基于立体场景分层的虚实融合技术研究

**摘要：**虚实融合技术以图像处理、计算机图形学、计算机视觉技术为核心，将计算机生成的虚拟物体和真实场景相结合，实现对真实世界信息的补充和增强。随着技术的进步发展，人们对虚实融合技术的要求也越来越高，如何实现虚拟物体和真实场景的逼真融合成为研究新热点。本文研究了虚实融合的流程，设计出一套基于立体场景分层的虚实融合框架，根据视差和景深的关系实现了虚拟物体和部分前景物体的正确遮挡效果。

**关键词：**立体视觉；图像匹配；虚实融合

**VIRTUAL REALITY FUSION TECHNOLOGY BASED ON stereo scene layering**

**Abstract：**Virtual and real fusion technology takes image processing, computer graphics, computer vision technology as the core, and combines computer-generated virtual objects with real scenes to supplement and enhance real world information. With the progress and development of technology, people have higher and higher requirements for virtual and real fusion technology. How to realize the realistic fusion of virtual objects and real scenes has become a new research focus. This paper studies the process of virtual and real fusion, designs a virtual and real fusion framework based on 3d scene stratification, and realizes the correct occlusion effect of virtual objects and some foreground objects according to the relation between parallax and depth of field.

**Key Words:***[Stereo vision][Image matching][Virtual reality fusion]*

1 研究意义

增强现实（Augmented Reality,简称AR）是在虚拟现实技术（Virtual Reality，缩写为VR）的基础上发展出来的，通过计算机实时计算和多传感器融合，将计算机模拟出的现实生活中不存在的虚拟物体与真实场景叠加在同一个画面或空间中，两种信息相互补充，实现对现实的“增强”，给人超越真实世界的感受。随着技术的不断发展、成熟，AR技术在越来越多的领域投入使用，如医疗、教育、设计、广告等。

AR技术的目的是为用户提供包含虚拟物体信息的真实场景，因此虚实融合技术对AR技术的真实感起着尤其重要的作用，这也是当前AR技术的研究难点之一。

尽管国内外对虚实融合技术的研究都有了一定成果，但实际应用中仍存在着实时性难以满足、计算量大、融合效果不够真实等问题。人们大多根据遮挡关系确定物体的位置信息，大部分的虚实融合都只是简单地将虚拟物体叠加在真实场景之前，这样容易使人产生不真实感，难以很好地使虚拟物体和真实场景相融合。因此，为了更好地实现虚实融合，处理好虚实遮挡关系非常关键，本文设计出一套基于立体场景分层的虚实融合框架，通过基于Opencv的一系列算法对真实场景中的物体进行图形学处理，以达到较为真实的虚实融合效果。

2双目立体视觉

双目立体视觉是计算机视觉领域的重要研究领域。通过双目立体视觉技术，我们可以从两幅或者多幅由摄像机拍摄的图像中，计算出此二维图像上每个像素点对应着的三维世界真实物体的深度信息。利用深度信息，我们可以在二维图像中插入符合三维世界物体遮挡关系的虚拟物体，达到虚实融合的目的。

#### 2.1双目立体视觉原理

人类通过眼睛获得外界物体的信息由此产生视觉。外界物体反射的光经过角膜、晶状体等屈光介质后，会在视网膜表面成像，视网膜上的感光细胞收到刺激把光信号转化为电脉冲，经过视觉传导通路传到大脑的视觉中枢，由大脑对这些信息进行解码，从而构成人类的视觉系统。

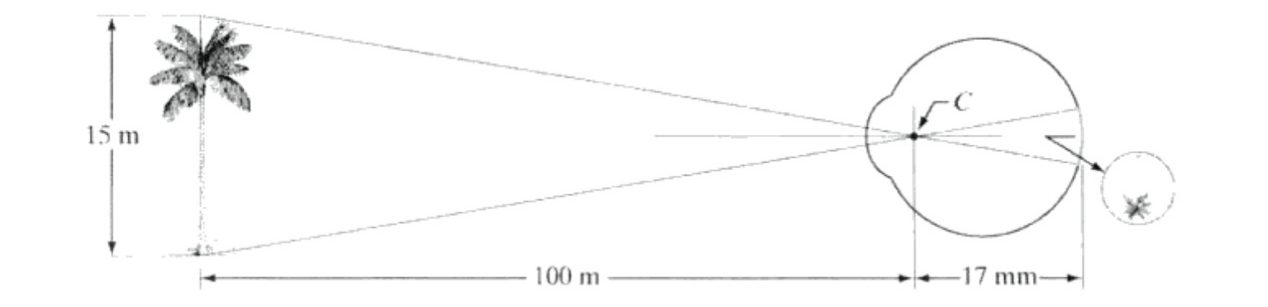


图 2-１ 人眼成像示意图（点C是晶状体位置）

摄像机的成像原理与人眼类似，唯一的区 别在于摄像机的聚焦是通过改变镜头和成像芯片之间的距离实现的，而人眼则是通过改变晶状体的形状，实现正确聚焦，在视网膜上获得清晰的像。

但是，当人看外界物体时，能自然而然地建立起物体和周围环境的三维立体关系，形成立体视觉，而普通的摄像机无法做到。这是因为立体视觉的形成需要基于视差。视差原理是让摄像机通过拍摄的图片建立立体视觉中的重要理论。

图2-2所示为简单的双目立体视觉的成像原理图。

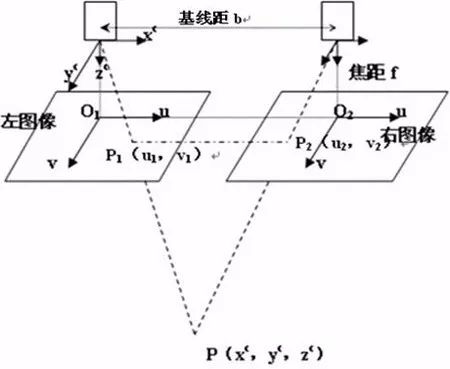


图2-2 双目视觉原理成像图

其中是人眼的焦距，是双眼之间的距离，*P*在三维世界的坐标是，左右图像分别为景物在成像平面（视网膜）上的成像，在人的左右眼的成像分别为和*，*是左眼光心*，*为右眼光心。实际上，光心在人眼内处于晶状体的位置，位于成像平面（视网膜）的前方，此图中为了更方便分析点和面之间的几何关系，将成像平面沿光心对称后置于光心前处。因为两只眼睛位于同一平面上，所以有。由三角形几何关系得到:

