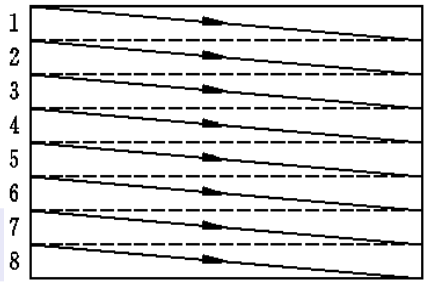
多媒体技术

回顾

- 卷积神经网络
  - 神经网络
    - 目标函数缩小：梯度下降法
    - 后向传播算法
  - 卷积神经网络
    - 卷积层
    - 池化层
    - 全连接层

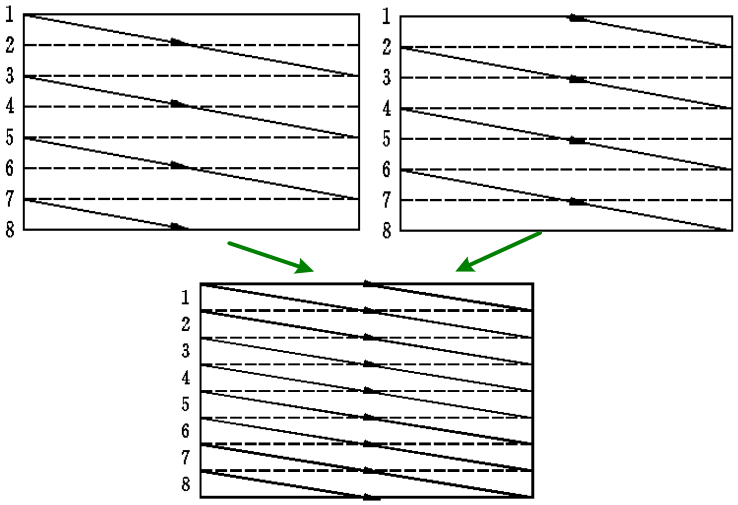
2.3.3 视觉媒体数字化

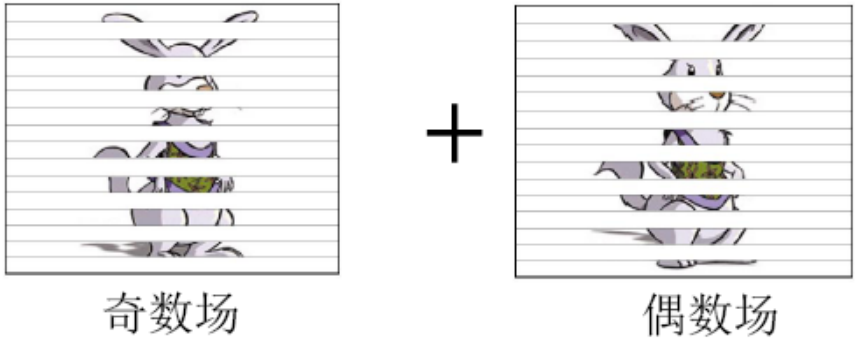
- 视频
  - 光栅扫描原理
    - 视频摄像机将图像转换为电信号，电信号是一维的，但图像是二维的，将二维图像转成为一维电信号是由光栅扫描的方法实现的。
    - 快速的扫描线从顶部开始，一行一行地向下扫描，直至显示器的最底部，然后再返回顶部的起点，重新开始扫描。
    - 这个过程产生的一个有序的图像信号集合，就组成了电视显示中的一幅图像，在此称为帧。连续不断的图像序列就形成了动态视频图像。
  - 逐行扫描/非隔行扫描/顺序扫描
    - 电子束从显示屏的左上角一行接一行地扫到右下角，在显示屏上扫一遍就显示一幅完整的图像。

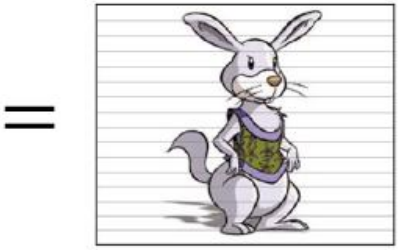


2.3.3 视觉媒体数字化

- 视频
  - 隔行扫描
    - 电子束扫完第1行后从第3行开始扫，接着扫第5行、7行、…，一直扫到最后一行的中间；
    - 一帧图像由两部分组成：由奇数行组成的奇数场；由偶数行组成的偶数场。







2.3.3 视觉媒体数字化

- 视频
  - 术语
    - 场频：每秒钟扫描的场数
      - 目的是使在屏幕上显示的图像看起来不会让人感觉到在闪烁，以及减低电网频率的干扰
    - 帧频：每秒扫描的帧数
      - 用“帧每秒(frames per second，fps)”做单位
      - PAL制和NTSC制电视的帧频分别为25 fps和30 fps。
    - 行频：每秒钟扫描的行数
    - 宽高比：扫描行的长度与在图像垂直方向上的所有扫描行所跨过的距离之比
      - 目前电视中的宽高比为4：3，新型电视的宽高比为16：9，有些电影系统的宽高比为2：1。

2.3.3 视觉媒体数字化

- 视频
  - 模拟视频
    - 由连续的模拟信号组成的视频图像
    - 电视制式
      - PAL、NTSC、SECAM
  - 数字视频
    - 以数字形式记录的视频

2.3.3 视觉媒体数字化

- 视频
  - PAL
    - 德国的彩色电视广播标准
      - 德国、英国等一些西欧国家，以及中国、朝鲜等国家采用
      - 由于使用的一些参数细节不同，有PAL-G、PAL-I和PAL-D等制式，我国大陆采用PAL-D制式
    - 主要特性
      - 图像的宽高比为4:3
      - 625条扫描线，隔行扫描，25帧每秒
      - 视像带宽至少4MHz
      - 使用YUV颜色模型
      - 声音使用调频制(FM)

2.3.3 视觉媒体数字化

- 视频
  - **NTSC**
    - 美国国家电视系统委员会(NTSC)制定的彩色电视广播标准
      - 美国、加拿大等大部分西半球国家以及日本、韩国、菲律宾和台湾地区采用
    - 主要特性
      - 图像宽高比4:3
      - 525条扫描线，隔行扫描，30帧每秒
      - 视像带宽为4.2MHz
      - 使用YIQ信号
      - 声音用调频制(FM)

2.3.3 视觉媒体数字化

- 视频
  - **SECAM**
    - 法国的彩色电视广播标准
      - 法国、俄罗斯、东欧和中东等约多个地区和国家使用
    - 主要特性
      - 图像的宽高比为**4:3**
      - **625**条扫描线，隔行扫描，**25**帧每秒
      - 视像带宽最高为**6MHz**
      - 使用**YUV**颜色模型

2.3.3 视觉媒体数字化

- 视频
  - 模拟视频： 线性编辑

信息存储的物理位置和接受信息的顺序是完全一致的，即录在前面的信息存储在磁带开头，录在后面的信息存储在磁带末尾。基于磁带的电子编辑系统，通常称为线性编辑系统。

局限：由于是复制磁带，成品带质量有所降低；
设备多，系统较复杂；
线性的编辑、查找、修改困难

2.3.3 视觉媒体数字化

- 视频
  - 数字视频

方法一：模拟视频转数字视频

处理图像和视频时，首先把连续的图像函数f(x,y)进行空间和幅值的离散化处理:

–采样

- 对连续图像彩色函数f (x,y)，沿着x方向等间隔采样，采样点数为M。
- 沿着y方向以等间隔采样，采样点数为N，得到一个M*N的离散样本阵列。
- 采样的结果即图像分辨率。

2.3.3 视觉媒体数字化

- 视频
  - 数字视频

方法一：模拟视频转数字视频

–量化

- 对每个离散点—像素的灰度或颜色样本进行数字化处理;
- 采样后的样本值的范围分为有限的多个区域，某一区域内的所有样本值用同一值表示，用有限的离散的数值量来代替无限的连续模拟量;
- 如：2级灰度构成二值图像，只有黑白之分，没有灰度层次。
- 如何量化彩色幅度，取决于所选的颜色空间表示。

