &= P_r(C_0) = \sum_{i=0}^s \sum_{j=0}^t \sum_{k=0}^q P_{ijk} = w_0(s, t, k) \quad (1) \\ u_0 &= (u_{0i}, u_{0j}, u_{0k})^T = \\ &= \left(\sum_{i=1}^s iP_r(i/C_0), \sum_{j=1}^t jP_r(j/C_0), \sum_{k=1}^q kP_r(k/C_0) \right)^T = \\ &= \left(\sum_{i=1}^s \sum_{j=1}^t \sum_{k=1}^q iP_{ijk}/w_0, \sum_{i=1}^s \sum_{j=1}^t \sum_{k=1}^q jP_{ijk}/w_0, \right. \\ &\quad \left. \sum_{i=1}^s \sum_{j=1}^t \sum_{k=1}^q kP_{ijk}/w_0 \right)^T \quad (2) \end{aligned}$$

前景 $C_1 = \{(i, j, k) \mid i > s, j > t, k > q\}$, 相应地,其类内概率和类内均值表示如下:

$$\begin{aligned} w_1 &= P_r(C_1) = \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} \sum_{k=q+1}^{L-1} P_{ijk} = w_1(s, t, q) \quad (3) \\ u_1 &= (u_{1i}, u_{1j}, u_{1k})^T = \\ &= \left(\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} iP_{ijk}, \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} jP_{ijk}, \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} kP_{ijk} \right)^T \quad (4) \end{aligned}$$

为了判定阈值的合适度,引入总均值指标:

$$\begin{aligned} u_1 &= (u_{1i}, u_{1j}, u_{1k})^T = \\ &= \left(\sum_{i=s+1}^L iP_r(i/C_1), \sum_{j=t+1}^L jP_r(j/C_1), \sum_{k=q+1}^L kP_r(k/C_1) \right)^T = \\ &= \left(\sum_{i=s+1}^L \sum_{j=t+1}^L iP_{ijk}/w_1, \sum_{i=s+1}^L \sum_{j=t+1}^L jP_{ijk}/w_1, \sum_{i=s+1}^L \sum_{j=t+1}^L kP_{ijk}/w_1 \right)^T \quad (5) \end{aligned}$$

所以,最优阈值 (s, t) 满足下列等式:

$$t_r S_B(s', t', k') = \max_{0 \leq s, t, k \leq L-1} \{t_r S_B(s, t, k)\} \quad (6)$$

其中,离散度矩阵为:

$$S_B = \sum_{k=0}^1 P_r(C_k) [(u_k - u_r)(u_k - u_r)^T] \quad (7)$$

由式(6)与式(7)可知,其类间方差为:

$$t_r S_B = w_0 [(u_{0k} - u_{T_i})^2 + (u_{0j} - u_{T_j})^2 + (u_{0k} - u_{T_k})^2] + w_1 [(u_{1i} - u_{T_i})^2 + (u_{1j} - u_{T_j})^2 + (u_{1k} - u_{T_k})^2] \quad (8)$$

在进行计算时,捕获并尝试所有 (s, t, k) 取值,使得离散度矩阵 S_B 达到最大的值即为所求的最优阈值。

虽然三维OTSU算法可以相当精准地求得最优阈值,但是其计算量过大,任意阈值需从 $(0, 0)$ 一直累加到 (s, t) ,存在大量不必要的运算。受二维OTSU算法改进方法的启发^[13],对式(1)~式(5)进行综合考量后,在此提出对主变量 w_0 和 w_1 进行递推处理的三维改进算法,以减少其计算量。

$$\begin{aligned} w_0(s, t, q) &= \sum_{i=0}^s \sum_{j=0}^t \sum_{k=0}^q P_{ijk} = \\ & \sum_{i=0}^{s-1} \sum_{j=0}^t \sum_{k=0}^q P_{ijk} + \sum_{j=0}^t \sum_{k=0}^q P_{sjk} = \\ w_0(s-1, t, q) &+ \sum_{j=0}^{t-1} \sum_{k=0}^q P_{sjk} + \sum_{k=0}^q P_{stk} = \\ w_0(s-1, t, q) &+ w_0(s, t-1, q) - \\ w_0(s-1, t-1, q) &+ P_{stq} + \sum_{i=0}^{k-1} P_{stik} = \\ w_0(s-1, t, q) &+ w_0(s, t-1, q) - \\ w_0(s-1, t-1, q) &+ P_{stq} + \\ w_0(s, t, q-1) &- w_0(s, t-1, q-1) - \\ w_0(s-1, t, q-1) &+ w_0(s-1, t-1, q-1) = \\ w_0(s-1, t, q) &+ w_0(s, t-1, q) + \\ w_0(s, t, q-1) &- w_0(s, t-1, q-1) - \\ w_0(s-1, t, q-1) &- w_0(s-1, t-1, q) + \\ w_0(s-1, t-1, q-1) &+ P_{stq} \\ w_1(s, t, q) &= \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} \sum_{k=q+1}^{L-1} P_{ijk} = \\ \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} \sum_{k=q+1}^{L-1} P_{ijk} &+ \sum_{j=t+1}^{L-1} \sum_{k=q+1}^{L-1} P_{(s+1)jk} = \\ w_1(s+1, t, q) &+ \\ \sum_{j=t+1}^{L-1} \sum_{k=q+1}^{L-1} P_{(s+1)jk} &+ \sum_{k=q+1}^{L-1} P_{(s+1)(t+1)k} = \\ w_1(s+1, t, q) &+ w_1(s, t+1, q) - \\ w_1(s+1, t+1, q) &+ P_{(s+1)(t+1)(k+1)} + \sum_{i=s+1}^{L-1} P_{(s+1)(t+1)k} = \\ w_1(s+1, t, q) &+ w_1(s, t+1, q) - \\ w_1(s+1, t+1, q) &+ P_{(s+1)(t+1)(q+1)} + \\ w_1(s, t, q+1) &- w_1(s, t+1, q+1) - \\ w_1(s+1, t, q+1) &+ w_1(s+1, t+1, q+1) = \\ w_1(s+1, t, q) &+ w_1(s, t+1, q) + \\ w_1(s, t, q+1) &- w_1(s, t+1, q+1) - \end{aligned}$$

