

本科生毕业设计

**基于视差和运动矢量的立体视频舒适度研究**

**Research of Visual Comfort of Stereoscopic Video Based on Parallax and Motion Vector**

|  |  |
| --- | --- |
| 学 生 姓 名 | 王家才 |
| 专 业 | 网络工程 |
| 学 号 | **150522126** |
| 指 导 教 师 | 权巍 |
| 学 院 | 计算机科学技术 |

**毕业设计（论文）原创承诺书**

1．本人承诺：所呈交的毕业设计（论文）《  》，是认真学习理解学校的《长春理工大学本科毕业设计（论文）工作条例》后，在教师的指导下，保质保量独立地完成了任务书中规定的内容，不弄虚作假，不抄袭别人的工作内容。

2．本人在毕业设计（论文）中引用他人的观点和研究成果，均在文中加以注释或以参考文献形式列出，对本文的研究工作做出重要贡献的个人和集体均已在文中注明。

3．在毕业设计（论文）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

4．本人完全了解学校关于保存、使用毕业设计（论文）的规定，即：按照学校要求提交论文和相关材料的印刷本和电子版本；同意学校保留毕业设计（论文）的复印件和电子版本，允许被查阅和借阅；学校可以采用影印、缩印或其他复制手段保存毕业设计（论文），可以公布其中的全部或部分内容。

以上承诺的法律结果将完全由本人承担！

作 者 签 名： 年 月日

# 摘要

立体视频从上世纪发展至今，在技术上已取得很多突破，人们随时能在影院体验到这一技术带来的身临其境的震撼效果。但由于舒适度问题的存在，立体视频技术的应用领域仍被限制在影院等场景，其在日常生活中的应用仍远不如传统平面视频。

立体视频的舒适度主要受视差和运动的影响。本文针对这两个影响因素，提出一种基于统计学方法的立体视频舒适度客观评价模型。该模型能以较快的速度识别出视频中存在不适的帧，并对整个视频的舒适度作出评价。主观评价试验的结果表明该模型对所给样本的视频舒适度估值的误差率在8%以内，具有较好的准确率。

**关键词：立体视频 舒适度 视差 运动矢量**

# Abstract

The technology of stereoscopic video has been evolving since last century and many breakthroughs has been made. Nowadays people can experience the impressive effect brought by the technology in cinema at any time. However, due to the existence of comfort problems, the application field of stereoscopic video technology is still limited to theaters and other scenes. Its application in daily life is still significantly less than traditional 2D video.

The comfort of stereoscopic video is mainly affected by parallax and motion. Based on these two influencing factors, this paper proposes an objective evaluation model of stereo video comfort based on statistical methods. The model can recognize the uncomfortable frames in the video and evaluate the comfort of the entire video at a fast speed. The results of the subjective evaluation test show that the error rate of the model on given sample can be limited within 8%. The model proves to be of good accuracy.

**Keywords：Stereoscopic Video; Comfort; Parallax; Motion Vector**

**目录**

[摘要 I](#_Toc10724274)

[Abstract II](#_Toc10724275)

[第一章 绪论 1](#_Toc10724276)

[1.1 立体视频舒适度研究目的及意义 1](#_Toc10724277)

[1.2 立体视频舒适度的研究现状 2](#_Toc10724278)

[1.3 本课题主要研究内容 3](#_Toc10724279)

[第二章 立体视觉原理和不适源分析 4](#_Toc10724280)

[2.1 双目立体视觉的形成原理 4](#_Toc10724281)

[2.2 立体视觉的不适源 5](#_Toc10724282)

[第三章 基于统计学的立体视频舒适度评价算法 7](#_Toc10724283)

[3.1 算法考虑的舒适度影响因素 7](#_Toc10724284)

[3.1.1 立体视频帧画面内视差 8](#_Toc10724285)

[3.1.2 立体视频帧间运动矢量 9](#_Toc10724286)

[3.2 影响因素的提取 10](#_Toc10724287)

[3.2.1 画面内视差的提取 10](#_Toc10724288)

[3.2.2 运动矢量的提取 11](#_Toc10724289)

[3.3 基于统计学的客观舒适度估计 11](#_Toc10724290)

[3.4 针对视差的舒适度提升尝试 17](#_Toc10724291)

[第四章 舒适度主观评价实验 20](#_Toc10724292)

[4.1 实验器材和受试者 20](#_Toc10724293)

[4.1.1 立体眼镜选择 20](#_Toc10724294)

[4.1.2 立体视频片段选择 20](#_Toc10724295)

[4.1.3 受试者选择 20](#_Toc10724296)

[4.2 实验步骤 21](#_Toc10724297)

[4.3 实验结果分析 22](#_Toc10724298)

[第五章 总结与展望 24](#_Toc10724299)

[参考文献 25](#_Toc10724300)

[致谢 27](#_Toc10724301)

[附录1 主观实验结果 28](#_Toc10724302)

[附录2 程序代码 29](#_Toc10724303)

# 绪论

传统平面视频技术目前已取得长足发展，从最初的CRT显示器全面过渡到了高分辨率、高刷新率和高色彩还原度的LCD显示器，人眼早已无法察觉显示面板的像素密度的进一步提升。而当传统平面显示技术的进步无法满足人们对更真实、更高还原度画面的追求，立体视频技术凭借其逼真的视觉体验和全新的交互体验，正逐渐成为人们的新宠，在教育、科研、军事、医疗、娱乐等众多领域得到越来越广泛的应用。但随着立体视频技术应用到越来越多领域，其舒适度问题也日益凸显出来。

## 立体视频舒适度研究目的及意义

立体显示技术早在上个世纪就已开始发展，人类史上第一部真正意义上的立体视频也早在1953年就已上映。但由于各种原因，这种具有全新视觉体验的视频形式却并没有很快在大众中流行开来。直到2009年，出自二十世纪福克斯之手的现象级3D科幻电影《阿凡达》的上映才使得立体视频技术为大众所熟知。但立体视频技术给人们带来身临其境的观影体验和交互体验的同时，也带来了各种舒适度问题。由于立体视频画面景深的持续变化，人眼不得不持续的进行辐辏调节以适应画面变化。长时间观看立体视频时，眼球一直持续进行微小运动，因此很容易产生眼疲劳、头晕目眩等不适症状。拍摄立体视频的技术早已趋向成熟，但由于舒适度问题的存在，目前的商业电影除部分特大制作之外，很少采用3D技术进行摄制。除影院外，立体视频技术目前的另一应用场景是VR一体机设备，各种VR游戏也层出不穷。搭载加速度传感器实时监测头部运动后，可以在立体显示的基础上实现360度的沉浸式交互体验。但这类VR设备与其他采用立体视频技术的设备一样，也同样存在长时间佩戴后导致眼部不适的问题[1]。

目前已有手机厂商成功实现了全球首个采用3D结构光技术的5G立体视频通话演示[2]。在不远的将来，随着低延迟、高带宽的5G网络的发展，移动端的3D视频通话、3D实景直播、在线VR等对网络延时和带宽要求较高的特殊应用在技术上将逐渐变得可行，而立体视频的舒适度问题则会成为制约立体视频技术在这类日常使用场景中的广泛应用的主要原因。若能通过研究立体视频舒适度问题，定位导致视觉不适的根源，消除或减轻长时间观看立体视频所产生的视觉疲劳症状，提升人们的立体视频观影体验质量，立体视频技术一定能在5G时代结合虚拟现实、增强现实或混合现实等应用，在更多日常应用场景中大放光彩。

## 立体视频舒适度的研究现状

立体视频技术的起源和发展较早，其舒适度问题也早已为人们所发觉，因此国内外已经针对立体视频的舒适度问题进行了为全面且有价值的研究。

目前的立体视频技术依赖于传统平面视频技术，显示面板的对比度、像素密度、分辨率、色彩还原度等参数会影响2D视频的观影质量，因此也会影响基于此的立体视频的体验质量。法国GIPSA实验室的Raluca Vlad等人的研究[3]中关于画面质量对立体内容的感知质量的影响的结论表明，用于产生立体感的2D图像的质量对3D内容的体验质量有影响，高质量的2D图像是产生高质量3D内容的前提。在国内，天津大学的刘畅等人的研究[4]针对立体视频的对比度和舒适度的关系进行了大量的主观评价实验，证明左右眼的对比度不一致会导致立体视频观影过程的不适感，并最终确定了导致视觉不适的归一化对比度差异门限值为1.18，最小值为-2.04。