对视频按照时间进行数字化所得到的图像序列，就构成了数字视频序列。

2.3.3 视觉媒体数字化

- 视频
  - 数字视频

方法二：数字化采集设备

高清摄像机、照相机、手机、摄像头等

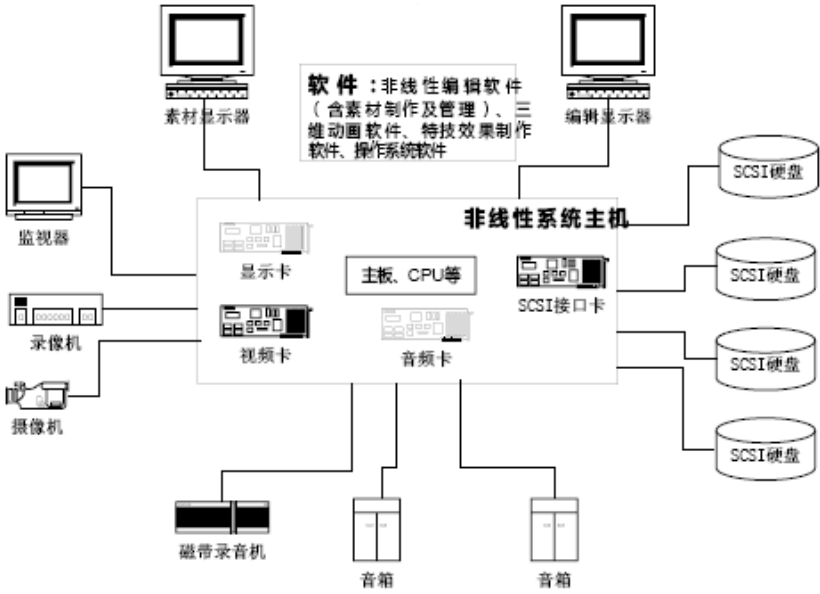
方法三：计算机图形绘制

好莱坞的电影

2.3.3 视觉媒体数字化

- 视频
  - 数字视频：非线性编辑

主要包括：计算机主机及显示器、视音频处理卡、硬盘阵列箱、接线盒、录像机、监视器、音箱、网络接口、软件，另外还有许多扩展设备如特技卡等。



非线性编辑软件：提供对原素材任意部分的随机存取、修改和处理

Adobe Premiere Pro

会声会影

2.3.3 视觉媒体数字化

- 视频文件格式
  - AVI格式
    - 常见的影音格式，图像质量比较高。
  - MOV格式
    - 由苹果公司开发的一种专业影音格式。
  - MPEG格式
    - 扩展名可以是MPG、MPE、MP4、MIV、M2V等。
  - DVDRip格式
    - 用MPEG-4来进行视频压缩，用MP3或AC3等压缩音频，同时结合字幕播放软件来外挂字幕。

2.3.3 视觉媒体数字化

- 视频文件格式
  - RM格式
    - Networks公司，根据不同的网络传输速率制定不同的压缩比率；在线播放。
  - RMVB格式
    - 打破原先RM格式那种平均压缩采样的方式，在保证平均压缩比的基础上合理利用比特率资源。
  - WMV格式
    - Microsoft公司推出的一种采用独立编码方式且可以直接在网上实时观看视频节目的文件压缩格式。

2.3.3 视觉媒体数字化

- **OpenCV**
  - 计算机视觉领域应用最广泛的开源工具包，基于C/C++，支持Linux/Windows/MacOS/Android/iOS，提供了Python、MATLAB和Java等语言的接口。
  - 特点
    - 拥有包括500多个C函数的跨平台的中高层API
    - 跨平台：Linux、Windows
    - 免费：无论对非商业应用和商业应用
    - 速度快
    - 使用方便

https://sourceforge.net/projects/opencvlibrary/

2.3.3 视觉媒体数字化

- 图像数据操作（内存分配与释放，图像复制、设定和转换）
- 图像/视频的输入输出（支持文件或摄像头的输入，图像/视频文件的输出）
- 矩阵/向量数据操作及线性代数运算（矩阵乘积、矩阵方程求解、特征值、奇异值分解）
- 多种动态数据结构（链表、队列、数据集、树、图）
- 基本图像处理（去噪、边缘检测、角点检测、采样与插值、色彩变换、形态学处理、直方图、图像金字塔结构）
- 结构分析（连通域/分支、轮廓处理、距离转换、图像矩、模板匹配、霍夫变换、多项式逼近、曲线拟合、椭圆拟合、狄劳尼三角化）
- 摄像头定标（寻找和跟踪定标模式、参数定标、基本矩阵估计、单应矩阵估计、立体视觉匹配）
- 运动分析（光流、动作分割、目标跟踪）
- 目标识别（特征方法、HMM模型）
- 基本的GUI（显示图像/视频、键盘/鼠标操作、滑动条）
- 图像标注（直线、曲线、多边形、文本标注）

2.3.3 视觉媒体数字化

- **OpenCV**
  - 例子
    - **Python下的OpenCV**
    - 从视频中截取帧，遍历一个指定文件夹下的所有视频，按照指定的间隔进行截屏保存

```python
import cv2
import os
import sys
# 第一个输入参数是包含视频片段的路径
input_path = sys.argv[1]
# 第二是输入参数是设定每隔多少帧截取一帧
frame_interval = int(sys.argv[2])
# 列出文件夹下所有的视频文件
filenames = os.listdir(input_path)
# 获取文件夹名称
video_prefix = input_path.split(os.sep)[-1]
# 建立一个新的文件夹，名称为原文件夹名称后加上_frames
frame_path = '{}_frames'.format(input_path)
if not os.path.exists(frame_path):
    os.mkdir(frame_path)
# 初始化一个VideoCapture对象
cap = cv2.VideoCapture()
# 遍历所有文件
for filename in filenames:
    filepath = os.sep.join([input_path, filename])
    # VideoCapture::open 函数可以从文件获取视频
    cap.open(filepath)
    # 获取视频帧数
    n_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
    for i in range(n_frames):
        ret, frame = cap.read()
        # 每隔frame_interval帧进行一次截屏操作
        if i % frame_interval == 0:
            imagename = '{}_{}_{:0>6d}.jpg'.format(video_prefix, filename.split('.')[0], i)
            imagepath = os.sep.join([frame_path, imagename])
            print('exported {}!'.format(imagepath))
            cv2.imwrite(imagepath, frame)
# 执行结束释放资源
cap.release()
```

2.3.3 视觉媒体数字化

- **OpenCV**
  - 视频的使用和操作：类和函数
    - 打开摄像头

```
CvCapture* cvCaptureFromCAM(camera_id=0);
```

    - 打开文件

```
CvCapture* cvCaptureFromFile(videofile_path);
CvCapture* cvCaptureFromAVI("inflie.avi");
```

    - 捕捉某一帧

```
cvGrabFrame(capture) //抓住一帧，为快速遍历视频帧
IplImage* img=cvRetrieveImage(capture);//把Grab的帧取出
IplImage* cvQueryFrame(capture);
```

    - 释放捕捉源

```
cvReleaseCapture(&capture);
```

2.3.3 视觉媒体数字化

- **OpenCV**
  - 视频的使用和操作：类和函数
    - 保存视频文件

typedef struct CvVideoWriter;

CvVideoWriter* cvCreateVideoWriter(const char* filename, int **fourcc**, double fps, CvSize frame_size, int is_color=1); 非0：彩色

int cvWriteFrame(CvVideoWriter* writer, const IplImage* image); 写入一帧到一个视频文件中

void cvReleaseVideoWriter(CvVideoWriter** writer);

如果不释放，可能会导致写视频失败

    - 获取/设置视频帧信息

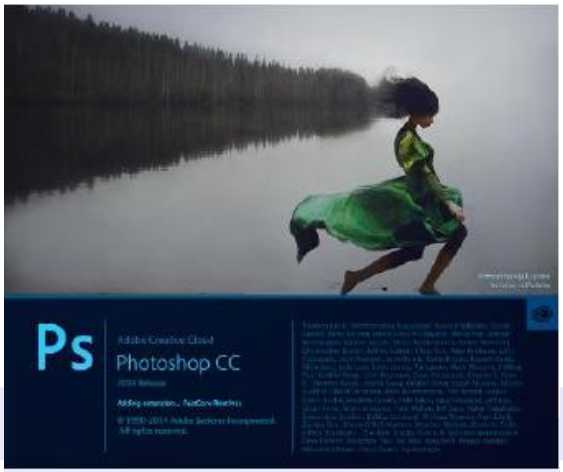
cvGetCaptureProperty(capture, property_id);

cvSetCaptureProperty(capture, property_id, value);