$$w_1(s+1, t, q+1) - w_1(s+1, t+1, q) + w_1(s+1, t+1, q+1) + P_{(s+1)(t+1)(q+1)}$$

这样,每次计算离散度矩阵的迹 t_r 时就无需从起始位置开始,只需从上一次递推初始处开始即可,减少了三重循环的次数,使得算法效率得以提高。

2 实验及分析

本文采用OptiTrack™DPC640/30立体视觉系统获取图像,该系统包含有3个镜头,分别为红外投射镜头、RGB彩色镜头和红外CMOS镜头,可同时采集到分辨率为 640×480 的彩色图像和深度图像。对于所获取的图像利用PC机上的Matlab软件和结合OpenCV的Microsoft Visual Studio 2010开发环境,对上述算法进行验证。软件运行环境为:Windows 7 64位, Matlab R2015a, PC机配置为Intel(R)Core(TM)i3-3240 CPU @ 3.40 GHz, 安装内存(RAM)为4.00 GB;立体视觉系统所安装的PC版驱动为:ZigJSOpenNI(v1.0.1)。实验情况如图5~图10所示。



图5 原始深度图

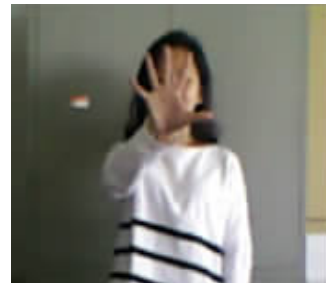


图6 原始RGB图



图7 YCrCb分割

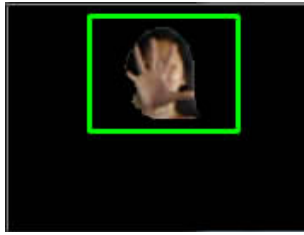


图 8 Grabcut 分割



图 9 未修复后分割



图 10 本文改进算法

图 5 和图 6 分别表示有类肤色背景(脸部)干扰时的原始手势的深度图像和 RGB 图像;图 7 表示将图 6 转换到 YCrCb 颜色空间中的分割结果,仅能大致辨出手势中 3 根手指的形状;图 8 表示对图 6 使用 Grabcut 算法之后的分割情况,图中绿色框表示便捷的人机交互区域,虽然分割效率很高(本次实验处理约 1 秒即可完成),且分割效果比前者要好,分割后手部信息保存较为完整,但仍未能准确地分割出手势,只得到了脸与手的混合分割结果;图 9 为获取的深度图像(即图 5)未经预处理(修复步骤)直接采用三维 OTSU 算法进行处理后的结果,表明了黑洞的存在会对分割工作产生影响,也证明了进行分割操作前对图像进行预处理的必要性;图 10 表示经过本文改进后的三维 OTSU 算法的分割效果,与图 7 和图 8 相比,消除了类肤色背景的干扰,使得手势分割更为精准。

为了进一步凸显改进后的三维 OTSU 算法运行效率更高,故选取多名实验者(10 名)再次进行对比测试,实验采用手势“1”、“2”和“3”,每人每个手势做 10 次,如图 11~图 13 所示(图中展示了其中 1 名实验者的 3 次手势实验)。

使用 Matlab 进行处理时,分别记录下处理对应手势

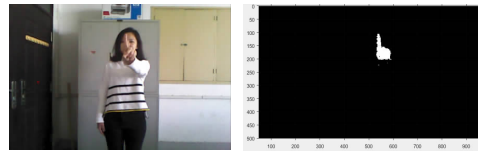


图 11 手势“1”

