

图像检索 与 *SURF*: *Speeded-Up Robust* *Features* 快速稳健特征点

Together with AVIC

中航光电

SURF适用的领域

- ▶ 1、图像检索
- ▶ 2、图像配准
- ▶ 3、目标识别

特征提取

- ▶ 1、低层，视觉特征
 - ▶ 颜色、纹理、形状等
- ▶ 2、高层，语义内容
 - ▶ 图像中包含的物体
 - ▶ 物体之间的语义关系

特征提取



SURF: 介绍

- **SURF** (Speeded-Up Robust Feature)是一种高稳定性的局部特征点检测和描述算法，可用来检测两幅图像之间的具有相同特征的点。



SURF: 可用领域

- ▶ 图像检索
- ▶ 目标识别
- ▶ 图像配准
- ▶ 3D重建

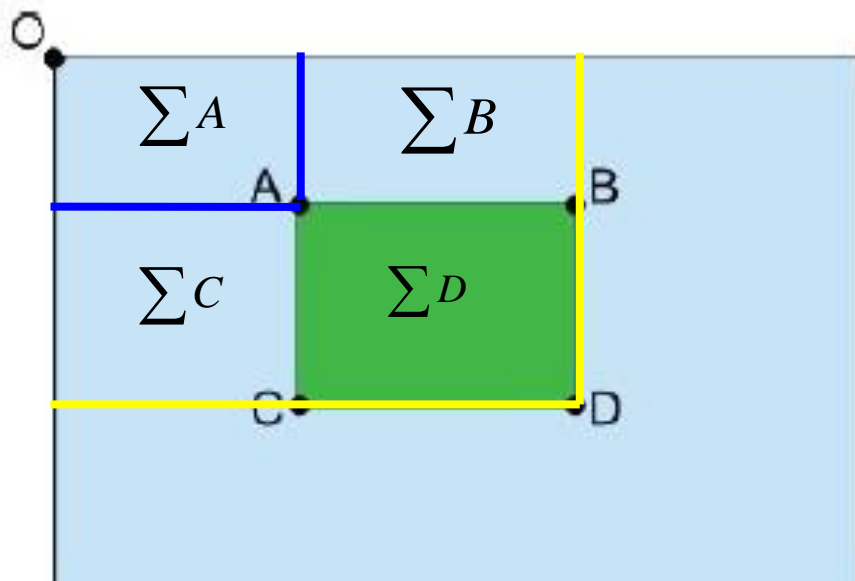
SURF: 作者目标

- ▶ 快速兴趣点检测
- ▶ 高鉴别度的兴趣点描述
- ▶ 快速描述子匹配
- ▶ 对常用图像变换的不变性
 - ▶ 图像旋转
 - ▶ 尺度变化
 - ▶ 亮度变化
 - ▶ 视角轻微变化

SURF: 兴趣点检测 - 积分图

- 积分图中任意一点 $I_{\Sigma}(i,j)$ 的值，等于原图像左上角区域像素的和：

$$I_{\Sigma}(i, j) = \sum_{i' < i, j' < j} p(i', j')$$



$$S_{ABCD} = D - B - C + A$$

SURF: 兴趣点检测 - 尺度变换

- ▶ 图像为 $I_0(X)$, $X = (x, y)$;
- ▶ 图像 $I(X, t)$ 为初始图像 $I_0(X)$ 经过尺度参数为 t ($t > 0$) 的尺度变换所得到的图像;
- ▶ 则算子 $T_t: I_0(X) \rightarrow I(X, t)$ 定义为尺度空间算子, 算子族 $\{T_t\}_{t>0}$ 就定义为尺度空间。

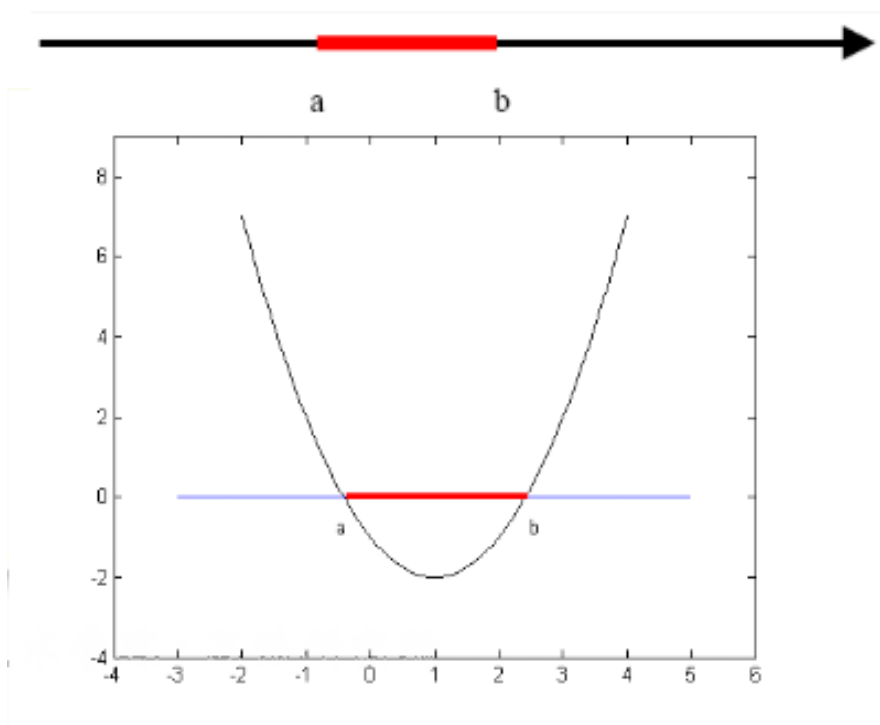


SURF: 兴趣点检测 - 高斯函数

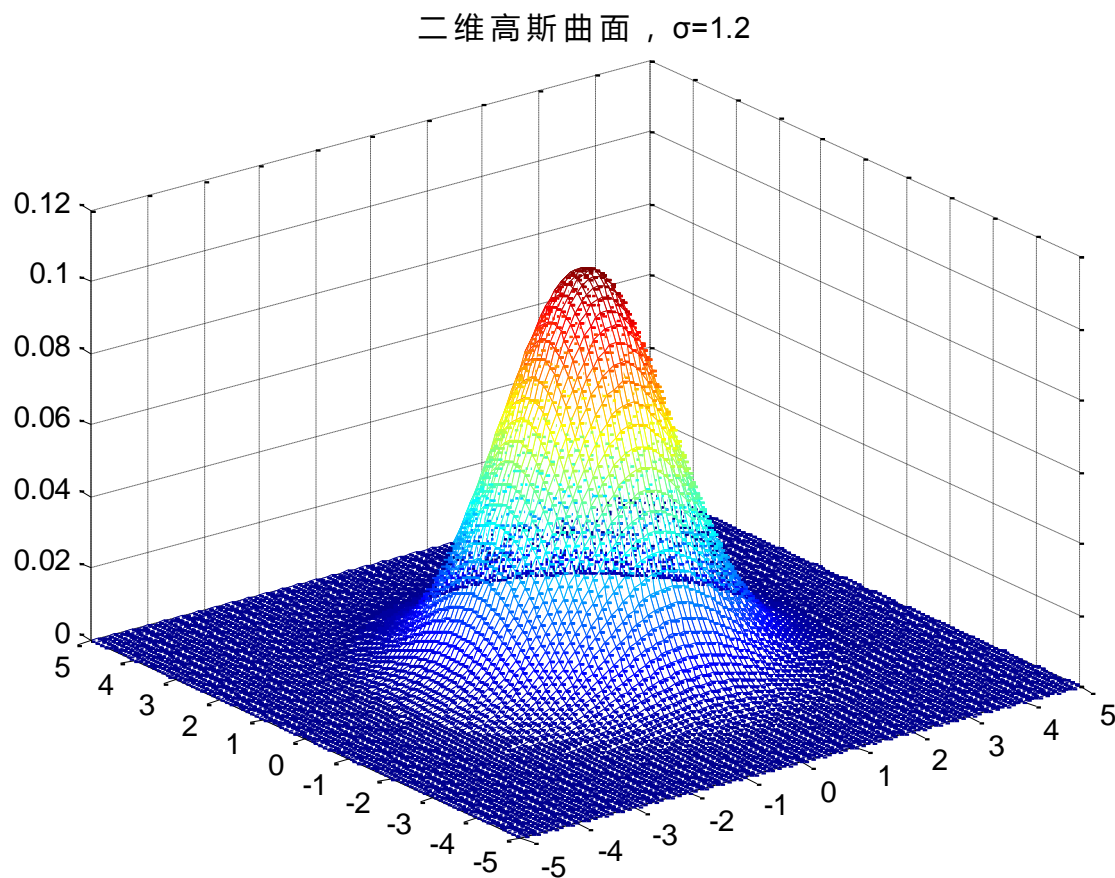
- ▶ 经过高斯变换得到的尺度空间称为高斯尺度空间；
- ▶ 高斯尺度空间的概念是由Witkin最早提出的，并经Koenderink等人的研究工作得到进一步的发展；
- ▶ Koenderink证明了高斯核函数是线性核函数，而Lindeberg又证明了高斯核函数是唯一的线性核函数。

