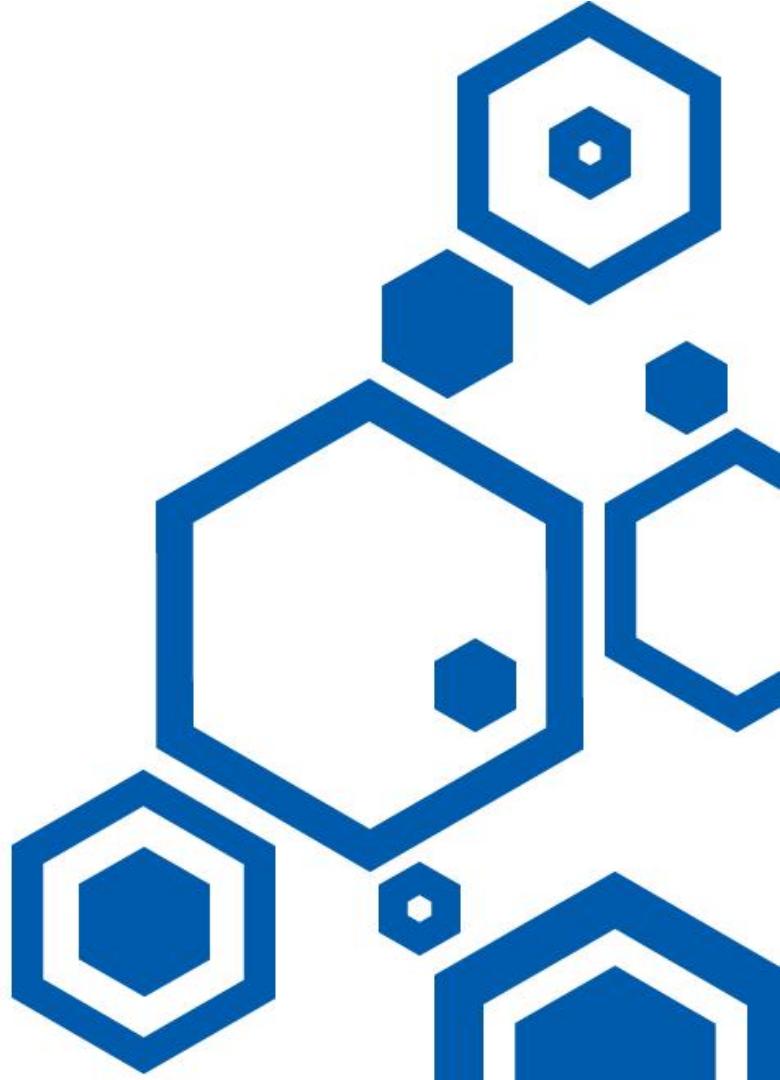


# 更高质量的成像

课程5-梯度域图像处理





# 课程资料来源及致谢

## 15-463, 15-663, 15-862 Computational Photography, Fall 2021



Time:	Mondays, Wednesdays 11:50 am - 1:10 pm ET
Location:	GHC 4303
Instructor:	Ioannis (Yannis) Gkioulekas
Teaching Assistants:	Alice Lai, Jenny Lin

许多图表引自2021年CMU计算摄影学课程讲义，因排版原因此后不再注明，感谢CMU及Ioannis教授的开源精神，  
课程链接：[http://graphics.cs.cmu.edu/courses/15-463/2021\\_fall/](http://graphics.cs.cmu.edu/courses/15-463/2021_fall/)

一些图像引自相关参考文献，在此讲义最后列出了参考阅读文献



## Removing Photography Artifacts using Gradient Projection and Flash-Exposure Sampling

Amit Agrawal

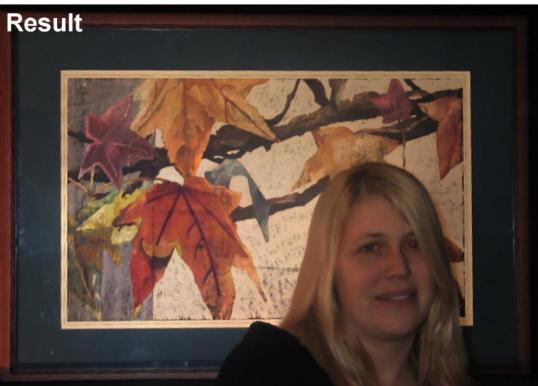
Mitsubishi Electric Research Labs (MERL), Cambridge, MA\*

Ramesh Raskar

Shree K. Nayar<sup>†</sup>

Yuanzhen Li

<sup>†</sup>Columbia University





# 目录

- ❖ 回顾闪光/无闪光图像融合的思想
- ❖ 关于梯度的基本理论知识
- ❖ 图像梯度的积分
- ❖ 一个基本的积分问题：泊松融合
- ❖ 泊松方程的加速求解算法
- ❖ 作业：用新的思想来解决闪光/无闪光图像融合
- ❖ 总结：梯度域图像处理



## 回顾第4课中的闪光-无闪光图像融合技术

开启闪光

关闭闪光





我们的想法是融合这两幅图像得到更好的结果





参考论文采用了双边滤波来获取图像的细节及进行去噪

## Digital Photography with Flash and No-Flash Image Pairs

Georg Petschnigg  
Richard Szeliski

Maneesh Agrawala  
Michael Cohen  
Microsoft Corporation

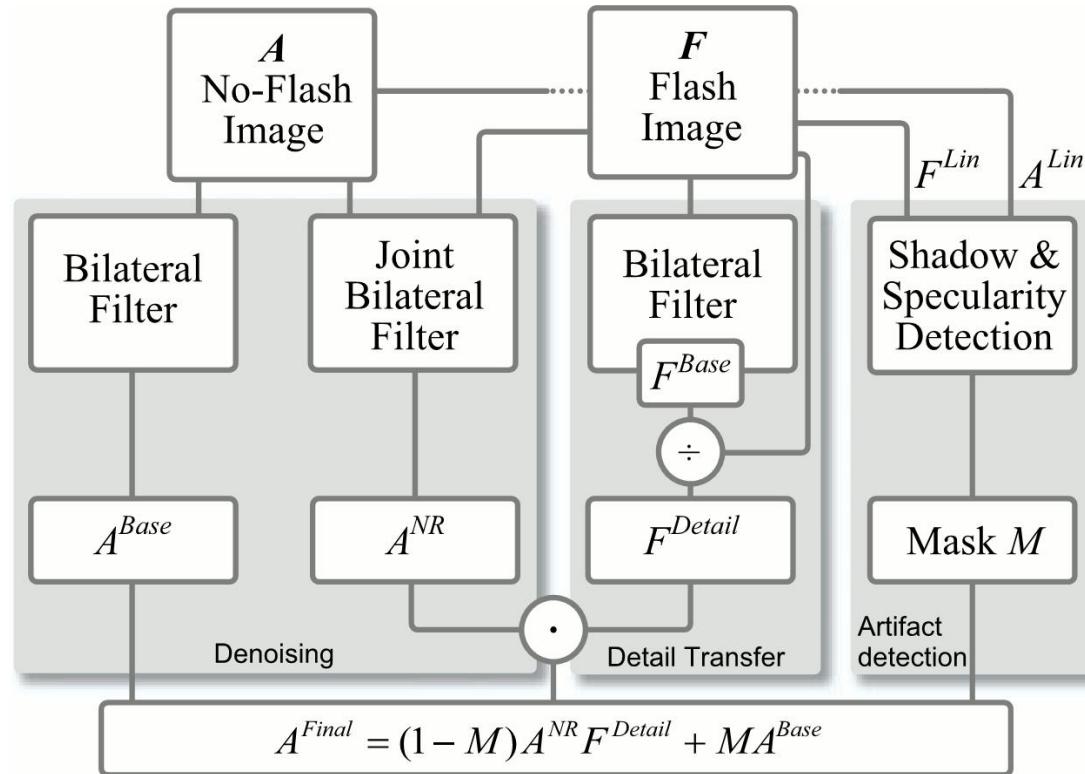
Hugues Hoppe  
Kentaro Toyama



Figure 1: This candlelit setting from the wine cave of a castle is difficult to photograph due to its low light nature. A flash image captures the high-frequency texture and detail, but changes the overall scene appearance to cold and gray. The no-flash image captures the overall appearance of the warm candlelight, but is very noisy. We use the detail information from the flash image to both reduce noise in the no-flash image and sharpen its detail. Note the smooth appearance of the brown leather sofa and crisp detail of the bottles. For full-sized images, please see the supplemental DVD or the project website <http://research.microsoft.com/projects/FlashNoFlash>.

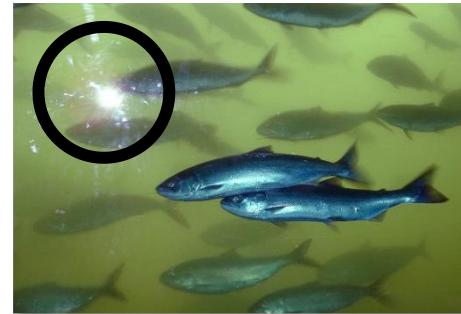


参考论文采用了双边滤波来获取图像的细节及进行去噪





## 并非所有闪光图像中的边缘都是真实的物体边缘



- 这会导致联合双边滤波的结果出现过模糊或者欠模糊
- 因此我们还需要处理这种缺陷



## 第4题：检测并处理闪光图像中的高光和阴影

**检测阴影的思想：**当被遮挡时产生阴影，此时目标像素的亮度应该非常接近无闪光图像

所以我们计算闪光图像和无闪光图像的差异，并从中选择小于某个阈值的像素作为阴影像素

**差异最好在线性空间进行，但我们仅提供了非线性的图像**

根据之前所讲的知识，请用下面的公式先把非线性的图像转为线性图像

$$C_{\text{linear}} = \begin{cases} \frac{C_{\text{non-linear}}}{12.92}, & C_{\text{non-linear}} \leq 0.0404482, \\ \left( \frac{C_{\text{non-linear}} + 0.055}{1.055} \right)^{2.4}, & C_{\text{non-linear}} > 0.0404482, \end{cases}$$



## 第4题：检测并处理闪光图像中的高光和阴影

**检测阴影的思想：**当被遮挡时产生阴影，此时目标像素的亮度应该非常接近无闪光图像

所以我们计算闪光图像和无闪光图像的差异，并从中选择小于某个阈值的像素作为阴影像素

然后用下面的公式来利用阈值进行阴影检测，得到一个Mask图像 $M^{Shad}$

$$M^{Shad} = \begin{cases} 1 & \text{when } F^{Lin} - A^{Lin} \leq \tau_{Shad} \\ 0 & \text{else} \end{cases}.$$

由于噪声的影响，其中很可能有许多小洞、孤岛区域等等，因此你还需要用形态学处理来处理 $M^{Shad}$ ，包括利用腐蚀操作去除鼓捣，并填充空洞等，最后再用膨胀操作来得到完整的阴影Mask



## 第4题：检测并处理闪光图像中的高光和阴影

**检测高光的思想：**在闪光线性图像 $F^{lin}$ 中高光会导致饱和，因此通过检测其中大于95%最大值的像素来得到高光区域

由于噪声的影响，其中很可能有许多小洞、孤岛区域等等，因此你还需要用形态学处理来处理高光区域，包括利用腐蚀操作去除鼓捣，并填充空洞等，最后再用膨胀操作来得到完整的高光Mask



## 第4题：检测并处理闪光图像中的高光和阴影

**合并高光和阴影：**直接对阴影Mask和高光Mask取并集，并对结果做高斯模糊以羽化边缘，避免不自然的边缘，最后得到一个合并后的 $M$

联合双边滤波的无闪光图

第3题的细节图像如下：

$$A^{\text{Detail}} = A^{\text{NR}} \frac{F + \epsilon}{F^{\text{Base}} + \epsilon},$$

基本双边滤波的闪光图像

你要按下面的公式得到第4题的最终图像：

基本双边滤波的无闪光图像

$$A^{\text{Final}} = (1 - M) \cdot A^{\text{Detail}} + M \cdot A^{\text{Base}}.$$

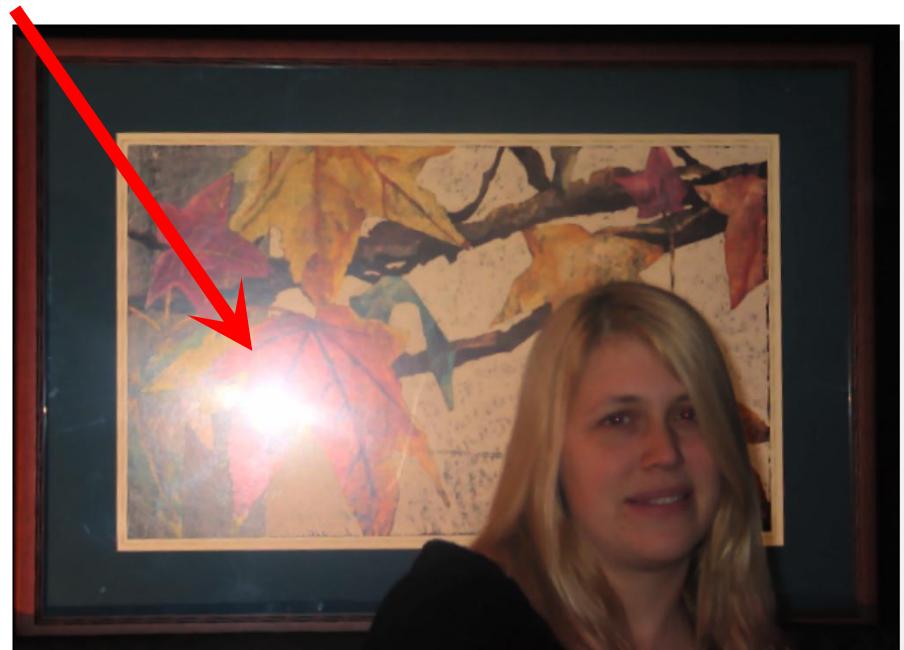
细节图

- ◆ 第4课所用的技术难以处理下面的问题



## 强烈的反光区域

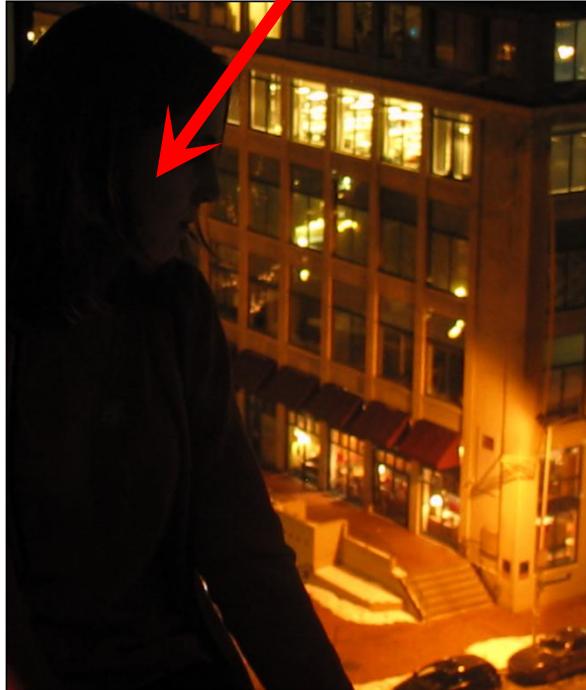
闪光灯导致的高光热点





## 闪光图像在玻璃上形成的反射

无闪光图像的前景欠曝



闪光图像导致的反射





## 无闪光图像的自反射与闪光图像的高光同时存在导致更加困难的问题





该如何解决这些问题？



# 第5课的核心参考论文，也是我们第5课作业的重要来源

## Removing Photography Artifacts using Gradient Projection and Flash-Exposure Sampling

Amit Agrawal

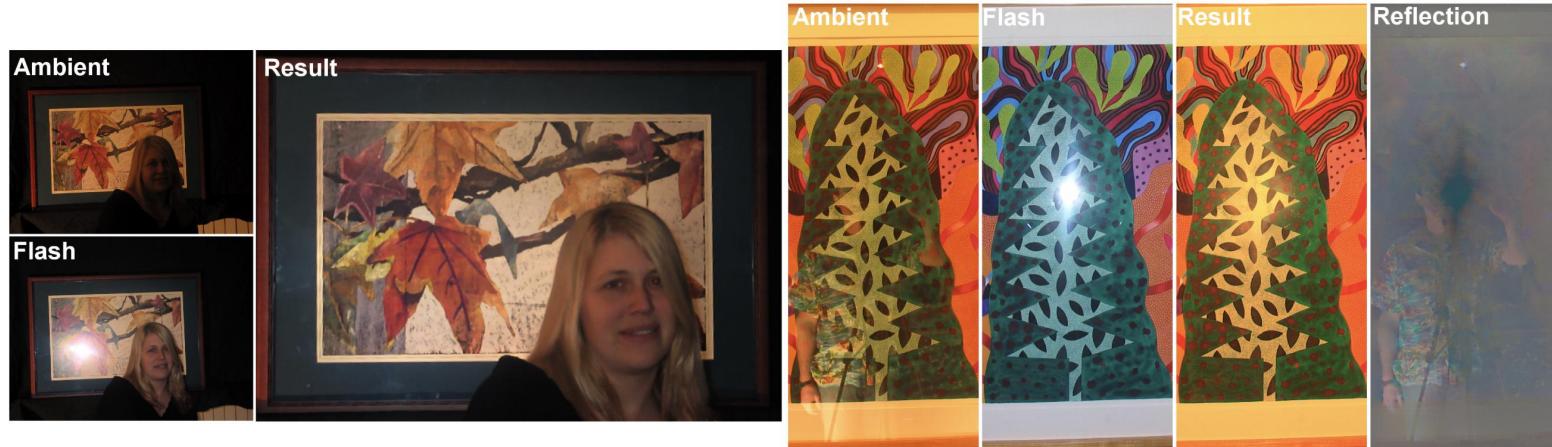
Mitsubishi Electric Research Labs (MERL), Cambridge, MA\*