CV_CAP_PROP_POS_MSEC - video capture timestamp
CV_CAP_PROP_POS_FRAMES - 0-based index of the frame
CV_CAP_PROP_POS_AVI_RATIO - relative position of video(0-start, 1-end)
CV_CAP_PROP_FRAME_WIDTH - width of frames in the video stream
CV_CAP_PROP_**FRAME_HEIGHT** - height of frames in the video stream
CV_CAP_PROP_**FPS** - frame rate
CV_CAP_PROP_**FOURCC** - 4-character code of codec CV_CAP_PROP_FRAME_COUNT - number of frames in video file

2.3.3 视觉媒体数字化

- 动画
  - 计算机动画方法
    - 2D帧动画
      - 步骤1：创建每一帧的图像；
      - 步骤2：在时间轴的每个帧上逐帧绘制不同内容，连续播放成动画
    - 优点：表现细腻的动画
    - 缺点：工作量大，输出的文件量更大



补间动画：只创建关键帧，由软件自动生成过渡帧

软件：Photoshop CC(Creative Cloud)

2.3.3 视觉媒体数字化

- 动画
  - 计算机动画方法
    - 传统3D计算机动画
      - 前期准备 -> 建模制帧 -> 后期处理
    - 局限
      - 在三维动画制作软件中人工建模或调整虚拟角色动作的工作方式建模时间长、成本高、对建模人员技术要求高
    - 解决方案
      - 动作捕捉技术：在运动物体的关键部分设置跟踪器，由动作捕捉系统捕捉跟踪器位置，再经过计算机处理后向用户提供可以在动画制作中应用的数据。
      - 当数据被计算机识别后，动画师可以在计算机产生的镜头中调整、控制运动的物体。

2.3.3 视觉媒体数字化

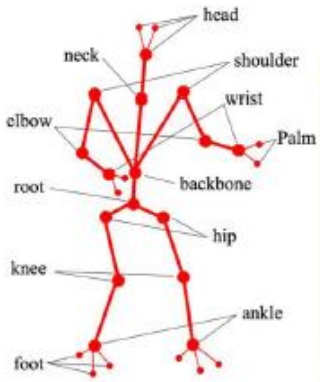
- 运动捕捉系统
  - 传感器
    - 它将向动作捕捉系统提供运动物体运动的位置信息
  - 信号捕捉设备
    - 负责位置信号的捕捉
    - 对于光学动作捕捉系统则是指高分辨率红外摄像机

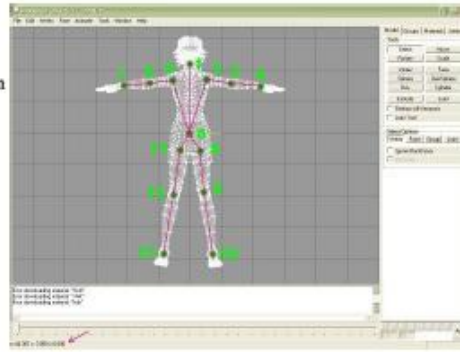
多个高速相机从不同角度对目标特征点监视和跟踪，理论上对于空间任一点，只要同时为两部相机所见，就可以确定这一时刻在空间的位置，当相机以足够高的速率连续拍摄时，从图像序列中可以得到该点的运动轨迹

2.3.3 视觉媒体数字化

- 运动捕捉系统
  - 数据传输设备
    - 将大量的运动数据从信号捕捉设备快速准确的传输到计算机系统进行处理

  - 数据处理设备
    - 经过动作捕捉系统捕捉到的数据需要修正、处理还要与三维模型绑定才能完成计算机动画制作的工作





Autodesk MotionBuilder（动作捕捉） / Maya（手工）

2.3.3 视觉媒体数字化

- 动画文件格式
  - GIF格式
    - Graphics Interchange Format，图形交换格式。80年代出现。
    - 压缩比高，利于网络传输。
    - 增加了渐显方式，先看到图像的大致轮廓，然后随着传输过程的继续逐步看清图像中的全部细节。
    - Internet上大量采用的彩色动画文件多为这种格式的文件。

2.3.3 视觉媒体数字化

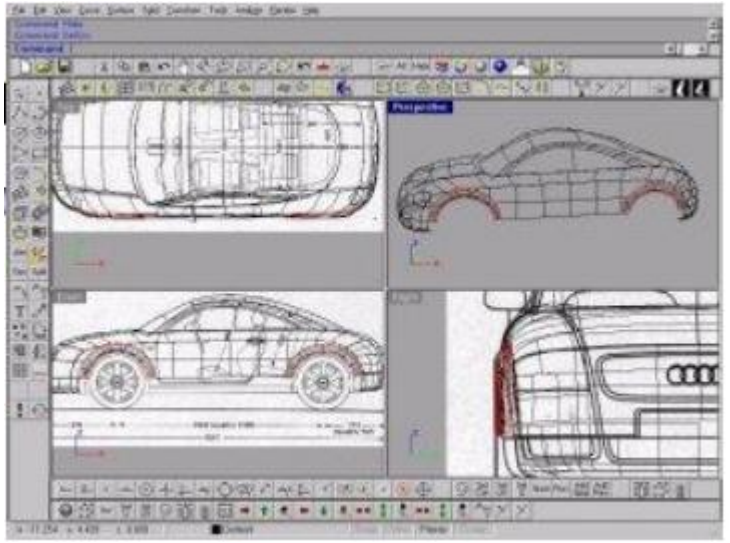
- 动画文件格式
  - FLIC（FLI/FLC）格式
    - 由Autodesk公司研制，其产品Autodesk Animator、Animator Pro、3D Studio等均使用这种文件格式。
  - SWF格式
    - 是MicroMedia公司软件Flash的动画格式。
    - 能够用比较小的体积表现丰富的多媒体形式，可以与HTML达到水乳交融的境界。
    - 利用矢量技术制作，画面放大时清晰流畅，不影响质量。

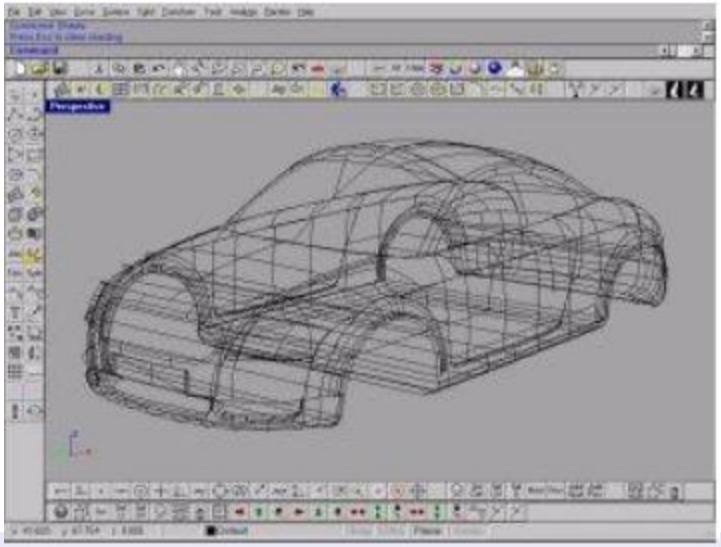
2.3.3 视觉媒体数字化

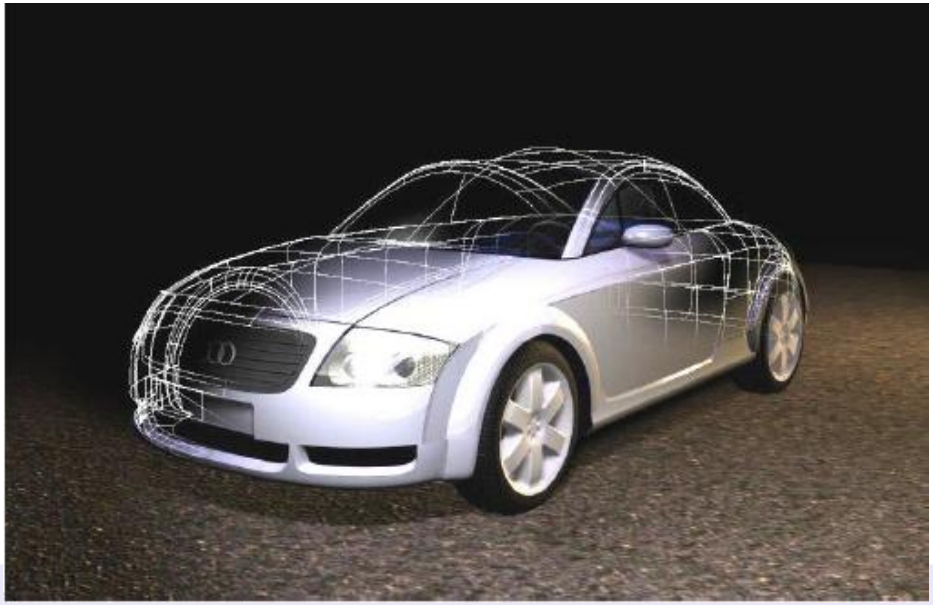
- 图形
  - 分类
    - 维度：2D和3D
    - 处理技术
      - 线条信息表示，如工程图、等高线地图、曲面的线框图等
      - 明暗图，通常所说的真实感图形
  - 表示方法
    - 参数表示
      - 方程、样条等
      - 精确表示和计算
    - 枚举出图形中所有顶点（几何信息）和顶点之间的连接关系（拓扑信息）