|  |  |  |
| --- | --- | --- |
|  |  | （2-1） |

由上式可以看出，在左右眼内的成像位置和是不同的。将上式中和做差可得三维世界中某一点在左右两个成像平面上相应点的位置差，将其定义为视差[1]。

|  |  |  |
| --- | --- | --- |
|  |  | （2-2） |

由此可见，人的左眼和右眼之间存在视差，两只眼睛看到的是不同的图像。人眼把有差别的两幅图像传给大脑之后，大脑可以根据两幅图像之间的差别重建起景物的三维信息，使人能感知到外界物体之间的深度信息。同理，把人眼的视差原理应用在双目摄像机上后，我们也可以根据两个摄像机拍摄的二维图像，通过计算机运算，重新构建出图像场景中的三维信息。

#### 2.2视差与物体深度信息的关系

由上一节我们大致了解到计算机也可以基于视差原理还原出摄像机拍摄图像中物体的三维信息，模拟人的双目立体视觉。本节将对如何把二维图像中获得的视差转换为物体在三维空间中的深度信息进行说明。

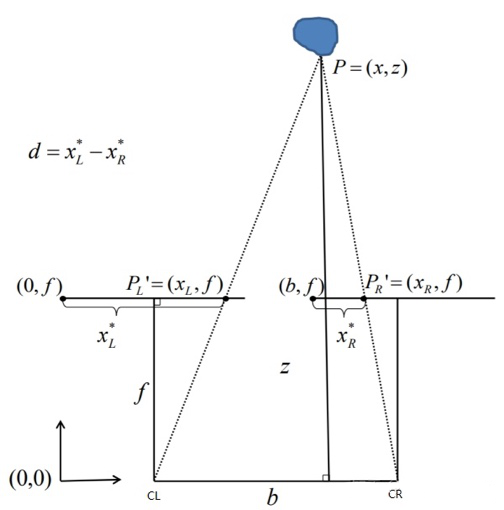


图2-3 视差和深度的关系图

如图 2-3所示，和分别是左摄像机和右摄像机的光心，是左右摄像机之间的距离，是摄像机的焦距，和分别是左右摄像机的成像平面，相当于人眼的视网膜，是我们用摄像机拍摄的一个三维物体上的一点，和分别是点在左右两个成像平面上的成像位置，即点在双目摄像机拍摄出的图像中的位置，和两点间的视差 *，*是摄像机到物体的距离，即为点的深度值。从图2-3中可以看出，所以可以得到：

|  |  |  |
| --- | --- | --- |
|  |  | （2-3） |

由图中几何关系可得到的表达式：

|  |  |  |
| --- | --- | --- |
|  |  | （2-4） |

其中是投影平面的宽度。把代入式（ 1‑1 ）中，可得深度与视差的关系：

|  |  |  |
| --- | --- | --- |
|  |  | （2-5） |
|  |  | （2-6） |

其中和均为双目摄像机参数，为常数，深度与视差成正比关系。因此，想要重构出点的深度信息，只需要在双目摄像机拍摄出的两幅图像中获得视差值就可以了。

#### 2.3立体匹配算法概述

立体视觉技术是基于双目立体视觉原理提出的计算机视觉领域技术。该技术目标是根据摄像头拍摄的图像或者视频，在计算机内部的三维坐标系中重新构建目标物体，从而便于进行后续的各类处理。立体视觉技术经过多年发展，目前已成为较为成熟的一门技术，广泛应用于自动驾驶、工业检测、增强现实等多种领域中。立体视觉技术分为图像获取、摄像机标定、特征提取、立体匹配、三维恢复等步骤。其中立体匹配[2]是立体视觉技术中的关键技术之一，实现由二维图像计算物体相应点之间视差的目的。

早在20世纪50年代，计算机视觉就已逐步开始发展。当时研究者把这一技术归入模式识别的技术中，技术内容主要集中于识别图中的信息，分析和解释二维图像。直到20世纪70年代，David Marr教授提出计算机视觉理论第一个较为完善的视觉系统框架，他随后将其应用在双目匹配上，从两张视差图中重构出有深度信息的立体图形，为后来立体视觉的发展奠定了理论基础[3]。

立体匹配的本质就是一个通过优化模型求解最优解的过程。立体匹配算法是在两幅二维图像中以最小代价寻找相似点，即三维物体各点分别在两幅图像上的成像点。其算法的有效性主要依赖于匹配基元、匹配准则和匹配算法3个因素[4]，分别对应构建最优化模型中的决策变量、目标函数和约束条件以及求解过程3个部分。

匹配基元是指用于进行立体匹配的图像特征，可以是像素点或区域块的灰度值或者是边缘。灰度值这一特征信息最直观，但对光照变化敏感，对图像的要求较高。边缘是像素发生灰度值突变的点集，对光照不同造成的灰度值变化不敏感，是图像匹配的重要特征信息。根据基元的不同，立体匹配算法可分为区域立体匹配算法、基于特征的立体匹配算法、基于相位的立体匹配算法。区域立体匹配算法能根据图像灰度信息直接获得稠密视差图，但在选择约束窗口的大小和形状时比较困难。基于特征的立体匹配算法基于图像的特征进行匹配，但只能获得稀疏视差图，想要得到稠密视差图需要通过计算量较大的插值估计。基于相位的立体匹配算法是根据合成频率的相位展开方法提取相位，在相位相同的地方匹配相应点。匹配基元的选择很大程度上影响了后续算法的构建方式。

匹配准则确定立体匹配过程中的约束条件。立体匹配是条件数很大的病态问题，如果没有条件进行约束，当输入数据有小的误差，会引起问题解的相对误差很大。对于不同的立体匹配算法使用的约束条件也不尽相同，常用到的约束条件包括：极线约束、几何相似性约束、场景约束等等[5]。