除从传统视频继承的舒适度影响因素外，立体视频的观影舒适度同时也受到其他各种复杂因素的影响。在观看传统2D视频时，人眼内晶状体的焦距以及眼球的辐辏都是固定的，而在观看立体视频时，人眼的辐辏需要随时根据视频画面的景深进行调节。当左右画面不一致时，人眼无法调节到使左右眼画面相匹配的状态，这种情况会使人脑产生强烈的不适感。法国波尔多大学的Jérémy Frey等人[5]通过研究受试者在观看立体视频的过程中脑电图的变化来找出人脑认为不适的视频帧，在视差发生变化的1秒内以74%的准确率识别到人脑的异常，由此确定了立体视频的视差与舒适度的相关性，这与西班牙马德里理工大学的Rodrigo, J.A.等人的研究[6]具有一致的结论。除这种借助复杂手段进行的客观舒适度研究之外，也有许多其他基于主观评价测试的舒适度研究。国外相对较全面的有西班牙马德里理工大学的Juan Pedro López等人的研究[7]，其基于立体视频帧的景深图和运动信息，针对一组立体视频内容进行了一系列主观评价测试，较为全面的分析了视频帧内的视差信息、帧间的运动信息以及二者的组合对观看者的舒适度影响，得出了一些常见的不适场景。较大负视差画面给观影者带来物体出屏的真实感的同时，也常常引起辐辏-调节冲突，超过舒适区的负视差是导致视觉不适的一个重要因素。实验表明画面同时存在运动和负视差时会导致舒适度较大程度的降低。此项研究同时也发现视频帧间的高速运动与观看者产生不适感之间的较高相关性。在国内，西安电子科技大学的常波[8]通过软件合成一些具有不同视觉特征的非自然立体视频片段，有针对性的研究了立体视频舒适度与各个不同立体视频特性之间的关联，总结出了几个影响舒适度的关键因素。该研究的主观实验结果表明空域和时域上的视差变化均会导致舒适度下降，视差的变化幅值也会影响舒适度。除视差外，研究还注意到立体视频的前景物尺寸与主观实验得出的舒适度的正相关性，但这种相关性在前景物尺寸足够大时减弱。吉林大学的姜秀红等[9,10]分两个研究路线分别建立了舒适度评价模型。第一条路线按照人眼注意区域将视差细分为了前景、背景、帧间三类视差，同时考虑视频画面的运动，用这四个因素构建了立体视频舒适度评价模型。第二条路线选择水平视差作为立体视频舒适度评价模型的一个主要因素，在此基础上加入视频画面本身的对比度信息作为模型的补偿因素，基于深度感知理论以及最小二乘拟合法建立了另一个舒适度评价模型。主观评价测试结果表明两种模型都具有一定的可行性。

整体来看，目前国内外针对立体视频的舒适度研究已取得比较丰硕的进展，已确定许多对立体视频舒适度有影响的特征。在实用性方面，基于这些已确定的舒适度影响因素，各项研究已能够对立体视频的创建、后期处理、传输直到观看的过程给出提高立体视频舒适度的实用性意见。虽然学术界对舒适度的研究已较为完善，但所提出的各种舒适度客观评价模型为了追求计算结果的准确性，大多需要参考多个舒适度影响因素，模型较为复杂，对计算性能的要求较高，在计算资源的消耗和计算结果的准确率之间不能很好的取舍。

## 本课题主要研究内容

本课题选取立体视频中的视差和运动矢量两个信息作为评估立体视频舒适度的主要因素，按照已有研究所证实的视差和运动矢量对舒适度的影响方式，通过统计学方法来计算立体视频的舒适度估计值。

本文首先介绍立体视觉的形成原理，分析可能引起人眼不适的特征，引入视差和景深等概念，在此基础上论证视差和运动对立体视频观看舒适度的影响，证明选取这两个因素作为舒适度客观评价算法的主要指标的合理性。然后根据视差的特点和视差对舒适度的影响方式，对从视频帧中获取的视差信息进行统计学计算并通过计算结果判断画面视差的统计学特征是否符合已有研究所识别的引起不适的视差特征，并尝试进行视差调整，估计出视差重映射对舒适度的提升程度。在通过视差估计该帧画面的舒适度值的同时，将该帧与前一帧进行比较，计算出该帧相对上一帧的运动矢量，并分析该帧的运动信息在统计学上的特征是否符合已有研究所确定的导致舒适度降低的运动信息特征。结合视差和运动矢量信息，通过算法估计出每一帧的客观舒适度，在此基础上进一步估计整个立体视频序列的客观舒适度，并将计算结果与主观评价实验的结果进行比对，验证算法的准确性。

# 立体视觉原理和不适源分析

## 双目立体视觉的形成原理

人眼可以感知物体的远近，这得益于人的双眼和大脑的密切合作。在视物时，人的左眼和右眼所看到的画面虽然大致相同，但在细节上存在很多差异，大脑会将这种差异识别成景深信息。在生理结构上，人眼的晶状体可以简化为一个焦距在一定范围内可变的光学镜片。虽然视网膜实际呈球面型，但在视物时，前景物会投影在视网膜中央的黄斑区，球面上的这一小块区域可以近似认为是一个平面。因此，可以将双眼近似的等效为如下图2-1所示的光学系统。

图 2‑1 双目等效光学系统

左眼

右眼

物体

A

A**’**

B

在上图简化后的光学系统中，物体散射的光线经过晶状体汇聚后分别成像到左眼视网膜的A点和右眼视网膜的B点。当物体在足够远位置时，物体的成像点A和B都处于视网膜正中间，此时左眼成像点A所对应的右眼位置A'与右眼成像点B的位置重合，A'B的距离为0。当物体逐渐靠近时，其在左眼的成像点A逐渐左移，右眼成像点B则逐渐右移，A'B的长度逐渐增大。图中A'B的长度与物体的远近相关，物体越近，A'B长度越大，反之亦然。因此，大脑可以简单的通过对比左右眼所观察到的画面中对应点在两眼成像位置的差异大小来估计出物体到人眼的距离。实际上大脑获取物体景深信息的原理和过程相当复杂，但在研究立体视频所依赖的立体视觉形成原理时，可以简单的认为人脑所感知的物体远近只取决于物体在两眼间的成像位置差异，即只与图中的A'B的长度有关。因此，立体视觉本质上是由于同一物体在左右眼视网膜上的成像位置的差异所导致的。这种差异越大，人脑感知到的物体位置越近。

## 立体视觉的不适源

基于上述立体视觉形成的基本原理，针对左右眼展示从不同位置拍摄的同一画面，便可“欺骗”大脑产生立体感。这种方式虽然能够在左右眼画面上较好的模拟人眼的真实视物体验，但其他方面仍存在一些与实际视物体验不一致的地方。

以左右3D格式的立体视频为例，如下图2-2所示，在观看某一帧左右3D格式的画面时，借助于偏振式眼镜，显示屏的左半边画面和右半边画面分别只成像于左眼和右眼视网膜。人脑能从左右眼差异中感知到画面所表达的物体的位置，并转动眼球调整视轴使其交汇于人脑感知的物体位置。但为了在视网膜上投影出清晰的画面，晶状体必须聚焦于屏幕上。此时晶状体聚焦的点被调节在显示屏上，视轴却会聚到了显示屏之后。而在实际视物时，人眼的晶状体聚焦的点和视轴汇聚的点是一致的。

图 2‑2 辐辏-调节冲突

左眼

右眼

感知物体位置

显示屏

观看立体视频时，这种晶状体和眼球视轴的调节不一致被称为辐辏-调节冲突。根据日本早稻田大学的Shibata Takashi的研究[11]，一定范围内的辐辏-调节冲突并不会导致明显不适，这个范围被称为舒适区（ZoC）。在观看距离为0.3米（一般移动设备的建议观看距离）时，舒适区定义下的视轴汇聚点距离约为0.2到0.6米，在观看距离为3米（一般电视的建议观看距离）时，视轴汇聚点的舒适区距离约为1.2米到无穷大。超过舒适区的辐辏-调节冲突被认为会导致不适。

当人脑所感知到的物体位置越近，眼球所要转动的角度越大，眼球越容易疲劳。当视差为正，即某物体在右半画面中的相对位置比在左画面中的相对位置靠右时，物体的感知位置较远，人脑认为物体沉入屏幕内。反之，视差为负时，感知位置较近，人脑认为物体浮于屏幕外。较大的负视差会使辐辏-调节冲突落在舒适区之外，导致舒适度的降低。

当左右眼画面差异较大时，例如左眼画面与右眼画面完全不相关，或左右眼画面存在垂直视差，此时无论如何调节眼球，人脑都无法对左右眼画面进行匹配，此时会产生强烈的不适感。

# 基于统计学的立体视频舒适度评价算法

本章介绍一种基于统计学的立体视频舒适度评价算法。算法主要考虑立体视频的视差和运动矢量信息，利用统计学方法总结出视频帧内的视差分布规律和帧间的运动规律，通过视差和运动矢量的统计学特征来以较低的时间复杂度得出该视频帧大体上的舒适度估计值，在短时间内对视频序列整体上的观看体验给出客观评价。算法在对每一帧进行舒适度估计值解算的同时，若某帧的视差特征被识别为可能导致不适，算法尝试对其进行视差调节并计算通过视差重映射所能提高的舒适度值。

## 算法考虑的舒适度影响因素

现有研究已确定的影响立体视频舒适度的因素多种多样，在数据传输上主要有立体视频的非保真编码和有损传输等，在图像本身客观参数上主要有分辨率、亮度、对比度等，在3D内容特性上有景深变化范围、视差变化范围、辐辏-调节冲突、画面运动等各种因素。在用于快速估计立体视频舒适度的计算模型中，显然不可能考虑所有影响因素。当前多数关于立体视频舒适度的研究认为视差是与立体视频舒适度相关性最强的一个因素，多数舒适度评价模型也都考虑了视差对舒适度的影响。本文为保证舒适度最终估计值的准确性，考虑到特征提取算法的计算资源消耗，在视差的基础上，选择较为容易通过计算获取的帧间运动矢量信息作为视差对舒适度影响的补充。下面分两个小结分别对选取视差和运动矢量作为舒适度主要影响因素的合理性和正确性进行论述。