SURF: 兴趣点检测 - 高斯函数

- 模式分类，SVM，数据在低维线性不可分时，使用高斯核函数，映射到高维后可分。



SURF: 兴趣点检测 - 高斯函数



SURF: 兴趣点检测 - Hessian 矩阵

- 对图像中的点 $X = (x, y)$, 尺度为 σ 的 Hessian 矩阵 $H(X, \sigma)$ 定义为:

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix}$$

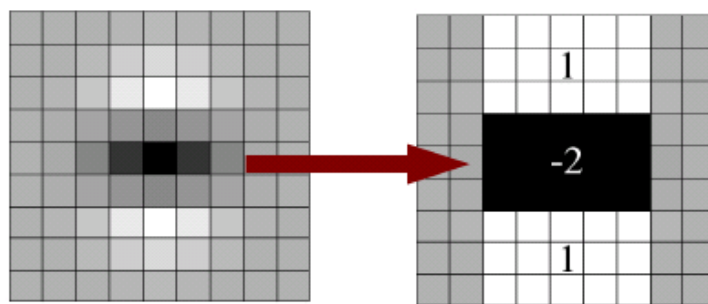
- 其中 $L_{xx}(X, \sigma)$ 是尺度为 σ 的二维高斯函数的二阶倒数与图像 I 在 X 点处的卷积:

$$L_{xx} = \frac{\partial^2 g(\sigma)}{\partial x^2} * I(x, y)$$

SURF: 兴趣点检测 - Hessian 矩阵

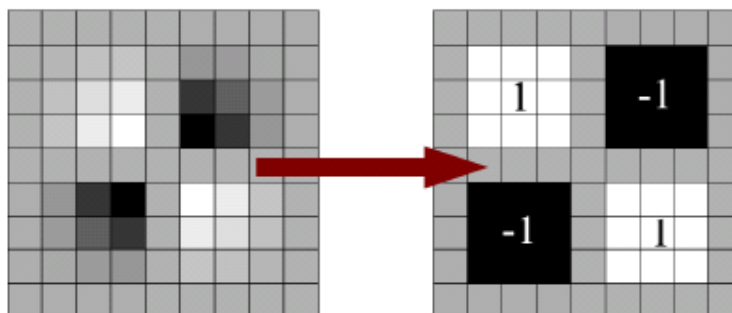
► 高斯二阶倒数的离散化和近似:

► $L_{yy} \rightarrow D_{yy}$



L_{yy}

► $L_{xy} \rightarrow D_{xy}$



L_{xy}

SURF: 兴趣点检测 - Hessian 矩阵

- ▶ 用矩形滤波近似高斯函数，其近似Hessian行列式为：

$$\det(\mathcal{H}_{\text{approx}}) = D_{xx}D_{yy} - (wD_{xy})^2$$

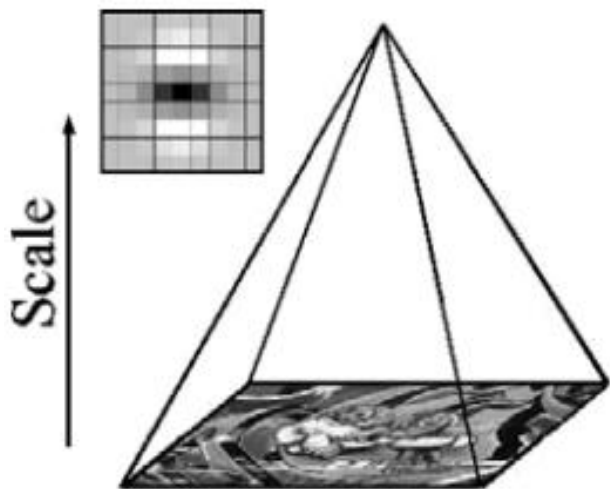
- ▶ **w** is needed for the energy conservation between the Gaussian kernels and the approximated Gaussian kernels

$$w = \frac{|L_{xy}(1.2)|_F |D_{yy}(9)|_F}{|L_{yy}(1.2)|_F |D_{xy}(9)|_F} = 0.912... \simeq 0.9$$

- ▶ 计算过程可用积分图进行加速

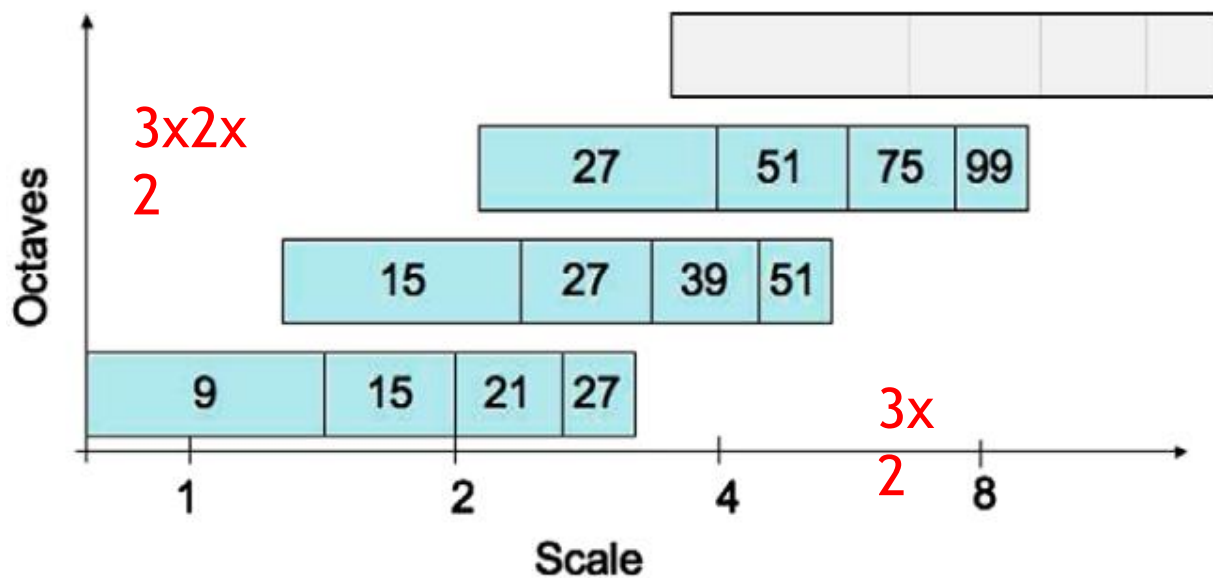
SURF: 兴趣点检测 - 尺度空间

- Lowe的SIFT中通过不断缩小图像得到金字塔
- SUFT中只是增大高斯矩形滤波器，计算量小



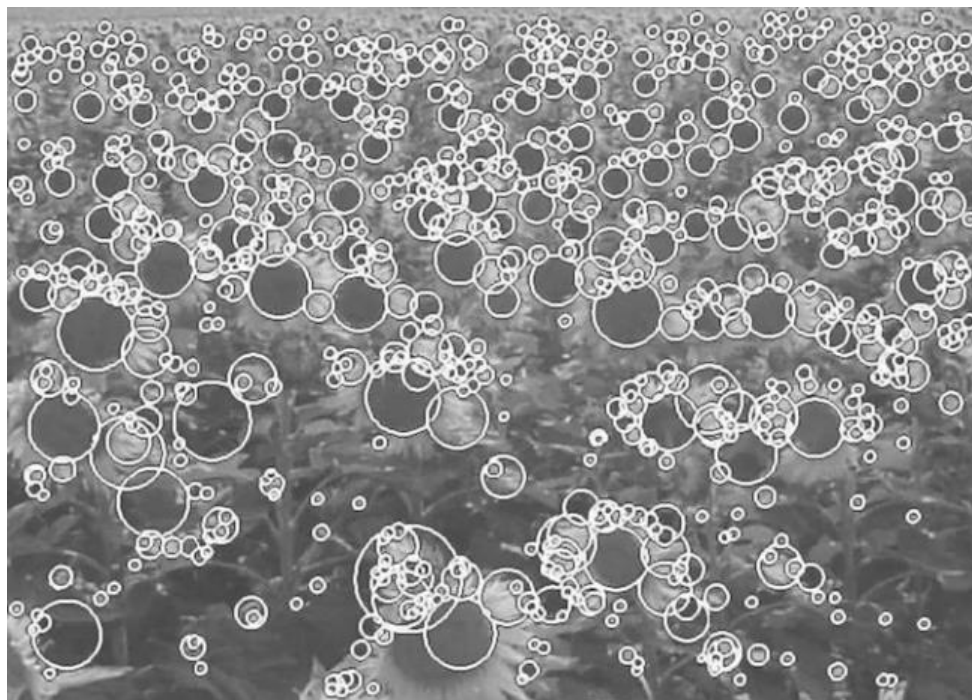
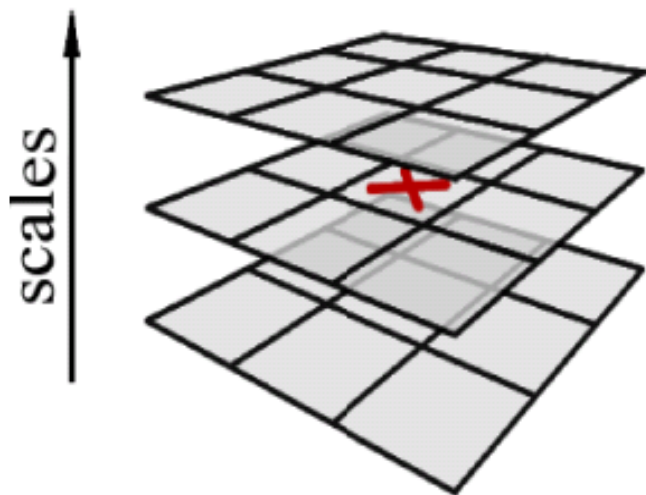
SURF: 兴趣点检测 - 尺度空间

