



半导体智能制造：基于模型的显示量测智能仪器快速开发实践

郑增强 Mike Zheng



MATLAB EXPO 2021

Agenda

- 精测产品介绍
- 精测基于模型设计流程介绍
- 基于模型设计的产品创新
- 基于模型设计的好处

发展历程

Company Introduction

质量检测与良率控制，贯穿了显示、半导体、新能源领域高端制造的整个生命周期，是核心中的核心。



发展历程

Company Introduction



武汉显示产业基地



苏州显示产业基地



上海半导体产业基地



北京半导体产业基地



美国研发基地



日本研发制造基地

产品介绍

Product Introduction

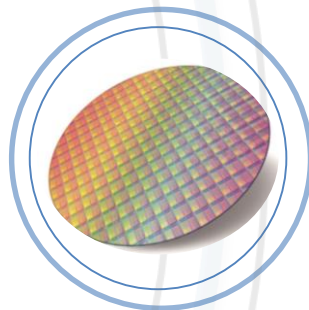
公司从定制业务入手，深入垂直行业，形成相对标准化的核心部件，定制端到端的技术解决方案。



发展历程

Company Introduction

No.1



- 国内**唯一**同时提供显示三大制程检测设备的厂商，面板检测行业模组段设备全球占比**第一**，CELL和ARRAY段检测设备国内厂商占比**第一**
- **第一家**推出基于生产和信息化处理的人工智能液晶模组生产线
- 针对Micro LED量产需求，推出国内**第一条**集Gamma、DeMura、AOI于一体的自动化产线
- 推出业内**第一台**Micro LED点灯检查机
- 国内**第一家**推出拥有完全自主知识产权的半导体前道检测侧设备（eVIEW™全自动晶圆缺陷检测设备）
- 国内**第一家**高功率电子负载模块化推行者
- 推出国内**第一台**通道式电池模拟器

发展历程

Company Introduction

精测电子合作伙伴遍布三大领域



*以上企业排名不分先后

显示量测智能仪



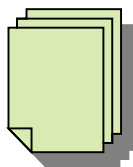
- Imaging Flicker Meter(JC-IFM-31CF)



- Imaging Color Meter (JC-ICM-31CF)

基于模型设计的优势 Model-Based Design

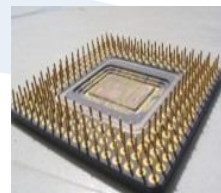
需求和详细指标



设计



实现

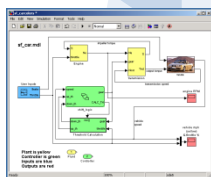


测试验证



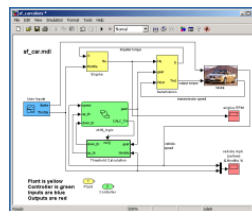
Model elaboration

Continuous verification



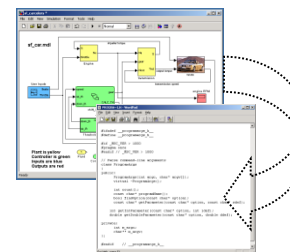
可执行的模型

- 无二义性
- only "one truth"