### 立体视频帧画面内视差

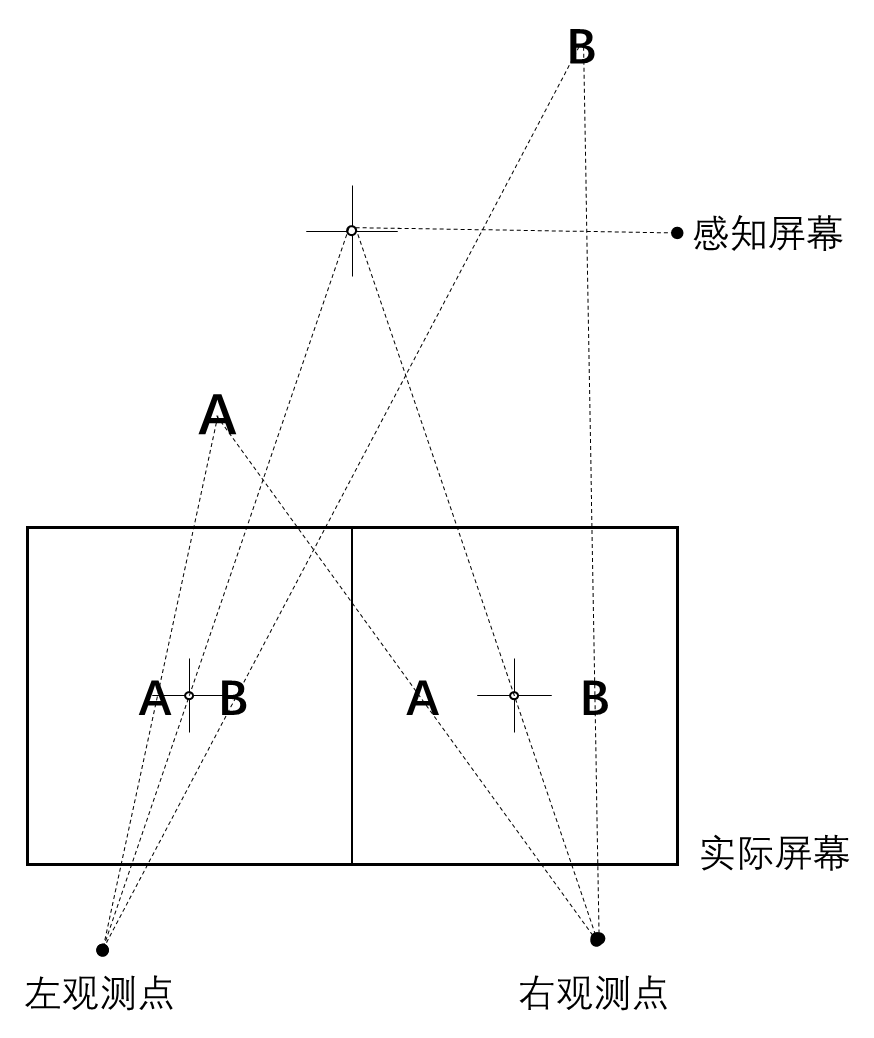
根据双目立体视觉的形成原理，左右眼的视差是携带景深信息的载体，是产生立体感的根源。同时，研究也指出视差是导致不适感的一大源头。在立体视觉中，视差是指以左眼和右眼作为不同观测点观看同一物体时，物体在左右眼视野中的位置的差异。观测点离物体越近，或两个观测点之间的距离越远，这种差异就越大。由于人的左右眼距离恒定，同一物体在左右眼视野中的位置差异就只取决于物体的距离。在观看立体视频时，人眼感知视差来源于视频帧画面中的对应像素点的距离。如下图3-1所示，人脑感知的立体视频画面中心在实际屏幕后方，由于屏幕中文字A在右画面中的相对位置比其在左画面中的相对位置偏左，人脑感知的文字A的位置在屏幕前，人脑认为文字A的位置较近，这种感觉可描述为“出屏感”。文字A所具有的这种视差称为负视差。文字B的情况相反，右画面中文字B的位置相对偏右，具有正视差，感知到的距离较远，这种感觉称为“入屏感”。

图 3‑1 视差与景深关系

通过理论分析可以知道，存在较大负视差的画面具有较强的出屏感，使人脑感知的物体距离过近，控制眼球转动的肌肉需要持续紧张以使眼球转动较大的角度，使视轴汇聚到该物体，因而负视差过大理论上会导致眼疲劳。

除负视差外，视差影响舒适度的另一个特征是视差的变化范围。若画面中物体的视差各不相同且分布较为离散，人眼的聚焦点较难从一个感知深度较近的物体立刻转移到另一个感知深度较远的物体，此时也容易产生不适。

### 立体视频帧间运动矢量

若单独选取立体视频帧内的视差作为舒适度评价算法的参考因素，所估计出的舒适度都只针对某一特定帧，这种方法用于评估立体静态图的舒适度较为合理，但不适用于立体视频的舒适度评估。从本质上讲，立体视频虽也是由一帧一帧的静态图组成的，但立体视频的舒适度并不只取决于组成视频的所有视频帧的舒适度，视频帧的运动同样会对立体视频的舒适度产生影响。立体视频画面中物体位置的运动会引起眼球的运动，若视频画面内的物体一直保持运动，眼球同样必须一直进行辐辏调节，使物体通过眼球晶状体所成的像落在视网膜的黄斑区以得到物体清晰的画面。因而若立体视频帧间存在较大的运动，会导致眼球同样产生较大的辐辏调节运动，导致舒适度降低。

立体视频的帧间运动矢量并不完全独立的影响视频的舒适度。前述Juan Pedro López等人的研究[7]表明，视差和运动虽都能独立的影响立体视频的舒适度主观评价实验结果，但二者的在影响立体视频舒适度时存在一定的“协同关系”，即视差和运动对舒适度的影响分别均为1时，二者同时出现后对立体视频舒适度的影响却有“1+1＞2”的效果，即二者会在原来的基础上加强各自对舒适度的影响。研究认为当某视频画面内某物体的视差为负，且同时该物体在该帧的位置相对其在上一帧的位置存在运动时，会导致较为明显的舒适度降低。具有正视差的物体由于感知距离较远，其位置变化在人眼看来并不明显，因而其运动并不会引起较大的辐辏调节。而具有负视差的物体本身会使眼球的辐辏调节角度过大，当其存在运动时，即使是较小的运动，也会引起眼球视轴较大的转动，因而相比正视差和运动的组合，负视差与运动的组合更容易引起明显的舒适度降低。

## 影响因素的提取

### 画面内视差的提取

立体视频帧内的视差是指对应像素点在左右画面中的相对位置的差异程度。对立体视频帧内视差的提取本质上是对左右画面内像素点的匹配。双目立体视觉的匹配算法多种多样，也各有其优缺点。应用较为广泛的有总绝对差值算法（SAD算法）和半全局块匹配算法（SGBM算法）。SAD算法的核心思路是针对左画面的每一个像素块，在右画面中寻找差异最小的像素块。块内像素点差异的绝对值之和越小，算法就认为两个块的匹配程度越高。如下图3-2所示，当算法在视频帧的左画面执行块匹配到文字“A”处时，在右画面中扫描到文字“A”出现的块时，左右像素块的相似度最高，差异最小，因而其总绝对差值近似为0。而右画面中其他任意位置的像素块均与左画面的像素块存在一定差异，两个像素块之间的SAD大于0。可以简单认为总绝对差值最小的两个像素块对应视野内的同一空间位置。这种算法实现简单，时间复杂度较低，匹配结果的准确性也在可接受范围内，因而在一些准确性要求较低的场景下应用较为广泛。

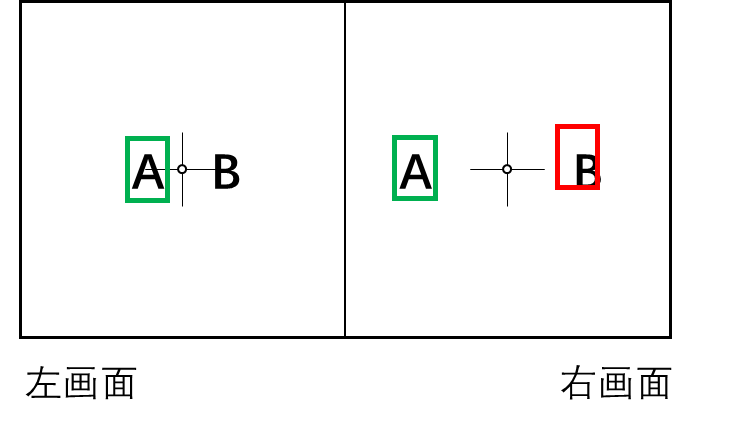


图 3‑2 SAD算法匹配过程

由于舒适度的准确估计依赖于视差信息的准确获取，因而SAD算法虽然在双目立体视觉匹配上的速度较快，但准确性难以满足要求。本文的舒适度评价算法采用openCV所实现的SGBM算法来获取立体视频帧的景深图。SGBM算法主要参考了德国航空航天中心的Heiko Hirschmüller的研究成果[12]，基于互信息和全局平滑约束的近似来执行双目视觉的像素点匹配。这种算法的时间复杂度相比SAD算法要高，但实际测试表明其匹配结果比SAD算法更准确，且运行时间也在可接受的范围内。

### 运动矢量的提取

立体视频所携带的运动信息是其与立体静态图像的根本不同之处，在立体视频的每一帧静态图像的基础上评价立体视频的舒适度，就必须考虑视频帧间所包含的运动信息。视频的运动矢量的提取也有多种方法，由于对立体视频的舒适度估计需要考虑的视频运动信息只涉及到判断画面是否存在引起不适的较大运动，因而对运动信息提取的准确性要求不高。本文采用openCV中的光流场法来计算视频的帧间运动信息。openCV中的稠密光流算法实现参考了瑞典林雪平大学计算机视觉实验室的Gunnar Farnebäck [13]所提出的基于多项式展开的两帧运动估计算法。实测该算法可以在较短时间内对视频两帧之间的运动信息进行较为准确的提取。