Ramesh Raskar

Shree K. Nayar<sup>†</sup>

Yuanzhen Li

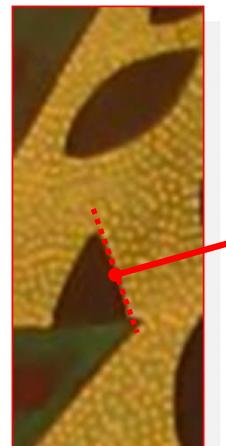
<sup>†</sup>Columbia University





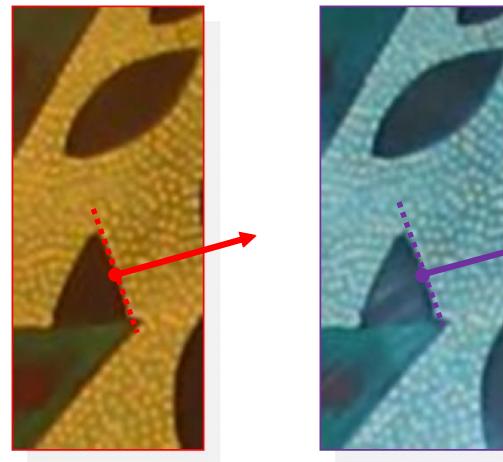
## 让我们观察闪光图像和无闪光图像的梯度信息

在非强反射区域  
它们具有一致的梯度方向



闪光图像的梯度向量

无闪光图像的梯度向量



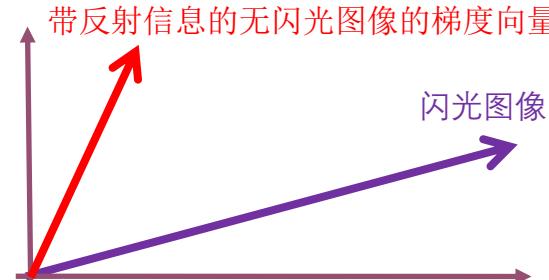
→



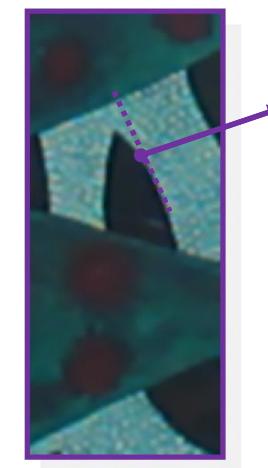
## 让我们观察闪光图像和无闪光图像的梯度信息

在强反射区域

它们具有不同的梯度方向



闪光图像的梯度向量





## 采用梯度向量投影的思想来消除反射



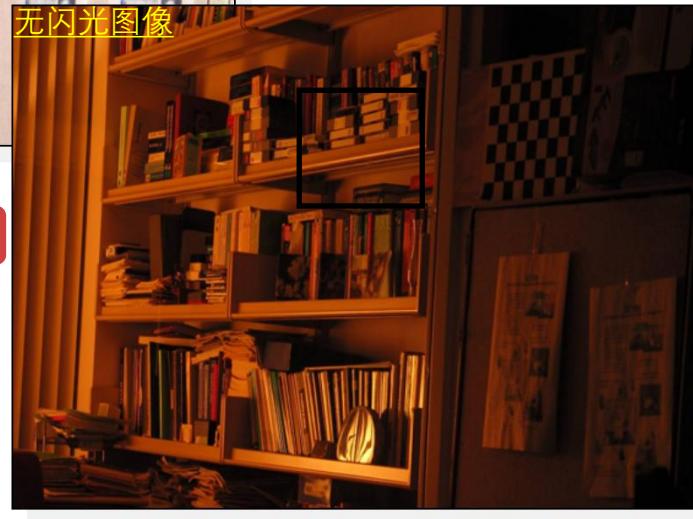


梯度向量投影的思想可以解决更多的问题

闪光图像



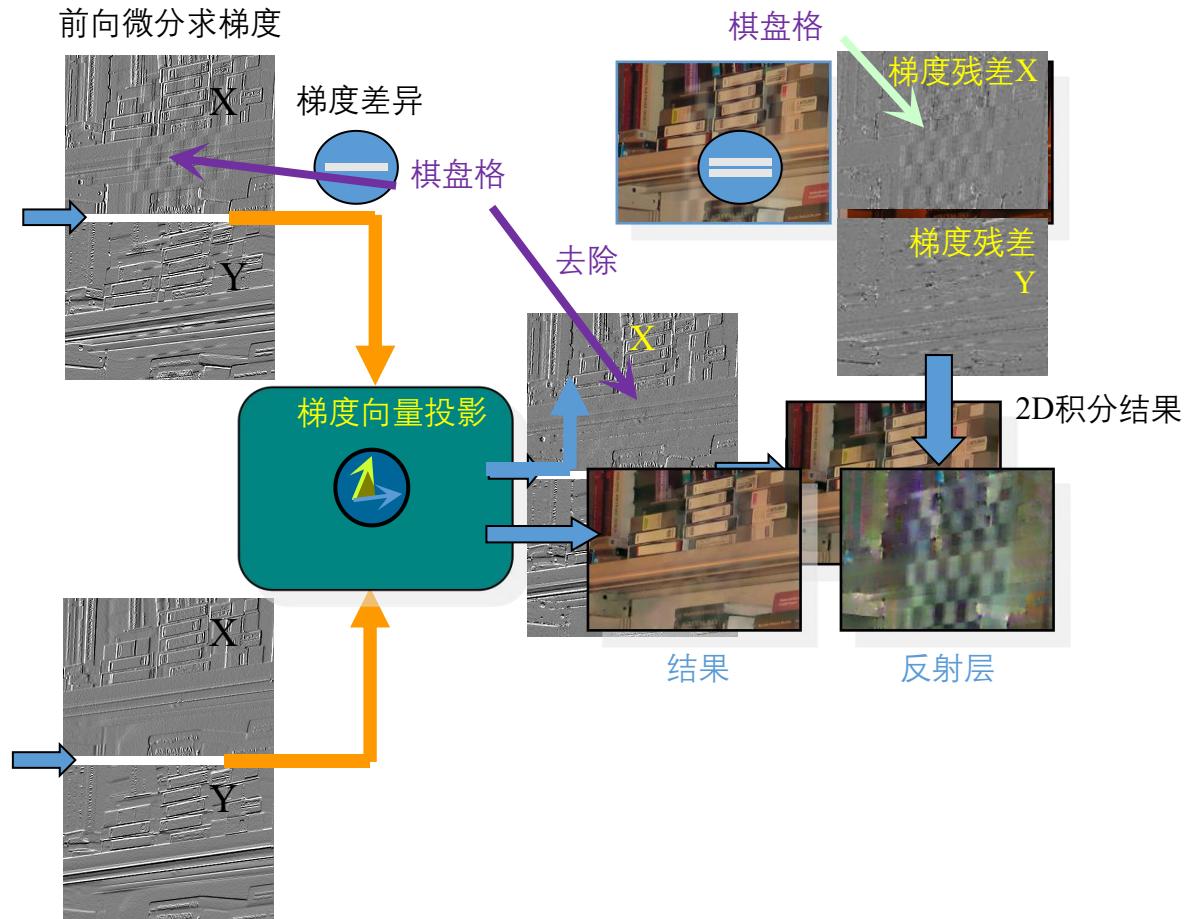
无闪光图像



玻璃外的棋盘格

玻璃上的反射图案

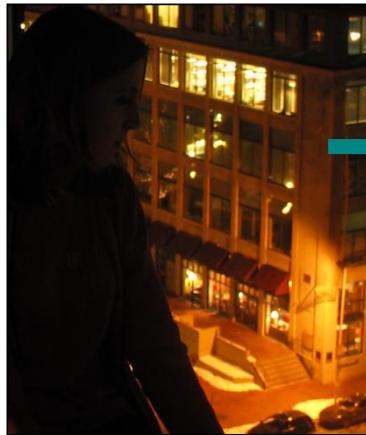
◆ 闪光图像及其梯度





# 消除闪光带来的反射

无闪光图像



闪光图像



梯度向量投影



Result

无闪光+闪光



反射  
消除



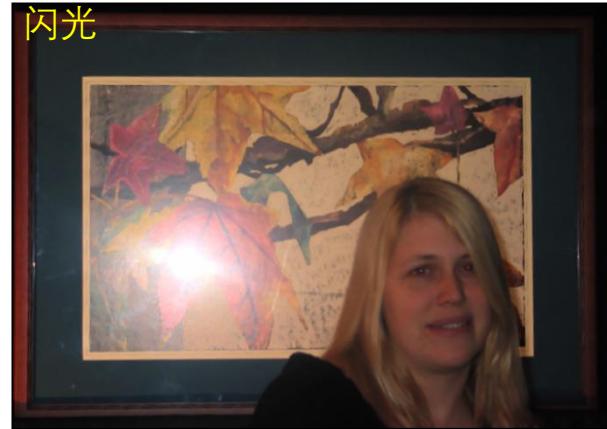
## 两种图像的梯度并非处处一致，需要小心处理



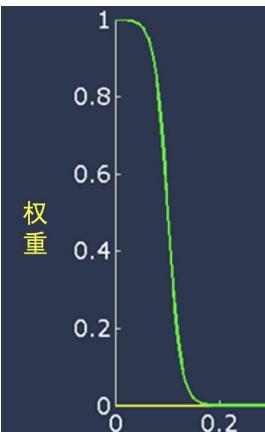
无闪光



闪光



结果

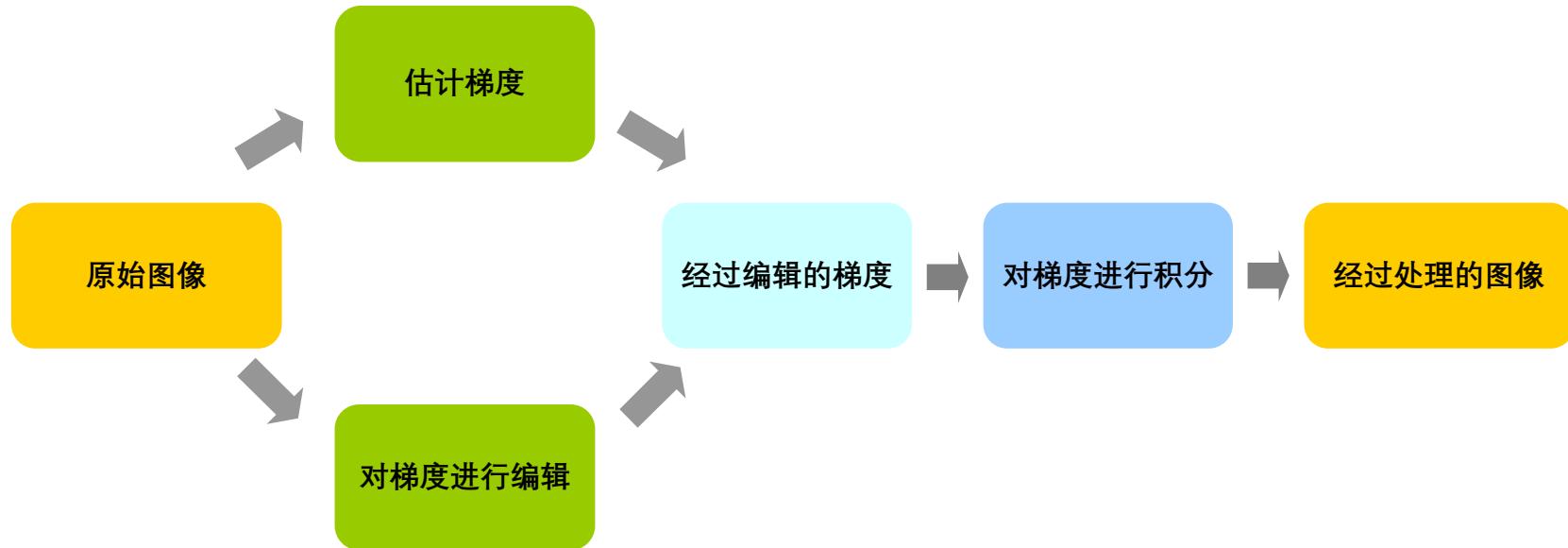


梯度方向一致性约束  
闪光图像的梯度

- 前面讲的本质上是一种梯度域的图像处理



## 梯度域图像处理的思想



例如前面讲的，就是对梯度  
进行投影，去除与基础图像  
梯度方向不一致的梯度信息



# 目录

- ◆ 回顾闪光/无闪光图像融合的思想
- ◆ 关于梯度的基本理论知识
- ◆ 图像梯度的积分
- ◆ 一个基本的积分问题：泊松融合
- ◆ 泊松方程的加速求解算法
- ◆ 作业：用新的思想来解决闪光/无闪光图像融合
- ◆ 总结：梯度域图像处理



## 2D图像的标量场和向量场

**标量场：**在空间中为每个点赋以一个值

$$I(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$$

**向量场：**在空间中为每个点赋以一个向量

$$[u(x, y) \quad v(x, y)] : \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

**示例：**

- 一个单色灰度图像是一个标量场
- 一个两通道的图像是一个向量场
- 一个3通道图像（例如RGB图）也是一个向量场，不过是3维的



## Nabla操作相关的知识

**Nabla (或del): 向量微分算子**

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix}$$

你可以把它看做一个2维向量



## Nabla操作相关的知识

**Nabla (或del):** 向量微分算子

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix}$$

你可以把它看做一个2维向量

**梯度-Gradient (grad):** Nabla操作和标量场的乘积

$$\nabla I(x, y) = ?$$

**散度-Divergence:** Nabla和向量场的内积

$$\nabla \cdot [u(x, y) \ v(x, y)] = ?$$

**旋度-Curl:** Nabla和向量场的叉积

$$\nabla \times [u(x, y) \ v(x, y)] = ?$$



## Nabla操作相关的知识

**Nabla (或del):** 向量微分算子

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix}$$

你可以把它看做一个2维向量

**梯度-Gradient (grad):** Nabla操作和标量场的乘积

$$\nabla I(x, y) = \begin{bmatrix} \frac{\partial I}{\partial x}(x, y) & \frac{\partial I}{\partial y}(x, y) \end{bmatrix}$$

维度?

**散度-Divergence:** Nabla和向量场的内积

$$\nabla \cdot [u(x, y) \ v(x, y)] = \frac{\partial u}{\partial x}(x, y) + \frac{\partial v}{\partial y}(x, y)$$

维度?

**旋度-Curl:** Nabla和向量场的叉积

$$\nabla \times [u(x, y) \ v(x, y)] = \left( \frac{\partial v}{\partial x}(x, y) - \frac{\partial u}{\partial y}(x, y) \right) \hat{k}$$

维度?