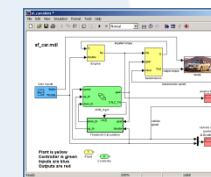
仿真

- 减少依赖于真实样机的昂贵仿真
- 系统级分析



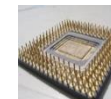
自动代码生成

- 最小化编码错误



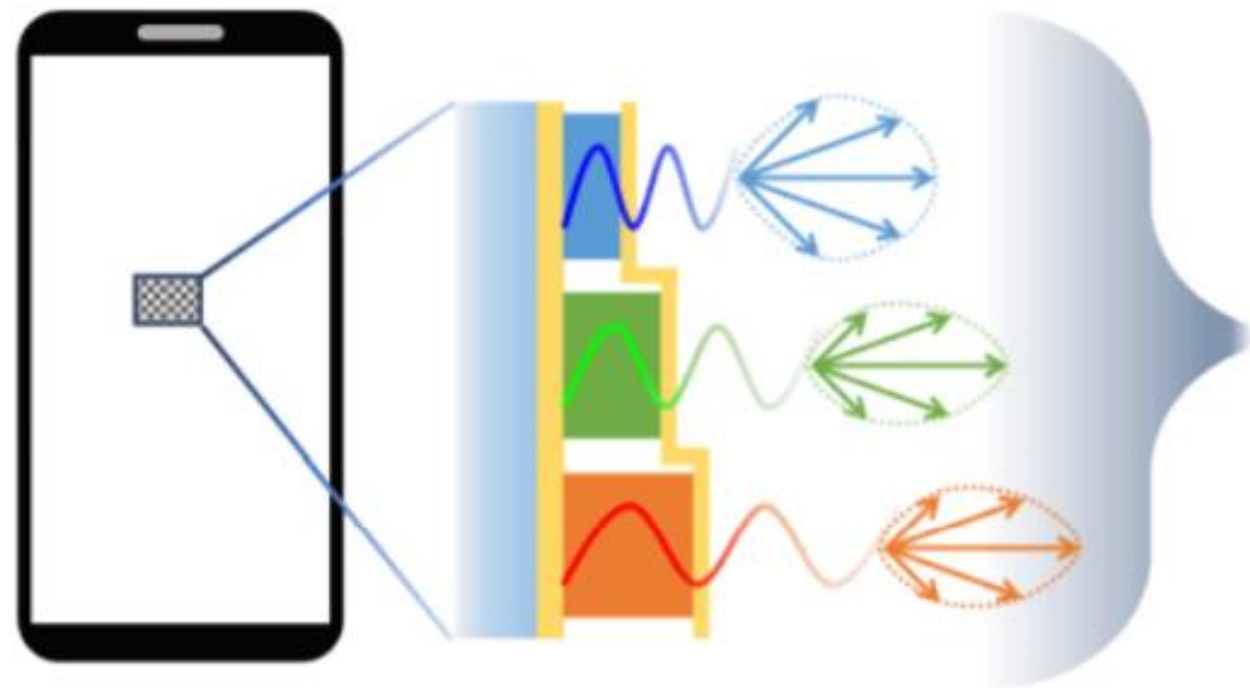
设计的测试验证

- 更早的发现错误



需求描述与理论计算模型