- SURF将尺度空间划分成若干组(Octaves),一个组代表了固定数目的、逐步放大的滤波模板对同一个输入图像的一系列响应。



SURF: 兴趣点检测 - 尺度空间

► 3 x 3 x 3 非最大化抑制



SURF: 兴趣点描述

- ▶ 通过计算尺度空间的Haar小波响应在x, y方向的偏导数, 来生成64D的描述子
- ▶ 1、以兴趣点为中心, 以 $6s$ (s 为兴趣点所在尺度) 为半径的圆形区域内, **确定主方向**, 以实现旋转不变性;
- ▶ 2、构建一个沿主方向的正方形, 并从中**提取SURF描述子**。

SURF: 兴趣点描述 - 主方向

- ▶ 以兴趣点为中心， $6s$ 为半径的圆形区域内，计算Haar小波响应，小波边长为 $4s$ ；
- ▶ 用中心位于兴趣点的 $\sigma=2s$ 的高斯滤波器加权

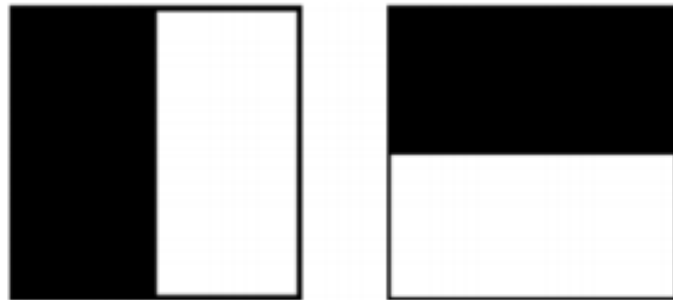
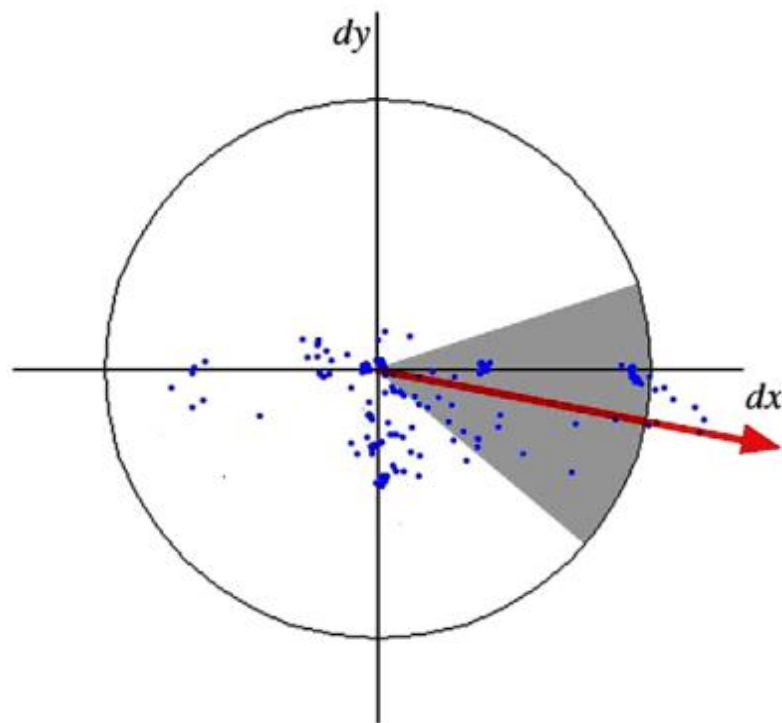


Fig. 9. Haar wavelet filters to compute the responses in x (left) and y direction (right). The dark parts have the weight -1 and the light parts $+1$.