## Nabla操作相关的知识

**Nabla (或del):** 向量微分算子

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix}$$

你可以把它看做一个2维向量

**梯度-Gradient (grad):** Nabla操作和标量场的乘积

$$\nabla I(x, y) = \begin{bmatrix} \frac{\partial I}{\partial x}(x, y) & \frac{\partial I}{\partial y}(x, y) \end{bmatrix}$$

2维向量场

**散度-Divergence:** Nabla和向量场的内积

$$\nabla \cdot [u(x, y) \ v(x, y)] = \frac{\partial u}{\partial x}(x, y) + \frac{\partial v}{\partial y}(x, y)$$

标量场

**旋度-Curl:** Nabla和向量场的叉积

$$\nabla \times [u(x, y) \ v(x, y)] = \left( \frac{\partial v}{\partial x}(x, y) - \frac{\partial u}{\partial y}(x, y) \right) \hat{k}$$

3维向量场



## Nabla操作相关的知识

**Nabla (或del):** 向量微分算子

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix}$$

你可以把它看做一个2维向量

**梯度-Gradient (grad):** Nabla操作和标量场的乘积

$$\nabla I(x, y) = \begin{bmatrix} \frac{\partial I}{\partial x}(x, y) & \frac{\partial I}{\partial y}(x, y) \end{bmatrix}$$

2维向量场

**散度-Divergence:** Nabla和向量场的内积

$$\nabla \cdot [u(x, y) \ v(x, y)] = \frac{\partial u}{\partial x}(x, y) + \frac{\partial v}{\partial y}(x, y)$$

标量场

**旋度-Curl:** Nabla和向量场的叉积

$$\nabla \times [u(x, y) \ v(x, y)] = \left( \frac{\partial v}{\partial x}(x, y) - \frac{\partial u}{\partial y}(x, y) \right) \hat{k}$$

黄色部分是一个标量场



## Nabla相关的组合操作

梯度的旋度：

$$\nabla \times \nabla I(x, y) = ?$$

梯度的散度：

$$\nabla \cdot \nabla I(x, y) = ?$$



## Nabla相关的组合操作

梯度的旋度：

$$\nabla \times \nabla I(x, y) = \frac{\partial^2}{\partial y \partial x} I(x, y) - \frac{\partial^2}{\partial x \partial y} I(x, y)$$

梯度的散度：

$$\nabla \cdot \nabla I(x, y) = \frac{\partial^2}{\partial x^2} I(x, y) + \frac{\partial^2}{\partial y^2} I(x, y) \equiv \Delta I(x, y)$$



## Nabla相关的组合操作

梯度的旋度：

$$\nabla \times \nabla I(x, y) = \frac{\partial^2}{\partial y \partial x} I(x, y) - \frac{\partial^2}{\partial x \partial y} I(x, y)$$

梯度的散度：

$$\nabla \cdot \nabla I(x, y) = \frac{\partial^2}{\partial x^2} I(x, y) + \frac{\partial^2}{\partial y^2} I(x, y) \equiv \Delta I(x, y)$$

拉普拉斯: 标量微分算子

Nabla算子的自身内积  $\Delta \equiv \nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$



# 简洁的记号

**Nabla (或del):** 向量微分算子

$$\nabla = [ \quad_x \quad_y ]$$

你可以把它看做一个2维向量

**梯度-Gradient (grad):** Nabla操作和标量场的乘积

$$\nabla I = [I_x \quad I_y]$$

2维向量场

**散度-Divergence:** Nabla和向量场的内积

$$\nabla \cdot [u \quad v] = u_x + v_y$$

标量场

**旋度-Curl:** Nabla和向量场的叉积

$$\nabla \times [u \quad v] = (v_x - u_y) \hat{k}$$

黑体部分为标量场



# 简洁记号

梯度的旋度：

$$\nabla \times \nabla I = I_{yx} - I_{xy}$$

梯度的散度：

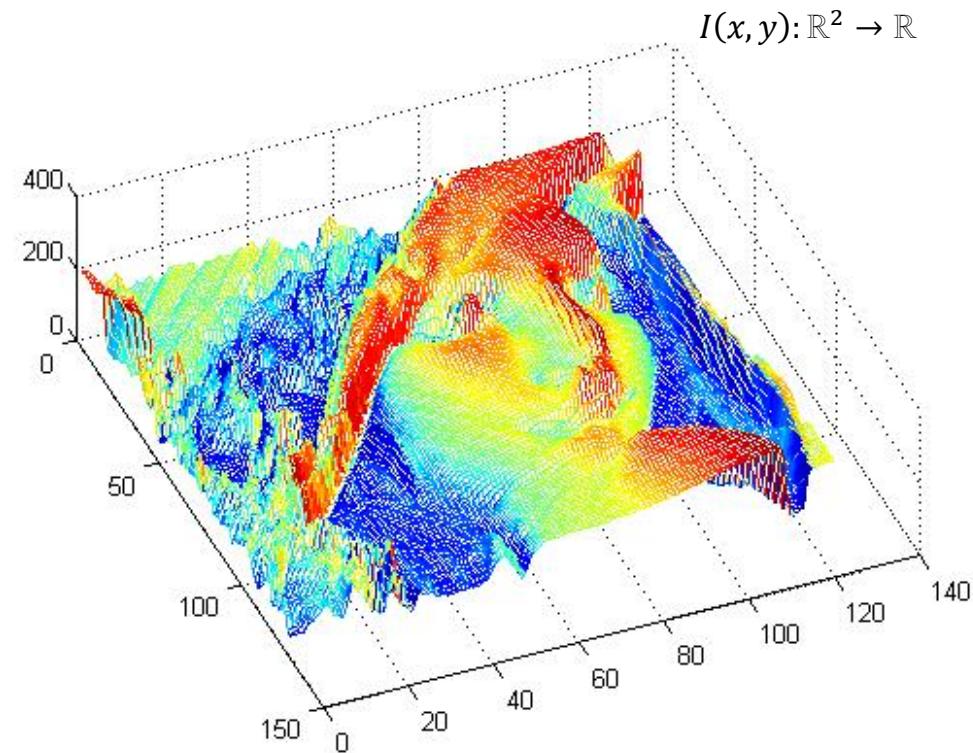
$$\nabla \cdot \nabla I = I_{xx} + I_{yy} \equiv \Delta I$$

拉普拉斯: 标量微分算子

Nabla算子的自身内积     $\Delta \equiv \nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$



单通道图像本身是一个2维标量函数

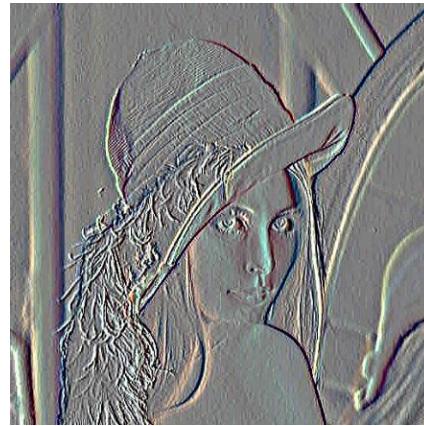




## 图像梯度操作将标量场转换为了向量场



标量场  $I(x, y): \mathbb{R}^2 \rightarrow \mathbb{R}$



向量场  $\nabla I(x, y) = \left[ \frac{\partial I}{\partial x}(x, y) \quad \frac{\partial I}{\partial y}(x, y) \right]$





## 离散数字图像的梯度用有限微分来近似

对于连续函数，求导可用前向微分来计算

$$\frac{\partial I}{\partial x}(x, y) = \lim_{h \rightarrow 0} \frac{I(x + h, y) - I(x, y)}{h}$$

而对于离散向量场，我们设 $h=1$ 来求前向微分

$$\frac{\partial I}{\partial x}(x, y) = I(x + 1, y) - I(x, y)$$

这个操作可用上一节课提到的卷积来完成



## 离散数字图像的梯度用有限微分来近似

对于连续函数，求导可用前向微分来计算

$$\frac{\partial I}{\partial x}(x, y) = \lim_{h \rightarrow 0} \frac{I(x + h, y) - I(x, y)}{h}$$

而对于离散向量场，我们设 $h=1$ 来求前向微分

$$\frac{\partial I}{\partial x}(x, y) = I(x + 1, y) - I(x, y)$$

X方向偏导滤波

1	-1
---	----

通常采用中心微分实现，不过本课我们采用邻域微分来实现

$$\frac{\partial I}{\partial x}(x, y) = \frac{I(x + 1, y) - I(x - 1, y)}{2}$$

Y方向偏导滤波

1
-1



## 离散图像的拉普拉斯操作

如何实现图像的拉普拉斯:

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



# 离散图像的拉普拉斯操作

如何实现图像的拉普拉斯:

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

采用多个偏导滤波器来实现

$$\begin{array}{c} \boxed{1} \quad \boxed{-1} \\ \boxed{1} \quad \boxed{-1} \end{array} * \begin{array}{c} \boxed{1} \quad \boxed{-1} \end{array} + \begin{array}{c} \boxed{1} \\ \hline \boxed{-1} \end{array} * \begin{array}{c} \boxed{1} \\ \hline \boxed{-1} \end{array} = ?$$

?



# 离散图像的拉普拉斯操作

如何实现图像的拉普拉斯:

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

特别注意:

- 拉普拉斯和微分需采用一致的核来实现
- 对于图像的边缘需要用补齐(Padding)等方式仔细处理

采用多个偏导滤波器来实现

拉普拉斯滤波器

$$\underbrace{\begin{array}{|c|c|} \hline 1 & -1 \\ \hline \end{array} * \begin{array}{|c|c|} \hline 1 & -1 \\ \hline \end{array}}_{\frac{\partial^2 I}{\partial x^2}(x, y)} + \underbrace{\begin{array}{|c|c|} \hline 1 \\ \hline -1 \\ \hline \end{array} * \begin{array}{|c|c|} \hline 1 \\ \hline -1 \\ \hline \end{array}}_{\frac{\partial^2 I}{\partial y^2}(x, y)} = \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

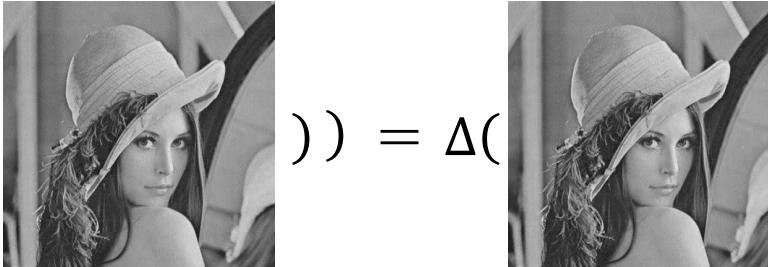


# 正确实现微分算子和拉普拉斯算子

- 拉普拉斯和微分需采用一致的核来实现
- 对于图像的边缘需要用补齐(Padding)等方式仔细处理

正确实现的拉普拉斯算子和微分算子需要通过如下测试（除了上下左右四边）

完善的图像处理库可能包括以不同的方式实现的这些算子

$$\nabla \cdot (\nabla( \underbrace{\text{梯度}}_{\text{散度}} )) = \Delta( \underbrace{\text{拉普拉斯}}_{\text{散度}} )$$




那么有可能从图像的梯度反过来恢复原始图像吗？



标量场  $I(x, y): \mathbb{R}^2 \rightarrow \mathbb{R}$



$$\text{梯度场 } \nabla I(x, y) = \left[ \frac{\partial I}{\partial x}(x, y) \quad \frac{\partial I}{\partial y}(x, y) \right]$$



那么有可能从图像的梯度反过来恢复原始图像吗？

这涉及到一个问题：向量场的积分



标量场  $I(x, y): \mathbb{R}^2 \rightarrow \mathbb{R}$



$$\text{梯度场 } \nabla I(x, y) = \left[ \frac{\partial I}{\partial x}(x, y) \quad \frac{\partial I}{\partial y}(x, y) \right]$$



# 目录

- ◆ 回顾闪光/无闪光图像融合的思想
- ◆ 关于梯度的基本理论知识
- ◆ 图像梯度的积分
- ◆ 一个基本的积分问题：泊松融合
- ◆ 泊松方程的加速求解算法
- ◆ 作业：用新的思想来解决闪光/无闪光图像融合
- ◆ 总结：梯度域图像处理



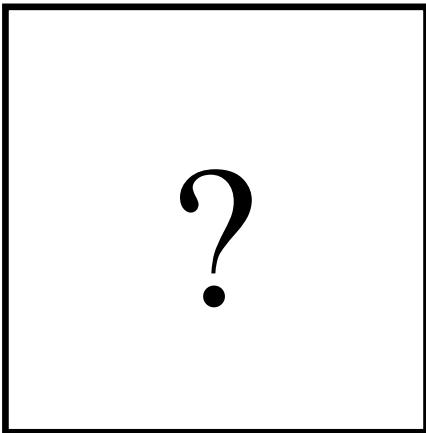
## 关于向量场积分的两个基础问题

- 什么时候可以对一个向量场进行积分？
- 如何实现对向量场的积分？



# 可积分的向量场

- 给定任意的向量场( $u, v$ ), 我们是否总是可以通过积分转换为标量场 $I$ ?



$$I(x, y): \mathbb{R}^2 \rightarrow \mathbb{R}$$



$$u(x, y): \mathbb{R}^2 \rightarrow \mathbb{R}$$



$$v(x, y): \mathbb{R}^2 \rightarrow \mathbb{R}$$

这里

$$\frac{\partial I}{\partial x}(x, y) = u(x, y)$$

$$\frac{\partial I}{\partial y}(x, y) = v(x, y)$$



## 2次可微函数的性质

这种函数满足梯度场的旋度为0

$$\nabla \times \nabla I = I_{yx} - I_{xy} = 0$$

从直觉上，这意味着什么？



## 2次可微函数的性质

这种函数满足梯度场的旋度为0

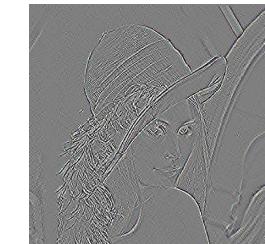
$$\nabla \times \nabla I = I_{yx} - I_{xy} = 0$$

从直觉上，这意味着什么？ 意味着结果与微分的顺序无关

$$I_{yx} = I_{xy}$$



# 示例

 $I$  $I_x$  $I_y$  $\Delta I$  $\nabla \times \nabla I$  $I_{xy}$  $I_{yx}$  $=$



## 2次可微函数的性质

这种函数满足梯度场的旋度为0

$$\nabla \times \nabla I = I_{yx} - I_{xy} = 0$$

从直觉上，这意味着什么？意味着结果与微分的顺序无关

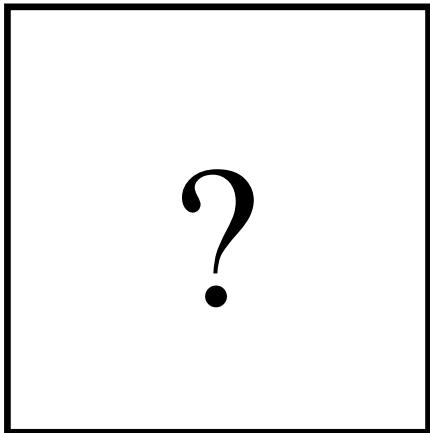
$$I_{yx} = I_{xy}$$

如何利用这个性质推导出“可积分”的条件呢？



# 可积分的向量场

- 给定任意的向量场( $u, v$ ), 我们是否总是可以通过积分转换为标量场I?



$$I(x, y): \mathbb{R}^2 \rightarrow \mathbb{R}$$



$$u(x, y): \mathbb{R}^2 \rightarrow \mathbb{R}$$



$$v(x, y): \mathbb{R}^2 \rightarrow \mathbb{R}$$

这里

$$\frac{\partial I}{\partial x}(x, y) = u(x, y)$$

$$\frac{\partial I}{\partial y}(x, y) = v(x, y)$$

仅当:

$$\nabla \times \begin{bmatrix} u(x, y) \\ v(x, y) \end{bmatrix} = 0 \Rightarrow \frac{\partial u}{\partial y}(x, y) = \frac{\partial v}{\partial x}(x, y)$$



## 关于向量场积分的两个基础问题

- 什么时候可以对一个向量场进行积分? **利用旋度检验混合2阶偏导是否相等**
- 如何实现对向量场的积分?