2.3.3 视觉媒体数字化

- 图形
  - 参数表示
    - CAD/CAM广泛的应用于建筑设计、机械产品设计，飞机、汽车等外形设计







2.3.3 视觉媒体数字化

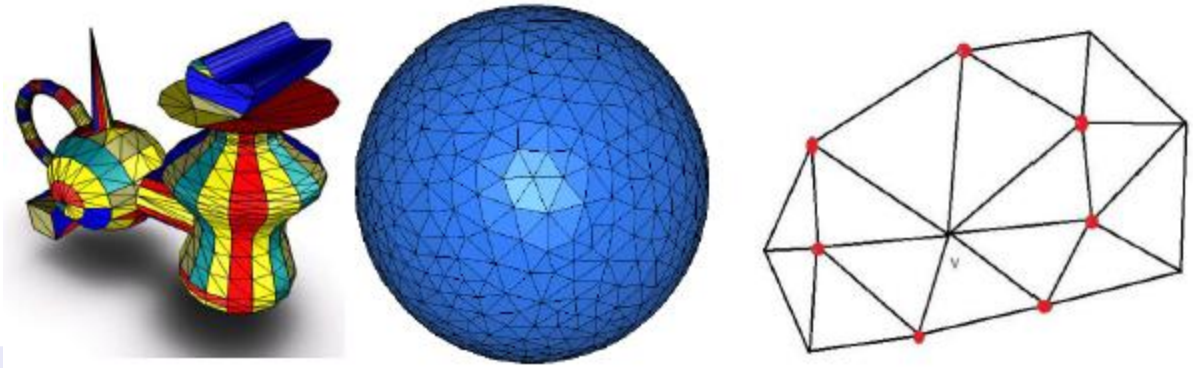
- 图形
  - 枚举出图形中所有顶点（几何信息）和顶点之间的连接关系（拓扑信息）

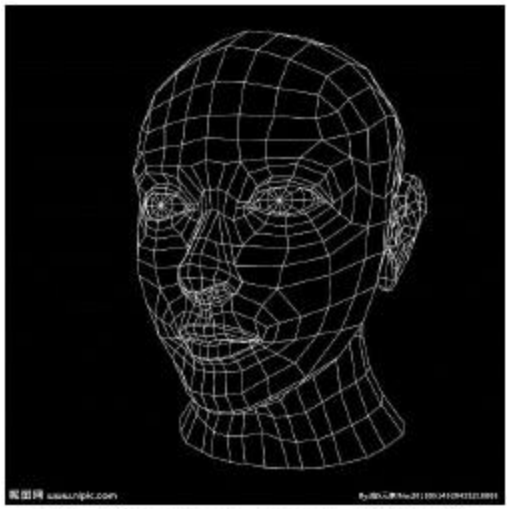
```
pDC->MoveTo(10, 10);
pDC->LineTo(100,100);
```

➢2D图形：顶点坐标（x, y）

➢3D图形：顶点坐标（x, y, z）

例如：三角网格模型，用离散的三角网格结构去模拟现实中物体的形状

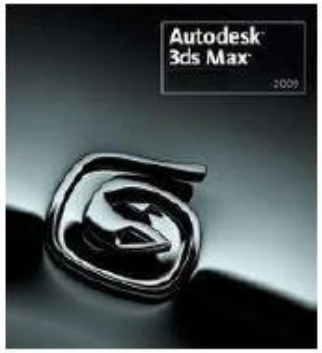


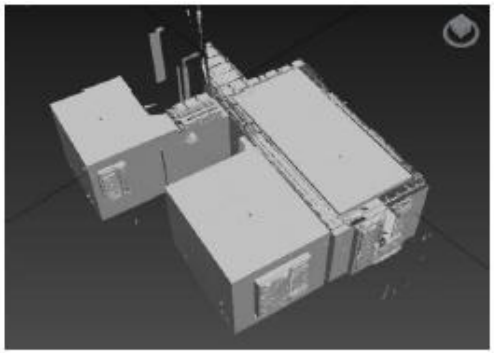


多边形网格模型

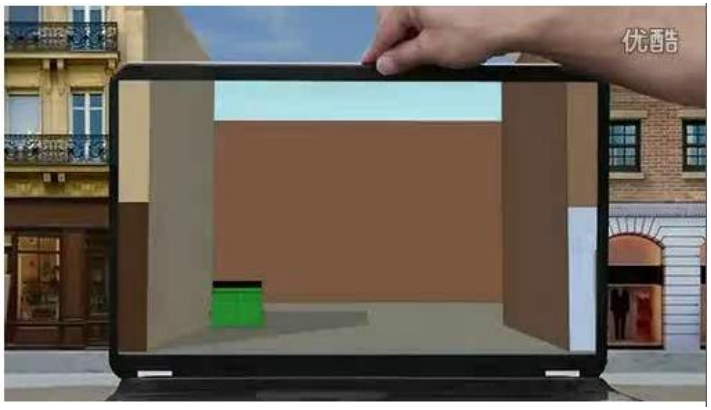
2.3.3 视觉媒体数字化

- 图形建模方法
  - 1、三维软件建模





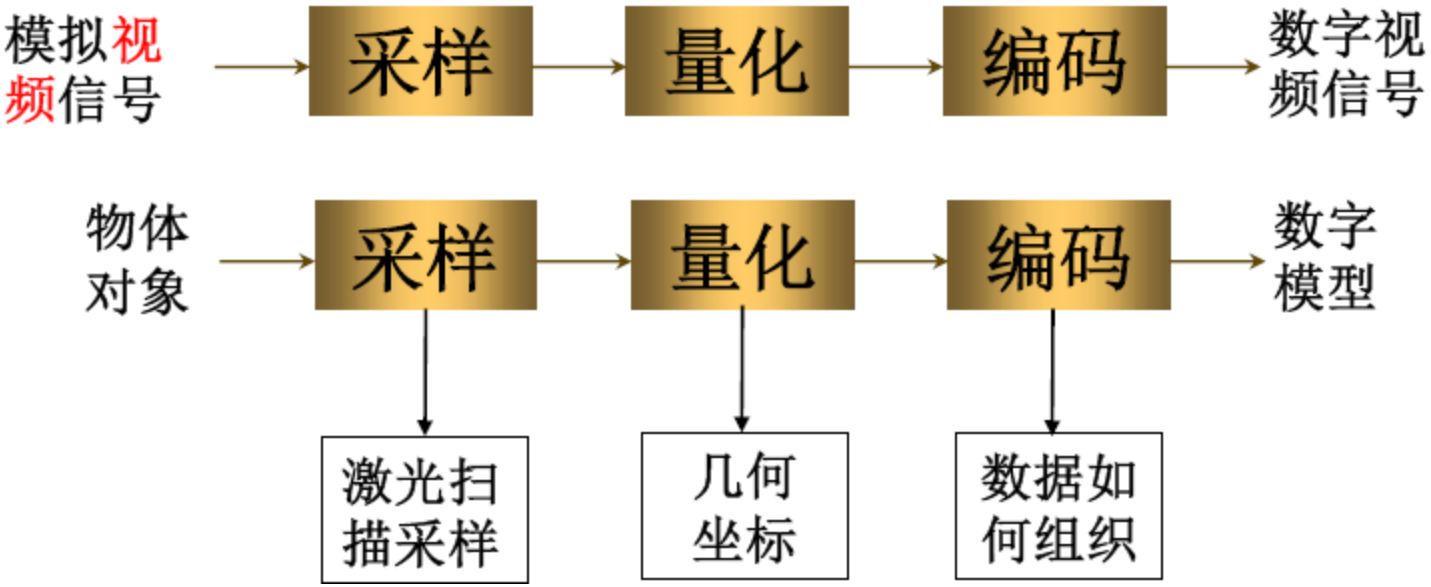
3Ds Max软件



SketchUp软件

2.3.3 视觉媒体数字化

- 图形建模方法
  - 2、三维激光扫描技术



基本原理：由扫描设备向目标物体表面发射可见光（激光），然后再通过扫描设备的CCD镜头捕捉反射回来的激光，最后采用脉冲测距或三角测距等方法计算物体表面顶点的三维坐标。