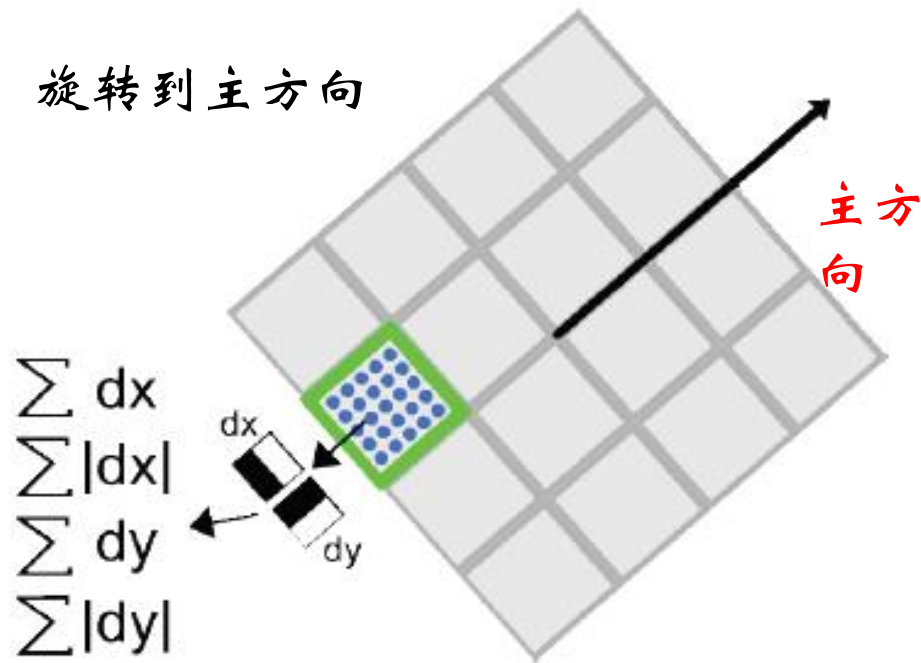
SURF: 兴趣点描述 - 主方向

- 以兴趣点为中心，用中心角为 $\pi/3$ 的滑动窗口，计算窗口内的X、Y方向加权小波响应的和，输出一个向量，向量长度最长的方向，即为主方向



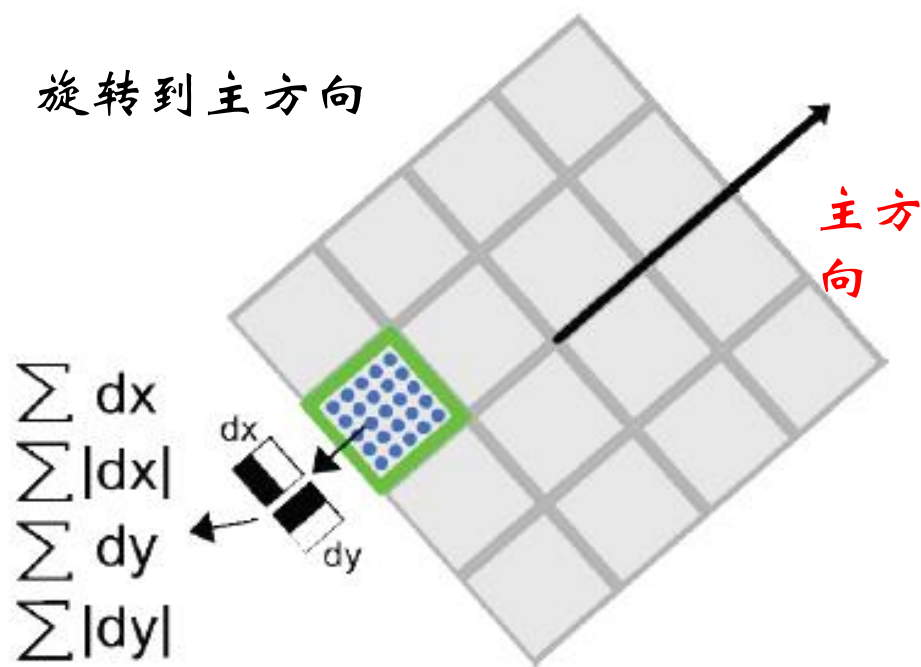
SURF: 兴趣点描述 - 描述子

- 以兴趣点为中心，沿主方位构建一个边长为 $20s$ 的正方形，再等分为 4×4 的子区域
- 在每个子区域等间距选取 5×5 个采样点



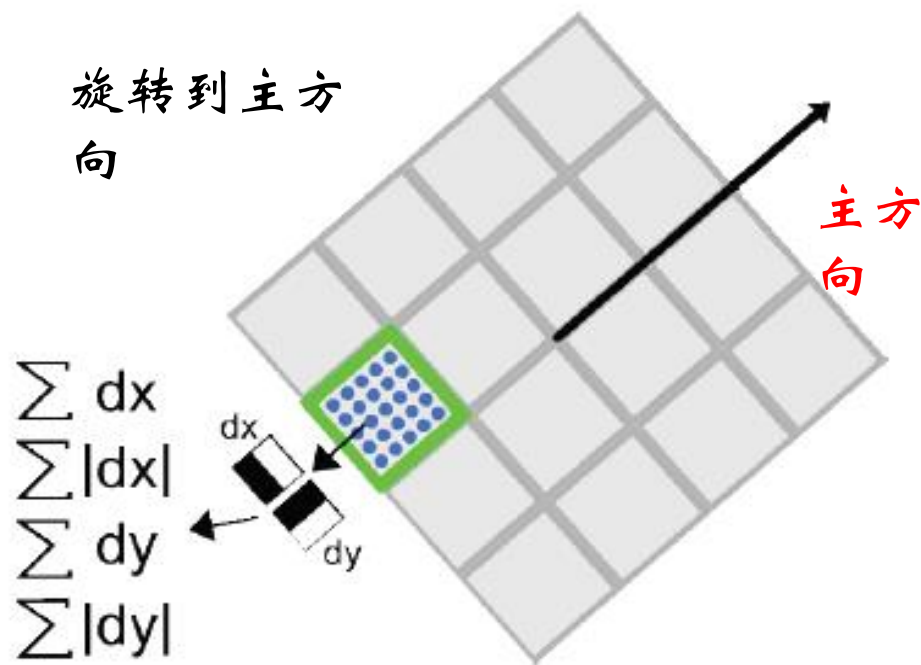
SURF: 兴趣点描述 - 描述子

- ▶ 计算这些点的相对于主方向的Haar小波响应（边长为 $2s$ ），得到 dx , dy
- ▶ 用 $\sigma=3.3s$ 的高斯滤波器加权

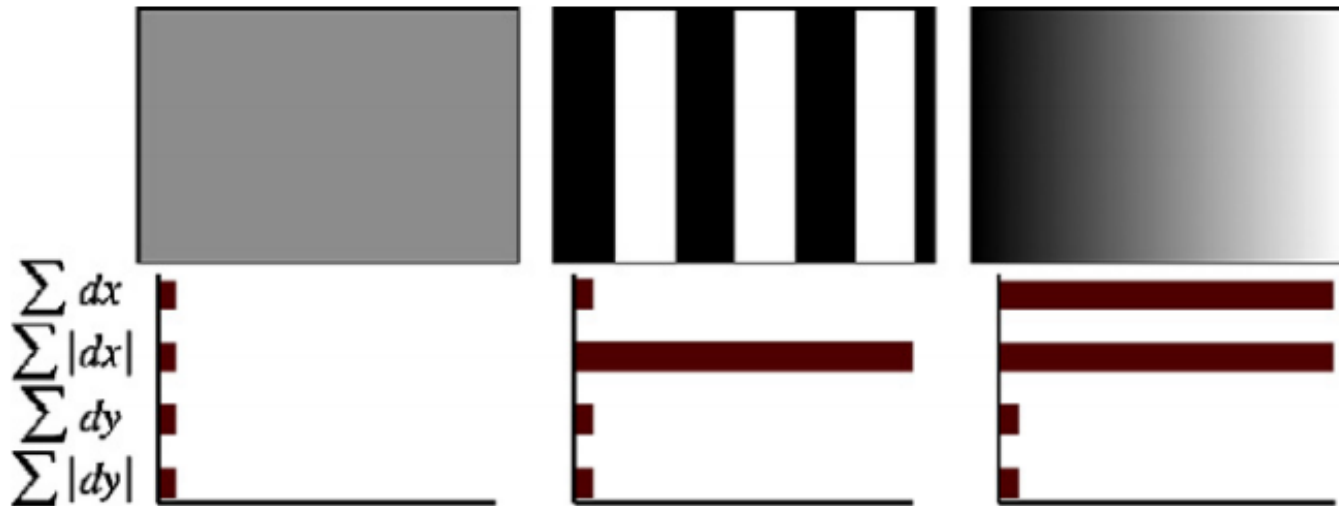


SURF: 兴趣点描述 - 描述子

- ▶ 每个4x4子区域的4D描述向量如下，一共64D
- ▶ $v = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$

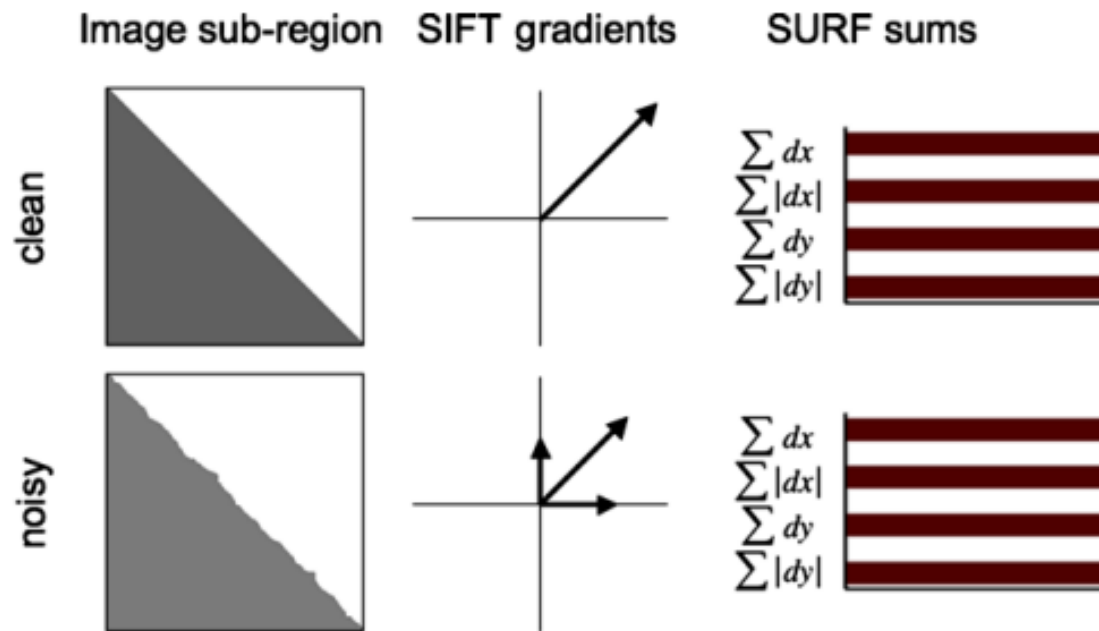


SURF: 示例



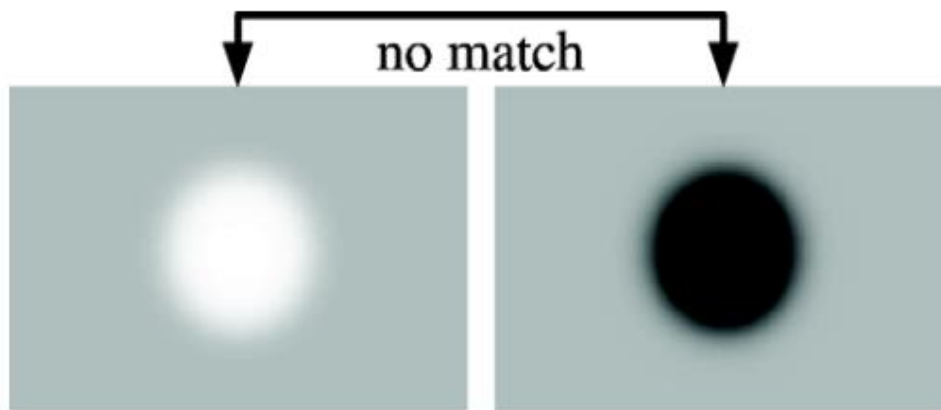
SURF: 示例

- SIFT的计算基于梯度方向
- SURF基于主方向，抗噪声性更好



SURF: 快速匹配

- 兴趣点检测时，计算了Hessian矩阵，可用拉普拉斯算子，即Hessian矩阵的迹，排除一大部分不匹配的点。



SURF: 匹配 - 欧氏距离

► 欧氏距离 (Euclidean)

► 如果马氏距离中的协方差矩阵 Σ 为单位阵, 则马氏距离简化为欧氏距离

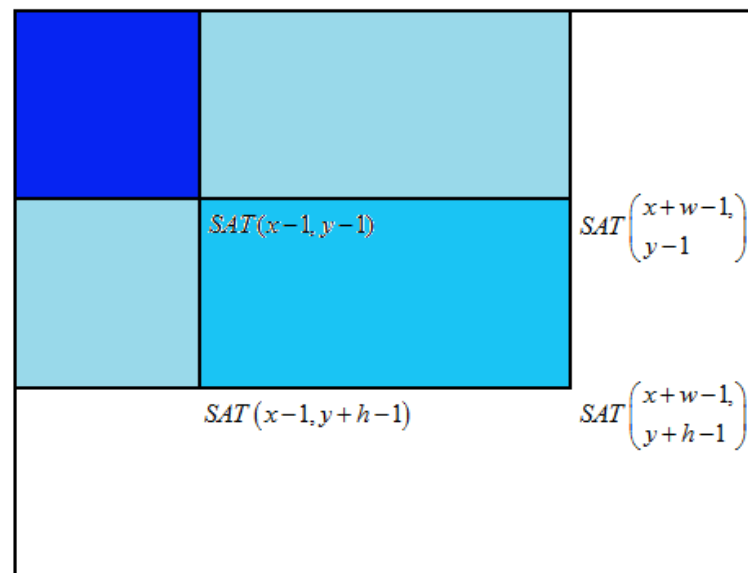
$$d(x, y) := \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