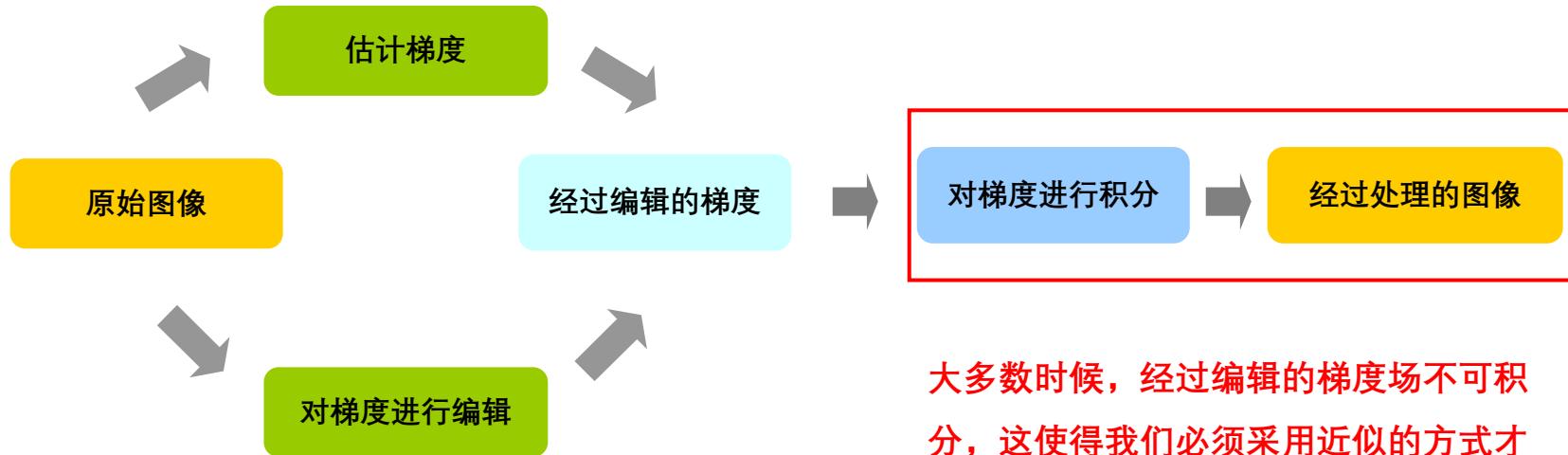


## 关于向量场积分的两个基础问题

- 什么时候可以对一个向量场进行积分？利用旋度检验混合2阶偏导是否相等
- 如何实现对向量场的积分？这里将遇到重要的挑战



# 通过梯度积分恢复图像的应用和挑战



大多数时候，经过编辑的梯度场不可积分，这使得我们必须采用近似的方式才能恢复图像

例如前面讲的，就是对梯度进行投影，去除与基础图像梯度方向不一致的梯度信息



# 目录

- ❖ 回顾闪光/无闪光图像融合的思想
- ❖ 关于梯度的基本理论知识
- ❖ 图像梯度的积分
- ❖ 一个基本的积分问题：泊松融合
- ❖ 泊松方程的加速求解算法
- ❖ 作业：用新的思想来解决闪光/无闪光图像融合
- ❖ 总结：梯度域图像处理



## 泊松融合的应用



原图



直接拷贝粘贴融合



泊松融合



泊松融合的核心思想是保持融合区域的梯度平滑



原图



目标图



直接拷贝粘贴融合



泊松融合



## 定义和符号



把 $g$ 插入到 $S$ 的 $\Omega$ 区域，插入后用 $f$ 来描述，背景区域用 $f^*$ 来描述

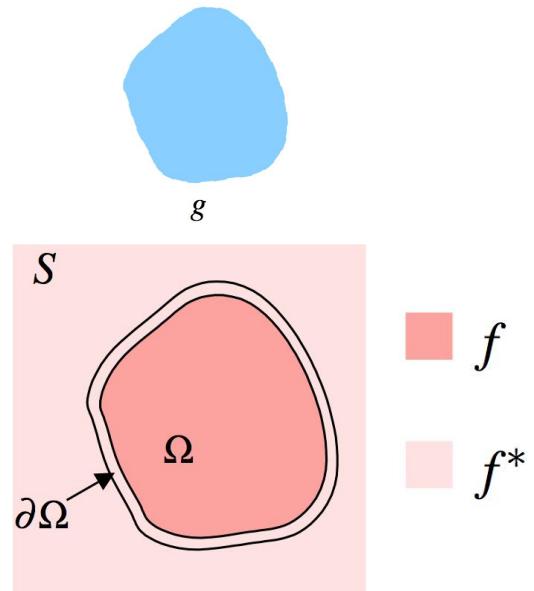
$g$ : 源图

$S$ : 目标图

$\Omega$ : 目标区域

$f$ : 插入后的函数表示

$f^*$ : 背景区域的函数表示





## 定义和符号



把 $g$ 插入到 $S$ 的 $\Omega$ 区域，插入后用 $f$ 来描述，背景区域用 $f^*$ 来描述

$g$ : 源图

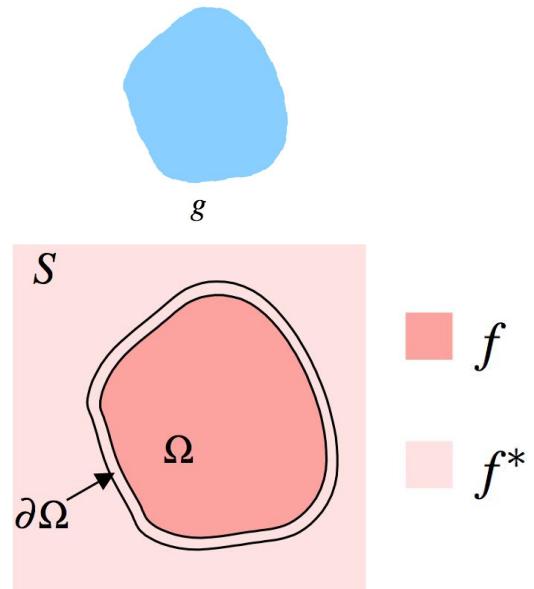
$S$ : 目标图

$\Omega$ : 目标区域

$f$ : 插入后的函数表示

$f^*$ : 背景区域的函数表示

这里 $f$ 是未知的





# 定义和符号



把 $g$ 插入到 $S$ 的 $\Omega$ 区域，插入后用 $f$ 来描述，背景区域用 $f^*$ 来描述

$g$ : 源图

$S$ : 目标图

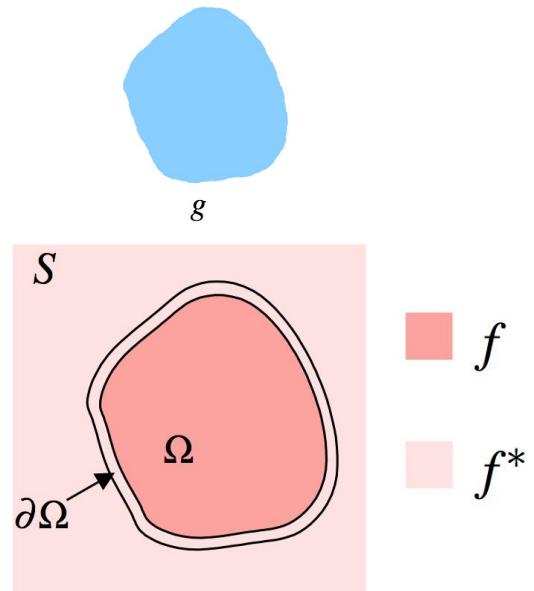
$\Omega$ : 目标区域

$f$ : 插入后的函数表示

$f^*$ : 背景区域的函数表示

如何求解 $f$ ?

- 它应该像 $g$ 吗?
- 或者它应该像 $f^*$ 吗?



$f$   
 $f^*$



# 定义和符号



把 $g$ 插入到 $S$ 的 $\Omega$ 区域，插入后用 $f$ 来描述，背景区域用 $f^*$ 来描述

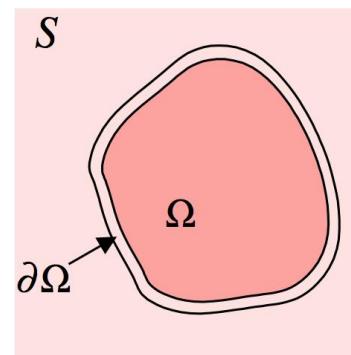
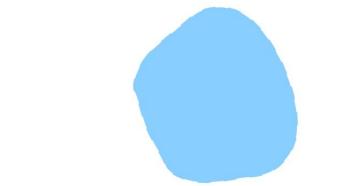
$g$ : 源图

$S$ : 目标图

$\Omega$ : 目标区域

$f$ : 插入后的函数表示

$f^*$ : 背景区域的函数表示



$f$ 应该满足：

- 在  $\Omega$  内部,  $\nabla f = \nabla g$ , 这样它保留了 $g$ 的特征
- 在边缘  $\partial\Omega$  处,  $f = f^*$ . 这样融合边界又很平滑



# 泊松融合就是对 $\nabla g$ 在狄利克雷边界条件下进行积分来恢复图像的过程



把 $g$ 插入到 $S$ 的 $\Omega$ 区域，插入后用 $f$ 来描述，背景区域用 $f^*$ 来描述

$g$ : 源图

$S$ : 目标图

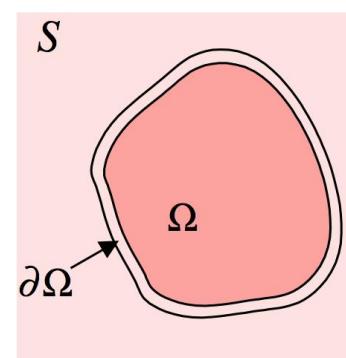
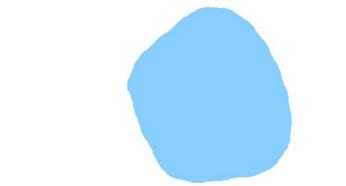
$\Omega$ : 目标区域

$f$ : 插入后的函数表示

$f^*$ : 背景区域的函数表示

$f$ 应该满足：

- 在  $\Omega$  内部,  $\nabla f = \nabla g$ , 这样它保留了 $g$ 的特征
- 在边缘  $\partial\Omega$  处,  $f = f^* \rightarrow$  狄利克雷 Dirichlet 边界条件



$f$

$f^*$



## 利用最小二乘积分求解泊松问题



## 把前面的问题转换为最小二乘积分来求解，这是一个变分优化问题

“变分”指的是未知数是整个函数的最优化

变分问题

$$\min_f \iint_{\Omega} |\nabla f - \mathbf{v}|^2 \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

?

?

回忆一下：

Nabla算子的定义

$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

这是什么

$$\mathbf{v} = (u, v)$$



## 把前面的问题转换为最小二乘积分来求解，这是一个变分优化问题

“变分”指的是未知数是整个函数的最优化

变分问题

$$\min_f \iint_{\Omega} |\nabla f - \mathbf{v}|^2 \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

f的梯度与向量场v相近

在边界处f与f\*相等

回忆一下：

Nabla算子的定义

$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

是的，这是我们已知的g的梯度场

$$\mathbf{v} = (u, v)$$



## 变分问题的极值点是带Dirichlet边界条件的泊松方程的解

用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

参考论文：Pérez et al., “Poisson Image Editing,” SIGGRAPH 2003.

回忆一下 ...

拉普拉斯

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

输入梯度场

$$\mathbf{v} = (u, v)$$

散度

$$\operatorname{div} \mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$$



## 变分问题的极值点是带Dirichlet边界条件的泊松方程的解

用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

f的拉普拉斯等于梯度场v的散度

回忆一下 ...

拉普拉斯  $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$

输入梯度场

$$\mathbf{v} = (u, v)$$

散度  $\operatorname{div} \mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$



## 变分问题的极值点是带Dirichlet边界条件的泊松方程的解

用欧拉-拉格朗日方程可以得出

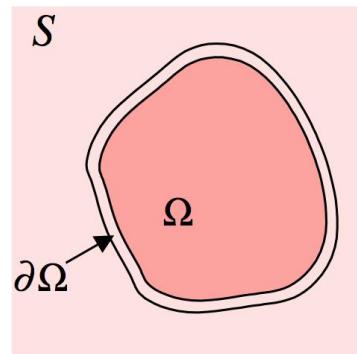
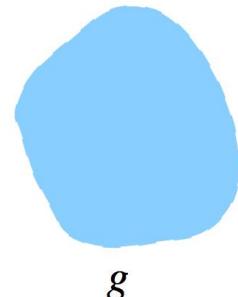
$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

$f$ 应该满足：

- 在  $\Omega$  内部， $\nabla f = \nabla g$ ，这样它保留了  $g$  的特征
- 在边缘  $\partial\Omega$  处， $f = f^*$ 。这样融合边界又很平滑

在泊松融合中输入梯度场应该是？

$$\mathbf{v} = (u, v) =$$





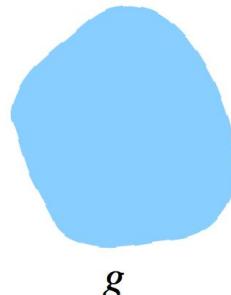
## 变分问题的极值点是带Dirichlet边界条件的泊松方程的解

用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

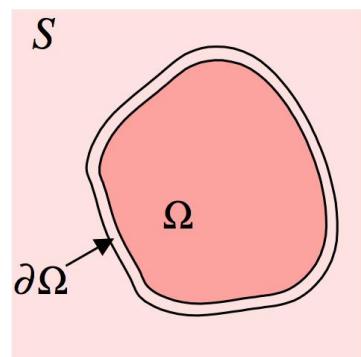
$f$ 应该满足：

- 在  $\Omega$  内部， $\nabla f = \nabla g$ ，这样它保留了  $g$  的特征
- 在边缘  $\partial\Omega$  处， $f = f^*$ 。这样融合边界又很平滑



在泊松融合中输入梯度场应该是

$$\mathbf{v} = (u, v) = \nabla g$$





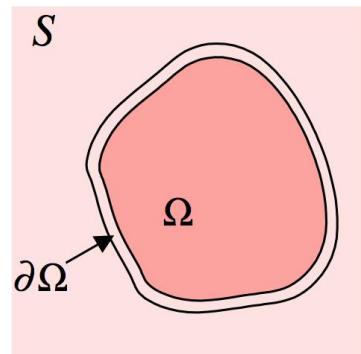
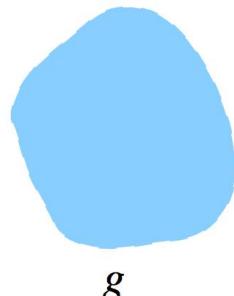
## 变分问题的极值点是带Dirichlet边界条件的泊松方程的解

用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

$f$ 应该满足：

- 在  $\Omega$  内部， $\nabla f = \nabla g$ ，这样它保留了  $g$  的特征
- 在边缘  $\partial\Omega$  处， $f = f^*$ 。这样融合边界又很平滑



在泊松融合中输入梯度场应该是

$$\mathbf{v} = (u, v) = \nabla g$$

在泊松融合中输入梯度场的散度是？

$$\operatorname{div} \mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$$



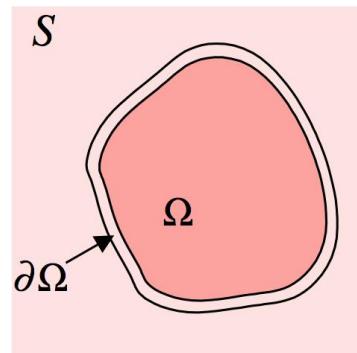
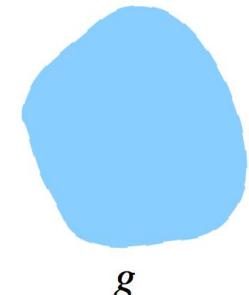
## 变分问题的极值点是带Dirichlet边界条件的泊松方程的解

用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

$f$ 应该满足：

- 在  $\Omega$  内部， $\nabla f = \nabla g$ ，这样它保留了  $g$  的特征
- 在边缘  $\partial\Omega$  处， $f = f^*$ 。这样融合边界又很平滑



在泊松融合中输入梯度场应该是

$$\mathbf{v} = (u, v) = \nabla g$$

在泊松融合中输入梯度场的散度是？

$$\operatorname{div} \mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = \Delta g$$



## 变分问题的极值点是带Dirichlet边界条件的泊松方程的解

用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

$f$ 应该满足：

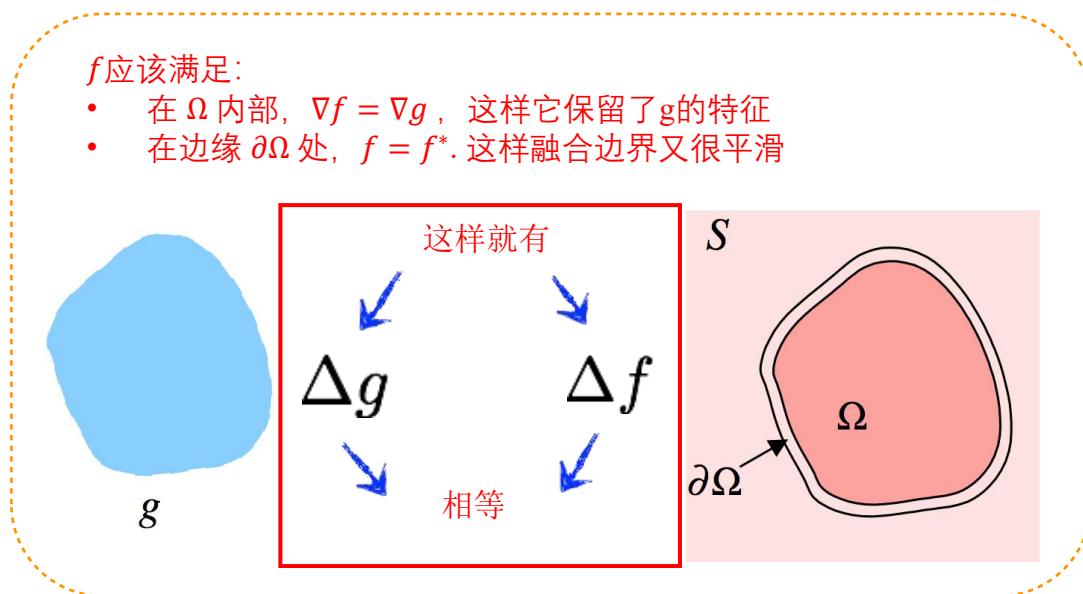
- 在  $\Omega$  内部， $\nabla f = \nabla g$ ，这样它保留了  $g$  的特征
- 在边缘  $\partial\Omega$  处， $f = f^*$ 。这样融合边界又很平滑

在泊松融合中输入梯度场应该是

$$\mathbf{v} = (u, v) = \nabla g$$

在泊松融合中输入梯度场的散度是？

$$\operatorname{div} \mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = \Delta g$$





# 变分问题的极值点是带Dirichlet边界条件的泊松方程的解

那么如何求解泊松方程呢

用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

回忆一下 ...