匹配算法根据算法运行时约束的作用范围，通常被分为局部立体匹配算法和全局（半全局）立体匹配算法两大类[6]。局部立体匹配算法在其中一幅图像上定一个参考点，在参考点的领域设定一个合适大小、形状和权重的窗口，然后在另一图像相同的窗口内寻找于参考点相似度最高的子窗口最为匹配点并计算视差。局部立体匹配方法匹配精度较低，对视差不连续区域和遮挡区域的匹配效果不佳，但计算量低，适合对实时性要求高的应用。全局（半全局）立体匹配算法定义了一个全局能量函数，其中包含一个数据项，用于描述匹配程度，以及一个平滑项，体现场景约束。全局（半全局）匹配算法事实上就是一个能量函数的优化问题，此问题在二维空间极难解决，因此计算速度较慢。目前部分算法将能量优化问题通过近似来降维，使计算复杂度降低，算法性能得到提升，全局立体匹配算法的应用也变得更为广泛。本文就是利用其中一种全局（半全局）立体匹配算法进行视差图的计算，具体算法原理将在下一节详细说明。

立体匹配算法发展至今已有许多对算法本身的优化，同时人们也在积极尝试在立体匹配算法中加入新的技术以使算法的性能更加优秀。目前人工智能和深度学习是技术发展的大势，部分将神经网络、深度学习与立体匹配算法结合的尝试取得了不错的成果，在Middlebury测试平台的立体匹配算法性能排行榜上排名靠前。立体匹配算法仍在不断发展，未来将有更新、更高效的算法应用于更多领域。

### 3基于场景分层的虚实融合技术研究

根据视差值和深度值的关系，我们可以通过设置视差值范围将立体场景按深度分层，根据虚拟物体和真实场景的遮挡关系，将虚拟物体放到合适的深度位置，从而实现前景物体对虚拟物体的遮挡。同时，为了更好的融合效果，前景物体的边缘也要尽可能清晰。下面是本文采用的虚实融合技术的框架：

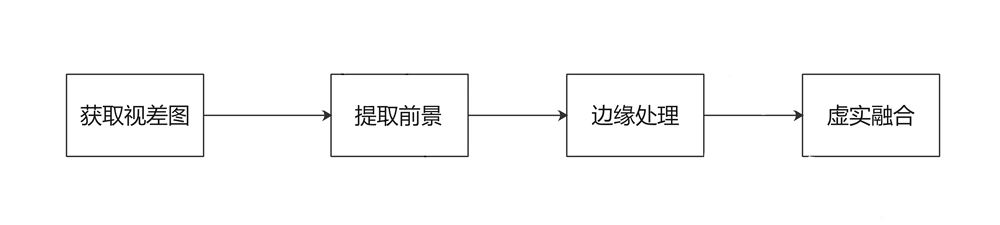


图3-1 本文采用虚实融合技术框架

#### 3.1获取立体场景视差图

通过选用不同的立体匹配算法可以得到不同的视差图。本文主要对SGM(Semi-Global Matching)立体匹配算法[7]进行了研究。

SGM立体匹配算法最早源于Heiko Hirschmuller所著的《Stereo Processing by Semi-Global Matching and Mutual Information》一文，是一种经典的半全局立体匹配算法。SGM立体匹配算法将全局能量函数由二维简化到8至16个一维问题，大大提升了算法的计算速度当前许多性能好的立体匹配算法都由SGM算法优化而来。

SGM算法分为匹配代价计算、代价聚合、视差计算和后处理四个部分。下文将对此四部分进行详细说明。

## 3.1.1匹配代价计算

匹配代价是指参考图像上的每个像素点，用规定视差范围内的所有视差值，与匹配目标图像上每个像素点进行匹配所需要的代价，其计算值常被储存在一个*W\*H\*D*的三维数组中。*W*指的是图像宽度，*H*是图像高度，*D*是视差范围内的所有视差值。计算匹配代价实际上是在衡量参考像素与目标图像中候选像素之间的相关性，当代价越小时，两像素点在对应视差值下相关性越高。

计算匹配代价的方法有很多种，在SGM算法创始者Heiko Hirschmuller的论文里使用的是基于互信息的匹配代价算法。但由于互信息法代价计算的原理比较复杂，且需要迭代，计算效率不高，在实际运用中往往使用计算速度更快的基于Census变换法[8]来计算匹配代价。

Census变换属于非参数图像变换的一种，其本质就是将图像中的像素灰度值转换成为二进制码流。匹配代价通过计算参考图像和目标图像二进制码流的汉明距离得到。匹配代价计算的全过程示意图如下：

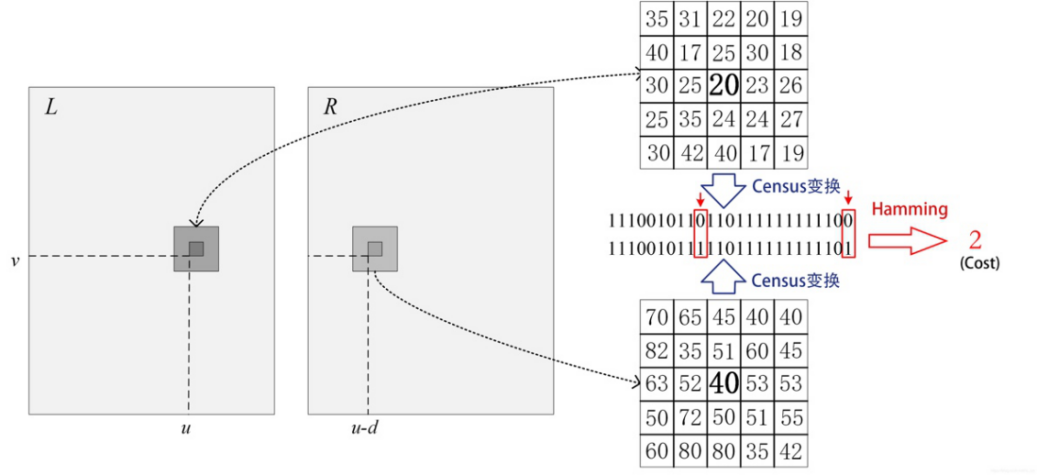


图3-2 基于Census变换的匹配代价计算过程示意图

Census变换的过程是：在左右图像中定义一个合适大小的矩形窗口，此窗口遍历左右两幅图像，取每次的窗口中心处像素灰度值作为参考值，窗口中其他位置的像素灰度值都与参考值比较，如果小于等于则该位置记为0，如果大于则该位置记为1。由此，图像就由灰度值变为了二进制值。设为变换窗口中心像素，表示窗口内其他像素，是Census变换值，基于Census变换的匹配代价算法过程可用公式表示为：

|  |  |  |
| --- | --- | --- |
|  |  | （3-1） |

经Census变换后左图像记为，右图像记为，两幅图像之间的匹配代价用汉明距离计算，如下方表达式所示：

|  |  |  |
| --- | --- | --- |
|  |  | （3-2） |

基于Census变换的匹配代价计算利用像素灰度值之间的相互关系，降低了左右两幅图像对光照一致的要求，局部窗口的设置也易于计算机进行并行运算，是较为高效的匹配代价计算方式。