子课题一. 计算输入图像的积分图

- ▶ 输入：原始彩色图像
- ▶ 输出：图像的积分图
- ▶ 流程：
 - ▶ 把输入图像变成灰度图像
 - ▶ 积分图像中，每个点存储是其左上方所有像素之和

- ▶ 计算图像的积分图：

$$I_{\Sigma}(x) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j)$$



- ▶ 任意矩形区域内的点的求和只需要计算三个加(减)法运算

子课题二. 计算图像的det值

- ▶ 输入：图像的积分图
- ▶ 输出：图像det值得向量
- ▶ 流程：
 - ▶ 建立图像的response层，即向上取样，如下图所示。
 - ▶ 计算每一层次图像像素点的det值
 - ▶ 对层次极值点进行插值
- ▶ 计算图像像素点det值是这个算法最耗时的部分。为了在目标影像上确定SURF特征点，我们使用了 $3*3*3$ 的模板在3维尺度空间进行非最大化抑制，根据预设的Hessian阈值 H ，当 h 大于 H ，而且比临近的26个点的响应值都大的点才被选为兴趣点。最后进行插值精确。

子课题三. 计算图像兴趣点的方向

► 输入：兴趣点向量

► 输出：兴趣点的方向

► 流程：

► 计算兴趣点以 $6s$ 为半径圆的内的 x 方向和 y 方向Harr小波变换值

► 计算哪个方向上 x 方向和 y 方向harr值的和最大，

► 最大值方向就是兴趣点的方向。

子课题四.计算图像的特征矢量

- ▶ 输入：方向矢量和det矢量
- ▶ 输出：兴趣点的特征矢量
- ▶ 流程：
 - ▶ 计算 5×5 区域harrx和haary
 - ▶ 计算Gaussian权值
 - ▶ 计算局部的dx、abs(dx)、dy和abs(dy)
 - ▶ 全局的dx、abs(dx)、dy和abs(dy)，加上局部的dx、abs(dx)、dy和abs(dy)，5返回1，直到计算完4个局部的dx、abs(dx)、dy和abs(dy)

子课题五.匹配图像对应点

- 对于两个特征点描述子的相似性度量，我们采用欧式距离进行计算：

$$\text{Dis}_{ij} = \left[\sum_{k=0}^{k=n} (X_{ik} - X_{jk})^2 \right]^{1/2}$$

式中， X_{ik} 表示待配准图中第 i 个特征描述子的第 k 个元素， X_{jk} 是参考图中第 j 个特征描述子的第 k 个元素， n 表示特征向量的维数。

- 输入：图像兴趣点特征值矢量
- 输出：匹配对应点矢量
- 过程：2层循环搜索2个图像特征矢量对应点的距离如果从内层循环搜索完成，判断搜索得到的对应点距离比值小于阈值则加入匹配点矢量中。下图是匹配2幅静态图像的结果。



五、Tile-based SURF



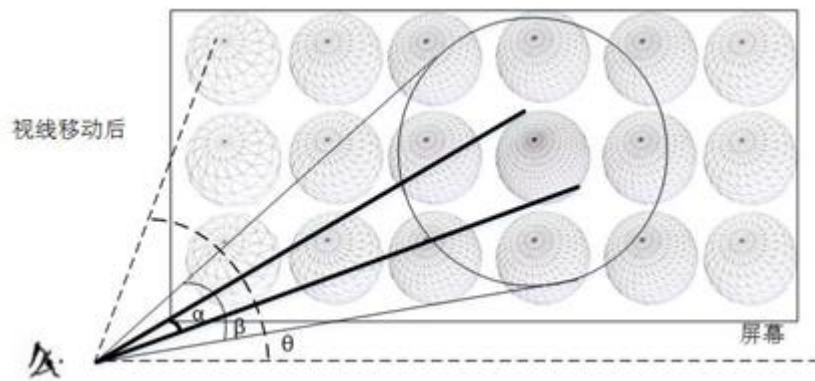
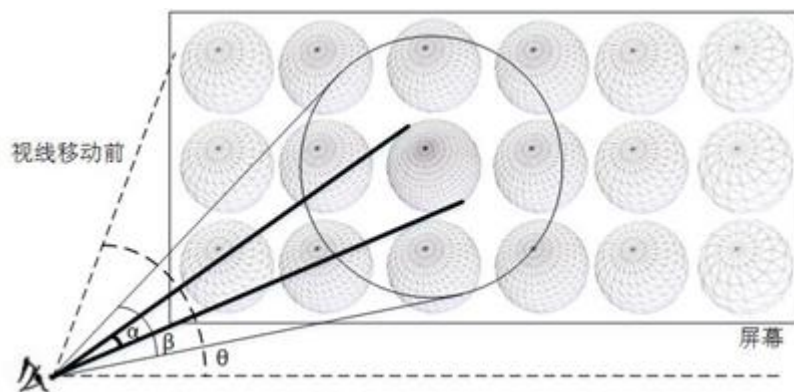
OpenGL 与基于视线焦点 追踪的低功耗图形处理

Together with Tencent

腾讯

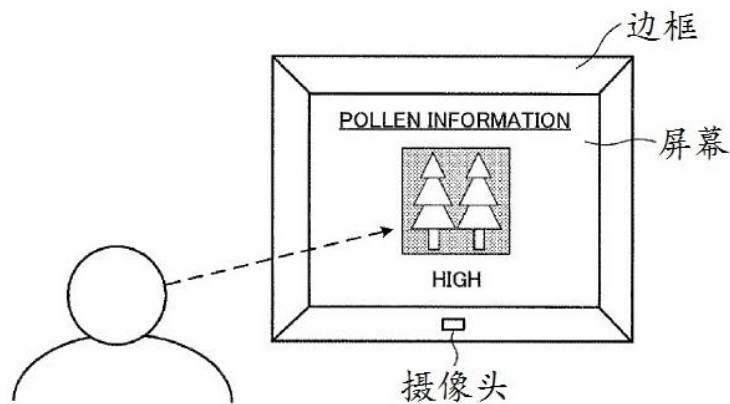
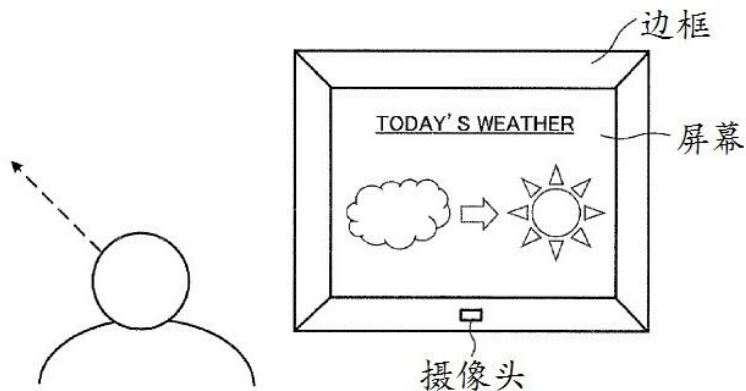
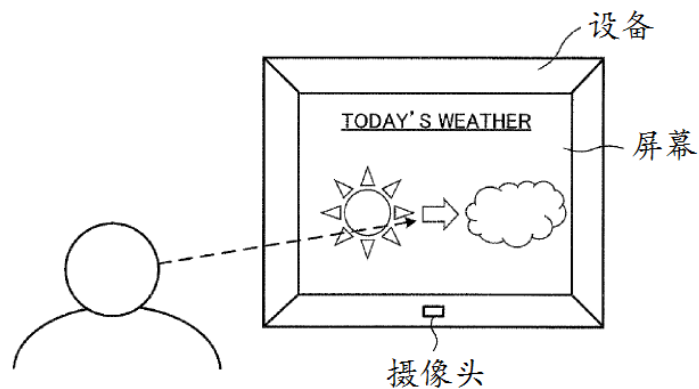
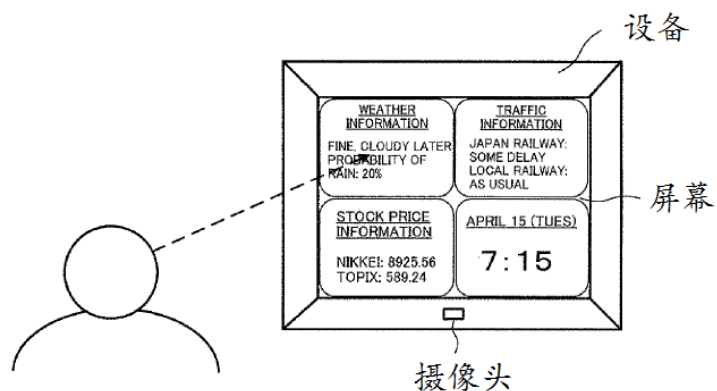
创意