拉普拉斯  $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$

输入梯度场

$$\mathbf{v} = (u, v)$$

散度  $\operatorname{div} \mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$



# 离散化泊松方程

用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

拉普拉斯滤波器

0	1	0
1	-4	1
0	1	0

x方向偏导滤波器

1	-1
---	----

Y方向偏导滤波器

1
-1

对每个像素有

$$(\Delta f)(x, y) = (\nabla \cdot \mathbf{v})(x, y)$$

对离散图像有：

$$\begin{aligned} & -4f(x, y) + f(x+1, y) + f(x-1, y) \\ & \quad + f(x, y+1) + f(x, y-1) \\ & = u(x+1, y) - u(x, y) + v(x, y+1) - v(x, y) \end{aligned}$$



# 离散化泊松方程

用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

拉普拉斯滤波器

0	1	0
1	-4	1
0	1	0

x方向偏导滤波器

1	-1
---	----

Y方向偏导滤波器

1
-1

写得更紧凑一些，那么对每个像素有

$$(\Delta f)_p = (\nabla \cdot \mathbf{v})_p$$

对离散图像有：

$$-4f_p + \sum_{q \in N_p} f_q = (u_x)_p + (v_y)_p$$



可以把方程进一步重写如下，并用矩阵乘法表示

P个变量的线性方程

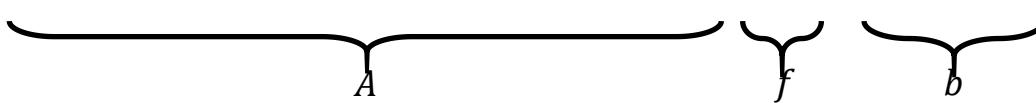
$$-4f_p + \sum_{q \in N_p} f_q = (u_x)_p + (v_y)_p$$

1个像素一个方程  
 $p = 1, \dots, P$

用矩阵乘法表示

每个像素会添加一个稀疏的行

$$\left[ \begin{array}{ccccccccc} 0 & \cdots & 1 & \cdots & 1 & -4 & 1 & \cdots & 1 & \cdots & 0 \end{array} \right] \cdot \begin{bmatrix} f_1 \\ \vdots \\ f_{q_1} \\ \vdots \\ f_{q_2} \\ f_p \\ f_{q_3} \\ \vdots \\ f_{q_4} \\ \vdots \\ f_P \end{bmatrix} = \begin{bmatrix} (\nabla \cdot \mathbf{v})_1 \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_1} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_2} \\ (\nabla \cdot \mathbf{v})_p \\ (\nabla \cdot \mathbf{v})_{q_3} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_4} \\ \vdots \\ (\nabla \cdot \mathbf{v})_P \end{bmatrix}$$



A                    f                    b



可以把方程进一步重写如下，并用矩阵乘法表示

P个变量的线性方程

$$-4f_p + \sum_{q \in N_p} f_q = (u_x)_p + (v_y)_p$$

1个像素一个方程  
 $p = 1, \dots, P$

用矩阵乘法表示

拉普拉斯矩阵

每个像素会添加一个稀疏的行

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

$$\begin{bmatrix} f_1 \\ \vdots \\ f_{q_1} \\ \vdots \\ f_{q_2} \\ f_p \\ f_{q_3} \\ \vdots \\ f_{q_4} \\ \vdots \\ f_P \end{bmatrix} = \begin{bmatrix} (\nabla \cdot \mathbf{v})_1 \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_1} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_2} \\ (\nabla \cdot \mathbf{v})_p \\ (\nabla \cdot \mathbf{v})_{q_3} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_4} \\ \vdots \\ (\nabla \cdot \mathbf{v})_P \end{bmatrix}$$

A

f

b



## 进一步观察拉普拉斯矩阵

对于 $m \times n$  图像，我们可以将该矩阵重新组织为块三对角线形式，如下所示：

$$A_{mn \times mn} = \begin{bmatrix} D & I & 0 & 0 & 0 & \cdots & 0 \\ I & D & I & 0 & 0 & \cdots & 0 \\ 0 & I & D & I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & I & D & I & 0 \\ 0 & \cdots & \cdots & 0 & I & D & I \\ 0 & \cdots & \cdots & \cdots & 0 & I & D \end{bmatrix}$$

这里要求把像素以列优先的形式排列

$I_{m \times m}$ 是 $m \times m$ 的单位矩阵

$$D_{m \times m} = \begin{bmatrix} -4 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -4 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -4 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & -4 & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 & -4 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 & -4 \end{bmatrix}$$



# 离散化泊松方程

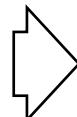
用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

离散化后的矩阵表示

$$\begin{bmatrix} D & I & 0 & 0 & 0 & \cdots & 0 \\ I & D & I & 0 & 0 & \cdots & 0 \\ 0 & I & D & I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & I & D & I & 0 \\ 0 & \cdots & \cdots & 0 & I & D & I \\ 0 & \cdots & \cdots & \cdots & 0 & I & D \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_{q_1} \\ \vdots \\ f_{q_2} \\ f_p \\ f_{q_3} \\ \vdots \\ f_{q_4} \\ \vdots \\ f_P \end{bmatrix} = \begin{bmatrix} (\nabla \cdot \mathbf{v})_1 \\ (\nabla \cdot \mathbf{v})_{q_1} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_2} \\ (\nabla \cdot \mathbf{v})_p \\ (\nabla \cdot \mathbf{v})_{q_3} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_4} \\ \vdots \\ (\nabla \cdot \mathbf{v})_P \end{bmatrix}$$

简写为线性方程



$$Af = b$$

特别小心: 边界像素要特殊对待, 一般设置为目标的边界像素值等于源像素值



# 离散化泊松方程

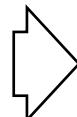
用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

离散化后的矩阵表示

$$\begin{bmatrix} D & I & 0 & 0 & 0 & \cdots & 0 \\ I & D & I & 0 & 0 & \cdots & 0 \\ 0 & I & D & I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & I & D & I & 0 \\ 0 & \cdots & \cdots & 0 & I & D & I \\ 0 & \cdots & \cdots & \cdots & 0 & I & D \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_{q_1} \\ \vdots \\ f_{q_2} \\ f_p \\ f_{q_3} \\ \vdots \\ f_{q_4} \\ \vdots \\ f_P \end{bmatrix} = \begin{bmatrix} (\nabla \cdot \mathbf{v})_1 \\ (\nabla \cdot \mathbf{v})_{q_1} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_2} \\ (\nabla \cdot \mathbf{v})_p \\ (\nabla \cdot \mathbf{v})_{q_3} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_4} \\ \vdots \\ (\nabla \cdot \mathbf{v})_P \end{bmatrix}$$

简写为线性方程



$$Af = b$$

如何求解呢?

特别小心: 边界像素要特殊对待, 一般设置为目标的边界像素值等于源像素值



## 求解线性方程

将系统转换为线性最小二乘问题：

$$E_{LLS} = \|\mathbf{A}f - \mathbf{b}\|^2$$

Matlab:

$$\mathbf{f} = \mathbf{A} \setminus \mathbf{b}$$

展开误差

$$E_{LLS} = \mathbf{f}^\top (\mathbf{A}^\top \mathbf{A}) \mathbf{f} - 2\mathbf{f}^\top (\mathbf{A}^\top \mathbf{b}) + \|\mathbf{b}\|^2$$

最小化误差：

$$(\mathbf{A}^\top \mathbf{A})\mathbf{f} = \mathbf{A}^\top \mathbf{b}$$

设置导数为0

求解

$$\mathbf{f} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b} \quad \longleftarrow$$

一般我们不求解矩阵的逆



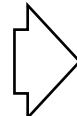
# 离散化泊松方程

用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

离散化的矩阵表示

$$\begin{bmatrix} D & I & 0 & 0 & 0 & \cdots & 0 \\ I & D & I & 0 & 0 & \cdots & 0 \\ 0 & I & D & I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & I & D & I & 0 \\ 0 & \cdots & 0 & I & D & I & 0 \\ 0 & \cdots & \cdots & 0 & I & D & I \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ \vdots \\ f_{q_1} \\ \vdots \\ f_{q_2} \\ \vdots \\ f_p \\ \vdots \\ f_{q_3} \\ \vdots \\ f_{q_4} \\ \vdots \\ f_P \end{bmatrix} = \begin{bmatrix} (\nabla \cdot \mathbf{v})_1 \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_1} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_2} \\ \vdots \\ (\nabla \cdot \mathbf{v})_p \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_3} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_4} \\ \vdots \\ (\nabla \cdot \mathbf{v})_P \end{bmatrix}$$



简写为线性方程

$$Af = b$$

因为A的尺寸为PxP, 非常大

特别小心: 边界像素要特殊对待, 一般设置为目标的边界像素值等于源像素值



# 泊松方程的多种求解方法

- 泊松求解(也就是刚才讲的最小二乘求解方法)

- + 适用范围广
- 但矩阵A可能很大，导致求解困难

- 多种加速求解方法

- + (共轭) 梯度下降求解
- + 多网格算法
- + Pre-conditioning(预处理)
- ...



# 目录

- ❖ 回顾闪光/无闪光图像融合的思想
- ❖ 关于梯度的基本理论知识
- ❖ 图像梯度的积分
- ❖ 一个基本的积分问题：泊松融合
- ❖ 泊松方程的加速求解算法**
- ❖ 作业：用新的思想来解决闪光/无闪光图像融合
- ❖ 总结：梯度域图像处理



# 重新审视我们的变分优化问题

变分问题

$$\min_f \iint_{\Omega} |\nabla f - \mathbf{v}|^2 \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

f的梯度与向量场v相近

在边界处f与f\*相等

输入向量场

$$\mathbf{v} = (u, v)$$

回忆一下：

Nabla算子的定义

$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$



# 重新审视我们的变分优化问题

## 变分问题

$$\min_f \iint_{\Omega} |\nabla f - \mathbf{v}|^2 \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

f的梯度与向量场v相近

在边界处f与f\*相等

输入向量场

$$\mathbf{v} = (u, v)$$

回忆一下：

Nabla算子的定义

$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

对于离散图像

X方向偏导

1	-1
---	----

Y方向偏导

1
-1



# 重新审视我们的变分优化问题

我们可以用梯度近似来  
离散化这个变分问题

离散问题  
 $G, f$ , and  $v$ 分别是?

$$\min_f \|Gf - v\|^2$$

此时先忽略边界条件

回忆一下:

Nabla算子的定义

$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

对于离散图像

X方向偏导

1	-1
---	----

Y方向偏导

1
-1



# 重新审视我们的变分优化问题

我们可以用梯度近似来  
离散化这个变分问题

离散问题

离散梯度叠加形成的矩阵G

$$\min_f \|Gf - v\|^2$$

未知图像的向量化表示

目标梯度场的向量化表示

此时先忽略边界条件

回忆一下：

Nabla算子的定义

$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

对于离散图像

X方向偏导

1	-1
---	----

Y方向偏导

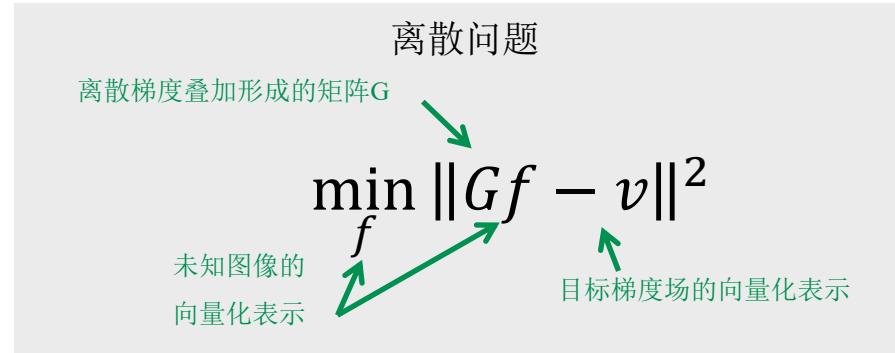
1
-1



# 重新审视我们的变分优化问题

如何求解这个离散版本的变分问题呢？

我们可以用梯度近似来  
离散化这个变分问题



此时先忽略边界条件

回忆一下：

Nabla算子的定义

$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

对于离散图像

X方向偏导

1	-1
---	----

Y方向偏导

1
-1



## 方法1：计算驻点

给定损失函数

$$E(f) = \|Gf - v\|^2$$

我们计算偏导数

$$\frac{\partial E}{\partial f} = ?$$



## 方法1：计算驻点

给定损失函数

$$E(f) = \|Gf - v\|^2$$

我们计算偏导数

$$\frac{\partial E}{\partial f} = G^T G f - G^T v$$



## 方法1：计算驻点

给定损失函数

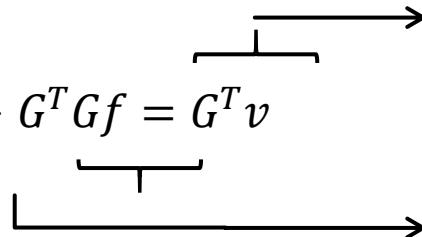
$$E(f) = \|Gf - v\|^2$$

我们计算偏导数

$$\frac{\partial E}{\partial f} = G^T G f - G^T v$$

设导数为0

$$\frac{\partial E}{\partial f} = 0 \Rightarrow G^T G f = G^T v$$



它相当于我们之前推导出的  
向量b!

它相当于我们之前推导的拉  
普拉斯矩阵A

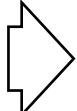


# 离散化泊松方程相当于离散化变分优化问题

用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

离散化后的矩阵表示

$$\begin{bmatrix} D & I & 0 & 0 & 0 & \cdots & 0 \\ I & D & I & 0 & 0 & \cdots & 0 \\ 0 & I & D & I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & I & D & I & 0 \\ 0 & \cdots & \cdots & 0 & I & D & I \\ 0 & \cdots & \cdots & \cdots & 0 & I & D \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_{q_1} \\ \vdots \\ f_{q_2} \\ f_p \\ f_{q_3} \\ \vdots \\ f_{q_4} \\ \vdots \\ f_P \end{bmatrix} = \begin{bmatrix} (\nabla \cdot \mathbf{v})_1 \\ (\nabla \cdot \mathbf{v})_{q_1} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_2} \\ (\nabla \cdot \mathbf{v})_p \\ (\nabla \cdot \mathbf{v})_{q_3} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_4} \\ \vdots \\ (\nabla \cdot \mathbf{v})_P \end{bmatrix}$$


线性方程组  
 $Af = b$

相同的方程组  
 $G^T G f = G^T v$

无论是离散化连续泊松方程还是变分优化问题，我们得到的都是同一个方程组



## 方法1：计算驻点

给定损失函数

$$E(f) = \|Gf - v\|^2$$

我们计算偏导数

$$\frac{\partial E}{\partial f} = G^T G f - G^T v$$

设导数为0

因此这个式子的求解复杂度跟之前泊松求解方式一致

$$\frac{\partial E}{\partial f} = 0 \Rightarrow G^T G f = G^T v$$



## 方法2：使用梯度下降法

给定损失函数

$$E(f) = \|Gf - v\|^2$$

我们计算偏导数，并得到残差

$$\frac{\partial E}{\partial f} = G^T G f - G^T v = A f - b \equiv -r$$



## 方法2：使用梯度下降法

给定损失函数

$$E(f) = \|Gf - v\|^2$$

我们计算偏导数，并得到残差

$$\frac{\partial E}{\partial f} = G^T G f - G^T v = A f - b \equiv -r$$

然后我们迭代地计算它的解

$$f^{i+1} = f^i + \eta^i r^i \quad i = 0, 1, \dots, N, \text{ 其中 } \eta^i \text{ 是正步长}$$