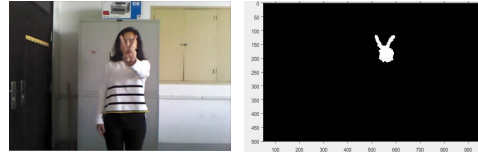


图 12 手势“2”

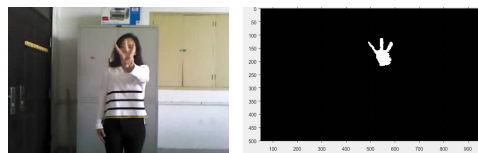


图 13 手势“3”

所用的时间,再分别取其平均值,定义改进率 $\delta = (\text{原始} - \text{改进}) / \text{原始}$,以百分比形式表示,得到表 1,进行对比。表 1 的数据表明,改进后的算法运行效率普遍提高近 20% 左右,证实了改进算法的有效性。

表 1 算法运算时间对比

手势	算法运算时间 t/s		改进率 $\delta / \%$
	原始 OTSU 算法	改进后算法	
“1”	3.0192	2.4154	19.99
“2”	3.0413	2.4991	17.83
“3”	3.0837	2.4355	21.02

3 结束语

本文提出了一种改进三维 OTSU 分割算法,通过立体视觉系统分别获取彩色图像和深度图像,与其他方法进行对比分析。实验结果表明,本文的改进算法不仅能够排除类肤色干扰精准分割手势,而且比原始 OTSU 算法在计算上得以优化,效率得以提高,为手势的后续识别等应用打下了坚实的基础。

参考文献:

[1] ZHU Y, YANG Z, YUAN B. Vision based hand gesture recognition[C]//Proceedings of the 2013 International Conference on Service Sciences(ICSS'13), Shenzhen, April 11-13, 2013:260-265.

[2] 曹昕燕,赵继印,李敏.基于肤色和运动检测技术的单目视觉手势分割[J].湖南大学学报:自然科学版, 2011,38(1):78-83.

- [3] 曹建秋,王华清,蓝章礼.基于改进 YCrCb 颜色空间的肤色分割[J].重庆交通大学学报:自然科学版,2010,29(3):488-492.
- [4] BOYKOV Y, JOLLY M P. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images[C]//Proceeding of 8th IEEE International Conference on Computer Vision, Vancouver, BC, July 7-14, 2001:105-112.
- [5] 徐秋平,郭敏,王亚荣.基于分水岭变换和图割的彩色图像快速分割[J].计算机工程,2009,35(19):210-212.
- [6] ROTHER C, KOLMOGOROV V, BLAKE A. GrabCut: interactive foreground extraction using iterated graph cuts[J]. ACM Transactions on Graphics, 2004, 23(3):309-314.
- [7] 伊聪聪,吴斌,张红英.一种改进的 Grabcut 图像分割方法[J].小型微型计算机系统,2014,35(5):1164-1168.
- [8] 周良芬,何建农.基于 Grabcut 改进的图像分割算法[J].计算机应用,2013,33(1):49-52.
- [9] 赵志平,陈雷月.基于深度传感器图像分割技术的研究[J].信息技术,2014(1):109-112.
- [10] 王艳,张奇志.基于 Kinect 深度信息的手势识别[J].北京信息科技大学学报:自然科学版,2013,28(1):22-26.
- [11] 徐鹏飞,张红英.基于 Kinect 深度图像信息的手势分割和指尖检测算法[J].西南科技大学学报,2014,29(1):49-54.
- [12] 彭诚,孙新柱.一种改进的深度图像修复算法研究[J].重庆工商大学学报:自然科学版,2016,33(1):65-69.
- [13] 张智丰,关坤,金文标,等.二维 Otsu 法的降维递推综合改进算法[J].杭州电子科技大学学报:自然科学版,2014,34(6):18-22.

Research of Hand Gesture Segmentation Algorithm Based on the Stereo Vision System

FANG Wen, SUN Xinzhu, FAN Xiancheng, LIU Mengyue

(College of Electrical Engineering, Anhui Polytechnic University, Wuhu 241000, China)

Abstract: The segmentation of the gesture is one of the important steps to achieve accurate and efficient human-computer interaction, but it is often effected by skin color background and other factors of the external disturbance in this procession, which make the segmentation operation incomplete. To solve this problem, the OptiTrack™ DPC640/30 stereo vision system with introducing the depth image to guide the gesture segmentation is used, and focusing on the large miscellaneous operations and computations of the original three-dimensional OTSU algorithm in segmentation, the operation of the main function is simplified, in which the main variables W_0 and W_1 were recurrence so that each trace of a scatter matrix calculation can start from the position of the last recurrence instead of the initial position to reduce the number of triplet loop effectively. The experimental results show that the proposed algorithm is more effective than YCrCb based color space segmentation and Grabcut segmentation method in dealing with the class of skin color background interference, and the algorithm execution efficiency can be improved compared with the original OTSU algorithm.

Key words: gesture; color image segmentation; depth image segmentation; three-dimensional OTSU algorithm