- ▶ 人的视线和注意力往往聚焦在某些局部区域（视线焦点）
- ▶ 利用这一特点，本创意优化视线焦点附近区域而非全屏幕的显示效果
- ▶ 满足甚至提高人主观视觉的感受
- ▶ 极大降低图形处理器（GPU）渲染图形的功耗，节约电能并增加电子设备的续航能力。



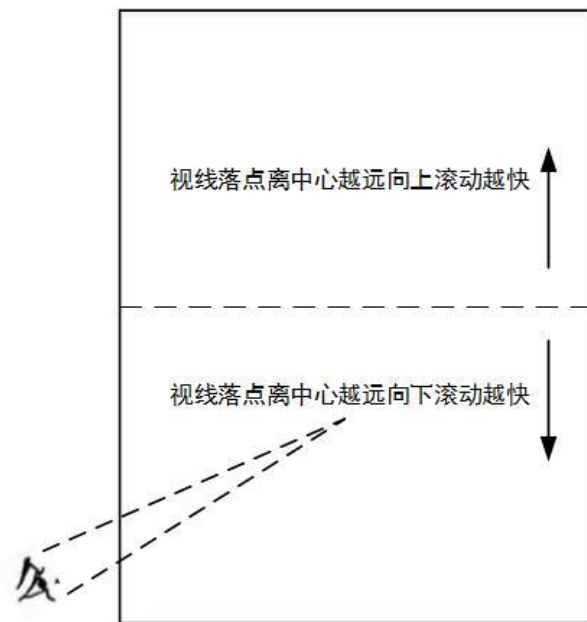
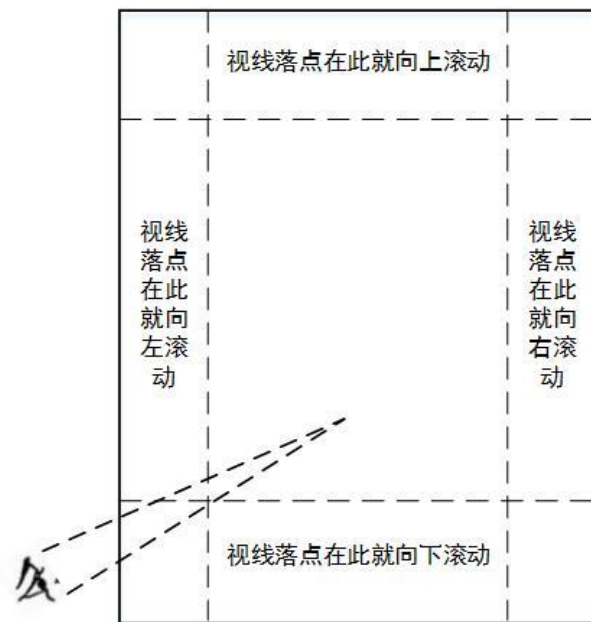
类似专利介绍

► 眼球追踪并控制显示内容



相关专利

► 阅读体验



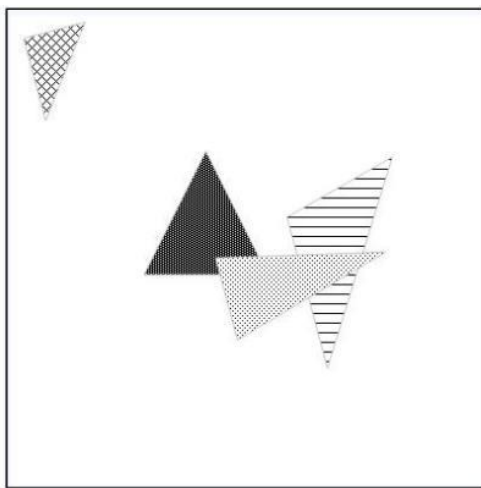
► 视频控制

► 智能屏保

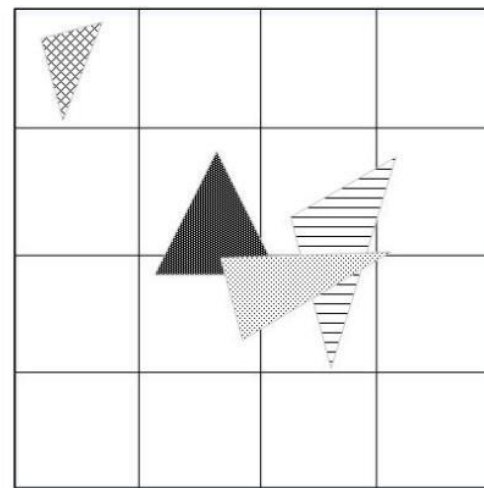
Tile-based GPU

(PowerVR from Imagination)

- ▶ GPU的渲染管线包括：顶点转化、图元建立、栅格化、帧缓冲处理（包含纹理贴图、染色、透明混合处理，光影处理等）
- ▶ 桌面GPU：每一帧重新渲染，带宽需求大、功耗大、但效果性能好
- ▶ 嵌入式GPU：片上缓存，只保留中间数据，每次只计算需要更新的内容，节省带宽
- ▶ 多大的片上缓存？
- ▶ 切分渲染区域
- ▶ 去隐藏