2.3.3 视觉媒体数字化

- 图形文件格式
  - obj文件
    - 3D模型文件格式，由Wavefront公司为3D建模和动画软件“Advanced Visualizer”开发的一种标准

编码规范：顶点几何、纹理、法线，以及顶点间的连接关系

（a）顶点坐标:**v** x y z

```
v -0.5  -0.5   0.5
v 0.5   -0.5   0.5
v -0.5   0.5   0.5
v 0.5    0.5   0.5
v -0.5   0.5  -0.5
v 0.5    0.5  -0.5
v -0.5  -0.5  -0.5
v 0.5   -0.5  -0.5
```

（b）顶点纹理坐标:**vt** u v

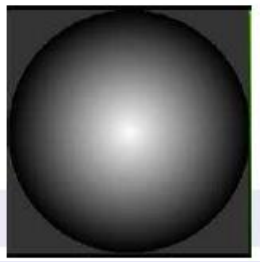
```
vt 0.000000 0.000000
vt 1.000000 0.000000
vt 0.000000 1.000000
vt 1.000000 1.000000
vt 0.000000 2.000000
vt 1.000000 2.000000
vt 0.000000 3.000000
vt 1.000000 3.000000
```

（c）顶点法线坐标:**vn** x y z

```
vn 0.000000 0.000000 1.000000
vn 0.000000 0.000000 1.000000
vn 0.000000 0.000000 1.000000
vn 0.000000 0.000000 1.000000
vn 0.000000 1.000000 0.000000
vn 0.000000 1.000000 0.000000
vn 0.000000 1.000000 0.000000
vn 0.000000 1.000000 0.000000
```

（d）拓扑关系:**f** 顶点索引、纹理索引、法线索引

```
f 1/1/1 2/2/2 4/4/3 3/3/4
f 3/3/5 4/4/6 6/6/7 5/5/8
f 5/5/9 6/6/10 8/8/11 7/7/12
```



2.3.3 视觉媒体数字化

- **OpenGL**
  - 一个功能强大的开放图形库，是一组绘图命令的API集合
  - 利用这些API能够方便的描述二维和三维几何物体，并控制这些物体按某种方式绘制到显示缓冲区中
  - API提供了物体描述、平移、旋转、缩放、光照、纹理、材质、像素、位图、文字、交互以及提高显示性能等方面的功能，基本涵盖了开发二、三维图形程序所需的各个方面

设置视点：

```
gluLookAt(eyex,eyey,eyez,aimx, aimy, aimz, upx, upy, upz )
```

正交平行投影 `glOrtho(-1.0,1.0,-1.0,1.0,-1.0,1.0);`

透视投影：`gluPerspective(θ , (GLfloat) w/(GLfloat) h, near, far);`

2.3.4 视觉媒体的三维立体显示

1、立体显示原理

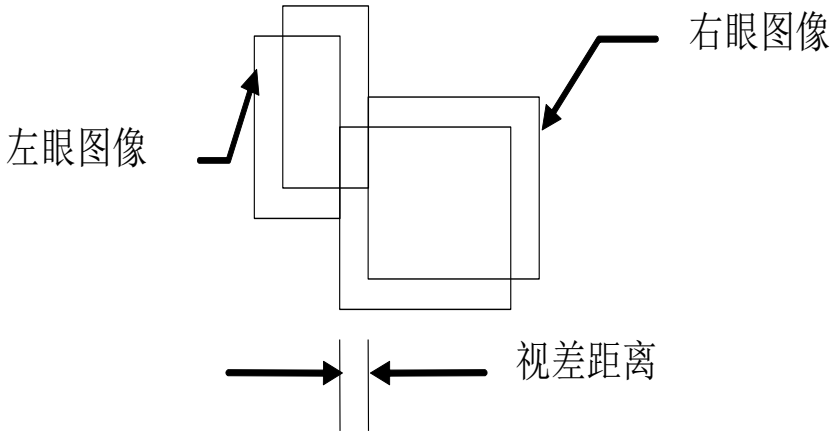
- 三维显示实际上具有两个含义:
  - 一个是指物体的三维图像在平面上的显示，特别是三维图形的显示，这是图形学的重点。
  - 另一个含义是指所显示的图像确确实实是立体的，是“浮”在空间中的，和我们所看的立体电影一样。这是我们讨论的重点。

2.3.4 视觉媒体的三维立体显示

2、立体视觉形成因素

- 生理因素：视差
  - 眼睛看到的是两幅图像：左眼一幅、右眼一幅，这两幅图像被称为立体图像对。
  - 视差（parallax）是投影到人眼视网膜上图像上两点间的水平距离。

优势：周围物体间的距离、深度、凸凹等都能分辨出来，这样成的像就是立体的像



视差的种类大致分为四种：零视差、正视差、负视差和发散视差。

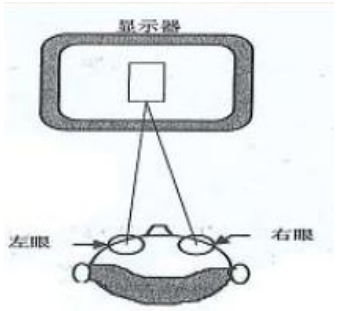
2.3.4 视觉媒体的三维立体显示

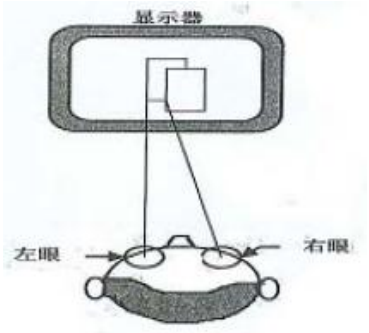
零视差：当显示的左右眼图像之间没有缝隙时，视差为零，就称为零视差。

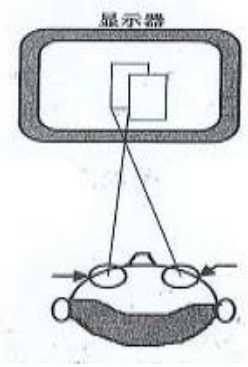
正视差：一旦计算机显示的立体图像对的视差大于0，我们就可以看到深度。正视差差值大于0且小于等于人眼之间的距离。

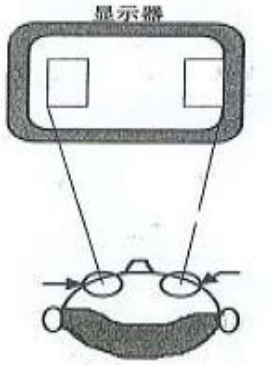
负视差：当两眼的目光交叉时，就会产生负视差。这时，所观察的对象将会浮现在两眼与显示器之间的空间中。

发散视差：视差的值比两眼之间的距离还要大。









2.3.4 视觉媒体的三维立体显示

立体图像对的生成

（1）时分法

–时分法技术原理是在发送端用两台摄像机，模拟人的左、右两眼，产生一对视差图像信号，编码成一路信号进行传送。

–接收端解码成两路信号。第一次刷新时播放左眼画面，快门眼镜滤掉右眼画面，下一次刷新时播放右眼画面，快门眼镜滤掉左眼画面。

–将两套画面以极快的速度切换，在人眼视觉暂留特性的作用下合成了连续的画面在屏幕上同时显示两幅图像。

2.3.4 视觉媒体的三维立体显示

主流技术:

