

云南大学本科教学

《计算机图形学实验》课程教学大纲

一、课程基本信息

课程名称	计算机图形学实验						
	Computer Graphics Test						
课程代码	INFO3L3005			课程性质	必修		
开课院部	信息学院			课程负责人	徐丹		
课程团队	徐丹, 钱文华, 袁国武, 吴昊						
授课学期	第3学期			学分/学时	1/27		
课内学时	27	理论学时		实验学时	27	实训(含上机)	
		实习		其他			
课外学时							
适用专业	计算机科学与技术						
授课语言	中文						
先修课程	数据结构; 离散数学; 程序设计; 线性代数; 高等数学						
后续课程	图像处理						
课程简介	<p>《计算机图形学实验》是为计算机科学与技术专业开设的一门应用型学科基础实验课, 是与专业必修课《计算机图形学》同时开设的一门实践课程。本课程以图形的显示与生成等应用为主, 既有对理论知识点的验证, 也包括针对具体问题的设计和研究。学生在理论学习的基础上, 通过编程工具实现和验证图形学基本算法, 如基本光栅图形生成算法、图形几何变换、图形裁剪算法、消隐算法、真实感图形生成等。具备独立思考、算法设计、算法综合, 以及总结分析能力和实践能力的培养目标, 为今后图像处理, 虚拟现实, 计算机视觉等课程的学习奠定基础。此外, 为学生今后从事与计算机图形学及其应用方面的研究工作打下坚实基础。</p>						

二、课程目标及对毕业要求指标点的支撑

序号	课程目标	支撑毕业要求指标点	毕业要求
1	课程目标 1: 理解计算机图形学算法基本原理, 正确读懂算法流程图, 具备运用算法流程图进行图形学算法设计、分析及验证能力。	3.3 具有系统分析与设计、实施解决方案、完成工程任务, 归纳、整理、分析实验结果, 撰写报告和参与交流的能力。(完成设计/开发, 能分析和撰写)	毕业要求 3: 设计/开发解决方案: 能够设计针对复杂工程问题的解决方案, 设计满足特定需求的软件系统、模块/组件, 并能够在设计环节中体现创新意识, 考虑社会、健康、安全、法律、文化以及环境等因素。
2	课程目标 2: 实现对基本图形学算法的灵活应用。针对生成图形需求, 完成核心算法选择和设计; 实现功能模块连接, 设计综合图形生成算法流程。		
3	课程目标 3: 能够分析图形处理数据, 通过对运行结果的分析做出正确的判断、得出实验结论; 能采用准确、规范的语言撰写实验报告。		
4	课程目标 4: 培养学生对计算机图形学显示与生成的系统开发平台的应用能力; 实现图形软件开发包的安装、调试, 运用编程软件实现图形的生成、显示。	4.3 能够根据实验方案构建实验系统, 进行数据分析与解释, 正确地采集、整理实验数据。(能进行实验)	毕业要求 4: 研究: 能够基于计算机科学与工程的技术和方法对复杂工程问题进行分析与研究, 包括设计实验、分析与解释数据、并通过信息综合得到合理有效的结论。
5	课程目标 5: 熟练、正确使用常用计算机图形学编程软件, 能熟练阅读编程代码。		
6	课程目标 6: 具备复杂图形算法的编写、编译、运行能力, 理解常用的计算机图形学算法调试、分析、问题排查方法。		
7	课程目标 7: 理解编程软件的具体操作和工作原理, 能够根据参数等实际需求, 完成程序模块的设计。	5.1 理解计算机科学与技术相关领域常用的工具和设备的原则和使用方法, 能够针对计算机领域复杂工程问题, 开发、选择与使用恰当的技术、资源、	毕业要求 5: 使用现代工具: 能够针对计算机相关领域复杂工程问题, 开发、选择与使用恰当的技术、资源、