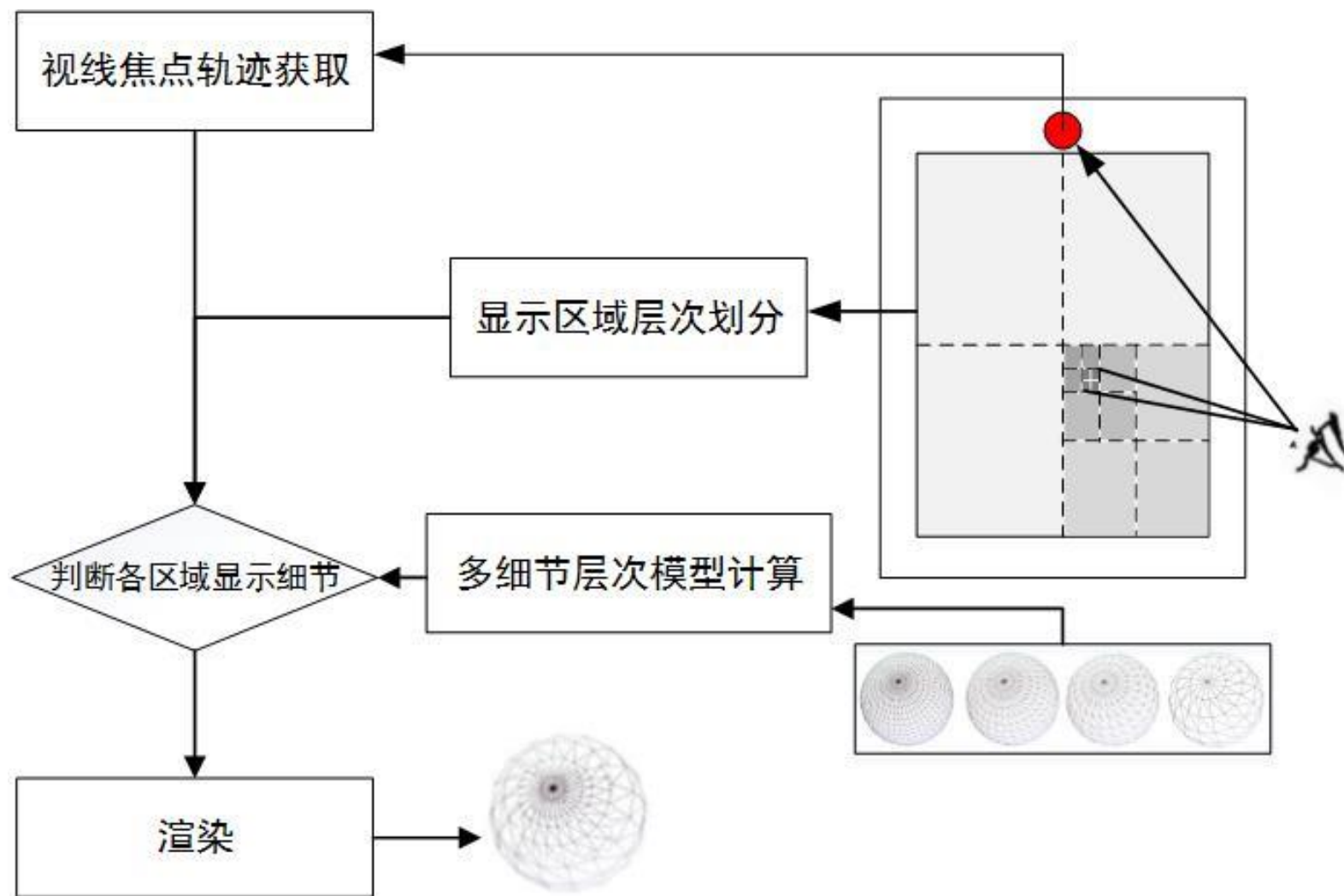


传统架构



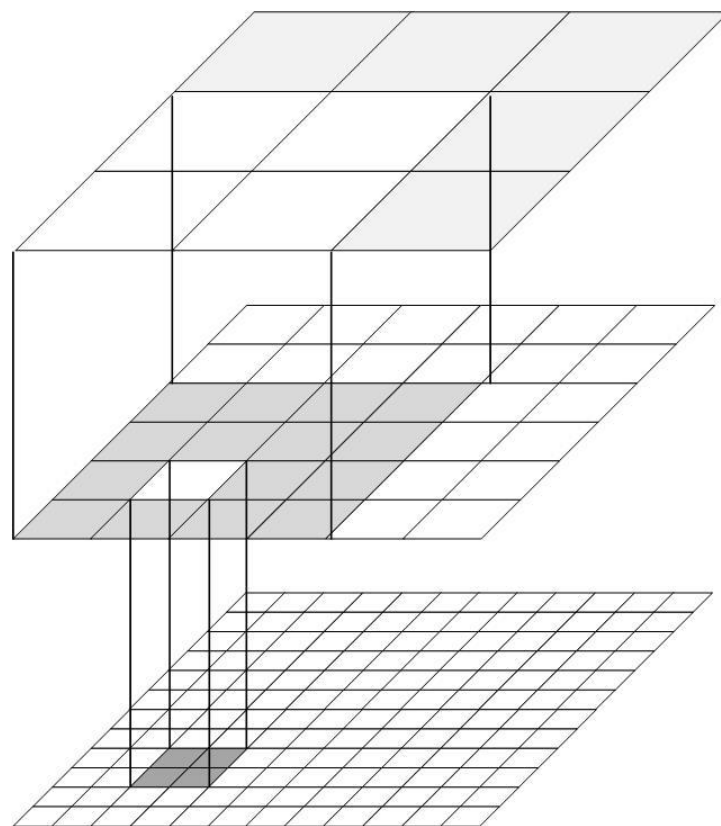
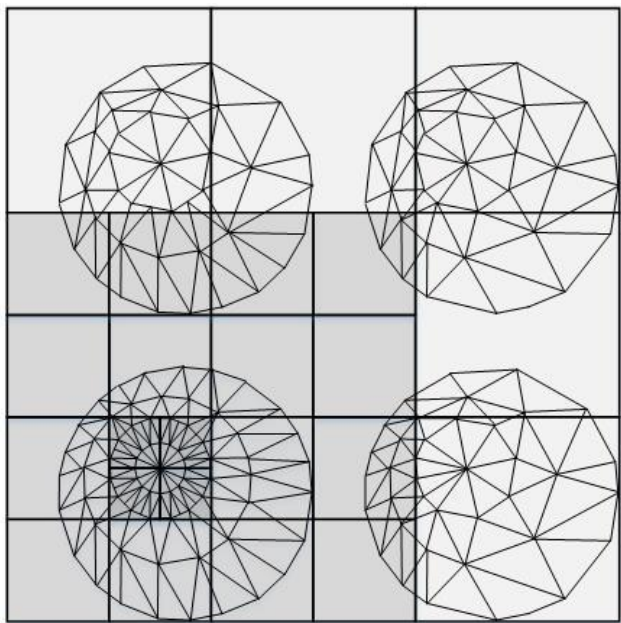
切片架构 (Tile-based)

基本技术



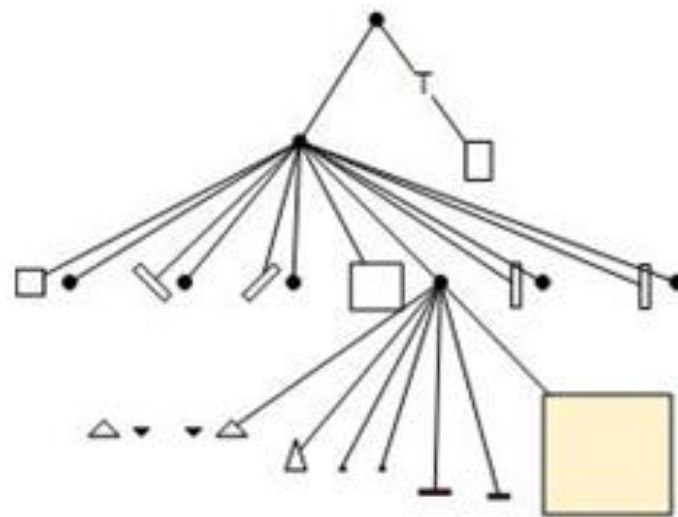
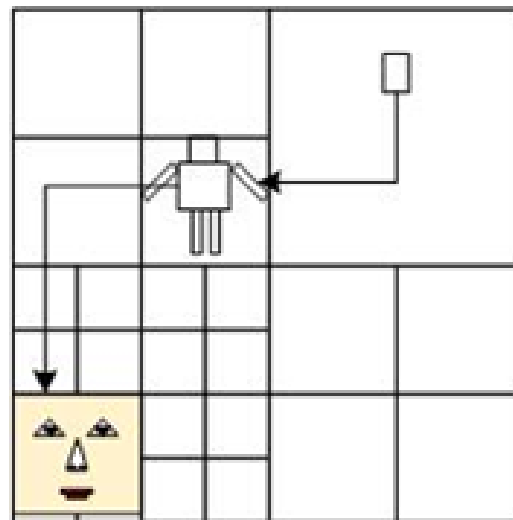
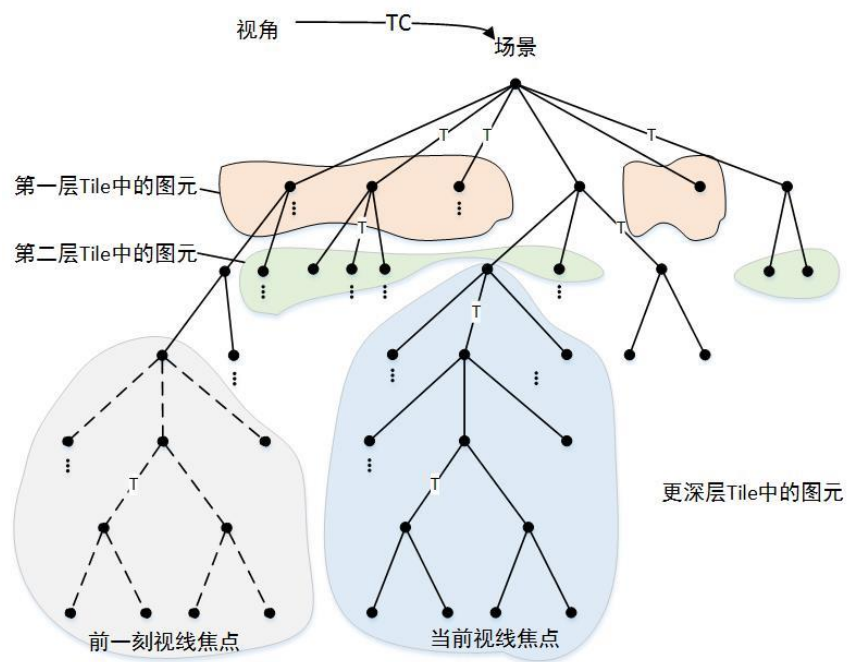
显示区域层次划分