在屏幕上以两倍的场频交互的显示左眼和右眼的影像，而快门眼镜则会动态的屏蔽使用者的左眼和右眼，利用人眼的视觉暂留机制，两眼影像叠加后产生双眼视差。

通常要达到120Hz，左右眼各60Hz

优点：成像优异，显示屏无改动

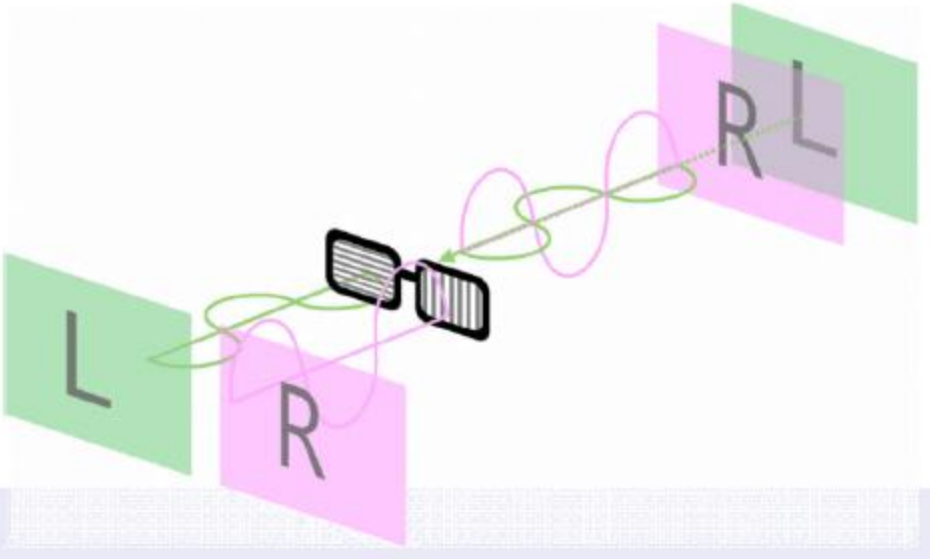
缺点：眼镜较贵且较重

2.3.4 视觉媒体的三维立体显示

（2）光分法

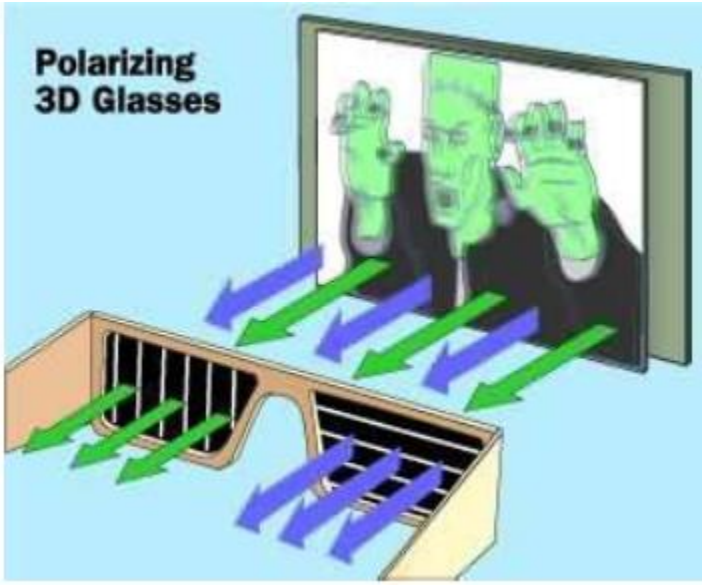
–利用光波震动的方向性来做左右眼影像的区分。

–利用偏光片（通过如百叶窗般排列的矽晶体涂料薄膜（偏光膜））过滤原本朝向不同方向震动的光线（偏振光），挡住与偏光膜方向垂直的光线，只让与偏光膜方向相同的光线通过，从而产生视差。



2.3.4 视觉媒体的三维立体显示

立体眼镜的左眼和右眼分别装上横偏振片和竖偏振片，横偏振光只能通过横偏振片，竖偏振光只能通过竖偏振片。这样就保证了左边相机拍摄的东西只能进入左眼，右边相机拍摄到的东西只能进入右眼，产生了立体图像对。



优点：宽视域、大景深，成像质量优异，眼镜价格较便宜；

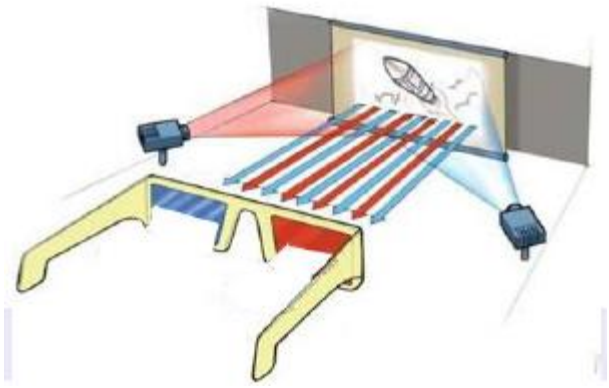
缺点：头部倾斜时无法过滤掉另一方向的光；成像画面偏暗

2.3.4 视觉媒体的三维立体显示

（3）色分法

–色分法是把两幅具有适当视差的同一景物分别用补色制成两幅图像（如红色和蓝色），再把这两幅图像组合在一起。

–红蓝立体形成的基本原理就是将左右图像用红蓝两种补色同时显示出来并用相应的补色观察，即分别用红色绘出左眼图像，用蓝色绘出右眼图像。



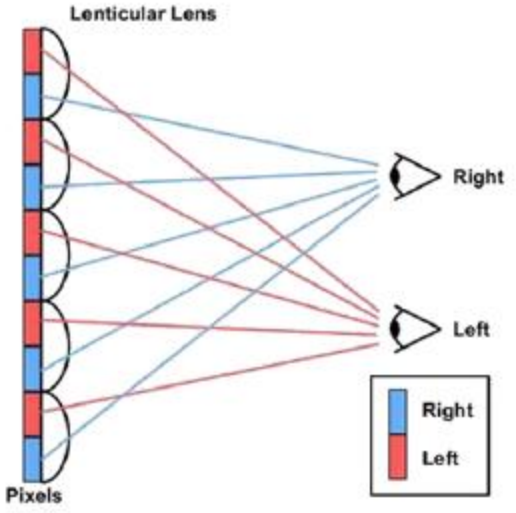
优点：实现3维简易，对视场和景深无严格的限制。

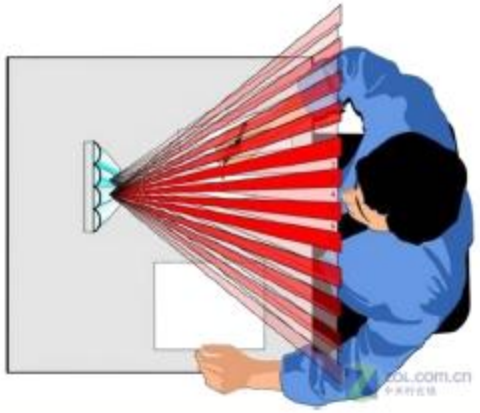
缺点：偏色、立体效果差但易引起眼部的疲劳。

2.3.4 视觉媒体的三维立体显示

未来的技术：裸眼3D显示

- 在屏幕表面上有一道道紧密排列的半圆形透镜，利用这些柱镜的折射角度不同，将左右眼的图像分开
- 优点：不需要眼镜，使用方便，制作成本相对较低，亮度较高
- 缺点：分辨率损失较严重，2D兼容性不好、观察视域有限





2.3.4 视觉媒体的三维立体显示

- 心理因素：增加心理感觉

朦胧透视

- 随着距离增大，景物的对比度降低从而产生深度感觉，这就是空气的透视作用。
- 当观看远处弥漫着烟雾的风景照片时，会有一种强烈的深度感觉。

遮挡和阴影

- 两个物体的前后位置关系会提供相应的深度线索
- 根据该物体颜色的变化也可以获得该物体深度变化的信息暗示

2.3.4 视觉媒体的三维立体显示

- 应用：3D电影

电影《阿凡达》。IMax摄像机装有两个相同的镜头，镜头间距正好是人眼间距65mm，两幅图像被分别记录到两条同步运行的空底片上。

2.4 触觉媒体技术

2.4.1 触觉媒体概述

- 概念
  - 皮肤可以感觉环境的温度、湿度，也可感觉压力，身体可以感觉振动、运动、旋转等，这些都是触觉在起作用，都可以作为传递信息的媒体。
  - 触觉媒体就是环境媒体，它描述了该环境中的一切特征和参数。
  - 人体在信息交流过程中起的作用最大的是人的头部、手部和整体躯干。
  - 与外界环境的触觉交互主要包括位置跟踪、力量反馈等方面。