8	<p>课程目标 8: 建立综合图形显示与生成的模块化设计思路。培养从需求出发, 逐步从发现问题、提出问题, 到解决问题的开发设计能力。能够根据需求对综合问题进行模块化、局部的功能模块划分。</p>	<p>现代工程工具、软/硬件开发工具、开源应用程序, 对复杂工程问题进行分析、计算预设。(理解工具原理和使用方法, 能选择和使用工具进行设计和开发)</p>	<p>现代工程工具、软/硬件开发工具、开源应用程序, 包括对计算机领域复杂问题的预测与模拟, 并能够理解开发技术和工具的局限性。</p>
---	--	--	--

三、教学内容及进度安排

序号	教学内容	学生学习预期成果	课时	教学方式	支撑课程目标
1	<p>内容: OpenGL、Visual studio 等开发平台使用介绍, 直线的生成算法。 重点: 实验代码编写规范, 实验要求, 计算机图形学算法常用实现工具。 难点: OpenGL 编程实验环境的使用</p>	<p>熟悉计算机图形学实验编程环境, 能够安装 OpenGL 编程软件, 读懂算法流程图, 根据算法流程图编写实验代码, 能够借助开发平台编程实现 DDA 算法、Bresenham 画线算法。</p>	2	<p>预习: 计算机图形学开发环境, OpenGL 编程软件安装方法。 课堂: 知识点讲解及演示。</p>	<p>课程目标 4: 培养学生对计算机图形学显示与生成的系统开发平台的应用能力; 实现图形软件开发包的安装、调试, 运用编程软件实现图形的生成、显示。 课程目标 5: 熟练、正确使用常用计算机图形学编程软件, 能熟练阅读编程代码。</p>
2	<p>内容: 圆与椭圆的生成算法实现及显示。 重点: Bresenham 画圆算法、椭圆的扫描转换算法实现。 难点: Bresenham 画圆算法、椭圆生成算法的优化方法。</p>	<p>熟悉 Bresenham 画圆算法、椭圆生成算法及优化方法, 能够读懂算法流程图, 根据算法流程图编程、调试程序, 实现圆和椭圆的生成。</p>	2	<p>预习: Bresenham 画圆算法、椭圆生成算法及优化方法。 课堂: 知识点讲解、演示、答疑。</p>	<p>课程目标 1: 理解计算机图形学算法基本原理, 正确读懂算法流程图, 具备运用算法流程图进行图形学算法设计、分析及验证能力。 课程目标 3: 能够分析图形数据, 通过对运行结果的分析做出正确的判断、得出实验结论; 能采用准确、规范的语言撰写实验报告。</p>

3	<p>内容：多边形区域填充算法实现。</p> <p>重点：多边形区域的扫描线填充算法，种子填充算法。</p> <p>难点：多边形区域扫描线填充顶点选取，种子填充算法加速算法。</p>	<p>熟悉多边形区域的扫描线填充算法，种子填充算法，能够读懂多边形区域填充算法流程图，根据算法流程图编程实现多边形区域的填充。</p>	2	<p>预习：扫描线填充算法、种子填充算法，多边形填充加速方法。</p> <p>课堂：知识点讲解、演示、答疑。</p>	<p>课程目标 3：能够分析图形数据，通过对运行结果的分析做出正确的判断、得出实验结论；能采用准确、规范的语言撰写实验报告。</p> <p>课程目标 4：培养学生对计算机图形学显示与生成的系统开发平台的应用能力；实现图形软件开发包的安装、调试，运用编程软件实现图形的生成、显示。</p>
4	<p>内容：二维图形的几何变换算法实现。</p> <p>重点：二维图形几何变换的矩阵操作，旋转及错切变换算法实现。</p> <p>难点：二维图形几何变换矩阵的复合运算。</p>	<p>熟悉二维图形几何变换矩阵运算，能够根据矩阵变换对二维图形进行编辑，能够完成二维图形的缩放、旋转、对称、平移、错切复合变换，能够根据要求分析、实现几何变换显示与生成。</p>	3	<p>预习：二维图形几何变换矩阵，缩放、旋转、对称、平移、错切变换矩阵及复合变换。</p> <p>课堂：知识点讲解、演示、答疑。</p>	<p>课程目标 2：实现对基本图形学算法的灵活应用。针对生成图形需求，完成核心算法选择和设计；实现功能模块连接，设计综合图形生成算法流程。</p> <p>课程目标 7：理解编程软件的具体操作和工作原理，能够根据参数等实际需求，完成程序模块的设计。</p>