## 选择最佳步长非常关键

使损失函数相对于 $\eta^i$ 的偏导数等于0：

$$E(f) = \|Gf - v\|^2$$

$$E(f^{i+1}) = \|G(f^i + \eta^i r^i) - v\|^2$$



## 选择最佳步长非常关键

使损失函数相对于 $\eta^i$ 的偏导数等于0：

$$E(f) = \|Gf - v\|^2$$

$$E(f^{i+1}) = \|G(f^i + \eta^i r^i) - v\|^2$$

$$\frac{\partial E(f^{i+1})}{\partial \eta^i} = [b - A(f^i + \eta^i r^i)]^T r^i = 0 \Rightarrow \eta^i = \frac{(r^i)^T r^i}{(r^i)^T A r^i}$$



## 梯度下降

给定损失函数

$$E(f) = \|Gf - v\|^2$$

迭代式求解：

$$r^i = b - Af^i, \quad \eta^i = \frac{(r^i)^T r^i}{(r^i)^T Ar^i}, \quad f^{i+1} = f^i + \eta^i r^i, \quad i = 0, \dots, N$$

这种方法是否相比之前的方法更高效？



## 梯度下降

给定损失函数

$$E(f) = \|Gf - v\|^2$$

迭代式求解：

$$r^i = b - Af^i, \quad \eta^i = \frac{(r^i)^T r^i}{(r^i)^T Ar^i}, \quad f^{i+1} = f^i + \eta^i r^i, \quad i = 0, \dots, N$$

这种方法是否相比之前的方法更高效？

我们永远不需要计算A，只需要计算它与向量f, r的乘积，而向量f和r都是图像



# 梯度下降

给定损失函数

$$E(f) = \|Gf - v\|^2$$

迭代式求解：

$$r^i = b - Af^i, \quad \eta^i = \frac{(r^i)^T r^i}{(r^i)^T Ar^i}, \quad f^{i+1} = f^i + \eta^i r^i, \quad i = 0, \dots, N$$

这种方法是否相比之前的方法更高效？

我们永远不需要计算A，只需要计算它与向量f, r的乘积，而向量f和r都是图像

因为A是拉普拉斯矩阵，所以可以使用与拉普拉斯核的卷积来高效地计算上述乘积



## 在实践中我们采用**共轭**梯度下降法

给定损失函数

$$E(f) = \|Gf - v\|^2$$

迭代式的优化求解

$$d^{i+1} = r^i + \beta^i d^i, \quad \eta^i = \frac{(r^i)^T r^i}{(d^i)^T A d^i}, \quad f^{i+1} = f^i + \eta^i d^i, \quad i = 0, \dots, N$$

$$r^{i+1} = r^i - \eta^i A d^i, \quad \beta^i = \frac{(r^{i+1})^T r^{i+1}}{(r^i)^T r^i}$$

- 更聪明的选择迭代更新的方向
- 关键计算都可以用卷积来完成
- 每次迭代只需要一次卷积



## 迭代求解的初始化

初始化 $f^0$  重要吗？



## 迭代求解的初始化

初始化 $f^0$  重要吗？

对于我们最终收敛到哪个 $f$ 来说并不重要，因为损失函数是凸函数

$$E(f) = \|Gf - v\|^2$$



## 迭代求解的初始化

初始化 $f^0$  重要吗？

对于我们最终收敛到哪个 $f$ 来说并不重要，因为损失函数是凸函数

$$E(f) = \|Gf - v\|^2$$

但它对收敛速度来说很重要



# 迭代求解的初始化

初始化 $f^0$  重要吗？

对于我们最终收敛到哪个 $f$ 来说并不重要，因为损失函数是凸函数

$$E(f) = \|Gf - v\|^2$$

但它对收敛速度来说很重要

进一步加速可以使用多分辨率方法：

- 求解分辨率非常低的 $f$ (例如 $2 \times 2$ )的初始问题
- 使用上面的解初始化梯度下降以获得更高的分辨率 $f$ (例如,  $4 \times 4$ )
- 使用上面的解初始化梯度下降以获得更高的分辨率 $f$ (例如,  $8 \times 8$ )
- ...
- 使用上一个解初始化具有原始分辨率 $P \times P$ 的 $f$ 的梯度下降



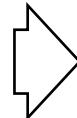
# 离散化泊松方程相当于离散化变分优化问题

用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

离散化后的矩阵表示

$$\begin{bmatrix} D & I & 0 & 0 & 0 & \cdots & 0 \\ I & D & I & 0 & 0 & \cdots & 0 \\ 0 & I & D & I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & I & D & I & 0 \\ 0 & \cdots & \cdots & 0 & I & D & I \\ 0 & \cdots & \cdots & \cdots & 0 & I & D \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_{q_1} \\ \vdots \\ f_{q_2} \\ f_p \\ f_{q_3} \\ \vdots \\ f_{q_4} \\ \vdots \\ f_P \end{bmatrix} = \begin{bmatrix} (\nabla \cdot \mathbf{v})_1 \\ (\nabla \cdot \mathbf{v})_{q_1} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_2} \\ (\nabla \cdot \mathbf{v})_p \\ (\nabla \cdot \mathbf{v})_{q_3} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_4} \\ \vdots \\ (\nabla \cdot \mathbf{v})_P \end{bmatrix}$$



线性方程组

$$Af = b$$

记住我们所做的等同于求解这个线性方程组



## Preconditioning – 预处理用于改善数值方法的收敛性能

我们求解这样的线性方程组

$$Af = b$$

对于任何可逆矩阵P，这等效于求解：

$$P^{-1}Af = P^{-1}b$$

什么时候解这个新的线性方程组更高效呢？



## Preconditioning – 预处理

我们求解这样的线性方程组

$$Af = b$$

对于任何可逆矩阵P，这等效于求解：

$$P^{-1}Af = P^{-1}b$$

什么时候解这个新的线性方程组更优呢？

- 理想情况下：如果A是可逆的，且P与A相同，则线性系统变得平凡！但是这在实际中几乎不可能实现，特别是计算A的逆比求解原始的线性系统更昂贵。
- 在实践中：如果矩阵P-1A有一个更好的条件数，或者它的奇异值分布更均匀，线性方程组就会变得更稳定，更易求解



## Preconditioning – 预处理

我们求解这样的线性方程组

$$Af = b$$

对于任何可逆矩阵P，这等效于求解：

$$P^{-1}Af = P^{-1}b$$

我们选择什么样的预处理方法呢？常见有：

- 雅可比预处理  $P_{\text{Jacobi}} = \text{diag}(A)$
- 不完全LU分解
- 不完全Cholesky分解
- 这是一个热门的研究领域



## Preconditioning – 预处理

我们求解这样的线性方程组

$$Af = b$$

对于任何可逆矩阵P，这等效于求解：

$$P^{-1}Af = P^{-1}b$$

我们选择什么样的预处理方法呢？常见有：

- 雅可比预处理
- 不完全LU分解
- 不完全Cholesky分解
- 这是一个热门的研究领域

$$P_{\text{Jacobi}} = \text{diag}(A)$$

在共轭梯度下降求解过程中可以引入预处理！



## 处理边界条件——再次回顾离散化泊松方程

用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

离散化后的矩阵表示

$$\begin{bmatrix} D & I & 0 & 0 & 0 & \cdots & 0 \\ I & D & I & 0 & 0 & \cdots & 0 \\ 0 & I & D & I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & I & D & I & 0 \\ 0 & \cdots & \cdots & 0 & I & D & I \\ 0 & \cdots & \cdots & \cdots & 0 & I & D \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_{q_1} \\ \vdots \\ f_{q_2} \\ f_p \\ f_{q_3} \\ \vdots \\ f_{q_4} \\ \vdots \\ f_P \end{bmatrix} = \begin{bmatrix} (\nabla \cdot \mathbf{v})_1 \\ (\nabla \cdot \mathbf{v})_{q_1} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_2} \\ (\nabla \cdot \mathbf{v})_p \\ (\nabla \cdot \mathbf{v})_{q_3} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_4} \\ \vdots \\ (\nabla \cdot \mathbf{v})_P \end{bmatrix}$$

简写为线性方程  
 $Af = b$

因为A的尺寸为PxP，非常大

特别小心：边界像素要特殊对待，一般设置为目标的边界像素值等于源像素值

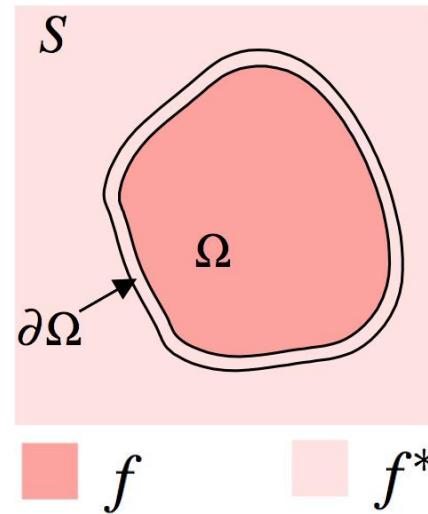


## 处理边界条件

- 构造一个掩膜图像B，其中0代表不需要改变的像素（包括 $S-\Omega$ 区域和 $\partial \Omega$ 区域），而1代表需求解的像素
- 对全图进行拉普拉斯滤波
- 使用(共轭)梯度下降法更新B为1的像素
- 这样新的算法为：

$$f^{i+1} = f^i + B\eta^i r^i \quad (\text{梯度下降})$$

$$f^{i+1} = f^i + B\eta^i d^i \quad (\text{共轭梯度下降})$$





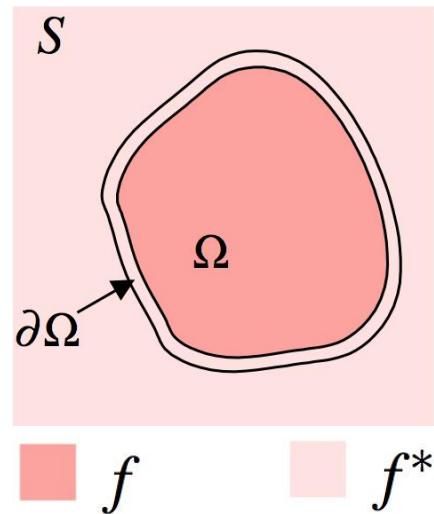
## 处理边界条件

- 构造一个掩膜图像B，其中0代表不需要改变的像素（包括 $S-\Omega$ 区域和 $\partial \Omega$ 区域），而1代表需求解的像素
- 对全图进行拉普拉斯滤波
- 使用(共轭)梯度下降法更新B为1的像素
- 这样新的算法为：

$$f^{i+1} = f^i + B\eta^i r^i \quad (\text{梯度下降})$$

$$f^{i+1} = f^i + B\eta^i d^i \quad (\text{共轭梯度下降})$$

在实践中，在(共轭)梯度下降的其他步骤中也需要掩膜，以处理无效的边界(例如，卷积时)





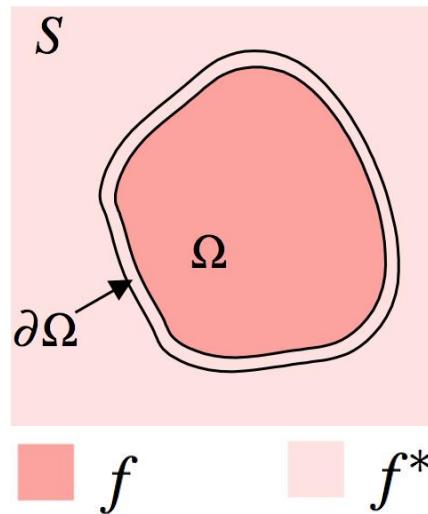
## 处理边界条件

- 构造一个掩膜图像B，其中0代表不需要改变的像素（包括 $S-\Omega$ 区域和 $\partial \Omega$ 区域），而1代表需求解的像素
- 对全图进行拉普拉斯滤波
- 使用(共轭)梯度下降法更新B为1的像素
- 这样新的算法为：

$$f^{i+1} = f^i + B\eta^i r^i \quad (\text{梯度下降})$$

$$f^{i+1} = f^i + B\eta^i d^i \quad (\text{共轭梯度下降})$$

在实践中，在(共轭)梯度下降的其他步骤中也需要掩膜，以处理无效的边界(例如，卷积时)，在作业描述中我会提到





# 目录

- ◆ 回顾闪光/无闪光图像融合的思想
- ◆ 关于梯度的基本理论知识
- ◆ 图像梯度的积分
- ◆ 一个基本的积分问题：泊松融合
- ◆ 泊松方程的加速求解算法
- ◆ 作业：用新的思想来解决闪光/无闪光图像融合
- ◆ 总结：梯度域图像处理

闪光图像



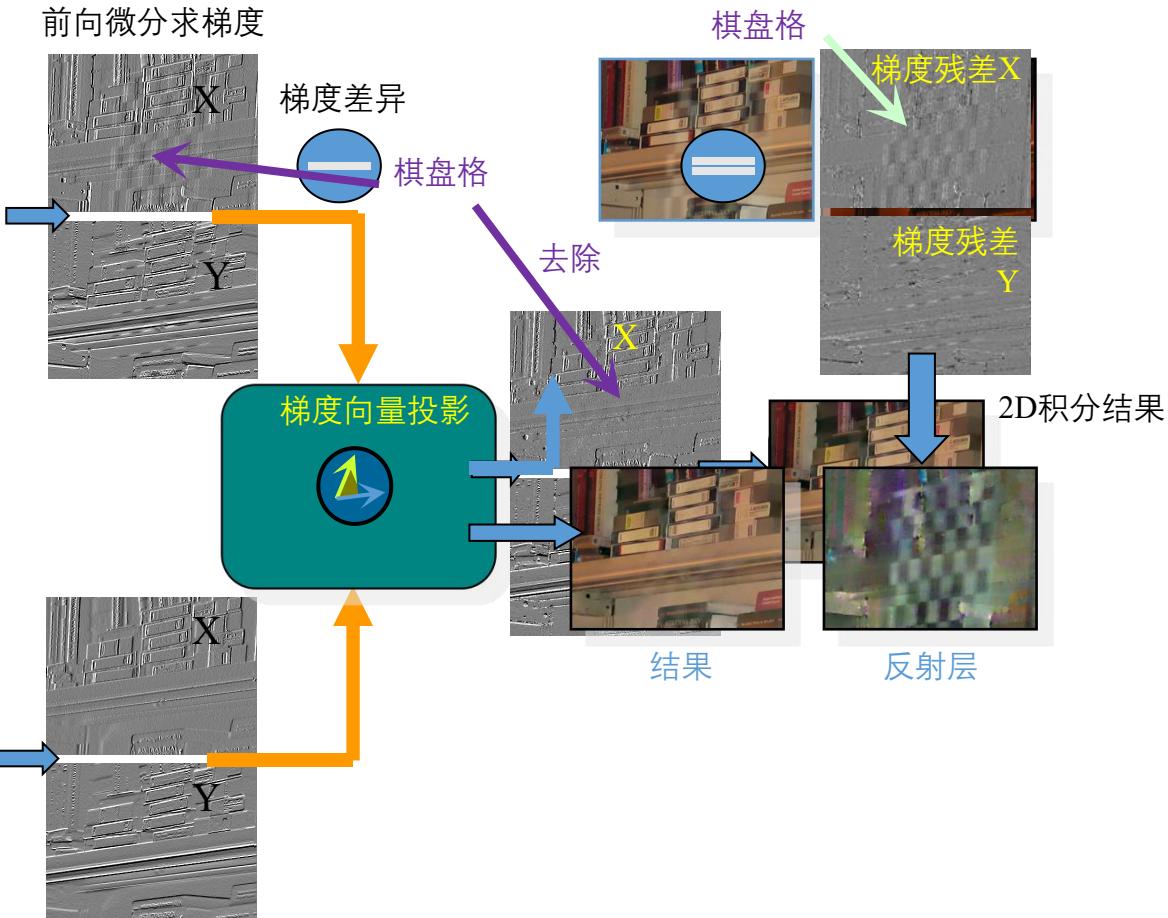
无闪光图像



玻璃外的棋盘格

玻璃上的反射图案

◆ 闪光图像及其梯度



◆ 无闪光图像及其梯度



我将带你用几个编程练习来掌握以下论文的核心思想

## Removing Photography Artifacts using Gradient Projection and Flash-Exposure Sampling

Amit Agrawal

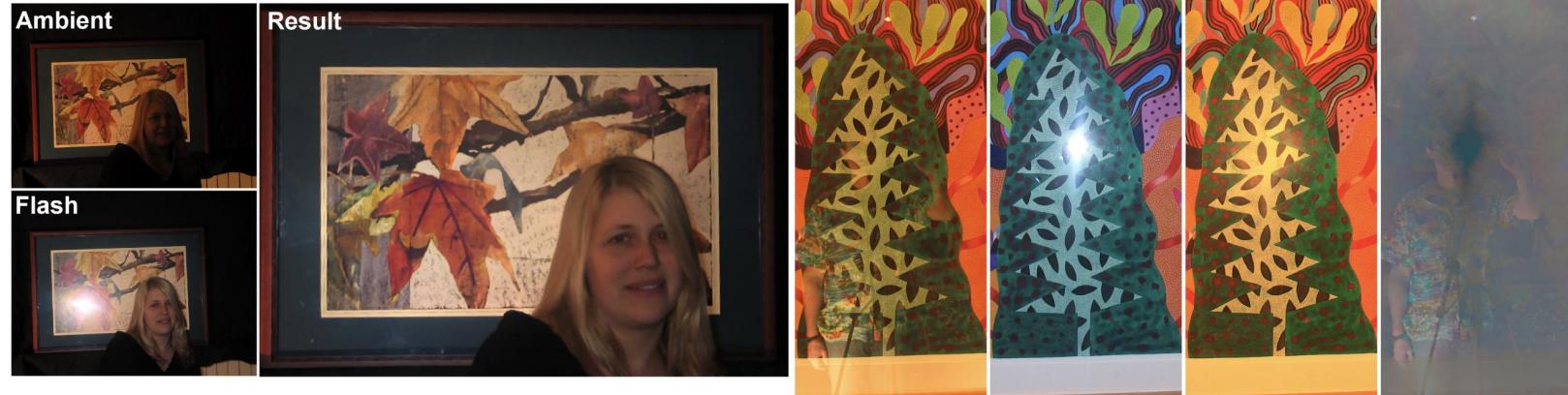
Mitsubishi Electric Research Labs (MERL), Cambridge, MA\*