## 3.1.2代价聚合

基于Census变换的匹配代价计算只考虑了局部的像素相关性，所以其结果容易受到噪声的影响，不能直接获得较好的视差图。在SGM立体匹配算法中用路径聚合法对匹配代价计算的结果进行优化，根据全局能量最优化的策略，寻找每个像素的最优视差使得整张图像的全局能量函数最小。全局能量函数表示为：

|  |  |  |
| --- | --- | --- |
|  |  | (3-3) |

其中为数据项，指的是像素窗口的匹配代价，是平滑项，对相邻像素视差变化超过一定阈值的地方进行惩罚，提高像素邻域相似性，以达到平滑视差图目的。平滑项在SGM立体匹配算法中分为两部分，一部分使用较小的数值对邻域像素视差连续区域（如平面处）进行惩罚，另一部分使用动态变化的大数值，保护真实场景中的视差非连续区域（如物体边缘处）。

能量函数在二维空间的计算较为困难，SGM立体匹配算法中提出路径聚合方法，在8至16个方向上对所有视差下的匹配代价进行一维聚合，大大降低所需计算量。所示的是路径聚合方向的示意图，黄色是8方向，红色是16方向。

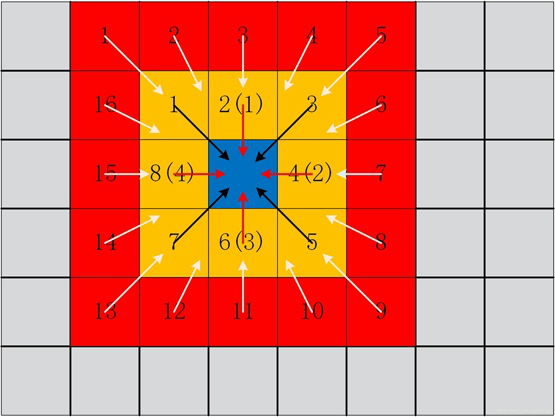


图3-3 路径聚合方向示意图

经过代价聚合后，各像素在各视差上的代价值更能反映出两幅图像之间像素的相关性，降低了噪声对最终视差图的影响。

## 3.1.3视差计算和后处理

视差计算使用赢家通吃（WTA，Winner Take All）策略来确定最小视差，即用某一像素点上经过代价聚合后最小的代价值对应的视差值作为该点的最优视差[9]。经过视差计算后的视差图是整像素精度的，在大多数应用中达不到需求精度，因此SGM立体匹配算法采用二次曲线内插的方法，将最优视差及其左右视差进行二次曲线拟合，把曲线极值点所对应的视差值作为新的子视差值，提高视差精度。

视差计算后得出的视差图往往因为遮挡点、噪声点、误匹配点仍存在一些问题，需要后处理对其进行一些修正。后处理使用左右一致性法（L-R Check）减少误匹配。因为左右图像中的像素点对应着现实世界中的一个物体，它们应该具有唯一的视差值，所以左右一致性法将左右图像互换，再进行一次立体匹配，将两次匹配得出的视差图进行对比，若对应点的视差值的差满足小于一定阈值的要求，则说明满足视差唯一性，该视差值保留，若不满足，则去掉该视差值。

至此，已能获得较为优秀的视差图用于进行后续的虚实融合。

## 3.1.4 SGM立体匹配算法发展与对比

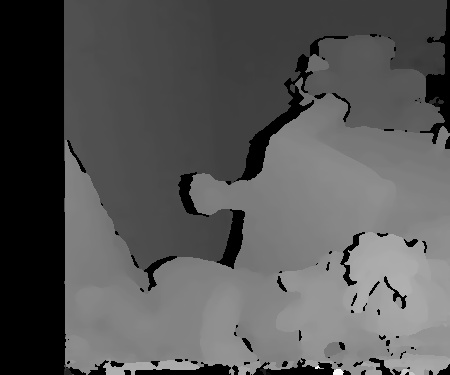
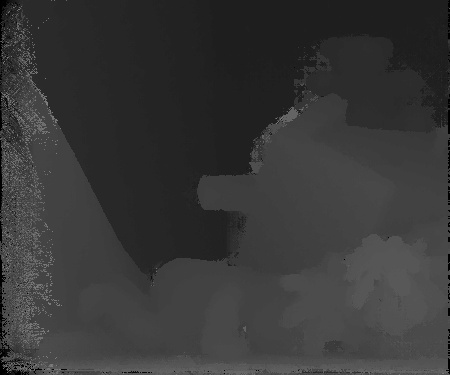
SGM立体匹配算法因其高效的性能受到研究学者的广泛关注，不乏许多人对其算法进行改进以符合更多应用场景。其中著名的计算机视觉库Opencv也将其优化后，作为SGBM（Semi-Global Block Matching）算法纳入库中。与原始的SGM算法相比，SGBM算法加入了Sobel算子对原图像进行预处理，并将处理后图像映射成一个新的图像，得到原图像的梯度信息，用于匹配代价计算。SGBM算法中的匹配代价计算也分为了计算梯度代价和灰度值代价两种。我们将两种算法获得的视差图进行了比较。