## 基于统计学的客观舒适度估计

在借助openCV完成了对立体视频帧内视差的准确提取和帧间运动信息的大致提取后，便可基于这两个信息对该立体视频序列进行客观舒适度估计。为保证算法的实现相对简单、时间复杂度和空间复杂度相对较低，本文拟选择关于视差和运动矢量的几个统计学特征值来对立体视频的舒适度进行估算。下面对立体视频的常见不适场景分析其统计学特征。

前述研究已证实具有负视差的画面会导致立体视频的观影舒适度降低。而具有负视差的画面在视差数据上的统计学特征为负视差像素点所占总像素点的比例较大，视差的平均值偏小。视差的平均值在一定程度上可以代表画面整体上的远近。若视差的平均值接近或小于0，说明画面内可能有半数物体具有“出屏感”，感知距离较近，也更容易导致辐辏-调节冲突落在Shibata舒适区之外，导致观影者产生视觉不适。而负视差像素点的比例更可以直接说明画面内具有负视差的物体的比例，通过简单理论分析可知负视差像素比例与舒适度之间的关系为负相关。

根据日本广播协会的Yamanoue Hirokazu等人的研究结论[14]，除负视差外，视频帧画面内的视差变化程度过大和视差分布范围过大都会导致舒适度的降低。统计学中可用标准差这一特征来衡量随机变量的离散程度，因而视差的标准差大小在一定程度上可以衡量该画面内的视差范围和视差的变化程度。标准差越大，说明视差的概率分布越离散，分布范围越广，该帧画面的视差变化程度越大，人眼需要在较大的景深范围内调节眼球辐辏角以看清视野内的不同物体，因而其舒适度估值应该较低。此外，法国南特大学的Jing Li等人的研究[15]发现虽然不同的人对视差和运动有不同的视觉不适敏感度，但负视差和运动的结合普遍会导致观看者的体验质量较大程度的下降。在算法中可以通过运动矢量信息提取出存在运动的像素点，然后对这些存在运动的像素点计算视差的平均值以判断该帧的运动区域视差是否为会导致不适的负视差。

过大的画面运动会引起眼球的大幅度辐辏调节，进而导致舒适度降低。与这种情况对应的统计学特征为运动矢量大小的平均值。若运动矢量大小的平均值偏大，说明画面存在较大幅度的运动，该帧的舒适度估计值应该相对较低。若画面内存在较为复杂的运动形式，比如不同物体同时具有不同的运动速度和方向，此时会给人一种“眼花缭乱”的感觉，人眼很难在混乱的运动中追踪到前景物。运动形式的复杂程度可以通过运动矢量的标准差来衡量。若运动矢量的标准差越大，说明各个像素点的运动形式差异越大，画面的运动形式则越复杂。参考Jing Li等人的研究[16]，过快的背景平面运动很可能引起视觉不适。运动矢量的众数可以较好的刻画这种特征。由于画面中背景所占的像素点一般较大，当背景静止时，运动矢量大小的众数必定为0。而当众数不为0时，背景一定存在某种形式的运动，众数的值越大，背景的运动越剧烈，越容易引起该立体视频帧的舒适度较大程度的降低。

根据以上分析，可以得出立体视频舒适度评价算法所参考的七个关于视差和运动矢量的统计学特征：视差的平均值()、标准差()，负视差像素点占所有像素点的比例()，运动区域的视差平均值()，运动矢量大小的平均值()、众数()、标准差()。各个统计学特征的值均存在一定的阈值，只有达到其产生不适的阈值后，才认为其会导致舒适度的降低。在各种影响舒适度的画面特征中，负视差和快速平面运动被认为是较为明显会导致视觉不适的，因而与之对应的统计学特征在评估舒适度时的权重也应较大。此外，各个统计学特征对舒适度的影响方式不同，某些特征达到一定阈值后，即认为其会导致舒适度一定程度的下降，这种影响是“全或无”(All-or-None)的，只要该特征值达到引起不适的阈值，便认为其会导致某种程度的不适，而不需考虑特征值的具体大小。而某些特征对舒适度的影响是线性(Linear)的，当这些特征值在一定范围内增大时，其对舒适度影响程度也逐渐增加。在前述七个统计学特征中，负视差像素点比例和运动矢量大小的众数两个特征对舒适度的影响为线性的，其导致舒适度下降的程度在一定范围内取决于其值的大小，而超过此范围后，特征值的进一步增大将不会导致舒适度继续降低。

通过理论分析对各个特征的权重和阈值进行估计，并将其应用到立体视频样本上进行实际测试并修正后，所确定的各个统计学特征对立体视频的舒适度的影响权重以及各个特征引起不适的启动阈值和截止阈值大致如下表3-1所示。

|  |  |  |  |
| --- | --- | --- | --- |
| **统计学特征** | **权重** | **启动阈值** | **截止阈值** |
|  | 0.15 | -10 | - |
|  | 0.15 | 130 | - |
|  | 0.15 | 0.2 | 0.4 |
|  | 0.15 | 5 | - |
|  | 0.1 | 20 | - |
|  | 0.2 | 0 | 30 |
|  | 0.1 | 20 | - |

表 3-1 各统计学特征权重和阈值

文献[15]的研究结果认为前景和背景的相对视差，以及画面运动速度是与舒适度相关的各种因素中，对立体视频舒适度产生最大影响的。表中对运动矢量的众数所给的权重值最大，而关于视差的统计学特征所占总权重达0.6，因而评价模型给出的最终舒适度评分也将较大程度的依赖于运动速度和相对视差的值，这与该研究的结论是一致的。

公式(3-1)给出了视频帧的舒适度估计值的计算方法。

(3-1)

其中表示第i个统计学特征值对舒适度的影响权重。是关于x的二值函数，当x大于或等于该统计学特征引起不适的阈值时函数值为1，否则为0。是关于y的分段线性函数，其对和的取值由下述公式(3-2)和公式(3-3)给出。

(3-2)

(3-3)

负视差像素点的比例在0.2以下时，认为画面的不存在较大负视差区域，不会导致不适，此时函数值为0。当负视差像素点比例从0.2增加到0.4时，观看者感知的近景物逐渐增多，对人眼产生不适的程度逐渐加深，因而认为P\_np的值与其对舒适度的影响程度为线性关系，结合其权重后，负视差像素比例从0.2到0.4对应的舒适度降低值为0.05到0.1。当该比例超过0.4时，认为该画面存在过大负视差区域，导致舒适度估计值降低0.15。类似的，运动矢量的众数在小于等于0时认为画面较为稳定，不影响舒适度。当众数值在0到30之间时，认为画面存在一定程度的背景运动，因而将其影响归一化为0.1到0.15。Mo\_mv突破30后，认为画面已经出现剧烈的背景运动，会明显导致不适，此时将导致舒适度估计值降低0.2。

根据上述客观舒适度评价公式(3-1)，可以计算每一帧的视差和运动矢量所决定的舒适度估计值。但在实际应用中，由于视频常见帧率能达到24fps(帧每秒)甚至更高，因而视频相邻帧之间的差异程度很小，相邻帧的视差和运动信息也近似，没有必要对其进行反复计算。因此，算法应对视频帧进行一定密度的取样，以避免对相似图像多次进行视差和运动矢量提取而耗费过多计算资源。考虑人眼视觉暂留时间约为0.1秒，每秒取样10帧是不会过多消耗计算资源的最大合理值。为保证对视频内快速运动的准确识别和对计算资源的高效利用，每隔0.2秒取样，即每秒取样5帧是一个较为平衡的方法，在不遗漏画面运动信息的前提下，不会出现重复提取高度相似画面的视差和运动矢量信息的情况，因而计算的效率较高。