5	<p>内容：二维多边形裁剪算法生成与实现。</p> <p>重点：Sutherland-Cohen 多边形裁剪算法编程实现。</p> <p>难点：梁有栋多边形裁剪算法编程实现。</p>	<p>熟悉二维多边形裁剪算法，熟悉 Sutherland-Cohen 多边形裁剪算法、梁有栋多边形裁剪算法。能够根据要求设计和改进裁剪算法，显示裁剪结果图形。</p>	3	<p>预习：二维多边形裁剪算法，Sutherland-Cohen 多边形裁剪算法、梁有栋多边形裁剪算法。</p> <p>课堂：知识点讲解、演示、答疑。</p>	<p>课程目标 1：理解计算机图形学算法基本原理，正确读懂算法流程图，具备运用算法流程图进行图形学算法设计、分析及验证能力。</p> <p>课程目标 2：实现对基本图形学算法的灵活应用。针对生成图形需求，完成核心算法选择和设计；实现功能模块连接，设计综合图形生成算法流程。</p>
6	<p>内容：三维图形的几何变换算法编程及实现。</p> <p>重点：三维图形几何变换的矩阵操作，三维旋转算法实现。</p> <p>难点：三维图形几何变换矩阵的复合运算编程实现。</p>	<p>熟悉三维图形几何变换矩阵运算，能够根据矩阵变换对三维图形进行编辑，能够完成三维图形的缩放、旋转、对称、平移、错切复合变换，能够根据要求分析、实现三维几何变换显示与生成。</p>	3	<p>预习：三维图形几何变换矩阵，缩放、旋转、对称、平移、错切变换矩阵及复合变换。</p> <p>课堂：知识点讲解、演示、答疑。</p>	<p>课程目标 2：实现对基本图形学算法的灵活应用。针对生成图形需求，完成核心算法选择和设计；实现功能模块连接，设计综合图形生成算法流程。</p> <p>课程目标 7：理解编程软件的具体操作和工作原理，能够根据参数等实际需求，完成程序模块的设计。</p>

7	<p>内容：曲线生成算法编程及实现。</p> <p>重点：三次样条曲线、Bezier 曲线、B 样条曲线生成算法理解及编程实现。</p> <p>难点：根据需求，完成不同曲线形状的编程实现。</p>	<p>熟悉曲线生成算法，三次样条曲线、Bezier 曲线、B 样条曲线生成算法；能够根据需求，设计参数和算法流程，完成不同曲线形状的生成与显示。</p>	3	<p>预习：常用曲线生成算法，如三次样条曲线、Bezier 曲线、B 样条曲线生成算法。</p> <p>课堂：知识点讲解、算法设计原理讲解、演示、答疑。</p>	<p>课程目标 2：实现对基本图形学算法的灵活应用。针对生成图形需求，完成核心算法选择和设计；实现功能模块连接，设计综合图形生成算法流程。</p> <p>课程目标 8：建立综合图形显示与生成的模块化设计思路。培养从需求出发，逐步从发现问题、提出问题，到解决问题的开发设计能力。能够根据需求对综合问题进行模块化、局部的功能模块划分。</p>
8	<p>内容：曲面生成算法编程及实现。</p> <p>重点：Bezier 曲面或 B 样条曲面生成与实现。</p> <p>难点：根据需求，完成不同曲面形状的编程实现。</p>	<p>熟悉 Bezier 曲面、B 样条曲面生成算法，能够对算法编程实现曲面生成与显示；能够根据需求，设计参数和算法流程，完成不同曲面形状的生成与显示。</p>	3	<p>预习：常用曲面生成算法，如 Bezier 曲面、B 样条曲面生成算法。</p> <p>课堂：知识点讲解、算法设计原理讲解、演示、答疑。</p>	<p>课程目标 6：具备复杂图形算法的编写、编译、运行能力，能够对常用的计算机图形学算法调试、分析问题排查。</p> <p>课程目标 8：建立综合图形显示与生成的模块化设计思路。培养从需求出发，逐步从发现问题、提出问题，到解决问题的开发设计能力。能够根据需求对综合问题进行模块化、局部的功能模块划分。</p>