我们选取了Middlebury网站上的一组图片进行测试：

（a）左视图 （b）右视图

图3-4 原始图像

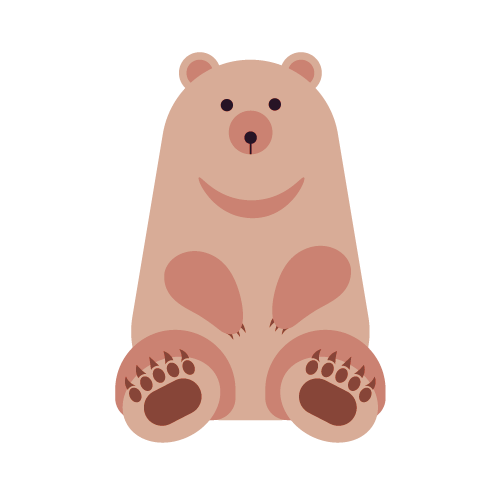
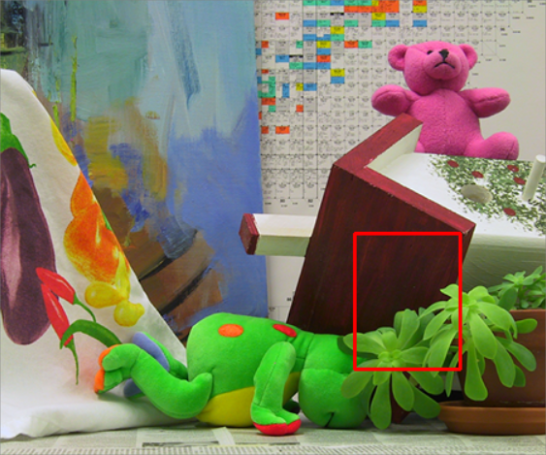
1. SGBM算法得到的视差图 （a）SGM算法得到的视差图

图3-5 视差图

两种算法都获得了较好的视差图， SGBM算法的计算速度更快，但物体边缘处会产生未匹配的黑边，影响本文后续的虚拟物体分层插入功能，且SGM算法的物体边缘更平滑，有利于提取出边缘清晰的前景物体，所以本文使用SGM算法获取视差图。

#### 3.2提取前景物体

现计划加入虚拟物体小熊，如图3-6(a)所示，加入位置为盆栽之后玩具房子之前，如图3-6(b)所示。所以提取的前景物体为盆栽。

(a)虚拟物体小熊 （b）放入位置

图3-6 虚实融合所需原始图

阈值分割法是一种基于区域的图像分割技术。我们采用的阈值分割原理针对于视差图的灰度特征，设定不同的特征阈值，将图像像素点分为若干类。针对于本实验中单一的静态目标图像，只需选取一个阈值，就可以通过视差图中灰度级的不同，来完成对前景和背景的分割。分割结果如下：

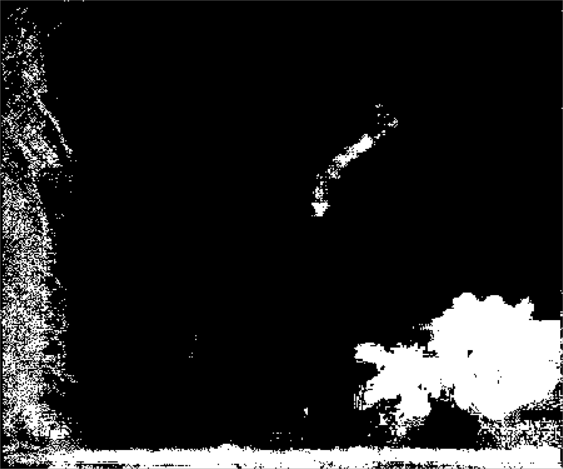


图3-7 直接使用阈值分割的前景物体视差图

由于盆栽的边缘的形状不规则，前景物体提取效果不是很好。本文尝试了K-means[10]和Grabcut[11]两种方式对前景提取方式进行改进。

## 3.2.1 K-means聚类

聚类是一个将数据集划分成若干个族（类）的过程，并且这个过程使同一族（类）的数据对象具有较高的相似度，不同族（类）的数据对象不相似。聚类分析应用广泛，在数据挖掘、数据分析、机器学习、模式识别、图像处理、统计学、市场分析等领域都可以看到它的身影。

从数据挖掘的角度来看，聚类分析分为以下几种：

1. 划分聚类：给定一个包含n个数据对象的数据集，将数据集划分为k个聚类，利用循环再定位技术即通过移动不同组中的对象改变划分的内容，要求同组中的对象彼此相关或相近，不同组中的对象彼此不同或较远。
2. 层次聚类：分解所给定的数据对象，以此来创建一个层次，层次方法分为自下而上和自上而下两种。
3. 基于密度的聚类：不断增长所获得的聚类直到邻近的数据对象密度超过一定阈值为止。
4. 基于网格的聚类：将对象空间划分为有限数目的单元，形成网格结构，在这一网格结构上进行后续的聚类操作。
5. 基于模型的聚类：假设一个模型给每个聚类，再去发现数据对象（符合对应模型）
6. means是划分聚类的代表，也称为K-均值聚类算法，是无监督的聚类算法，K-means算法原理比较简单，也比较容易实现、速度快、聚类效果较好，因此得到了广泛应用。K-means算法的算法思想是，对于给定的样本集，尽量使簇内的点紧密相连而让簇间的距离尽量的大。

它的处理流程一般是：

（1）输入聚类个数k,以及包含n个数据对象的数据库;

（2）从这n个数据对象中任意选择k个对象作为初始中心点;

（3）计算这些中心点与每个数据点的距离，数据点划分到离自己最近的那个中心点所在的组中;

（4）重新计算每个（有变化）的聚类的聚类中心（该聚类中所有对象的均值）;

（5）循环第（3）步和第（4）步直到标准测度函数开始收敛为止，一般标准测度函数采用均方差，具体定义如下：



(3-3)

其中*E*为数据库中所有对象的均方差之和；*p*为代表对象的空间中的一个点；*mi*为聚类*Ci*的均值。该公式表明各聚类本身尽可能紧凑，各聚类彼此尽可能分开。

使用k-means聚类处理过的图像如图3-8（a）所示（取k=5）：

|  |  |
| --- | --- |
| kmeans | grabcut_disparity |
| 1. K-means结果图 | (b)经过K-means处理的前景物体视差图 |

图3-8 K-means处理图

图3-8（a）中，k-means聚类将图像分割为五块，其中不同的颜色代表不同的分割区域，白色区域包含我们的目标前景，再结合阈值范围得出图3-8（b）。

由图3-8（b）可看出由于盆栽叶子阴影部分颜色和正常叶子的颜色差距过大，所以在聚类中被算成了其他类，无法分割，导致分割结果图中有大量空洞。

## 3.2.2 Grabcut算法

Grabcut图像分割算法是一种有效的从复杂背景中提取前景的交互算法，由微软剑桥研究院根据传统的Graphcut算法改进而来，是现阶段使用较为广泛的提取前景物体的方法之一。该算法在提取前景时需要少量的人机交互操作，用户使用矩形框标记前景物体，则计算机将矩形框内像素识别为前景，矩形框外像素识别为背景，再对图像建立高斯混合模型（GMM：Gaussian Mixture Model），使用 GMM 参数初始化每一个像素，并迭代进行mincut[12]算法从而实现前景分割。