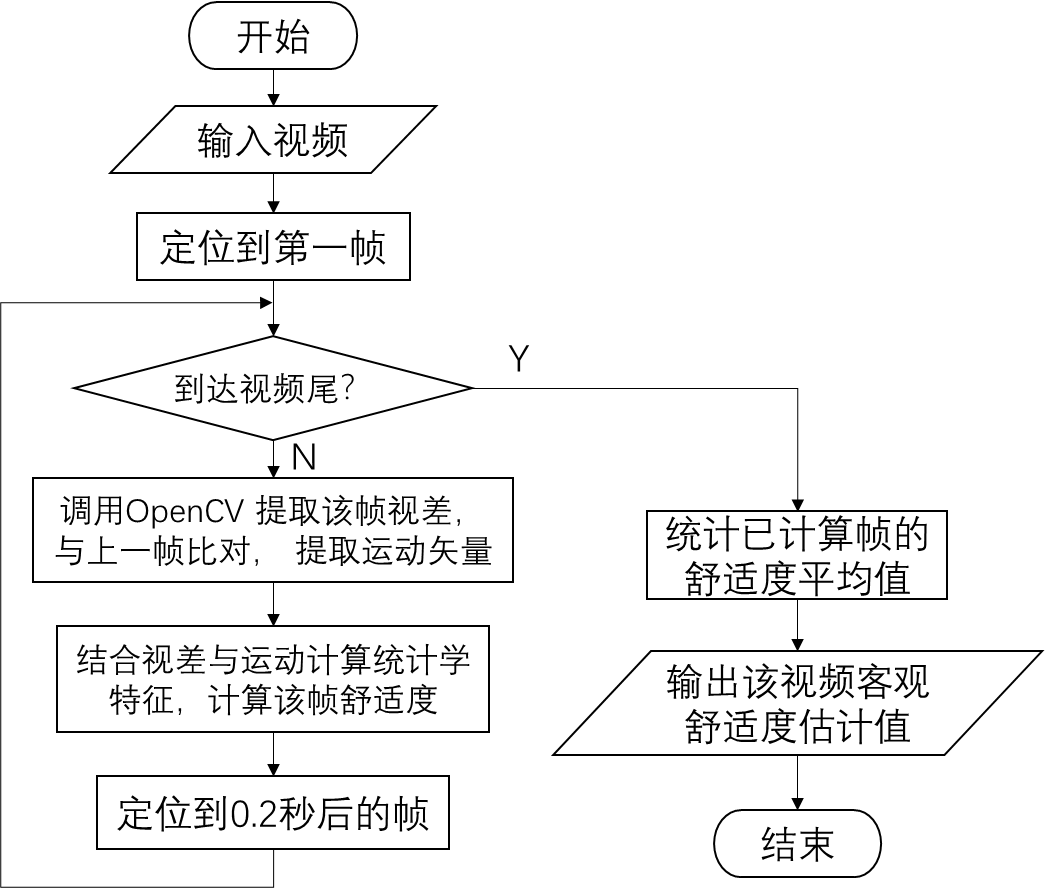
整个算法的流程如下图3-3所示。

图 3‑2 算法流程图

用户将待测试的立体视频序列输入程序，程序从第一帧开始，每隔0.2秒取一帧，将该帧分割为左右画面，通过openCV库的SGBM算法对该帧的左右画面进行半全局块匹配，提取出该帧画面的视差信息，然后将该帧的右画面转为灰度图，与上一取样帧的右画面灰度图一起执行稠密光流计算，提取出两帧之间的运动矢量信息，按照前述公式(3-1)，基于该帧视差和运动矢量的统计学特征，结合各个特征的权重，计算出该帧的舒适度估计值，最后计算出所取样的帧的舒适度平均值作为整个立体视频序列的客观舒适度估计值。

完成算法的编码后，实际执行过程中发现程序对视频帧内的视差提取效果较好，且对帧间的运动矢量提取效果也能达到预期。如下图3-4所示，在所采用的立体视频样本序列中，图3-4 b为程序对于图3-4 a取样帧的视差提取结果，在图中可以清晰的辨别出卡通人物的轮廓和不同景深信息。虽然在轮廓边缘的少数只存在于左画面中的像素块无法匹配到与其对应的右画面像素块而导致一定的噪声，但大体上的景深信息是较为准确的，可以用于估计该帧的舒适度值。

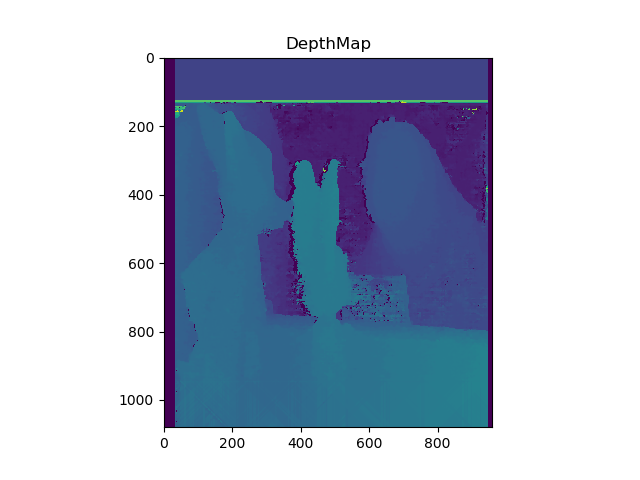


图 3‑4 c) 运动矢量图

图 3‑4 b) 视差图

图 3‑4 a) 《疯狂动物城》视频帧

图3-4 c为程序从图3-4 a中提取的该帧相对上一帧的运动矢量信息。运动矢量的方向和大小分别用色调和明度表示。不同的颜色表示不同的运动方向，该颜色亮度越高，表示其在该方向上的运动矢量的大小越大，运动越剧烈。从图中可以看出该帧的背景除噪声点之外基本不存在运动，且三个卡通人物的不同头部运动均能被较好的捕捉到，画面右侧的卡通人物的眨眼动作也能被较好的识别。

分析算法所提取出的景深图可以判断该画面的视差变化范围不大，三个卡通人物的景深差异较小，且背景较远，为正视差。在运动矢量的hsv表示图中可以看出该画面各个位置的颜色明度均不大，对应运动矢量大小均较小，且整个背景是静止的。算法所提取出的这些数据上的特征与实际观看时的体验大致一致，人眼对此帧的感知质量较高，不会导致不适。而该帧对应的各个统计学特征值如下表3-2，各个值均在引起不适的阈值以下，因而算法给出的该帧舒适度评分为1，认为其不会导致观看时的不适，这与观影者的主观感受一致。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| frameID |  |  |  |  |  |  |  | comfort |
| 310 | -81.96 | 91.01 | 0.174 | -45 | 7.92 | 0 | 16.76 | 1 |

表 3-2 《疯狂动物城》视频帧特征值及舒适度

在另一测试样本中，程序在下图3-5 a的视频帧中提取出的景深图如下图3-5 b，相比上一样本，此帧的视差分布显得更广，前景区域和背景区域的颜色差异更大，且前景区域的颜色也更亮，可以判断该画面存在较大的负视差像素比例和较大的视差标准差。此处可以注意到算法对画面底部的近景计算存在一定的噪点，但仍可对该处的植物有较好的景深识别。



图 3-5 a) 《阿凡达》视频帧

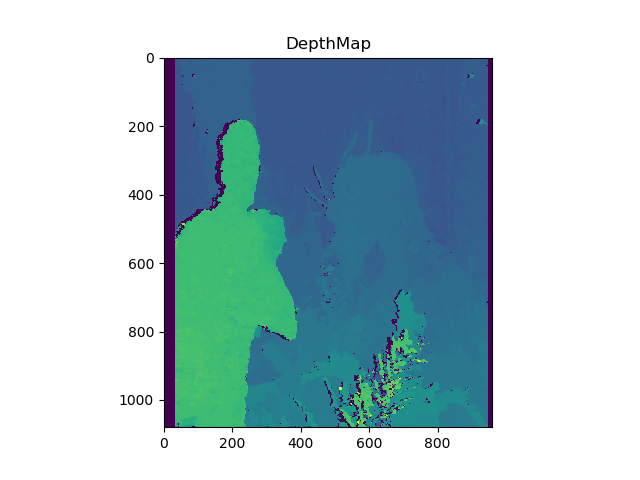
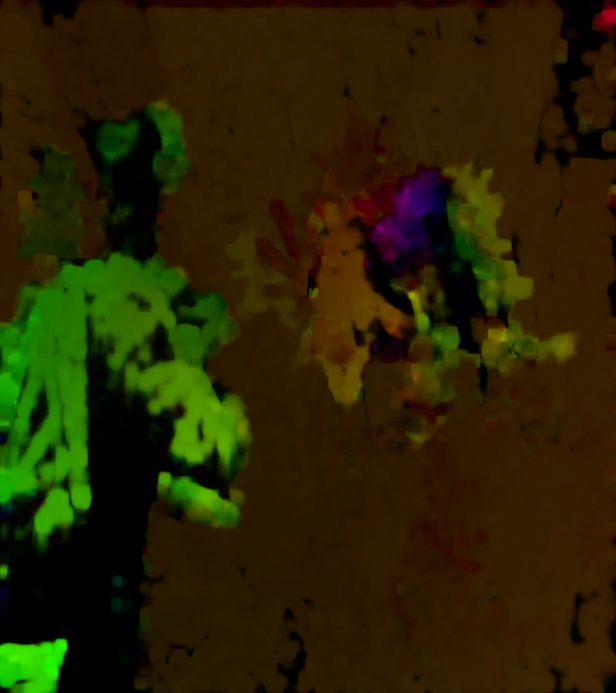


图 3-5 c) 运动矢量图

图 3-5 b) 景深图

该帧对应的运动矢量信息如图3-5 c，可以看出画面内各处均有较大的运动，且整个背景区域的运动信息趋于一致，具有负视差的前景物也有较大运动。可以判断该帧运动矢量的众数较大，且结合视差来看，负视差的区域存在一定的运动。该帧同时满足了视差变化范围大、前景物负视差、背景运动、负视差区域快速运动等各种不适特征，算法对该帧的统计学特征计算值见表，可见各个特征值都超过了阈值，因而算法对该帧的舒适度评价为0分，这也符合人眼对该帧的舒适度感受。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| frameID |  |  |  |  |  |  |  | comfort |
| 135 | 24.53 | 138.99 | 0.464 | 9 | 51.14 | 49 | 32.27 | 0 |

表 3-3 《阿凡达》视频帧特征值及舒适度

按照前述公式和算法流程完成程序并应用于实际立体视频序列后，程序在所给样本上均能给出与预期大致接近的舒适度估计值，对视频样本的舒适度评分能较为准确的粗略区分出好坏。但后续应通过较为完善的舒适度主观评价实验来进一步验证并改进该模型的准确性。

## 针对视差的舒适度提升尝试

立体视频的客观舒适度评价的最终目的是提高其观看舒适度，减少甚至消除视频观看过程中引起视觉不适的因素。但在忠于视频原始内容的前提下，通过减少视频的运动矢量来规避运动导致的不适是不合理的。若原视频本身包含较大的画面运动，后期处理就不应该对其进行慢放、裁切等操作来尝试规避运动所导致的舒适度降低。本文只考虑视差和运动矢量对立体视频舒适度的影响，因而对舒适度的提升尝试是通过视差优化来完成的。