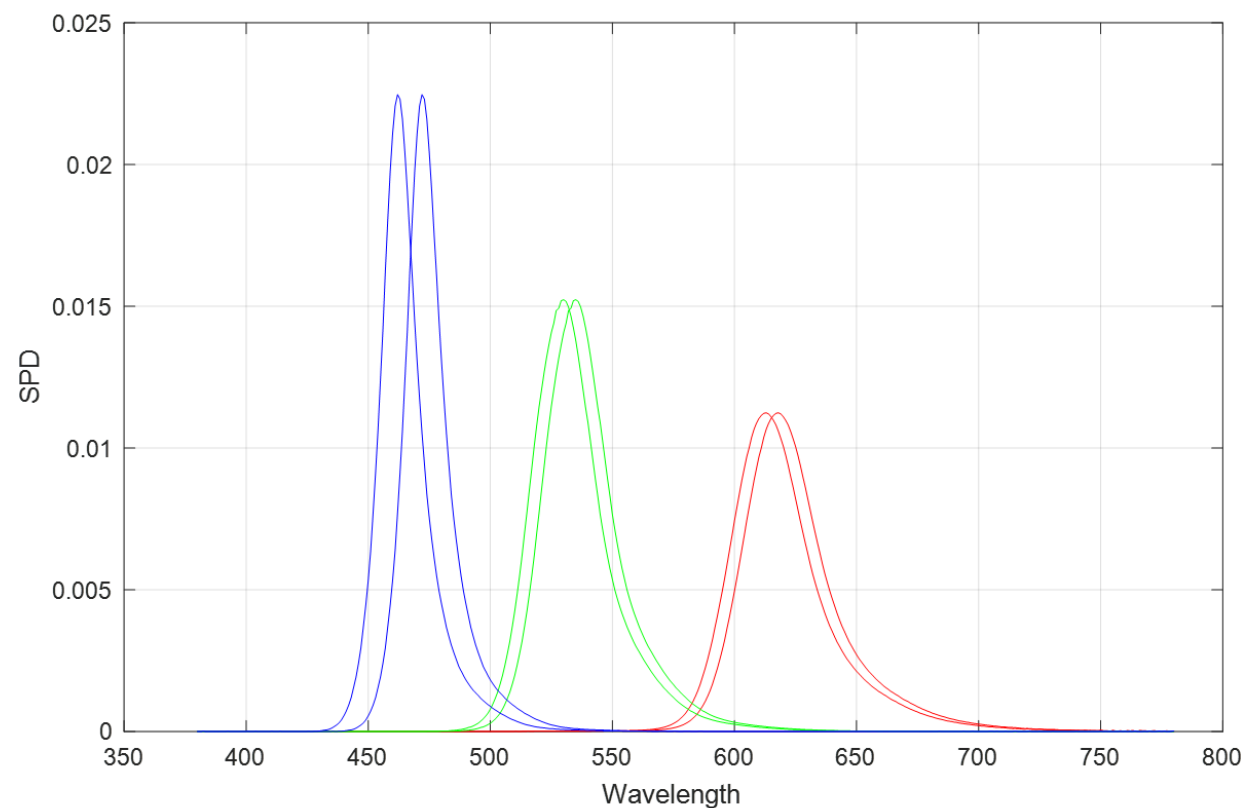
- 模拟显示面板发光特性
- 建立仪器测量理论模型



*图引用G. Tan , et al.Opt. Express 25,33629(2017)

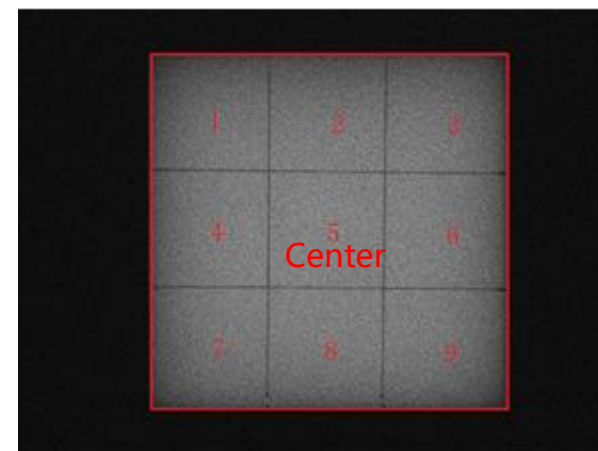
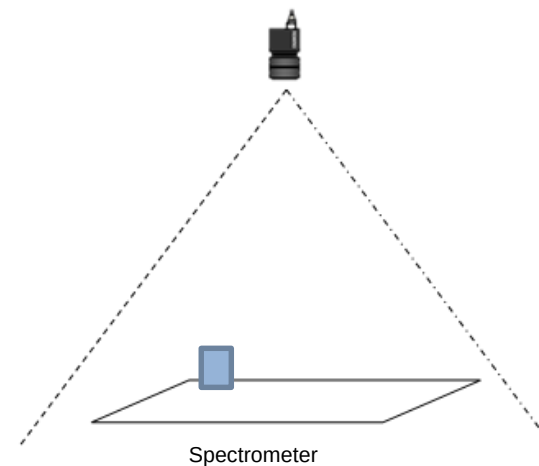
系统设计与误差设计