Grabcut算法将RGB图像映射成一个包含两个端点的无向图*s-t*，两个端点分别为源点*s*和汇点*t*，源点*s*表示前景，汇点*t*表示背景，原图中每个像素对应无向图中的每个点，将两个相邻的点（对应到原图中即为两个相邻像素）相连，生成的边称作*n-link*，边的权值代表相邻两点颜色的相似度，称为边界能量*V*；无向图中普通点和端点相连，生成的边称作*t-link*，边的权值代表该点和前景、背景的相似度，称为区域能量*U*。

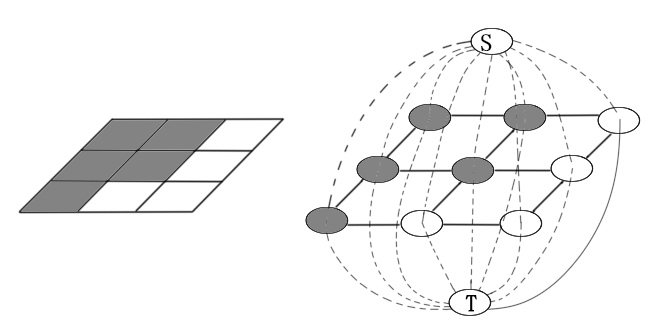


图3-9 s-t图

Grabcut算法将待分割图像表示为*Z=*{*z1，z2，…，zi，…zn*}（其中*n*为像素数，zi表示第*i*个像素的RGB值），使用一个包含了k个高斯分量的GMM（通常取k=5）分别对前景、背景进行建模，用来表示像素前后景的分布情况，由此产生了新向量*K=*{*k1，k2,…，ki，…，kn*}来表示每个像素属于哪个高斯分量。同时引入新向量={1，2，…,i,…n}，，取值为0代表该像素属于背景，取值为1代表该像素属于前景。由此得出Gibbs能量函数为：

** （3-4）

其中*U*函数的定义如下：

** （3-5）

*D*函数定义如下：

 （3-6）

其中函数*P（）*为高斯概率分布函数，函数*( )*为该高斯分量的混合权重系数。函数*P（）*表达式如下：

 （3-7）

其中为高斯函数均值向量，为高斯函数协方差矩阵，因此*D*函数表达式可写为：



 （3-8）

GMM参数可表示为：

 （3-9）

因此，在确定了，，三个参数后，当我们获得一个像素的RGB值时，就可以代入GMM计算，获取该像素属于前景和后景的概率，即可求出*t-link*的权值。

能量公式中，*V*函数的定义如下：

** （3-10）

其中**为定值50，*C*表示相邻像素对，*dis()*表示相邻像素间的欧氏距离，。由此能量*E*可求。

Grabcut算法计算流程如下：

1. 在原图中框选前景物体，将框内像素设为前景，框外像素设为背景；
2. 将框选结果作为参数，利用kmeans算法分别计算得到前景高斯混合模型和背景高斯混合模型的，，三个参数；
3. 分别计算前景像素或背景像素在K个高斯分量的概率密度，然后将其分配给概率密度最大的高斯分量;
4. 根据步骤（3），重新计算前景高斯混合模型和背景高斯混合模型的参数’，’，’。
5. 使用mincut算法进行图像分割；
6. 重复步骤（3）至步骤（5），直至能量*E*最小，完成最终分割。

利用Grabcut处理后的图片如下：

|  |  |
| --- | --- |
| grabcut1_color | grabcut_disparity |
| (a)Grabcut结果图 | （b）经过Grabcut处理的前景物体视差图 |

图3-10 Grabcut处理图

#### 3.3边缘处理

图3-10（b）中，提取的前景图像依旧有一些细小黑洞和白色噪点，所以采用形态学滤波对之进行处理，来得到较为平滑干净的前景视差图。本文中形态学滤波的主要过程是膨胀、腐蚀和中值滤波。

## 3.3.1 腐蚀与膨胀

腐蚀是一个局部过程，将要处理的图像与结构化元素进行卷积，选择一个锚点，求这个结构化元素以这个锚点为中心可以消除的最大值，换个说法，也就是求局部可以覆盖的最小值。这个结构化元素可以是任何形状和大小。

腐蚀的数学公式如下：

图片1 （3-11）

一般腐蚀操作对二值图进行处理，判断锚点颜色是否与周围领域的像素点颜色一样（即是否是白色点，即值是否为255），若一致，则保留，不一致则该点变为黑色（值即为0）。从表面理解，腐蚀就是腐蚀掉图像中的一部分，让目标区域变小的过程。

在Opencv中，我们直接利用erode函数进行腐蚀操作，可以去掉图像中的碎片和噪点。

膨胀与腐蚀一样，需要选择一个结构化元素和一个锚点。正好相反的是，膨胀是要求结构化元素以这个锚点为中心可以覆盖的最大值。

膨胀就是腐蚀的反操作，也可以说膨胀是对图像的负片进行腐蚀操作。也就是判断锚点颜色是否与周围领域的像素点颜色一样（即是否是白色点，即值是否为255），若一致，则保留，不一致则该点变为白色。显而易见，膨胀是让目标区域变大的过程。

膨胀的数学公式如下：

 （3-12）

在Opencv中，我们直接利用dilate函数进行膨胀操作，可以填充图像的细小空洞。

经测试，腐蚀、膨胀的先后顺序会影响最终结果。

对于图3-10（b），本文采取先膨胀再腐蚀的方式，结果如下：

|  |  |
| --- | --- |
| dilate | erode |
| （a）膨胀 | （b）腐蚀 |

图3-11 膨胀和腐蚀处理图

## 3.3.2 中值滤波

## 中值滤波是一种非线性滤波方法，是一种基于统计排序的方法。找到每个像素点符合模板大小的邻域，将邻域的灰度值进行排序，用排序结果决定的值代替中心值（也就是用邻域灰度值的中值代替该像素点的值）。

中值滤波首先需要选择一个模板大小，生成一个模板。这个模板从左上角开始在整幅图像上滑动，计算每个点的灰度值。模板的大小必须是大于1的奇数。模板尺寸越大，滤波后的图像越平滑越模糊。