在视差的各种特征中，负视差是对舒适度影响最为明显的一个特征。按照Shibata Takashi的研究结果[11]，各观看距离下的Shibata舒适区范围如下图3-6。图中可以明显的看出，在影院、电视等场景下的观影距离上，其对应的舒适区范围只存在最小值，也即较大负视差会导致不适，而在辐辏可调节的范围内，正视差无论多大都不会落在舒适区之外。在观看距离较近的情况下，舒适区的正视差范围相比负视差也要更大。针对Shibata舒适区的这一特征，在对立体视频帧进行调节时便可针对负视差画面增大其水平视差，使之较多的落在舒适区。

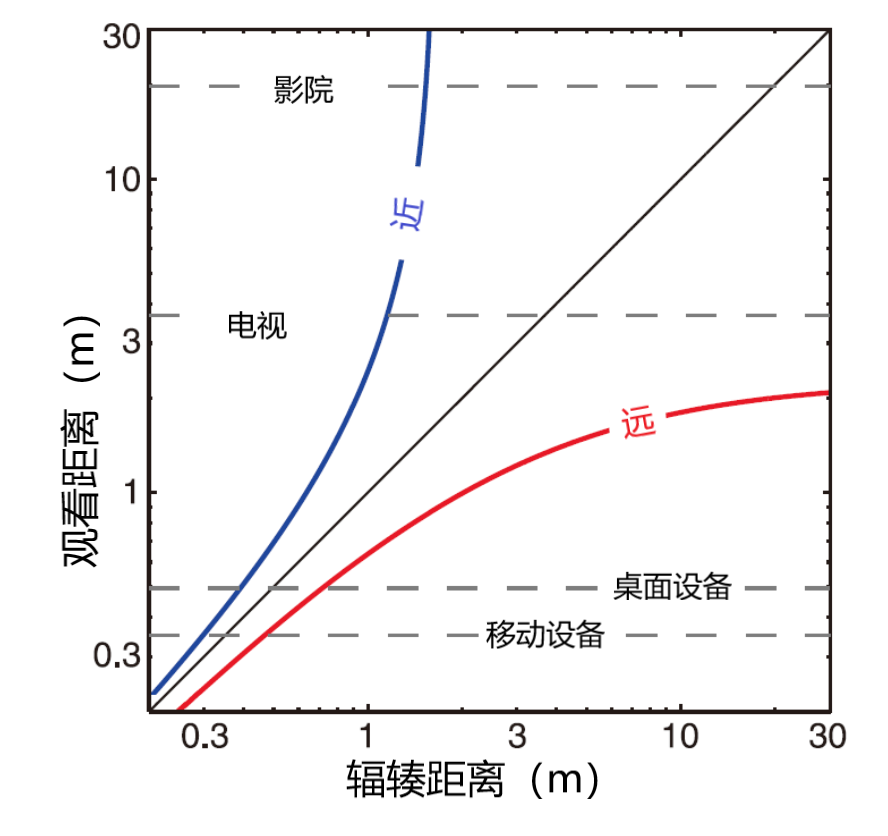


图 3-6 采用米制的Shibata舒适区

在舒适度提升算法的设计上，对水平视差的调节可通过平移左右画面来完成。实际测试后，认为在1920x1080分辨率的视频样本上，将负视差帧的右画面整体向右移动20个像素点可以获得较好的舒适度提升。在负视差导致舒适度降低的画面中，经过这种视差调整后的画面大多可获得1%到10%的整体舒适度提升。在所采用的立体视频样本序列《阿凡达》中，程序从时长为110秒的视频序列中共取样530帧，其中191帧因存在负视差而导致舒适度降低。经过视差调节后，在负视差较大的画面中，舒适度最大可提高7.5%，平均每帧的舒适度可以有5.2%的提高。从数据上看，通过视差调整可以使存在负视差的视频帧获得较为明显的舒适度提升。

# 舒适度主观评价实验

## 实验器材和受试者

目前大多数关于立体视频的舒适度评价研究都依赖于主观评价测试。由于舒适度研究的直接目的就是建立舒适度评价模型并使模型的评估结果符合人的主观感受，因而主观评价测试对于验证模型和算法的准确性是必不可少的，也是相对最可行的验证方案。本文所提出的算法同样采样主观评价试验来进行验证。

### 立体眼镜选择

市面上常见的立体眼镜主要有红蓝色差眼镜、偏振式眼镜和主动快门式眼镜，后两者效果相对较好，但对显示器的要求较高。考虑到实验成本，本文主观评价实验所采用的是谷歌推出的Cardboard VR眼镜盒，它在成像效果上虽不及影院级的效果，但其不会对观影者的舒适度产生较大影响，满足实验要求。Cardboard VR眼镜盒依赖于手机或其他设备的显示屏。在实验中，所用显示设备分辨率为2160x1080。在播放立体视频时，有效显示区域分辨率为1920x1080，左右眼视野均为横向960像素点，纵向1080个像素点。在实际观看过程中没有明显的颗粒感和画面变形，可以认为通过该眼镜盒体验到的立体视频质量与所播放的立体视频片段本身的质量无明显差别。

### 立体视频片段选择

立体视频技术目前较广泛的一个用途是观看3D电影。为保证受试者在实验中的体验与日常生活中接触到立体视频技术时的体验一致，本文主观评价实验所采用的视频序列均为从商业立体电影中截取的片段。用于实验的片段一共6个，分别来自《阿凡达》、《复仇者联盟3》、《驯龙高手》、《钢铁侠3》、《魔法王国》和《疯狂动物城》，涵盖了科幻片、纪录片、3D动画等各种题材。所有片段的分辨率均为1920x1080，帧速率为24帧每秒，可认为视频画质等因素不会对其舒适度产生影响，视频的舒适度主要受3D内容影响。

### 受试者选择

关于受试者的选择，天津大学的马辰阳的研究[17]对受试者的选定标准和数量均给出了指导性意见。为保证主观评价实验的结果准确性，本文所选受试者均为矫正视力正常、年龄20到23周岁之间的在校大学生，满足该研究所建议的受试者选择标准，且所有受试者均为计算机相关专业，对机器视觉也有一定程度的了解，可以认为受试者对所观看立体视频的舒适度评价是相对可信的。受限于实验时间和人脉，本次主观评价实验的受试者一共8人。

## 实验步骤

为使受试者熟悉立体视频的观影体验，在开始舒适度评价之前，让受试者先观看《钢铁侠3》，并告知受试者打分采用10分制，以该片为8分作为参考，对接下来的视频片段进行打分。在观看完每一片段后，立即对其进行打分，视频片段的得分只与受试者所体验到的不适程度有关，而不应考虑其对视频的内容是否感兴趣。视频的舒适度得分应该反映其舒适程度，受试者认为该视频片段引起不适的程度越高，对其的打分应越低。在受试者明确前述打分原则后，按相同的顺序让受试者观看剩下5个视频片段并打分。最后对8名受试者的打分情况进行统计和结果分析。

## 实验结果分析

为保证实验数据相对准确，对每个视频序列的8个得分去掉偏差最大的得分项。去除误差数据后所得实验结果如图4-1所示。

图 4-1 主观评价实验结果

从图中可以看出，去除误差数据后的每个测试视频的平均得分与算法估计值有较大的相似程度，可以认为该基于统计学特征的舒适度评价模型对立体视频的舒适度评价结果是可信的。从图中可以看出，受试者对某些视频的舒适度感受有较大差异，例如名为《魔法王国》的视频样本，即使去除误差数据后，其最低分为6分，而最高分达到了9分，这说明受试者对立体视频的舒适度感受是相当主观的。在实验过程中，多数受试者对于视频样本《阿凡达》的观影感受均为“运动过快，导致眼球无法快速追踪到前景物，产生画面模糊的感觉”，这与算法对该视频的评价一致。另外，所有受试者均认为画面舒适度受画面运动的影响最大，其次是前景物过大导致的“压迫感”，这与已有的基于主观评价测试的舒适度研究结果大致相符。

每个视频样本最终的主观评价得分平均值和算法给出的估计值如表4-1。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 平均得分 | 算法估值 | 误差 |
| 阿凡达 | 6.24 | 6.7 | 7.37% |
| 复仇者联盟 | 7.27 | 7.1 | 2.34% |
| 驯龙高手 | 8.77 | 9.3 | 6.04% |
| 魔法王国 | 8.16 | 7.8 | 4.41% |
| 疯狂动物城 | 9.1 | 9.3 | 2.20% |

表 4-1 主观评价得分及算法误差

单从实验结果来看，虽然视频样本的平均分与算法的估计值存在一定差异，但算法所给出的估值与实际得分之间的平均误差仅为4.47%，最小误差仅为2.19%，且算法所认为的视频样本的舒适度排序与视频样本的主观实验得分排序是一致的，也即，评价模型认为舒适度更佳的视频样本，在主观评价测试中也被认为是舒适度更佳的。这在一定程度上说明该基于视差和运动矢量的舒适度评价模型具有一定的可行性。

# 总结与展望

通过对立体视频舒适度影响因素的研究，本文选取了现有研究普遍认为对舒适度有较大影响的视差和运动矢量两个因素进行立体视频的客观舒适度评估。通过块匹配算法和稠密光流法对立体视频帧的视差和运动矢量进行提取后，根据这些信息的不同统计学特征与舒适度影响因素的对应关系，设计出了基于统计学特征的立体视频舒适度评价模型。主观评价试验的结果证明该模型具有较高的可行性，对各种不同场景的立体视频样本都能得出与实验结果大体一致的舒适度估计值，且在测试样本中的最大误差在8%以内。由于模型只参考了立体视频的视差和运动矢量信息，因而其对每一帧的舒适度估计是较快的。通过调整对视频帧的采样密度，可以很好的在计算速度和计算准确率之间进行平衡，以较快的速度找出视频中可能引起不适的帧。通过调整视差的方式提高不适帧的舒适度也能获得较好的结果，可使负视差帧的舒适度平均提升5.2%。