- 系统方案设计(包含公差的仪器测量模型)
- 系统同测量误差分析与设计



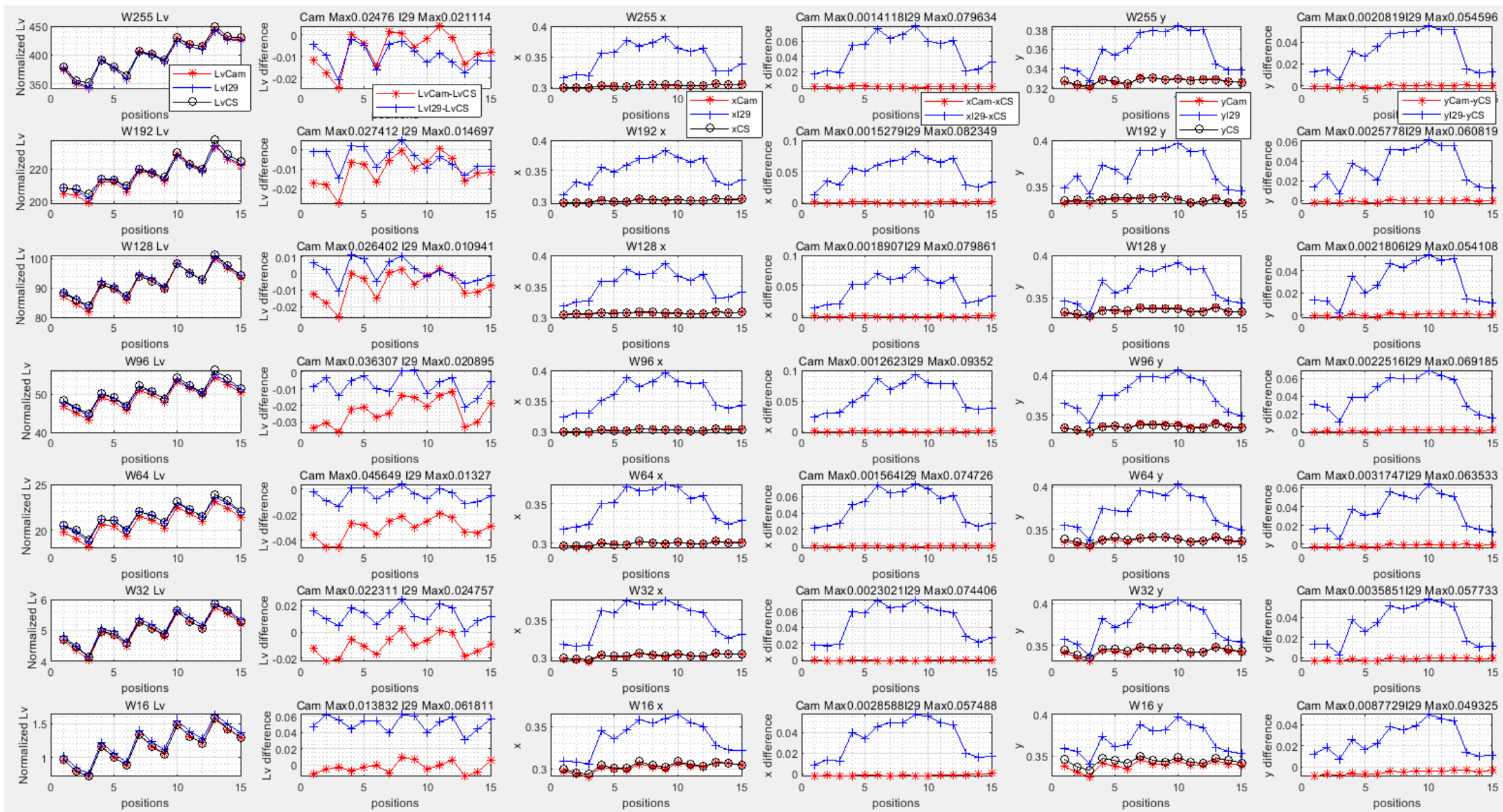
实验验证与原型算法开发

- 测量仪器模型的实验验证
- 校准和测量原型算法的开发
- MATLAB & ZEMAX 光学系统设计
- 机电系统设计和分析