Ramesh Raskar

Shree K. Nayar<sup>†</sup>

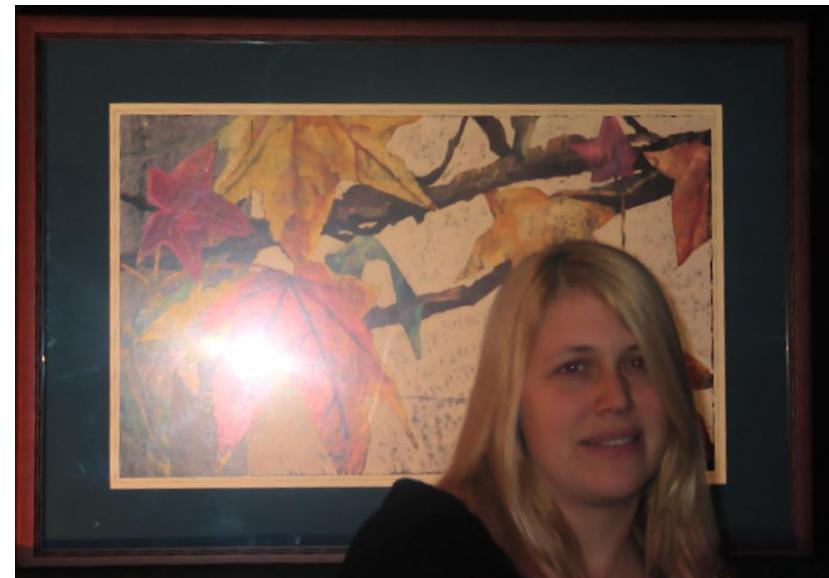
Yuanzhen Li

<sup>†</sup>Columbia University





你需要的素材图像来自于作业参考论文作者





## 作业第1题：对图像求导并用积分重建图像

- 任务：

输入任意一幅图像 $I$ ，它的梯度场为 $\nabla I$ ，你需要对这个梯度场进行积分构成一幅新的图像 $I^*$ ，并且使得它与原始图像一模一样



# 等效于求解离散化泊松方程

用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

离散化后的矩阵表示

$$\begin{bmatrix} D & I & 0 & 0 & 0 & \cdots & 0 \\ I & D & I & 0 & 0 & \cdots & 0 \\ 0 & I & D & I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & I & D & I & 0 \\ 0 & \cdots & \cdots & 0 & I & D & I \\ 0 & \cdots & \cdots & \cdots & 0 & I & D \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_{q_1} \\ \vdots \\ f_{q_2} \\ f_p \\ f_{q_3} \\ \vdots \\ f_{q_4} \\ \vdots \\ f_P \end{bmatrix} = \begin{bmatrix} (\nabla \cdot \mathbf{v})_1 \\ (\nabla \cdot \mathbf{v})_{q_1} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_2} \\ (\nabla \cdot \mathbf{v})_p \\ (\nabla \cdot \mathbf{v})_{q_3} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_4} \\ \vdots \\ (\nabla \cdot \mathbf{v})_P \end{bmatrix}$$

简写为线性方程  
 $Af = b$

因为A的尺寸为PxP，非常大

特别小心：边界像素要特殊对待，一般设置为目标的边界像素值等于源像素值



## 作业第1题：对图像求导并用积分重建图像

- 任务：

输入任意一幅图像 $I$ ，它的梯度场为 $\nabla I$ ，你需要对这个梯度场进行积分构成一幅新的图像 $I^*$ ，并且使得它与原始图像一模一样

- 任务：

因为A非常大，所以你需要使用共轭梯度下降法求解，而不是原始最小二乘方法求解



# 梯度下降

给定损失函数

$$E(f) = \|Gf - v\|^2$$

迭代式求解：

$$r^i = b - Af^i, \quad \eta^i = \frac{(r^i)^T r^i}{(r^i)^T Ar^i}, \quad f^{i+1} = f^i + \eta^i r^i, \quad i = 0, \dots, N$$

这种方法是否相比之前的方法更高效？

我们永远不需要计算A，只需要计算它与向量f, r的乘积，而向量f和r都是图像

因为A是拉普拉斯矩阵，所以可以使用与拉普拉斯核的卷积来高效地计算上述乘积



## 共轭梯度下降法

给定损失函数

$$E(f) = \|Gf - v\|^2$$

迭代式的优化求解

$$d^{i+1} = r^i + \beta^i d^i, \quad \eta^i = \frac{(r^i)^T r^i}{(d^i)^T A d^i}, \quad f^{i+1} = f^i + \eta^i d^i, \quad i = 0, \dots, N$$

$$r^{i+1} = r^i - \eta^i A d^i, \quad \beta^i = \frac{(r^{i+1})^T r^{i+1}}{(r^i)^T r^i}$$

- 更聪明的选择迭代更新的方向
- 关键计算都可以用卷积来完成
- 每次迭代只需要一次卷积



## 作业第1题：对图像求导并用积分重建图像

- 任务：输入任意一幅图像 $I$ ，它的梯度场为 $\nabla I$ ，你需要对这个梯度场进行积分构成一幅新的图像 $I^*$ ，并且使得它与原始图像一模一样
- 要求：因为A非常大，所以你需要使用共轭梯度下降法求解，而不是原始最小二乘方法求解
- 首先，你需要计算输入图像的梯度的散度，即 $\text{div}(\nabla I) = I_{xx} + I_{yy}$ ，这可以通过对图像做拉普拉斯滤波得到，因为 $\text{div}(\nabla I) \equiv \Delta I$
- 将上述散度图作为共轭梯度下降法的输入

$$r^i = b - Af^i = \Delta I - Af^i$$

刚才讲的共轭梯度下降法

$$d^{i+1} = r^i + \beta^i d^i, \quad \eta^i = \frac{(r^i)^T r^i}{(d^i)^T Ad^i}, \quad f^{i+1} = f^i + \eta^i d^i$$

$$r^{i+1} = r^i - \eta^i Ad^i, \quad \beta^i = \frac{(r^{i+1})^T r^{i+1}}{(r^i)^T r^i}$$



## 作业第1题：对图像求导并用积分重建图像

- 任务：输入任意一幅图像 $I$ ，它的梯度场为 $\nabla I$ ，你需要对这个梯度场进行积分构成一幅新的图像 $I^*$ ，并且使得它与原始图像一模一样
- 要求：因为A非常大，所以你需要使用共轭梯度下降法求解，而不是原始最小二乘方法求解
- 首先，你需要计算输入图像的梯度的散度，即 $\text{div}(\nabla I) = I_{xx} + I_{yy}$ ，这可以通过对图像做拉普拉斯滤波得到，因为 $\text{div}(\nabla I) \equiv \Delta I$
- 将上述散度图作为共轭梯度下降法的输入
- 你需要计算与原始图像尺寸一致的边界掩膜B，上下左右四边为0，其他为1，这样重构的图像上下左右四边与原图像一致，中间部分求解得到



## 作业第1题：对图像求导并用积分重建图像

- 任务：输入任意一幅图像 $I$ ，它的梯度场为 $\nabla I$ ，你需要对这个梯度场进行积分构成一幅新的图像 $I^*$ ，并且使得它与原始图像一模一样
- 要求：因为A非常大，所以你需要使用共轭梯度下降法求解，而不是原始最小二乘方法求解
- 首先，你需要计算输入图像的梯度的散度，即 $\text{div}(\nabla I) = I_{xx} + I_{yy}$ ，这可以通过对图像做拉普拉斯滤波得到，因为 $\text{div}(\nabla I) \equiv \Delta I$
- 将上述散度图作为共轭梯度下降法的输入
- 你需要计算与原始图像尺寸一致的边界掩膜B，上下左右四边为0，其他为1，这样重构的图像上下左右四边与原图像一致，中间部分求解得到
- 你需要初始化新图像为全0图像，然后设置它的上下左右四边与原始图像上下左右四边一致



# 作业第1题：你需要完整实现基于共轭梯度法的梯度场积分算法

**ALGORITHM 1:** Gradient field integration with conjugate gradient descent.

**Input:** target divergence  $D$ , initialization  $I_{\text{init}}^*$ , boundary mask  $B$ , boundary values  $I_{\text{boundary}}^*$ , convergence parameter  $\epsilon > 0$ , iteration number  $N$ .

**Output:** integrated image  $I^*$ .

/\* Initialization

```

1  $I^* \leftarrow B \cdot I_{\text{init}}^* + (1 - B) \cdot I_{\text{boundary}}^*$ ;
2  $r \leftarrow B \cdot (D - \text{LaplacianFiltering}(I^*))$ ;
3  $d \leftarrow r$ ;
4  $\delta_{\text{new}} \leftarrow \langle r, r \rangle$ ;
5  $n \leftarrow 0$ ;
/* Conjugate gradient descent iteration
6 while  $\sqrt{\langle r, r \rangle} > \epsilon$  and  $n < N$  do
7    $q \leftarrow \text{LaplacianFiltering}(d)$ ;
8    $\eta \leftarrow \frac{\delta_{\text{new}}}{\langle d, q \rangle}$ ;
9    $I^* \leftarrow I^* + B \cdot (\eta \cdot d)$ ;
10   $r \leftarrow B \cdot (r - \eta \cdot q)$ ;
11   $\delta_{\text{old}} \leftarrow \delta_{\text{new}}$ ;
12   $\delta_{\text{new}} \leftarrow \langle r, r \rangle$ ;
13   $\beta \leftarrow \frac{\delta_{\text{new}}}{\delta_{\text{old}}}$ ;
14   $d \leftarrow r + \beta \cdot d$ ;
15   $n \leftarrow n + 1$ ;
16 end
```

- D: 你刚才计算的散度 \*/
- B: 边界掩膜B, 0代表边界, 1代表其他 \*/
- $I_{\text{init}}^*$ : 初始化为全0图像
- $I_{\text{boundary}}^*$ : 图像的四个边, 设为原始图像的上下左右四边
- N: 迭代次数 \*/
- r: 残差图 \*/

刚才讲的共轭梯度下降法

$$d^{i+1} = r^i + \beta^i d^i, \quad \eta^i = \frac{(r^i)^T r^i}{(d^i)^T A d^i}, \quad f^{i+1} = f^i + \eta^i d^i$$

$$r^{i+1} = r^i - \eta^i A d^i, \quad \beta^i = \frac{(r^{i+1})^T r^{i+1}}{(r^i)^T r^i}$$

如果你的实现正确, 那么重构的图像应该和原始图像一模一样



## 作业第2题：计算梯度方向相干图

带反射信息的闪光图像的梯度向量



回顾一下梯度投影的思想



## 作业第2题：计算梯度方向相干图

带反射信息的闪光图像的梯度向量



梯度方向的一致性可以用梯度方向相干图来衡量



## 作业第2题：计算梯度方向相干图

带反射信息的闪光图像的梯度向量



梯度方向的一致性可以用梯度方向相干图来衡量

$$M = \frac{|\nabla\Phi' \cdot \nabla a|}{\|\nabla\Phi'\| \|\nabla a\|} = \frac{|\Phi'_x \cdot a_x + \Phi'_y \cdot a_y|}{\sqrt{\Phi'^2_x + \Phi'^2_y} \sqrt{a_x^2 + a_y^2}}.$$

论文中的符号表示：

- $\Phi'$ : 闪光图像
- $a$ : 无闪光环境光图像
- $\Phi^*$ : 融合后图像

第2题任务：编写代码求解M，注意这里的操作都是逐像素点的

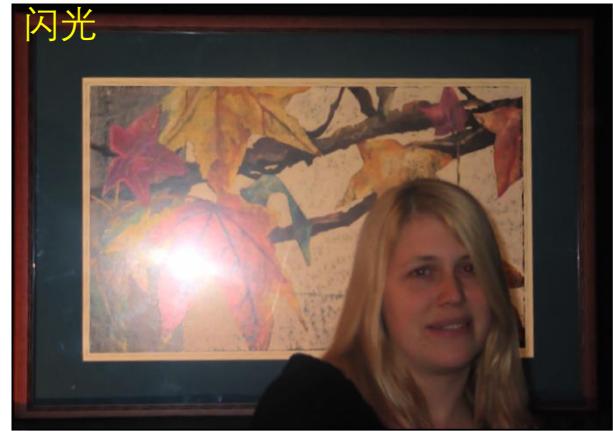


## 作业第3题：计算饱和权重图

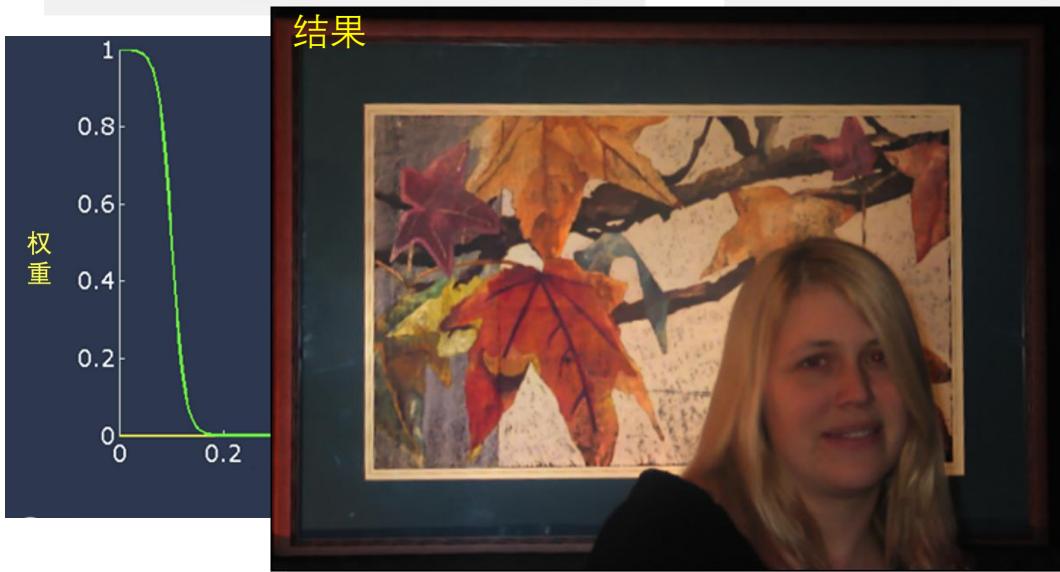
无闪光



闪光



结果



S和梯度方向一致性约束  
像和无闪光图像的梯度



## 作业第3题：计算饱和权重图

$$w_s = \tanh(\sigma \cdot (\Phi' - \tau_s))$$

作者的设置  $\sigma = 40$  and  $\tau_s = 0.9$

### 你需要

- ✓ 编写代码计算上面的饱和权重图
- ✓ 将它归一化到[0, 1]之间



## 作业第4题：利用M和w<sub>s</sub>融合闪光、无闪光图像的梯度

$$\nabla \Phi^* = w_s \cdot \nabla a + (1 - w_s) (M \cdot \nabla \Phi' + (1 - M) \cdot \nabla a)$$

论文中的符号表示：

- $\Phi'$ : 闪光图像
- $a$ : 无闪光环境光图像
- $\Phi^*$ : 融合后图像



$$\begin{bmatrix} \Phi_x^* \\ \Phi_y^* \end{bmatrix} = \begin{bmatrix} w_s \cdot a_x + (1 - w_s) (M \cdot \Phi'_x + (1 - M) \cdot a_x) \\ w_s \cdot a_y + (1 - w_s) (M \cdot \Phi'_y + (1 - M) \cdot a_y) \end{bmatrix}$$