但整个模型和主观实验仍存在一些问题。例如，模型中各种影响因素大部分被认为是“全或无”的，但实际上某些统计学特征在达到其明显引起不适的阈值前，就已经能对舒适度产生细微影响，简单的将其认为未达到阈值前影响为0是不合理的。此外，各个影响因素的阈值也是仅针对一个视频样本得出的，较为主观且不一定具有普适性。在主观评价实验过程中，也得到了很多有价值的反馈意见。有受试者反映所给的五个视频样本不具有足够的代表性，应该在实验中对更多的视频样本进行评估。在今后的工作中，这些不足之处都是有待改进的。

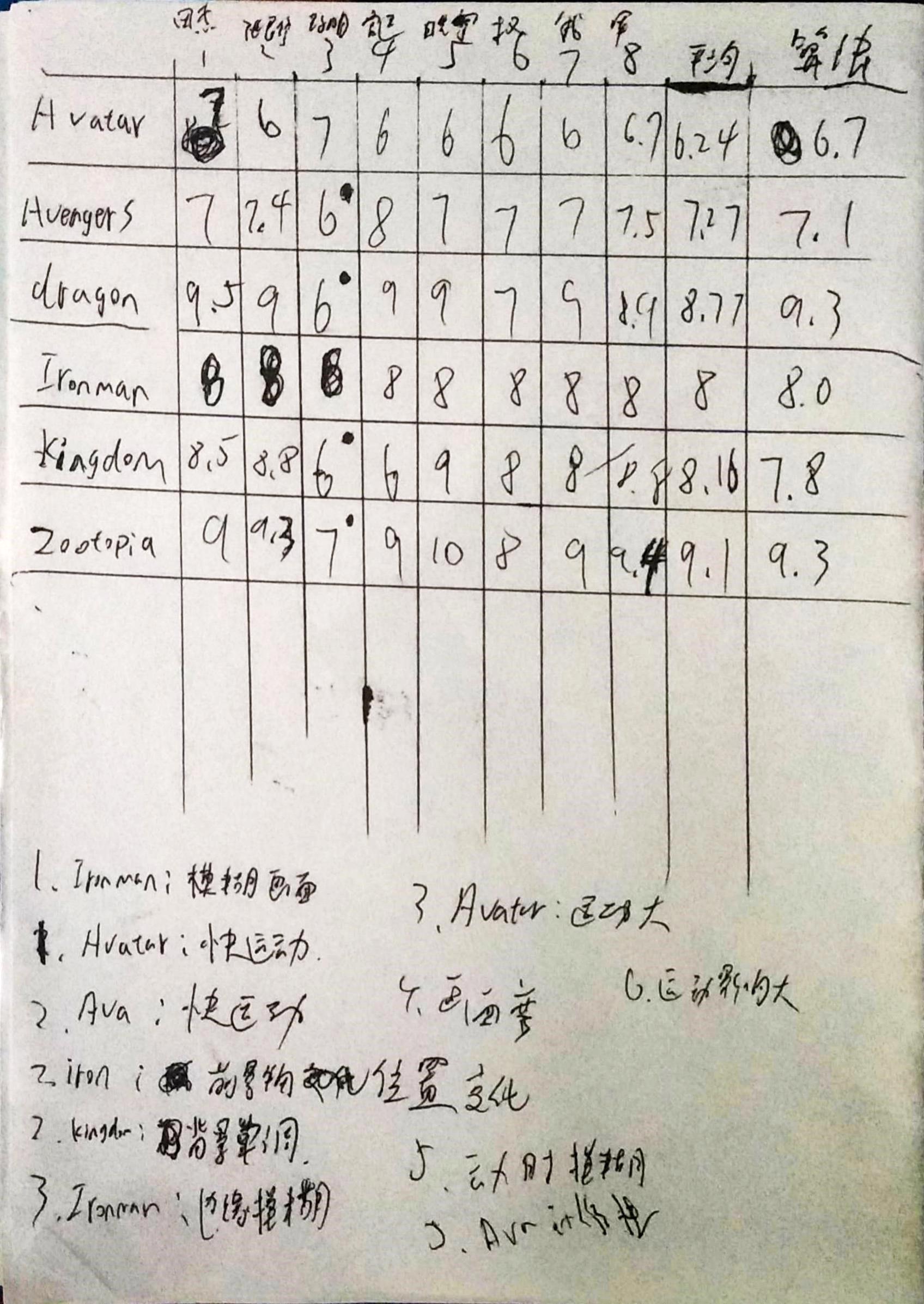
# 参考文献

1. 王舒晓.浅析全息投影、裸眼3D、虚拟现实技术的区别[J].电子测试,2018(22):107-109.
2. 包永刚.完成全球首个5G 3D视频通话演示，5G时代OPPO黑科技更值得期待. https://www.leiphone.com/news/201805/RUJKdXahAunpQ3Y9.html，2018-05-11
3. Raluca Vlad,Patricia Ladret,Anne Guérin. Three factors that influence the overall quality of the stereoscopic 3D content: image quality, comfort, and realism[P]. Electronic Imaging,2013.
4. 刘畅,李素梅.影响立体图像舒适度的对比度范围的测定[J].光电子.激光,2014,25(04):748-755.
5. Jérémy Frey, Aurélien Appriou, Fabien Lotte, Martin Hachet. Estimating Visual Comfort in Stereoscopic Displays Using Electroencephalography: A Proof-of-Concept[J].INTERACT,2015.
6. Rodrigo, J.A., Jimenez, D., Menendez, J.M.. Real-time 3-D HDTV depth cue conflict optimization[P]. Consumer Electronics - Berlin (ICCE-Berlin), 2011 IEEE International Conference,2011.
7. Juan Pedro López,Juan Antonio Rodrigo,David Jiménez,José Manuel Menéndez. Stereoscopic 3D video quality assessment based on depth maps and video motion[J]. EURASIP Journal on Image and Video Processing,2013.
8. 常波. 立体视频舒适度客观评价算法研究[D].西安电子科技大学,2014.
9. 姜秀红. 立体视频舒适度的研究[D].吉林大学,2016.
10. 姜秀红,王世刚,赵晓琳,鲁奉军,王晓燕.基于对比度补偿的立体视频舒适度评价[J].吉林大学学报(信息科学版),2016,34(02):192-195.
11. Shibata Takashi,Kim Joohwan,Hoffman David M,Banks Martin S. The zone of comfort: Predicting visual discomfort with stereo displays.[J]. Journal of Vision,2011,11(8).
12. Hirschmuller, H.. Accurate and efficient stereo processing by semi-global matching and mutual information[P]. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference,2005.
13. Farnebäck G.. Two-Frame Motion Estimation Based on Polynomial Expansion[J]. Image Analysis. SCIA 2003.
14. Yamanoue Hirokazu,Nojiri Yuji,Hnazato Atuo,Ide Sinji,Okano Fumio. Visual comfort and parallax distribution in stereoscopic HDTV[J]. ITE Technical Report,2002,26.51(0).
15. Jing Li, Barkowsky, M., Le Callet, P.. The influence of relative disparity and planar motion velocity on visual discomfort of stereoscopic videos[P]. Quality of Multimedia Experience (QoMEX), 2011 Third International Workshop,2011.
16. Jing Li,Marcus Barkowsky,Patrick Le Callet. Visual discomfort of stereoscopic 3D videos: Influence of 3D motion[J]. Displays,2014,35(1).
17. 马辰阳. 立体视频质量主客观评价方法研究[D].天津大学,2014.

# 致谢

本次论文的完成离不开各位老师、同学和朋友的帮助。指导老师在论文选题、模型建立、视差和运动的提取算法以及整个论文的编排上给出了很多参考意见。在本文的主观评价实验过程中，许多同学也对实验步骤和实验器材给出了改进建议，实验的完成离不开各位同学的积极参与。特别感谢王倩同学在毕业设计工作全程的监督和鼓励。在此对各位表示诚挚的谢意。

# 附录1 主观实验结果



附图1 主观实验结果

# 附录2 程序代码

import ffmpeg

import numpy as np

import matplotlib

import cv2

import os

import sys

from matplotlib import pyplot as plt

from scipy import stats

'''

TODO:

0:读取视频 √

1:获取视差 √

2:获取运动矢量 √

3:确定舒适度 √

4:加舒适度水印 （不做)

5:提高舒适度 √（估计可提高的值）

...