面型测量仪器实验示意图

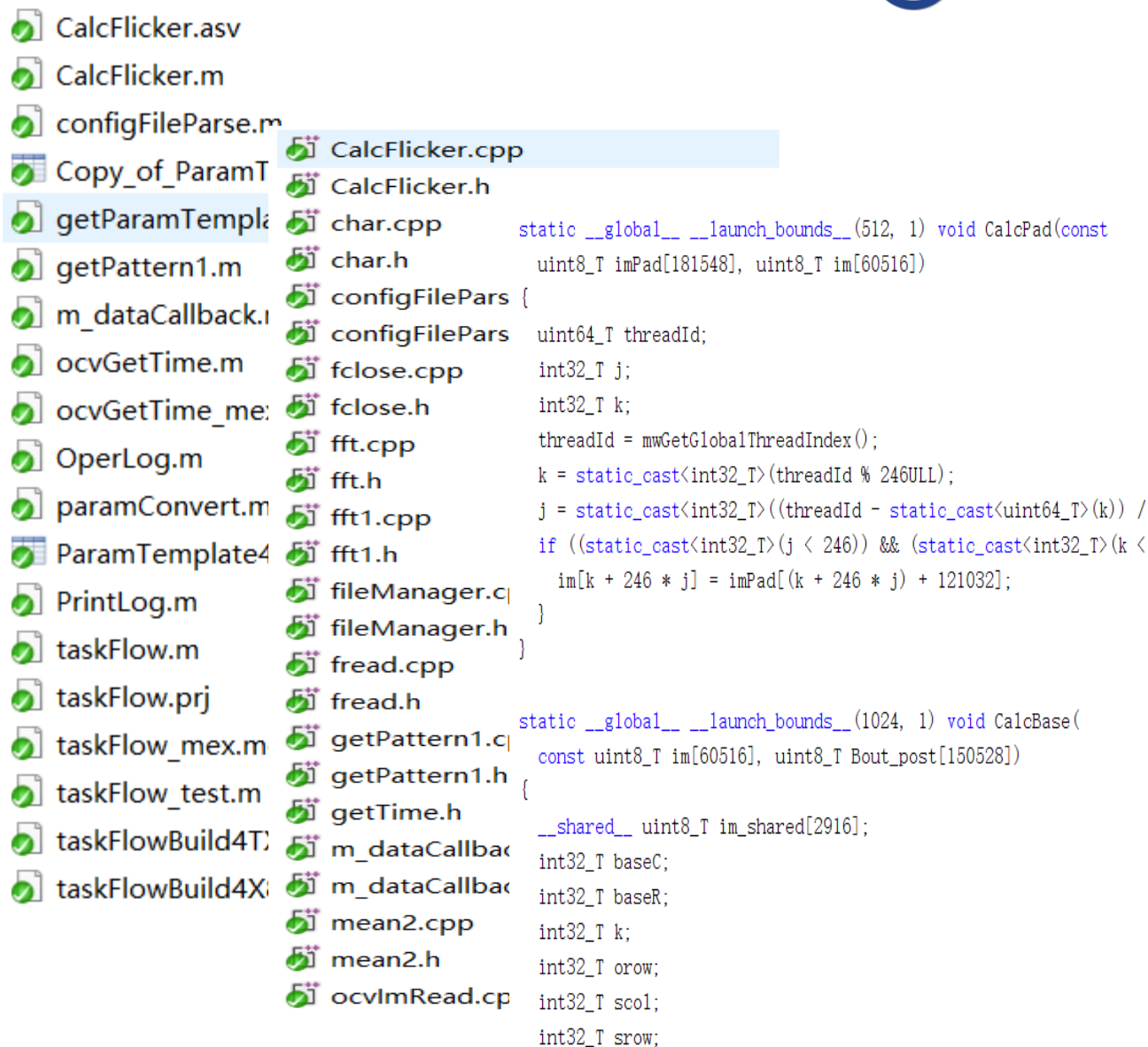
实验数据分析（色度）



算法部署到Jetson GPU

- 替换部分不支持的函数，生成GPU代码
- 在PC（GPU）上进行SIL测试
- 分析性能，优化代码：速度和内存
- 生成Jetson 代码及SO文件，远程测试PIL