2.4.2 简单指点设备与技术

- 指点的任务
  - 选择，定位，定向，路径，数量，操作
- 指点设备
  - 直接指点设备包括：光笔、触摸屏及输入笔等。
  - 间接指点设备包括：鼠标、跟踪球、控制杆和图形板。
  - 一些新型的指点设备，例如脚用鼠标器、视线跟踪器、凝视检测控制器等。

2.4.3 位置跟踪

- 手指动作测量和数字化
  - 对手部的跟踪采用一种称为数据手套的工具。
  - 对手指的测量主要采用在手套的手指部位装上能够测量手指弯曲、移动的检测器。
  - 对手指动作的测量和数字化，实际上更关心的是手指的相对位置。
  - 拇指和食指、食指和其他手指等的相对动作包含了许多含义。识别这些相对的动作，采用的方法就是建立手指动作模式库。

2.4.3 位置跟踪

- 空间位置跟踪
  - 在数据手套上有一个定位的装置，这是用于进行手部位置跟踪的。
  - 手部的位置是指手部在空中的相对位置，所以还需要一个坐标原点。事实上，无论是数据手套还是头部的位置跟踪其原理都是一样的。

2.4.4 力反馈与触觉反馈

- 这与位置跟踪正好相反，是由系统向参与者反馈力和运动的信息，如触觉刺激（物体的表面纹理等）、反作用力（推门的门重感觉）、运动感觉（摇晃、振动等）及温度、湿度等环境信息。
  - 力反馈：重力、阻力等的感知；
  - 触觉反馈：区别出不同物体的质感和纹理结构；
  - 热觉反馈：温度的反应。

第三章 多媒体数据压缩

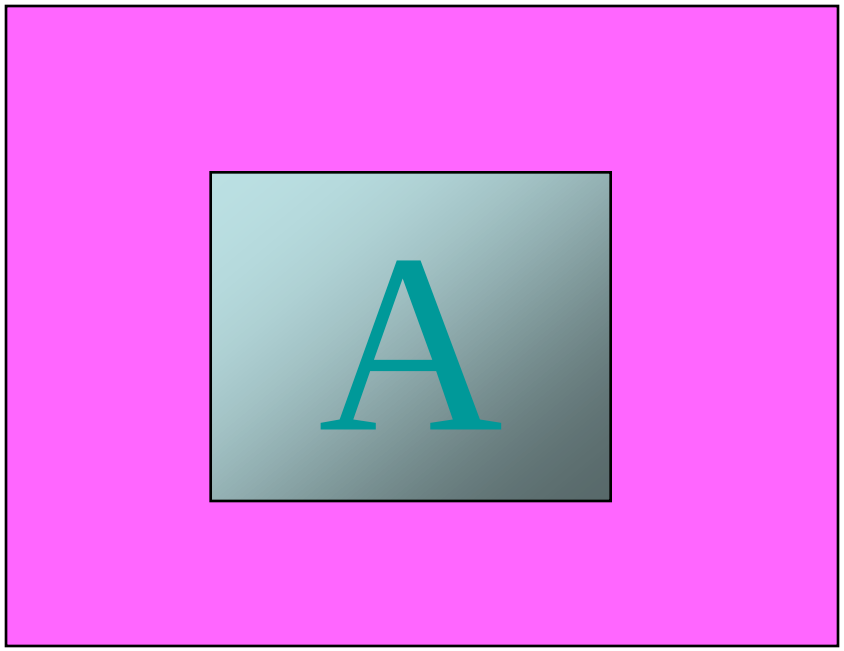
3.1 多媒体数据压缩技术概述

3.1.1 数据冗余的类型

- 数据：信息+冗余度
  - 空间冗余
    - 图像数据经常存在的一种冗余。
    - 像素点之间具有很强的相关性，在同一幅图像中，规则物体和规则背景的表面物理特性具有相关性，这些相关性的光成像结构在数字化图像中表现为数据冗余。

3.1.1 数据冗余的类型

- 数据：信息+冗余度
  - 空间冗余

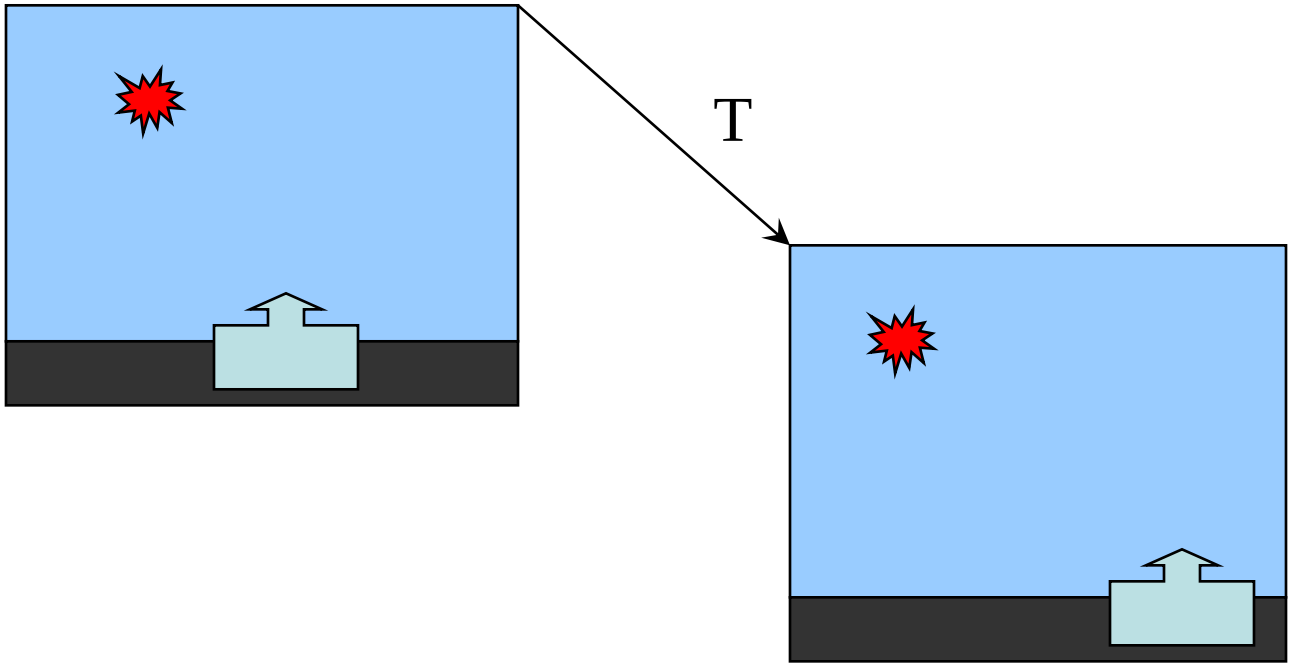


3.1.1 数据冗余的类型

- 数据：信息+冗余度
  - **时间冗余**
    - 活动图像的两个连续帧之间的冗余。
    - 两幅相邻的图像，后一幅图像与前一幅图像之间有较大的相关性，相邻帧往往包含相同的背景和移动物体，只不过移动物体所在空间位置有所不同。
    - 一帧图像中的某物体或场景可以由其它帧图像中的物体或场景重构出来。音频的前后样值之间也同样有时间冗余。

3.1.1 数据冗余的类型

- 数据：信息+冗余度
  - 时间冗余



3.1.1 数据冗余的类型

- 数据：信息+冗余度
  - **信息熵冗余**
    - 信息熵：一组数据所携带的信息量。
    - 如果图像中平均每个像素使用的比特数大于该图像的信息熵，存在冗余。

3.1.1 数据冗余的类型

- **决策量**
  - 在有限数目的互斥事件集合中，决策量是事件数的对数值。
  - 在数学上表示为

    $H_0=\log(n)$　　其中，n是事件数
  - 决策量的单位由对数的底数决定
    - **Sh (Shannon):** 用于以2为底的对数
    - **Nat (natural unit):** 用于以e为底的对数
    - **Hart (hartley):** 用于以10为底的对数

3.1.1 数据冗余的类型

- **信息量**
  - 具有确定概率事件的信息的定量度量
  - 在数学上定义为

$$I(x) = \log_2[1/\ p(x)] = -\log_2\ p(x)$$

其中，　$p(x)$ 是事件出现的概率

3.1.1 数据冗余的类型

- **信息量**
  - 举例：假设X={a,b,c}是由3个事件构成的集合，p(a)=0.5，p(b)=0.25，p(c)=0.25分别是事件a, b和c出现的概率，这些事件的信息量分别为，