对于膨胀后的图像进行中值滤波操作，Opencv中定义了medianBlur函数。

为了保护更多的细节，并且生成较为平滑的前景图像，本实验中采用了3\*3的模板。

对图3-11（b）进行处理后如下图所示：



图 3-12 中值滤波处理后的前景物体视差图

到这里就大致完成了前景视差图的提取，用二值化的前景视差图作为模板提取原图像中的前景部分，就可以得到较为平滑，边缘清晰的前景图。

#### 3.4虚实融合

在前景物体视差图中，遮挡住虚拟物体的区域视差值不为零。因此遍历虚拟物体的每个像素，对应到前景物体视差图中的相同坐标处，如果此处视差值为零，则显示虚拟物体；此处视差值不为零，则显示前景物体。融合效果如下图所示：

1. 左视图融合图 （b）右视图融合图

图3-13 融合图

#### 3.5融合结果检验

根据图3-13，可以看出盆栽遮挡了虚拟物体小熊，小熊遮挡了后面的玩具房子，实现了正确的遮挡关系，盆栽边缘也处理得相对平滑，实现了虚实融合。

为了验证小熊是否真的插在了盆栽和玩具房子之间，对融合图再计算一次视差图（此处为了快速检验使用了SGBM算法），结果如下图所示：



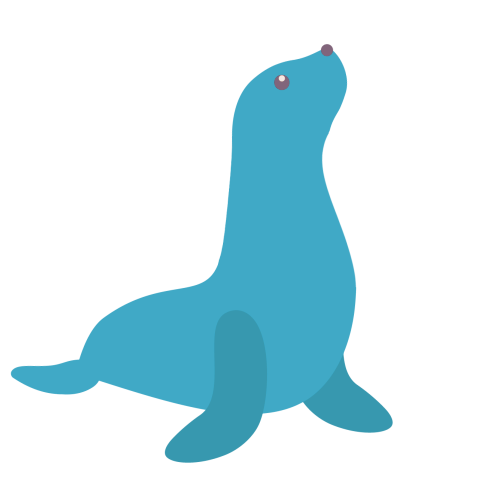
图3-14 检验视差图

由图可得小熊的视差在盆栽和房子之间，成功放入指定位置。

为了更好地验证融合效果，本文对另外两组图片以同样的方式进行虚实融合，详情见图3-15至图3-24。

(a) 左视图 （b）右视图

(c)目标位置 (d)虚拟物体

图3-15 测试图组1，目标位置为圆锥和面具之间，前景为圆锥

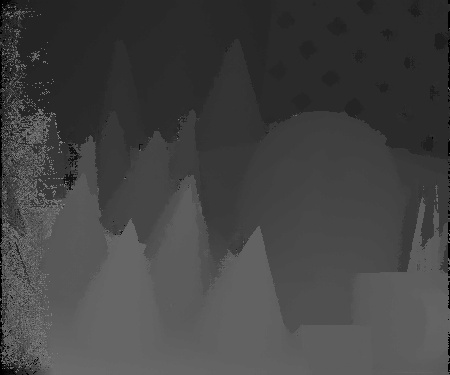
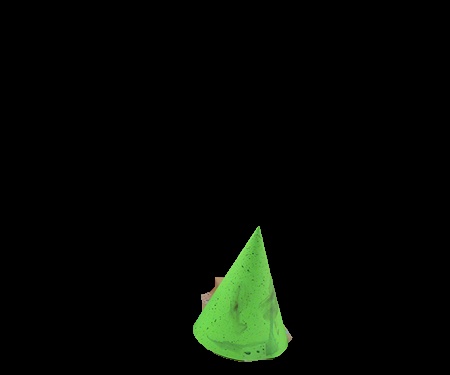
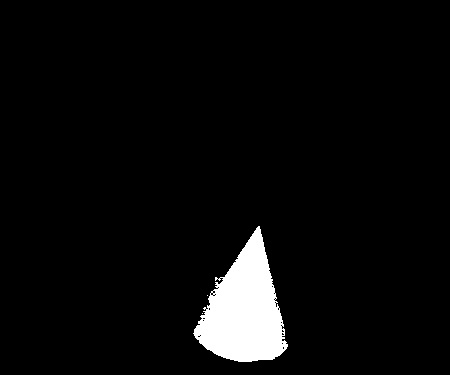


图3-16 视差图

1. Grabcut结果图 （b）经过Grabcut处理的前景物体视差图

图3-17 提取前景

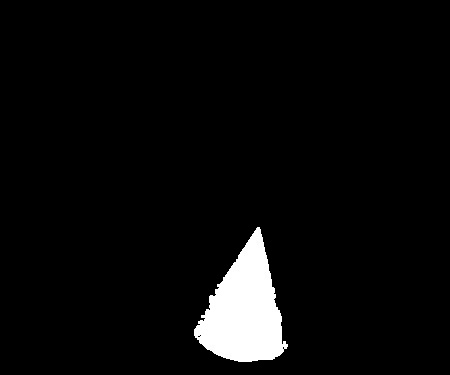
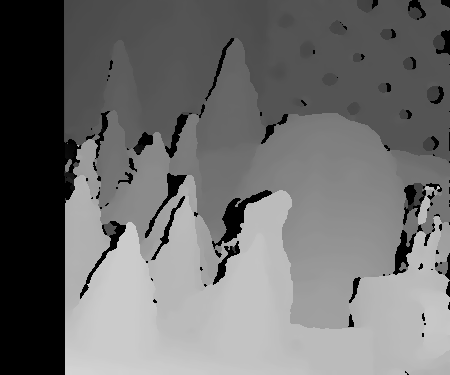