R2020b 速度比手写提升23.6%



统一寻址 (Unified Memory) : cudaMallocManaged

- Implicit copy on desktop, no copy in SoC
 - Shared host and GPU memory allocations (and variables)

```
float *a, *x, *y;  
float *gpu_a, *gpu_x, *gpu_y;
```

```
// allocate unified/managed memory
```

```
cudaMallocManaged(&gpu_a, msize, cudaMemLocationHost);  
cudaMallocManaged(&gpu_x, msize, cudaMemLocationHost);  
cudaMallocManaged(&gpu_y, msize, cudaMemLocationHost);
```

```
// compute a, x, y  
...
```

```
// copy input data from CPU to GPU
```

```
cudaMemcpy(gpu_a, a, msize, cudaMemcpyHostToDevice);  
cudaMemcpy(gpu_x, x, msize, cudaMemcpyHostToDevice);  
cudaMemcpy(gpu_y, y, msize, cudaMemcpyHostToDevice);
```

```
// run the kernel
```

```
saxpy<<<4096, 256>>>(msize, gpu_a, gpu_x, gpu_y);
```

```
// copy output data from GPU back to CPU
```

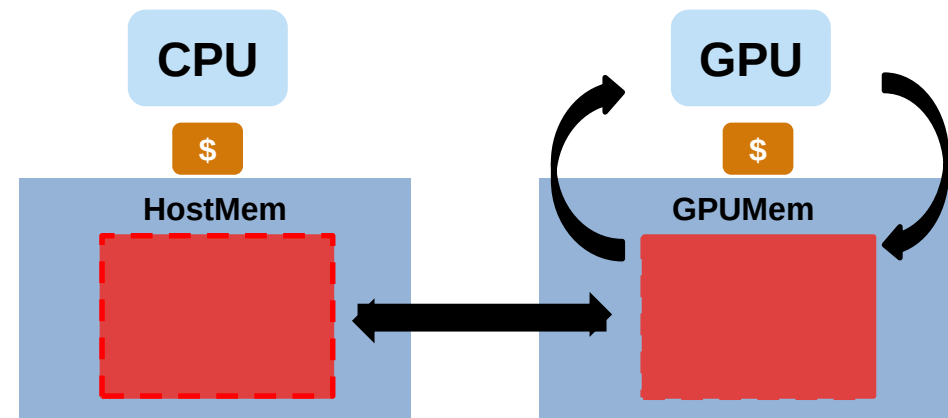
```
cudaMemcpy(y, gpu_y, msize, cudaMemcpyDeviceToHost);
```

```
cudaDeviceSynchronize();
```

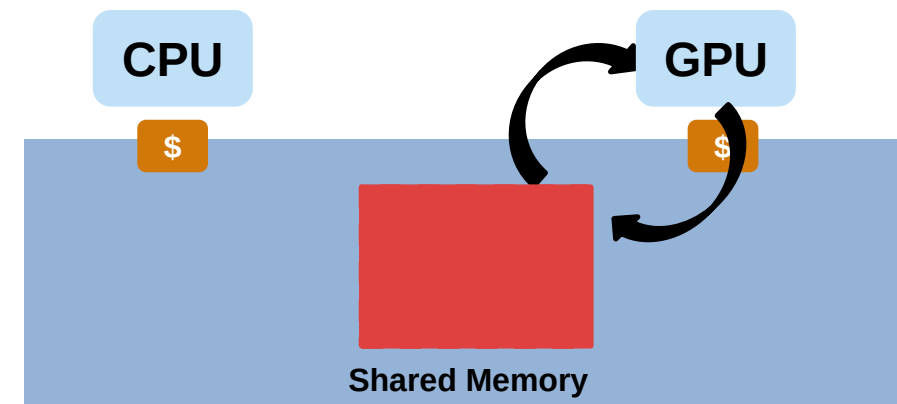
```
... = y;
```

Challenge: when to deviceSync?