- ▶ 在不同层的划分中进行不同粒度的渲染
- ▶ 图元大小分层

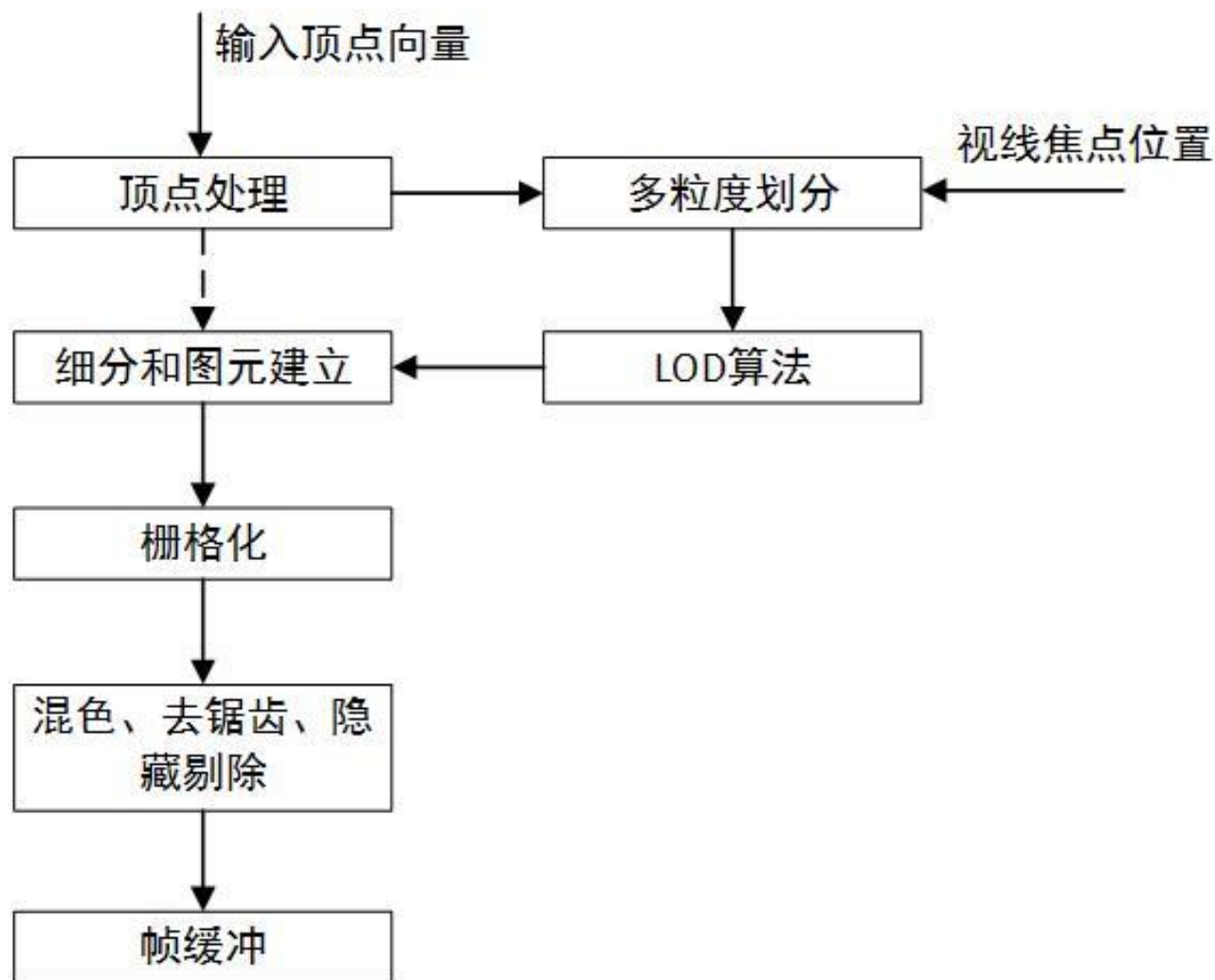


12:07:40

LOD 算法



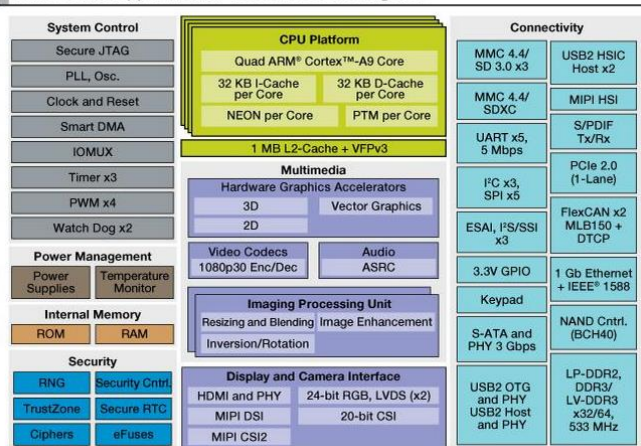
改变传统渲染管线



Freescal Sabrelite:i.MX6Q Development Board



i.MX 6Quad Applications Processor Block Diagram



Features:

CPU Complex

4x ARM® Cortex™-A9 up to 1.2 GHz per core

1 MB L2 cache

32 KB instruction and data caches

NEON SIMD media accelerator

GPU

GPU 3D

Vivante GC2000

200Mtri/s 1000Mpxl/s, OpenGL ES 3.0 & Halti, CL EP

GPU 2D(Vector Graphics)

Vivante GC355

300Mpxl/s, OpenVG 1.1

GPU 2D(Composition)

Vivante GC320

600Mpxl/s, BLIT

Memory

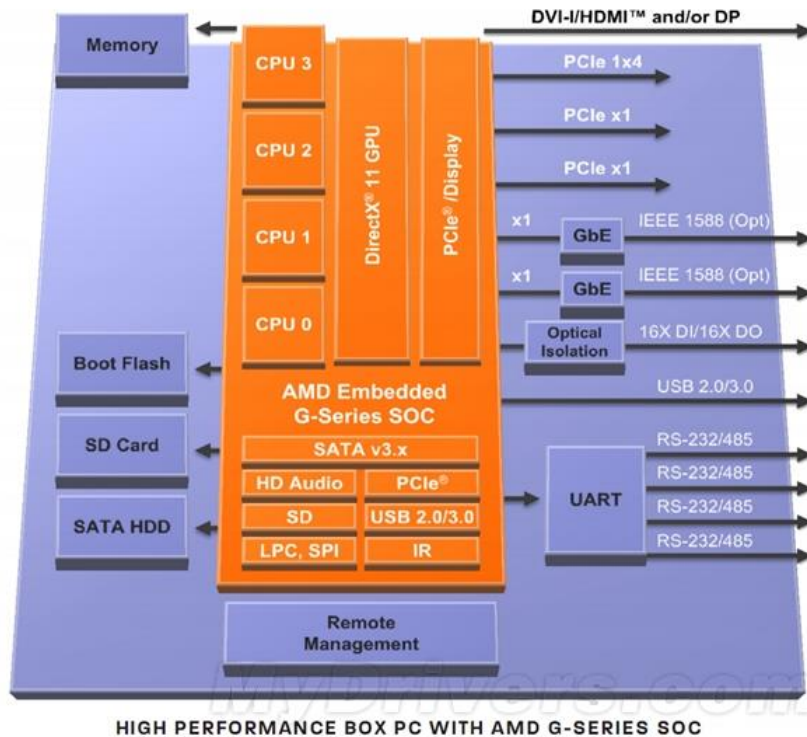
DDR

2x32 LP-DDR2, 1x64 DDR3 / LV-DDR3

NAND

SLC/MLC, 40-bit ECC, ONFI2.2, DDR

AMD GX-415GA APU SOC Development Board



1st generation APU SOC design

- Integrates Controller Hub functional block as well as CPU+GPU+NB
- 28nm process technology, FT3 and FT3b BGA packages, 24.5mm x 24.5mm
- Dual- or Quad- “Jaguar” or “Jaguar+” CPU cores with 2MB shared L2 cache

Features

Model: GX-415GA SOC

CPU Cores: 4(x86)

TDP(Thermal Design Power)(CPU, GPU,& SB): 15W

Shared L2 Cache : 2MB

CPU FREQ. : 1.5GHz

GPU FREQ.(GRAPHICS): 500MHz (HD 8330E)

DDR SPEED: DDR3-1600

TjC: 0-90° C

Support DirectX 11.1、OpenGL 4.2x、OpenCL 1.2

NVIDIA Tegra 3 Apalis T30

Features:

CPU: NVIDIA® Tegra™ 3 (ARM Cortex™-A9)

NVIDIA SoC:T30MQS

CPU Cores:4+1 4+1

L1 Instruction Cache (each core):32KByte

L1 Data Cache (each core):32KByte

L2 Cache (shared by cores) 1MByte

NEON support

Maximum CPU frequency (single core mode):1.4GHz

Maximum CPU frequency (quad core mode peak performance, time and temperature limited):1.3GHz

Memory

DDR3 RAM Size 2GByte

DDR3 RAM Speed (max) 1600MT/s

DDR3 RAM Memory Width 32bit

eMMC NAND Flash (8bit) 8GByte

GPU: Ultra-low Power NVIDIA GeForce GPU

OpenGL® ES 1.1 and 2.0 (depending on driver)

Dual Core (2 fragment shader pipe, vertex shader pipe)

OpenGL ES Shader Performance 16 GFLOPS

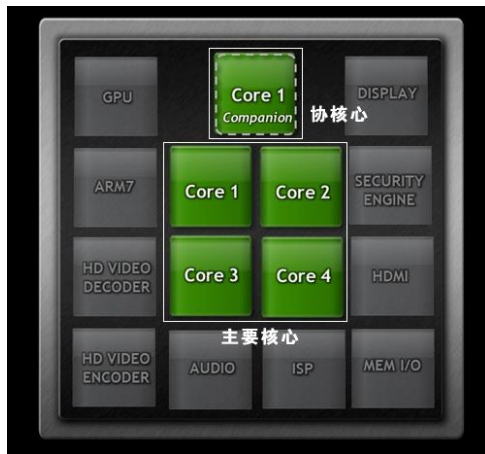
Programmable pixel shader

Programmable vertex and lighting

Anti-aliasing: 5x VCAA, 4xFSAA, or both

2K x 2K texture and 4K x 4K render resolutions supported

Advanced 2D and vector engine



Research: Soft Error in Multicore architecture

Together with 浪潮

芯片和系统的可靠性

► 软错误

► 原因：

- 宇宙射线、辐射击穿器件，导致逻辑错误；

► 特点：

- 短暂的错误，电路可以恢复

► 案例：

- Sun公司价值百万美元的旗舰服务器UltraSPARC-II由于软错误而崩溃；
- 惠普公司安装在美国国家实验室的2048-CPU核服务器因在高海拔地区，极易受到宇宙射线打击引起软错误、导致经常性宕机。
- 高空（宇宙）中的航电等电子设备更易出现故障



芯片和系统的可靠性

► 软错误

► 传统解决方案：

- 增加硬件冗余（如飞机有3套各自独立的电子控制系统），通过投票来做出决定：功耗，成本。具体，增加处理器核心数目。
- 增加冗余线程，在系统级完成检测和保护

► 研究目标：

- 利用CPU中多层次保护机制进行在线检测和恢复
- 利用GPU中多路数据通道进行在线检测和恢复