9	<p>内容：消隐算法编程及实现。</p> <p>重点：Z-buffer 消隐算法、ray-trace 算法编程及实现。</p> <p>难点：不同场景下的消隐算法编程与实现。</p>	<p>熟悉消隐算法原理，Z-buffer 消隐算法、ray-trace 等消隐算法；根据场景需求，设计参数和算法流程，完成不同场景下的物体消隐生成与显示。</p>	3	<p>预习：常用消隐算法，如 Z-buffer 消隐算法、ray-trace 消隐生成算法。</p> <p>课堂：知识点讲解、算法设计原理讲解、演示、答疑。</p>	<p>课程目标 6：具备复杂图形算法的编写、编译、运行能力，能够对常用的计算机图形学算法调试、分析问题排查方法。</p> <p>课程目标 8：建立综合图形显示与生成的模块化设计思路。培养从需求出发，逐步从发现问题、提出问题，到解决问题的开发设计能力。能够根据需求对综合问题进行模块化、局部的功能模块划分。</p>
10	<p>内容：真实感图形生成与显示，光照明模型算法实现。</p> <p>重点：真实感场景生成算法流程，Phong 光照明模型、Gouraud 光照明模型算法。</p> <p>难点：根据需求，不同场景下的真实感场景算法设计、编程与实现。</p>	<p>熟悉真实感图形生成步骤、流程与算法，Phong 光照明模型、Gouraud 光照明模型生成算法；根据真实感场景需求，设计光照明模型算法步骤、流程，通过代码编程、编译、调试与运行，完成不同场景下的真实感图形生成与现实。</p>	3	<p>预习：真实感图形生成步骤、流程，Phong 光照明模型、Gouraud 光照明模型生成算法。</p> <p>课堂：知识点讲解、算法设计原理讲解、演示、答疑。</p>	<p>课程目标 7：理解编程软件的具体操作和工作原理，能够根据参数等实际需求，完成程序模块的设计。</p> <p>课程目标 8：建立综合图形显示与生成的模块化设计思路。培养从需求出发，逐步从发现问题、提出问题，到解决问题的开发设计能力。能够根据需求对综合问题进行模块化、局部的功能模块划分。</p>

注：“学生学习预期成果”是描述学生在学完本课程后应具有的能力，可以用认知、理解、应用、分析、综合、判断等描述预期成果达到的程度。

四、课程考核

序号	课程目标（支撑毕业要求指标点）	评价依据及成绩比例		成绩比例 (%)
		作业	考试	
1	课程目标 1：理解计算机图形学算法基本原理，正确读懂算法流程图，具备运用算法流程图进行图形学算法设计、分析及验证能力。（支撑毕业要求指标点 3.3）	6	4	10
2	课程目标 2：实现对基本图形学算法的灵活应用。针对生成图形需求，完成核心算法选择和设计；实现功能模块连接，设计综合图形生成算法流程。（支撑毕业要求指标点 3.3）	6	4	10
3	课程目标 3：能够分析图形数据，通过对运行结果的分析做出正确的判断、得出实验结论；能采用准确、规范的语言撰写实验报告。（支撑毕业要求指标点 3.3）	6	4	10
4	课程目标 4：培养学生对计算机图形学显示与生成的系统开发平台的应用能力；实现图形软件开发包的安装、调试，运用编程软件实现图形的生成、显示。（支撑毕业要求指标点 4.3）	6	4	10
5	课程目标 5：熟练、正确使用常用计算机图形学编程软件，能熟练阅读编程代码。（支撑毕业要求指标点 4.3）	6	4	10
6	课程目标 6：具备复杂图形算法的编写、编译、运行能力，能够对常用的计算机图形学算法进行调试、分析、问题排查。（支撑毕业要求指标点 4.4）	6	4	10
7	课程目标 7：理解编程软件的具体操作和工作原理，能够根据参数等实际需求，完成程序模块的设计。（支撑毕业要求指标点 5.1）	12	8	20
8	课程目标 8：建立综合图形显示与生成的模块化设计思路。培养从需求出发，逐步从发现问题、提出问题，到解决问题的开发设计能力。能够根据需求对综合问题进行模块化、局部的功能模块划分。（支撑毕业要求指标点 5.1）	12	8	20