'''

# 打开视频文件

def openVid():

fileName = input("video path: ./vid/")

fileName = "./vid/" + fileName

while not os.path.isfile(fileName):

if os.path.isfile(fileName + ".mkv"):

fileName = fileName + ".mkv"

break

print("file doesn't exist!")

fileName = input("video path: ./vid/")

fileName = "./vid/" + fileName

cap = cv2.VideoCapture(fileName)

if cap.isOpened():

return cap

else:

print("cannot open video.")

sys.exit()

# 获取视频总帧数

def getFrameCount(cap):

if cap.isOpened():

return cap.get(cv2.CAP\_PROP\_FRAME\_COUNT)

else:

print("cannot open video.")

sys.exit()

# 获取帧速率

def getFrameRate(cap):

if cap.isOpened():

return cap.get(cv2.CAP\_PROP\_FPS)

else:

print("cannot open video.")

sys.exit()

# 给出左右画面，计算景深

def getDepthMap(imgL, imgR):

# stereo = cv2.StereoBM\_create(numDisparity = 32, blockSize = 3) # 速度快，准确性较低，单通道

stereo = cv2.StereoSGBM\_create(

minDisparity=-16, numDisparities=48, blockSize=5, P1=320, P2=1280) # 速度稍慢，准确性较高，多通道

return stereo.compute(imgL, imgR)

# 给出前后两帧，计算帧间运动矢量

def getMotionVector(prvs, next):

hsv = np.zeros\_like(imgR) # 将运动矢量按hsv显示，以色调h表示运动方向，以明度v表示运动位移

hsv[..., 1] = 255 # 饱和度置为最高

# 转为灰度以计算光流

prvs = cv2.cvtColor(prvs, cv2.COLOR\_BGR2GRAY)

next = cv2.cvtColor(next, cv2.COLOR\_BGR2GRAY)

flow = cv2.calcOpticalFlowFarneback(

prvs, next, None, 0.5, 3, 15, 3, 5, 1.2, 0) # 计算两帧间的光流，即运动矢量的直角坐标表示

mag, ang = cv2.cartToPolar(

flow[..., 0], flow[..., 1]) # 运动矢量的直角坐标表示转换为极坐标表示

hsv[..., 0] = ang\*180/np.pi/2 # 角度对应色调

hsv[..., 2] = cv2.normalize(mag, None, 0, 255, cv2.NORM\_MINMAX) # 位移量对应明度

return hsv

if \_\_name\_\_ == "\_\_main\_\_":

cap = openVid()

isDemo = int(input("is Demo(0/1)?"))

calcMod = int(input("calc optimize potential?"))

frameRate = getFrameRate(cap)

frameCount = getFrameCount(cap)

framesCalculated = 0

framesOptimized = 0

framesComfort = []

framesComfortOptimized = []

isSuccess, img = cap.read()

if not isSuccess:

print("video read error.")

sys.exit()

# 分割左右画面

imgL = np.split(img, 2, 1)[0]

imgR = np.split(img, 2, 1)[1]

prvs = imgR # 上一帧的右画面，用于运动矢量计算

# 每秒取5帧进行计算

for frameID in range(round(0), round(frameCount), round(frameRate/5)):

if frameID >= frameCount - 3:

frameID = frameCount - 3

cap.set(cv2.CAP\_PROP\_POS\_FRAMES, frameID)

isSuccess, img = cap.read()

if not isSuccess:

print("video read error.")

break

# 分割左右画面

imgL = np.split(img, 2, 1)[0]

imgR = np.split(img, 2, 1)[1]

next = imgR # 当前帧的右画面，用于运动矢量计算

hsv = getMotionVector(prvs, next)

hsv\_bak = hsv.copy()

# 计算深度图,disparity越大，景深越小，物体越近

disparity = getDepthMap(imgL, imgR)

framesCalculated += 1

comfort = 1

# 显示计算结果

print("frameID: ", frameID)

# 景深的平均值，偏大则意味着负视差（出屏感），可能不适

AVG\_depth = round(np.mean(disparity), 2)

print("AVG depth: ", AVG\_depth) # 大于-10时开始不适，权重为0.15

if AVG\_depth > -10:

comfort -= 0.15

# 运动矢量大小的平均值，可判断画面大致上是否稳定

AVG\_motionMag = round(np.mean(hsv[..., 2]), 2)

print("AVG motionMag: ", AVG\_motionMag) # 大于20时略不适，权重0.1

if AVG\_motionMag > 20:

comfort -= 0.1

# 景深的众数，由于景深基本不连续，众数意义不大

# print("Mode depth: ", stats.mode(disparity.reshape(-1))[0][0]) # 无明显阈值

# 运动矢量大小的众数，一般为0，若较大，说明画面中存在较大面积的快速运动，可能不适

Mode\_motionMag = stats.mode(hsv[..., 2].reshape(-1))[0][0]

# 大于0则不适，越大越不适，权重0.2，0到30归一化为0.1到0.15，大于30为0.2

print("Mode motionMag: ", Mode\_motionMag)

if Mode\_motionMag > 0:

if Mode\_motionMag > 30:

comfort -= 0.2

else:

comfort -= (Mode\_motionMag/600 + 0.1)

# 景深的标准差，若偏大说明景深范围较大，可能不适，但同时也是3D感更强的特征

STD\_depth = round(np.std(disparity), 2)

print("STD depth: ", STD\_depth) # 大于130时略不适，权重为0.15

if STD\_depth > 130:

comfort -= 0.15

# 运动矢量大小的标准差，若偏大说明各部分运动比较不一致，可能需要结合运动矢量的方向作进一步判断，若存在较复杂的运动形式，则可能不适

STD\_motionMag = round(np.std(hsv[..., 2]), 2)

print("STD motionMag: ", STD\_motionMag) # 大于20时略不适，权重为0.1

if STD\_motionMag > 20:

comfort -= 0.1

# 运动矢量方向的标准差，若偏大说明各部分运动比较不一致，可能需要结合运动矢量的大小作进一步判断，若存在较复杂的运动形式，则可能不适

# print("STD motionAng: ", round(np.std(hsv[...,0]),2)) # 无明显阈值

disparity\_Positive = disparity.copy()

disparity\_Positive[disparity\_Positive < 0] = 0

# 负视差的像素的所占比例，大于0.2时比较不适，权重0.15，0.2到0.4归一化为0.05到0.1，大于0.4为0.15

PCT\_disparity\_Positive = np.count\_nonzero(

disparity\_Positive)/disparity\_Positive.shape[0]/disparity\_Positive.shape[1]

print("close pixels percetage:", round(PCT\_disparity\_Positive, 3))

if PCT\_disparity\_Positive > 0.2:

if PCT\_disparity\_Positive > 0.4:

comfort -= 0.15

orgn\_cmft = -0.15

else:

comfort -= ((PCT\_disparity\_Positive - 0.2) / 4 + 0.05)

orgn\_cmft = -((PCT\_disparity\_Positive - 0.2) / 4 + 0.05)

if calcMod:

# 视差重映射并重新计算

# 实际并不写入文件，只估计此项提升值

trans = np.float32([[1,0,20],[0,1,0]])

imgR\_Mod = cv2.warpAffine(imgR, trans, imgR.shape[:2])

imgR\_Mod = imgR\_Mod.transpose((1,0,2))

disparity\_Mod = getDepthMap(imgL, imgR\_Mod)

disparity\_Positive = disparity\_Mod.copy()

disparity\_Positive[disparity\_Positive < 0] = 0

PCT\_disparity\_Positive = np.count\_nonzero(

disparity\_Positive)/disparity\_Positive.shape[0]/disparity\_Positive.shape[1]

print("Modified close pixels percetage:", round(PCT\_disparity\_Positive, 3))

if PCT\_disparity\_Positive > 0.2:

if PCT\_disparity\_Positive > 0.4:

mod\_cmft = -0.15

else:

mod\_cmft = -((PCT\_disparity\_Positive - 0.2) / 4 + 0.05)

else:

mod\_cmft = 0

comfort\_optimized = round(mod\_cmft - orgn\_cmft, 3)

if comfort\_optimized > 0:

framesOptimized += 1

framesComfortOptimized.append(comfort\_optimized)

print("comfort can be optimized by ", comfort\_optimized)

# 存在运动的像素点的视差平均值

movingPixels = hsv[..., 2]

movingPixels[movingPixels < 10] = 0 # 小于10的运动认为是静止

movingPixels[movingPixels > 0] = 1

movingDepth = np.multiply(disparity, movingPixels)

AVG\_movingDepth = round(np.sum(movingDepth) /

np.count\_nonzero(movingDepth))

print("AVG movingDepth: ", AVG\_movingDepth) # 大于5时不适，权重0.15

if AVG\_movingDepth > 5:

comfort -= 0.15

framesComfort.append(comfort)

comfort = round(comfort, 3)

print()

print("CurFrameComfort: ", comfort)

print("TotalComfort: ", round(sum(framesComfort)/framesCalculated, 2))

print()

# 当为demo模式时显示当前帧画面、运动矢量图和景深图

if isDemo:

# 显示当前帧

cv2.namedWindow("img", cv2.WINDOW\_NORMAL)

cv2.imshow('img', img)

cv2.waitKey(1)

# cv2.namedWindow("imgL", cv2.WINDOW\_NORMAL)

# cv2.imshow('imgL', imgL)

# cv2.namedWindow("imgR", cv2.WINDOW\_NORMAL)

# cv2.imshow('imgR', imgR)

# 显示当前帧的运动矢量的hsv表示

bgr = cv2.cvtColor(hsv\_bak, cv2.COLOR\_HSV2BGR) # hsv转为rgb用于显示

cv2.namedWindow("MotionVector", cv2.WINDOW\_NORMAL)

cv2.imshow("MotionVector", bgr)

cv2.waitKey(1)

# 显示当前帧的景深图

plt.title("DepthMap")

plt.imshow(disparity)

# plt.pause(0.1)

# input("press Enter to continue")

# 运动矢量的直方图，方便查看数值

# plt.title("MotionVector")

# plt.imshow(hsv[...,2])

# plt.show()

plt.pause(0.1)

doWrite = int(input("write？"))

if doWrite:

imgName = "./result/" + str(frameID) + ".png"

mvName = "./result/" + str(frameID) + "mv.png"

depName = "./result/" + str(frameID) + "depth.png"

cv2.imwrite(imgName, img)

cv2.imwrite(mvName, bgr)

plt.savefig(depName)

prvs = next # 当前帧覆盖上一帧，继续计算

print("TotalFrameCalculated: ", framesCalculated)

print("TotalComfort: ", round(sum(framesComfort)/framesCalculated, 2))

if calcMod:

print("estimated comfort optimization potential：", round(sum(framesComfortOptimized)/framesOptimized, 3))

print("success")