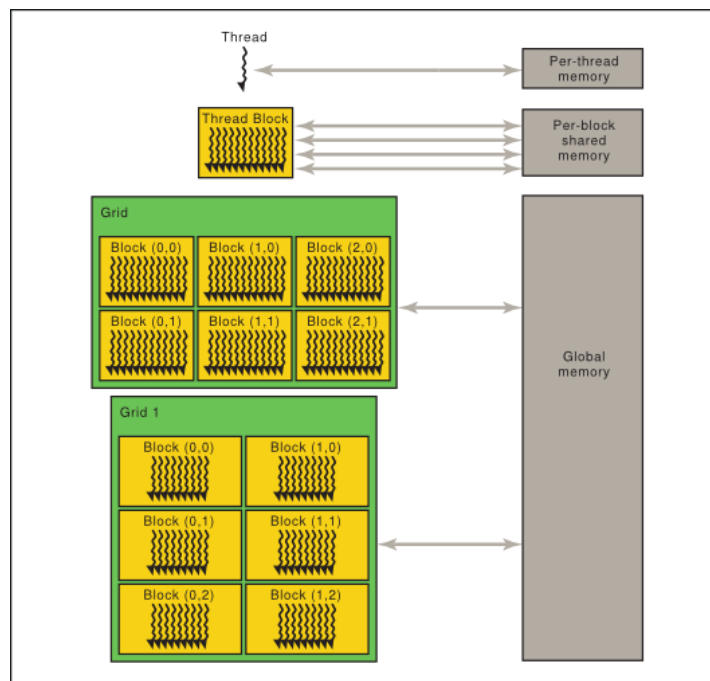
Desktop system



Embedded SoC system



GPU 内存优化是高性能的关键



Memory	Visibility	Heuristics/notes	GPU Coder support
Global memory	CPU + GPU	Share data b/w CPU and GPU	Yes
Local memory/registers	per GPU thread	Thread local data	Yes
Shared memory	per GPU block	Shared data between threads	Yes
Texture memory	CPU + GPU	Shared read-only data with 2D alignment	Future
Constant memory	GPU	Read-only constants	Yes

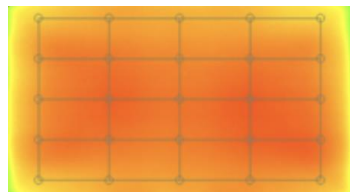
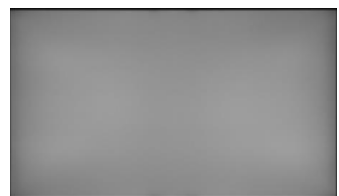
产品创新

- Deep Learning Toolbox
- MATLAB GPU Coder

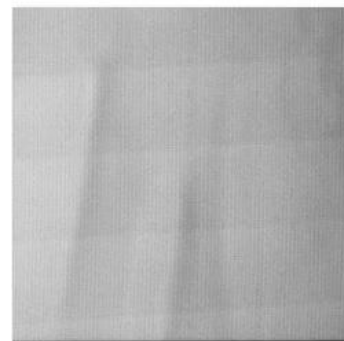


- FPGA+GPU协调处理
- 实现量与测的一体化
- 提升Tact Time 38.5%

亮度, 色度的计算和校准

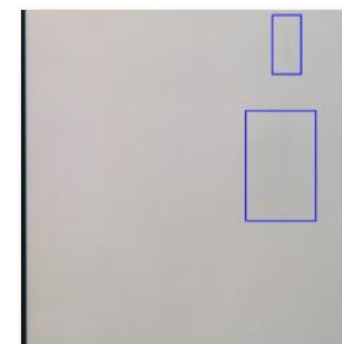
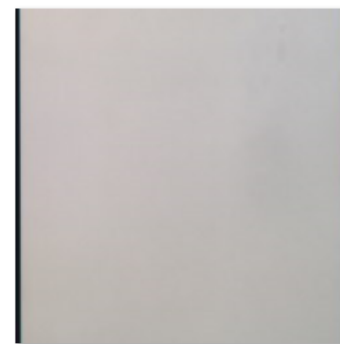


数据处理



图像处理算法运算

深度学习算法



基于深度学习的Mura检测

基于模型开发获的好处

产品研发管理



- MATLAB Graphics 给设计仿真、数据分析带来便利
- 快速的从理论模型到产品原型，缩短超过35%
- 产品工程化的质量(黄金参考，SIL和PIL)
- 简化流程和更小团队，减少约30%

MATLAB EXPO 2021

Thank you



© 2021 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.