合计	100	60	40
----	-----	----	----

五、教材及参考资料

(必含信息：教材名称，作者，出版社，出版年度，版次，书号)

- [1]刘倩，计算机图形学实验教程[M]，西南交通大学出版社，2013年。
- [2]徐文鹏等，计算机图形学基础(第2版) [M]，清华大学出版社，2014年。
- [3]孔令德，计算机图形学—基于MFC三维图形开发[M]，清华大学出版社，2014年。
- [4]D. P. Mukherjee，计算机图形学：算法与实现[M]，清华大学出版社，2012年。

大纲执笔人：钱文华

附录：各类考核评分标准表

实验评分标准

教学目标要求	评分标准				权重 (%)
	90-100	80-89	60-79	0-59	
课程目标 1：理解计算机图形学算法基本原理，正确读懂算法流程图，具备运用算法流程图进行图形学算法设计、分析及验证能力。（支撑毕业要求指标点 3.3）	熟练应用实验软件平台，根据算法流程图熟练运用编程语言编写代码，根据图形要求完成算法设计、分析及运行及测试。	应用实验软件平台，根据算法流程图运用编程语言编写代码，根据图形要求完成算法设计、分析及运行及测试。	基本应用实验软件平台，根据算法流程图运用编程语言基本能编写代码，根据图形要求完成算法设计、分析及运行及测试。	根据算法流程图，未能运用编程语言编写代码，根据图形要求未能完成算法设计、分析及运行及测试。	10
课程目标 2：实现对基本图形学算法的灵活应用。针对生成图形需求，完成核心算法选择和设计；实现功能模块连接，设计综合图形生成算法流程。（支撑毕业要求指标点 3.3）	熟练理解基本图形学算法，熟练设计综合图形生成核心算法和流程，熟练使用编程语言实现图形的生成和显示。	理解基本图形学算法，能设计综合图形生成核心算法和流程，能使用编程语言实现图形的生成和显示。	基本理解基本图形学算法，基本能设计综合图形生成核心算法和流程，基本能使用编程语言实现图形的生成和显示。	基本理解基本图形学算法，未能设计综合图形生成核心算法和流程，未能使用编程语言实现图形的生成和显示。	10
课程目标 3：能够分析图形数据，通过对运行结果的分析做出正确的判断、得出实验结论；能采用准确、规范的语言撰写实验报告。（支撑毕业要求指标点 3.3）	熟练应用编程环境，熟练运用编程环境工具查找代码错误，得出结论；实验报告撰写准确、规范。	应用编程环境，能运用编程环境工具查找代码错误，得出结论；实验报告撰写较准确、规范。	基本应用编程环境，基本能运用编程环境工具查找代码错误，基本得出结论；实验报告撰写准确性和规范性一般。	基本应用编程环境，未能运用编程环境工具查找代码错误，未能得出结论；实验报告撰写准确性和规范性较差。	10