    $I(a)=\log_2(1/0.50)=1$ sh

    $I(b)=\log_2(1/0.25)=2$ sh

    $I(c)=\log_2(1/0.25)=2$ sh
  - 一个等概率事件的集合，每个事件的信息量等于该集合的决策量。

3.1.1 数据冗余的类型

- **熵**
  - 按照香农(Shannon)的理论，在有限的互斥和联合穷举事件的集合中，熵为事件的信息量的平均值，也称事件的平均信息量。
  - 用数学表示为

3.1.1 数据冗余的类型

$$H(X)=\sum_{i=1}^{n}h(x_i)=\sum_{i=1}^{n}p(x_i)I(x_i)=-\sum_{i=1}^{n}p(x_i)\log_2 p(x_i)$$

其中，(1) $X=\{x_1,\cdots,x_n\}$是事件 $x_i(i=1,2,\cdots,n)$的集合，并满足 $\sum_{i=1}^{n}p(x_i)=1$；

(2) $I(x_i)=-\log_2 p(x_i)$表示某个事件 x_i 的信息量，其中 $p(x_i)$ 为事件 x_i 出现的概率，$0<p(x_i)\le 1$；$h(x_i)=-p(x_i)\log_2 p(x_i)$ 表示事件 x_i 的熵。

例如，$X=\{a,b,c\}$是由 3 个符号构成的集合，符号 a,b 和 c 出现的概率分别为 $p(a)=0.5$，$p(b)=0.25$，$p(c)=0.25$，那么符号 a,b 和 c 的熵分别等于 0.5，0.5，0.5，这个集合的熵为，

$$H(X)=p(a)I(a)+p(b)I(b)+p(c)I(c)=1.5\ \text{(Sh)}$$

3.1.1 数据冗余的类型

- **数据的冗余量**

在信息论中，数据的冗余量(R)定义为决策量(H_0)超过熵(H)的量，数学上表示为

$$R = H_0 - H$$

例如，令 $\{a,b,c\}$ 为 3 个事件构成的数据集，它们出现的概率分别为 $p(a)=0.5$，$P(b)=0.25$，$p(c)=0.25$，这个数据集的冗余量则为，

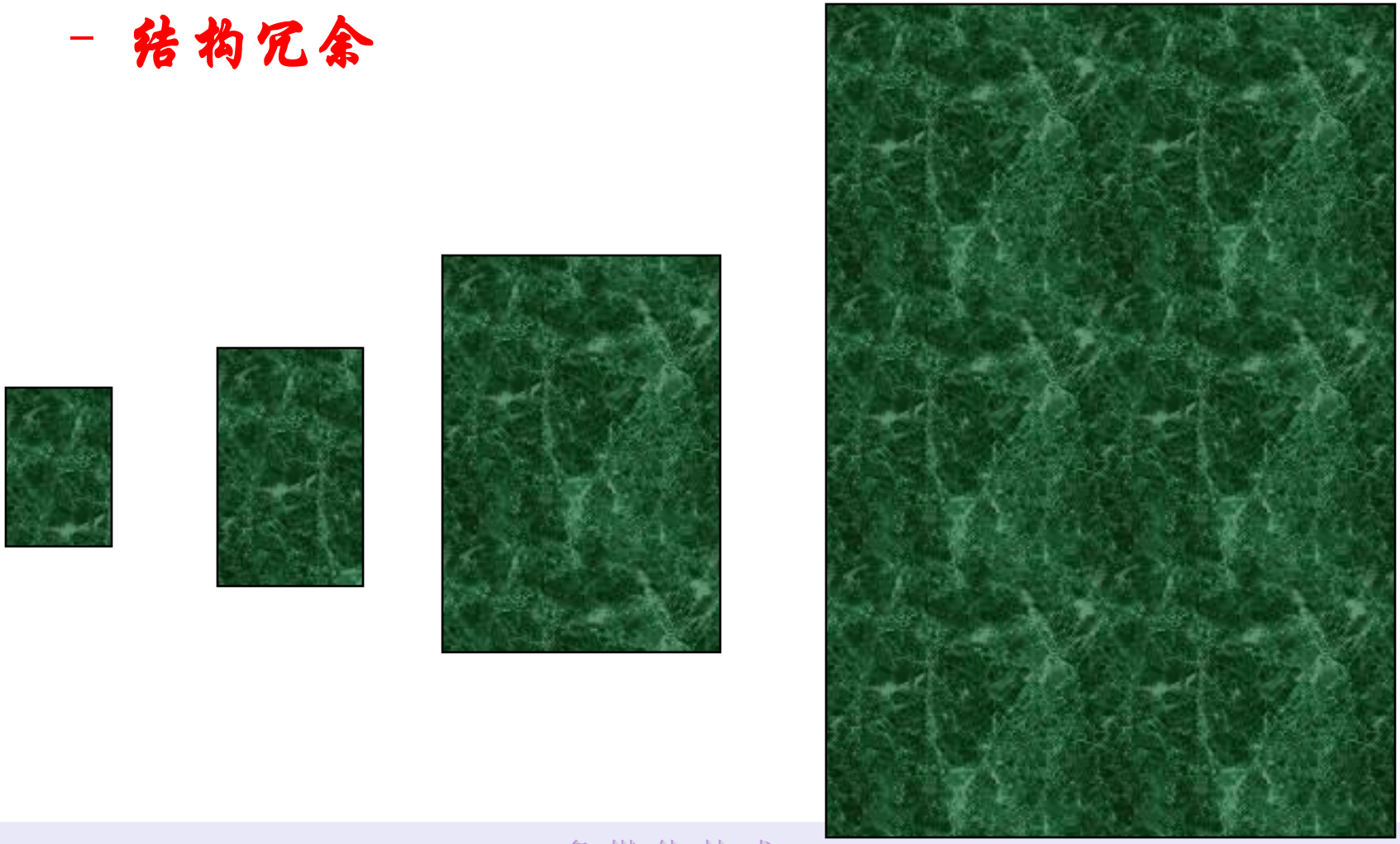
$$R = H_0 - H = \log_2 3 - \left[-\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)\right] = 1.58 - 1.50 = 0.08 \quad (\text{Sh})$$

3.1.1 数据冗余的类型

- 数据：信息+冗余度
  - **结构冗余**
    - 图像上的区域上存在非常强的纹理结构，在图像纹理区，图像的像素值存在明显的分布模式，如方格式的地板图案。

3.1.1 数据冗余的类型

- 数据：信息+冗余度
  - 结构冗余



3.1.1 数据冗余的类型

- 数据：信息+冗余度
  - 知识冗余
    - 许多图像的理解与某些基础知识有相当大的相关性。如人脸的图像有相同结构。
    - 是模型编码主要利用的特性。

3.1.1 数据冗余的类型

- 数据：信息+冗余度
  - 视觉冗余
    - 人类的视觉系统对图像场的敏感性是非均匀的和非线性的。
      - 人类视觉系统对亮度变化敏感，对色度变化相对不敏感；
      - 随着亮度的增加，视觉系统对量化误差的敏感度降低；
      - 对物体边缘敏感，内部区域相对不敏感；
      - 人类视觉系统对一般的分辨能力约为2^6灰度等级，而一般图像量化采用2^8灰度等级。

3.1.1 数据冗余的类型

- 数据：信息+冗余度
  - 听觉冗余
    - 人的听觉具有掩蔽效应。
    - 人耳对不同频率的声音的敏感性是不同的，并不能察觉所有频率的变化。

作业5

- 从AVI文件（bsd.avi）中得到视频流，对视频流进行Laplace边缘检测，并输出结果。

注：

1、Laplace边缘检测

$$f(x,y)=\frac{\partial^2 f}{\partial x^2}+\frac{\partial^2 f}{\partial y^2}$$

$$\begin{bmatrix}0 & 1 & 0\\ 1 & -4 & 1\\ 0 & 1 & 0\end{bmatrix}$$

2、主要思想

- 获取视频（cvCaptureFromAVI）
- 循环处理每帧（ cvQueryFrame ）
  1. 将多通道数组分割为多个单通道数组（cvCvtPixToPlane）
  2. 在每个通道内进行Laplace变换（ cvLaplace），计算图像的边缘信息
  3. 恢复多通道图像（ cvCvtPlaneToPix ）并显示

opencv + visual studio，2周之后将源代码（和运行说明）发给助教