## 作业第5题：调用你刚才开发的共轭梯度下降求解器，求解 $\Phi^*$

任务：

- 求梯度 $\nabla\Phi^*$ 的散度
- 初始化 $\Phi^*$ ，四边与原始图像四边一致，其他为0
- 初始化掩膜B，上下左右四边为0，其他为1
- 将上面的散度作为初始化值，调用刚才开发的共轭梯度下降算法求解 $\Phi^*$



## 和作者的结果做做比较

无闪光



闪光



结果





## 5道编程作业总结

- 编写基于共轭梯度下降法求解的求解器
- 编写代码计算梯度方向相干图
- 编写代码计算饱和权重图
- 编写代码得到融合梯度
- 编写代码，调用第1步的求解器，对融合梯度进行近似积分得到最终图像

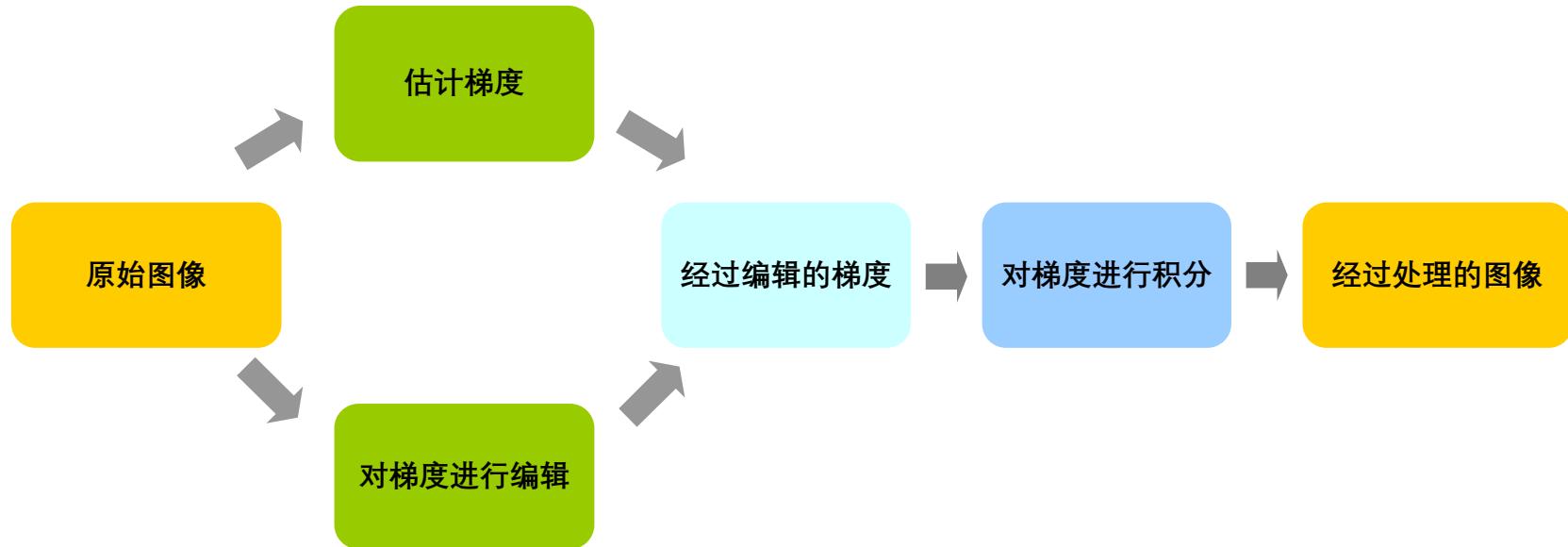


# 目录

- ◆ 回顾闪光/无闪光图像融合的思想
- ◆ 关于梯度的基本理论知识
- ◆ 图像梯度的积分
- ◆ 一个基本的积分问题：泊松融合
- ◆ 泊松方程的加速求解算法
- ◆ 作业：用新的思想来解决闪光/无闪光图像融合
- ◆ 总结：梯度域图像处理



## 梯度域图像处理的思想



例如前面讲的，就是对梯度  
进行投影，去除与基础图像  
梯度方向不一致的梯度信息



## 应用非常广泛 – 闪光/无闪光图像融合





## 应用非常广泛- 泊松融合



直接拷贝粘贴融合



泊松融合

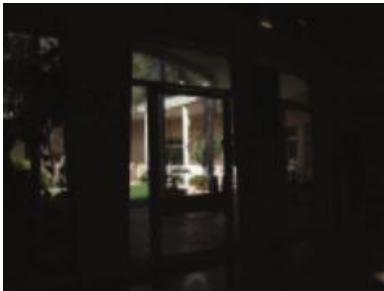
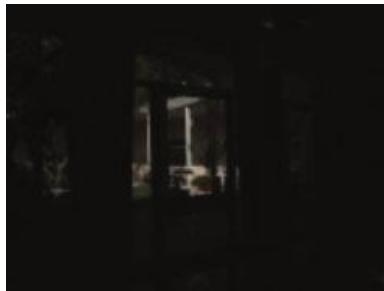


## 应用非常广泛 - 泊松图像编辑





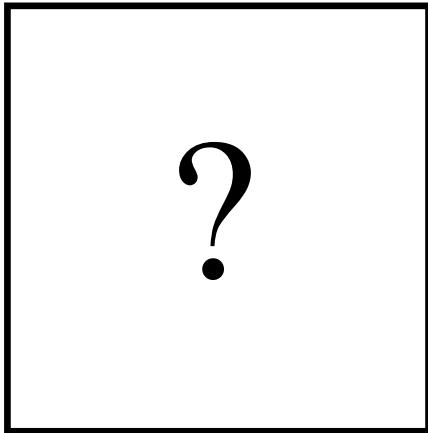
## 应用非常广泛 - 梯度域色调映射





# 不是任何梯度图像都可以积分，可积分的向量场满足旋度为0的条件

- 给定任意的向量场( $u, v$ ), 我们是否总是可以通过积分转换为标量场I?



$$I(x, y): \mathbb{R}^2 \rightarrow \mathbb{R}$$



$$u(x, y): \mathbb{R}^2 \rightarrow \mathbb{R}$$



$$v(x, y): \mathbb{R}^2 \rightarrow \mathbb{R}$$

这里

$$\frac{\partial I}{\partial x}(x, y) = u(x, y)$$

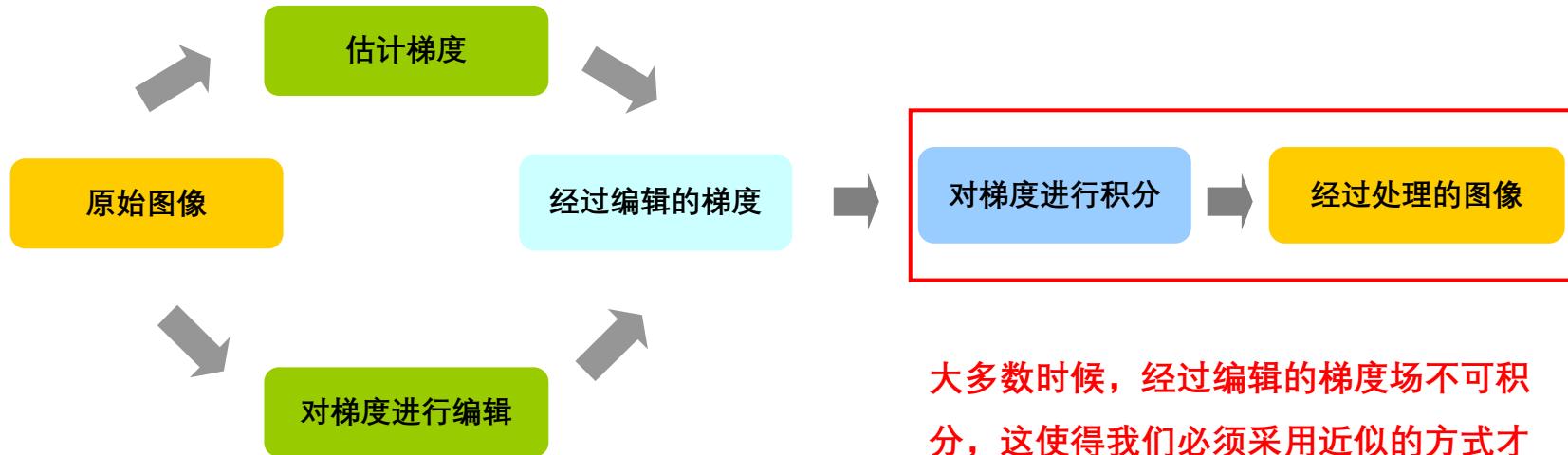
$$\frac{\partial I}{\partial y}(x, y) = v(x, y)$$

仅当:

$$\nabla \times \begin{bmatrix} u(x, y) \\ v(x, y) \end{bmatrix} = 0 \Rightarrow \frac{\partial u}{\partial y}(x, y) = \frac{\partial v}{\partial x}(x, y)$$



# 通过梯度积分恢复图像的应用和挑战



例如前面讲的，就是对梯度进行投影，去除与基础图像梯度方向不一致的梯度信息



## 近似的积分过程，变味了一个离散化泊松方程

用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

离散化后的矩阵表示

$$\begin{bmatrix} D & I & 0 & 0 & 0 & \cdots & 0 \\ I & D & I & 0 & 0 & \cdots & 0 \\ 0 & I & D & I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & I & D & I & 0 \\ 0 & \cdots & \cdots & 0 & I & D & I \\ 0 & \cdots & \cdots & \cdots & 0 & I & D \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_{q_1} \\ \vdots \\ f_{q_2} \\ f_p \\ f_{q_3} \\ \vdots \\ f_{q_4} \\ \vdots \\ f_P \end{bmatrix} = \begin{bmatrix} (\nabla \cdot \mathbf{v})_1 \\ (\nabla \cdot \mathbf{v})_{q_1} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_2} \\ (\nabla \cdot \mathbf{v})_p \\ (\nabla \cdot \mathbf{v})_{q_3} \\ \vdots \\ (\nabla \cdot \mathbf{v})_{q_4} \\ \vdots \\ (\nabla \cdot \mathbf{v})_P \end{bmatrix}$$

↗ 简写为线性方程  
 $Af = b$

$A$ 可能非常大，维度为 $P \times P$

特别小心：边界像素要特殊对待，一般设置为目标的边界像素值等于源像素值



## 把前面的问题转换为最小二乘积分来求解，这是一个变分优化问题

“变分”指的是未知数是整个函数的最优化

变分问题

$$\min_f \iint_{\Omega} |\nabla f - \mathbf{v}|^2 \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

f的梯度与向量场v相近

在边界处f与f\*相等

回忆一下：

Nabla算子的定义

$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

是的，这是我们已知的g的梯度场

$$\mathbf{v} = (u, v)$$



## 变分问题的极值点是带Dirichlet边界条件的泊松方程的解

用欧拉-拉格朗日方程可以得出

$$\Delta f = \operatorname{div} \mathbf{v} \quad \text{over } \Omega, \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

回忆一下 ...

拉普拉斯  $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$

输入梯度场

$$\mathbf{v} = (u, v)$$

散度  $\operatorname{div} \mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$



## 在实践中我们采用**共轭**梯度下降法求解

给定损失函数

$$E(f) = \|Gf - v\|^2$$

迭代式的优化求解

$$d^{i+1} = r^i + \beta^i d^i, \quad \eta^i = \frac{(r^i)^T r^i}{(d^i)^T A d^i}, \quad f^{i+1} = f^i + \eta^i d^i, \quad i = 0, \dots, N$$

$$r^{i+1} = r^i - \eta^i A d^i, \quad \beta^i = \frac{(r^{i+1})^T r^{i+1}}{(r^i)^T r^i}$$

- 更聪明的选择迭代更新的方向
- 关键计算都可以用卷积来完成
- 每次迭代只需要一次卷积



## 5道编程作业

- 编写基于共轭梯度下降法求解的求解器
- 编写代码计算梯度方向相干图
- 编写代码计算饱和权重图
- 编写代码得到融合梯度
- 编写代码，调用第1步的求解器，对融合梯度进行近似积分得到最终图像



# 阅读资料

## 基本参考资料:

- Szeliski textbook, Sections 3.13, 3.5.5, 9.3.4, 10.4.3.
- Pérez et al., “Poisson Image Editing,” SIGGRAPH 2003.

### 泊松图像编辑的论文

- Agrawal and Raskar, “Gradient Domain Manipulation Techniques in Vision and Graphics,” ICCV 2007 course,  
<http://www.amitkagrawal.com/ICCV2007Course/>

### 一个很棒的梯度域图像处理的课程

- Agrawal et al., “Removing Photography Artifacts Using Gradient Projection and Flash-Exposure Sampling,” SIGGRAPH 2005.

A paper on photography with flash and no-flash pairs, using gradient-domain image processing. 梯度域  
闪光、无闪光融合的最佳示例



# 阅读资料

## 额外阅读资料:

- Georgiev, "Covariant Derivatives and Vision," ECCV 2006.
  - An paper from Adobe on the version of Poisson blending implemented in Photoshop's "healing brush".
- Elder and Goldberg, "Image editing in the contour domain", PAMI 2001.
  - One of the very first papers discussing gradient-domain image processing.
- Frankot and Chellappa, "A method for enforcing integrability in shape from shading algorithms," PAMI 1988.
- Bhat et al., "Fourier Analysis of the 2D Screened Poisson Equation for Gradient Domain Problems," ECCV 2008.
  - A couple of papers discussing the (Fourier) basis projection approach for solving the Poisson integration problem.
- Agrawal et al., "What Is the Range of Surface Reconstructions from a Gradient Field?", ECCV 2006.
- Quéau et al., "Normal Integration: A Survey," JMIV 2017.
  - Two papers reviewing various gradient (and surface normal) integration techniques, including Poisson solvers.
- Szelistki, "Locally adapted hierarchical basis preconditioning," SIGGRAPH 2006.
- Krishnan and Szelistki, "Multigrid and multilevel preconditioners for computational photography," SIGGRAPH 2011.
- Krishnan et al., "Efficient Preconditioning of Laplacian Matrices for Computer Graphics," SIGGRAPH 2013.
  - A few well-known references on multi-grid and preconditioning techniques for accelerating the Poisson solver, with a specific focus on computational photography applications..
- Shewchuk, "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain," CMU TR 1994, <http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf>
  - A great reference on (preconditioned) conjugate gradient solvers for large linear systems.
- Briggs et al., "A multigrid tutorial," SIAM 2000.
  - A great reference book on multi-grid approaches.
- Bhat et al., "GradientShop: A Gradient-Domain Optimization Framework for Image and Video Filtering," TOG 2010.
  - A paper describing gradient-domain processing as a general image processing paradigm, which can be used for a broad set of applications beyond blending.
- Krishnan and Fergus, "Dark Flash Photography," SIGGRAPH 2009.
  - A paper proposing doing flash/no-flash photography using infrared flash lights.
- Kazhdan et al., "Poisson surface reconstruction," SGP 2006.
- Kazhdan and Hoppe, "Screened Poisson surface reconstruction," TOG 2013.
  - Two papers discussing Poisson problems for reconstructing meshes from point clouds and normals. This is arguably the most commonly used surface reconstruction algorithm.
- Lehtinen et al., "Gradient-domain metropolis light transport," SIGGRAPH 2013.
- Kettunen et al., "Gradient-domain path tracing," SIGGRAPH 2015.
- Hua et al., "Light transport simulation in the gradient domain," SIGGRAPH Asia 2018 course, [http://beltegeuse.s3-website-ap-northeast-1.amazonaws.com/research/2018\\_GradientCourse/](http://beltegeuse.s3-website-ap-northeast-1.amazonaws.com/research/2018_GradientCourse/)
  - In addition to *editing* images in the gradient-domain, we can *render* them directly in the gradient-domain.
- Tumblin et al., "Why I want a gradient camera?" CVPR 2005.
  - We can even directly *measure* images in the gradient domain, using so-called *gradient cameras*.
- Callenberg et al., "Snapshot difference imaging using correlation time-of-flight sensors," SIGGRAPH Asia 2017.
  - A form of camera with differential pixels.
- Koppal et al., "Toward wide-angle microvision sensors", PAMI 2013.
  - Gradient cameras using optical filtering.
- Chen et al., "ASP vision: Optically computing the first layer of convolutional neural networks using angle sensitive pixels," CVPR 2016.
  - Gradient cameras using angle-sensitive pixels.
- Kim et al., "Real-time 3D reconstruction and 6-DoF tracking with an event camera," ECCV 2016.
  - A paper on using event-based cameras for computer vision applications in very fast frame rates (best paper award at ECCV 2016!).



感谢聆听 !  
Thanks for Listening !