<p>课程目标4：培养学生对计算机图形学显示与生成的系统开发平台的应用能力；实现图形软件开发包的安装、调试，运用编程软件实现图形的生成、显示。 (支撑毕业要求指标点4.3)</p>	<p>熟练应用计算机图形学显示与生成的系统开发平台和软件，熟练应用图形软件开发包的安装、调试，熟练运用编程软件实现图形的生成、显示。</p>	<p>能够应用计算机图形学显示与生成的系统开发平台和软件，应用图形软件开发包的安装、调试，运用编程软件实现图形的生成、显示。</p>	<p>基本能够应用计算机图形学显示与生成的系统开发平台和软件，基本应用图形软件开发包的安装、调试，基本能够运用编程软件实现图形的生成、显示。</p>	<p>未能应用计算机图形学显示与生成的系统开发平台和软件，未能应用图形软件开发包的安装、调试，未能运用编程软件实现图形的生成、显示。</p>	<p>10</p>
<p>课程目标5：熟练、正确使用常用计算机图形学编程软件，能熟练阅读编程代码。 (支撑毕业要求指标点4.3)</p>	<p>熟练、正确运用图形软件编写图形生成代码，熟练并正确阅读编程代码。</p>	<p>能够运用图形软件编写图形生成代码，能够正确阅读编程代码。</p>	<p>基本能够运用图形软件编写图形生成代码，基本能够正确阅读编程代码。</p>	<p>未能运用图形软件编写图形生成代码，未能正确阅读编程代码。</p>	<p>10</p>
<p>课程目标6：具备复杂图形算法的编写、编译、运行能力，能对常用的计算机图形学算法调试、分析、问题排查。 (支撑毕业要求指标点4.4)</p>	<p>熟练、正确运用图形软件对复杂图形算法进行编写、编译、运行，熟练对常用的计算机图形学算法调试、分析、问题排查。</p>	<p>能够运用图形软件对复杂图形算法进行编写、编译、运行，对常用的计算机图形学算法调试、分析、问题排查。</p>	<p>基本能够运用图形软件对复杂图形算法进行编写、编译、运行，基本能对常用的计算机图形学算法调试、分析、问题排查。</p>	<p>未能运用图形软件对复杂图形算法进行编写、编译、运行，未能对常用的计算机图形学算法调试、分析、问题排查。</p>	<p>10</p>
<p>课程目标7：理解编程软件的具体操作和工作原理，能够根据参数等实际需求，完成程序模块的设计。 (支撑毕业要求指标点5.1)</p>	<p>熟练理解编程软件的具体操作和工作原理，能够根据参数等实际需求，熟练完成程序模块的设计。</p>	<p>理解编程软件的具体操作和工作原理，能够根据参数等实际需求，完成程序模块的设计。</p>	<p>基本理解编程软件的具体操作和工作原理，能够根据参数等实际需求，基本能够完成程序模块的设计。</p>	<p>未能理解编程软件的具体操作和工作原理，能够根据参数等实际需求，未能够完成程序模块的设计。</p>	<p>20</p>

<p>课程目标 8：建立综合图形显示与生成的模块化设计思路。培养从需求出发，逐步从发现问题、提出问题，到解决问题的开发设计能力。能够根据需求对综合问题进行模块化、局部的功能模块划分。（支撑毕业要求指标点 5.1）</p>	<p>熟练建立综合图形显示与生成的模块化设计思路；熟练发现问题、提出问题，解决问题；熟练根据需求对综合问题进行模块化、局部的功能模块划分。</p>	<p>能够建立综合图形显示与生成的模块化设计思路；能够发现问题、提出问题，解决问题；根据需求对综合问题进行模块化、局部的功能模块划分。</p>	<p>基本能够建立综合图形显示与生成的模块化设计思路；基本能够发现问题、提出问题，解决问题；基本能够根据需求对综合问题进行模块化、局部的功能模块划分。</p>	<p>未能建立综合图形显示与生成的模块化设计思路；未能够发现问题、提出问题，解决问题；未能根据需求对综合问题进行模块化、局部的功能模块划分。</p>	<p>20</p>
--	---	---	---	--	-----------