图3-18 形态学滤波处理边缘

（a）左视图融合图 （b）右视图融合图

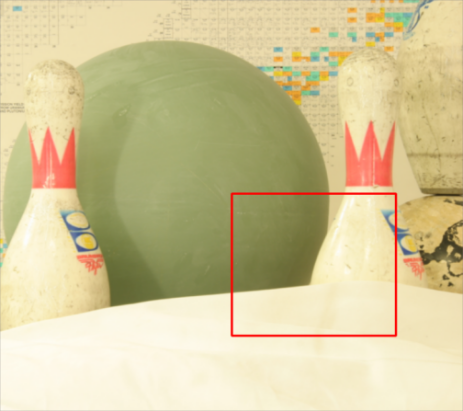
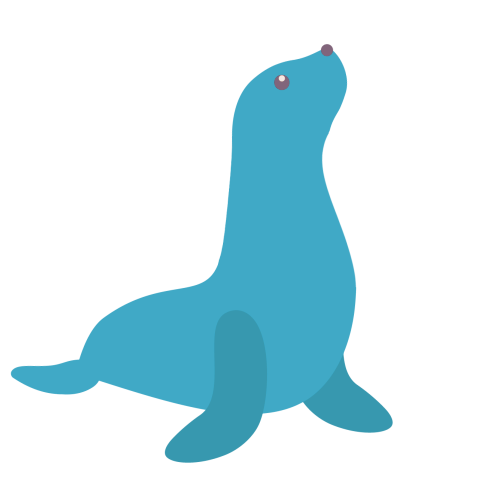


（c）检验视差图

图3-19 融合结果

(a)左视图 （b）右视图

(c)目标位置 (d)虚拟物体

图3-20 测试图组2，目标位置为布和保龄球之间，前景为布



图3-21 视差图

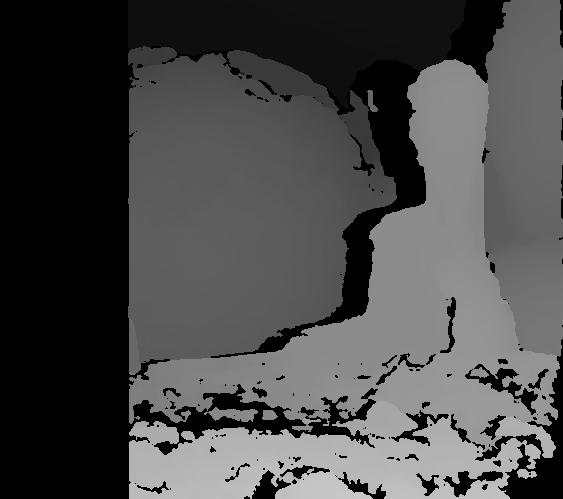
|  |  |
| --- | --- |
| grabcut3 | kmeans_disparity |
| （a）Grabcut结果图 | （b）经过Grabcut处理的前景物体视差图 |

图3-22 提取前景



图3-23 形态学滤波处理边缘

|  |  |
| --- | --- |
| outputL | outputR |
| （a）左视图融合图 | （b）右视图融合图 |



（c）检验视差图

图3-24 融合结果

通过检验视差图可以看到虚拟物体与前景物体的遮挡关系基本正确，虚实融合效果达到预期。

### 4 总结与展望

本文主要从研究意义、双目立体视觉、基于场景分层的虚实融合技术三个方面进行了阐述。其中，基于场景分层的虚实融合技术构造了“获取视差图—提取前景—边缘处理—虚实融合”的虚实融合流程，在获取视差图这一步中，比较了SGBM算法和SGM算法，由于SGM算法得到的视差图边缘更为顺滑流畅，选择了SGM算法；在提取前景这一步中，对K-means聚类算法和Grabcut算法进行比较，由于Grabcut算法能提取更为完整的前景，选择了Grabcut算法。在边缘处理这一步中，经过测试比较选择了先膨胀后腐蚀的方式。

虽然经过测试这种虚实融合的方式基本能达成预期效果，但依旧有许多不足之处需要完善：

1. SGM算法虽然可以得出边缘较为清晰的视差图，但是这种算法的运算量很大，当处理尺寸比较大的图片时，计算速度就很缓慢，在测试过程中出现报错的情况，需要对算法作出适当改进；
2. 当前景物体带有和主体颜色不同的花纹，或者前景物体的阴影部分和主体颜色相差很大时，采用K-means和Grabcut都不能完整地分离出前景物体。
3. 当需要提取的前景物体后面紧挨着颜色相似的背景干扰元素时，很难去除背景干扰元素。
4. 由于这种方式主要根据颜色分割前景，而不同物体的颜色大多不相同，所以目前只支持提取单独的前景物体，想同时提取多个前景物体就只能使用阈值分割，即使使用形态学滤波处理分割边缘，边缘也很难达到平滑的程度。

# 参考文献

1. 王欣,袁坤,于晓,章明朝. 基于运动恢复的双目视觉三维重建系统设计[J]. 光学精密工程,2014,22(05):1379-1387.
2. 曾文献,郭兆坤. 立体匹配算法研究综述[J]. 河北省科学院学报,2018,35(02):43-48.
3. 隋婧,金伟其. 双目立体视觉技术的实现及其进展[J]. 电子技术应用,2004,(10):4-6+12.
4. 游素亚. 立体视觉研究的现状与进展[J]. 中国图象图形学报,1997,(01):17-24.
5. 刘欢. 基于双目视觉立体匹配算法的研究与应用[D].哈尔滨工业大学,2018.
6. 王新艳,潘巍,王月莲,刘鑫玥. 基于全局差错能量函数的立体匹配算法[J]. 计算机工程,2017,43(07):244-249.
7. Hirschmüller Heiko. Stereo processing by semiglobal matching and mutual information.[J]. IEEE transactions on pattern analysis and machine intelligence,2008,30(2):.
8. 祝世平,闫利那,李政. 基于改进Census变换和动态规划的立体匹配算法[J]. 光学学报,2016,36(04):216-224.
9. 曹之乐,严中红,王洪. 双目立体视觉匹配技术综述[J]. 重庆理工大学学报(自然科学),2015,29(02):70-75.
10. 赵红丹,田喜平.基于K-means算法分割遥感图像的阈值确定方法研究[J].科学技术与工程,2017,17(09):250-254.
11. 梁耀,黎双文,刘鑫磊,李丰果.复杂背景下目标树叶自动分割的GrabCut算法[J].华南师范大学学报(自然科学版),2018,50(06):112-118.
12. 王联君,王静秋.基于GrabCut的磨粒图像分割方法研究[J].机械制造与自动化,2019,48(02):127-130+137.