

曲率滤波

龚元浩
gongyuanhao@gmail.com

source code at <https://github.com/YuanhaoGong>

图像处理问题通常是病态的



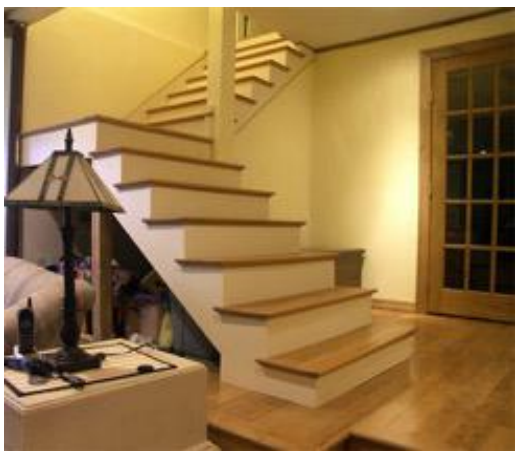
观测到的数据



恢复的图像

曲率正则项

- ▶ 病态问题需要正则项
- ▶ 常见的正则项包括TV（整体变分），MC（平均曲率），GC（高斯曲率）



分段常值



分段线性



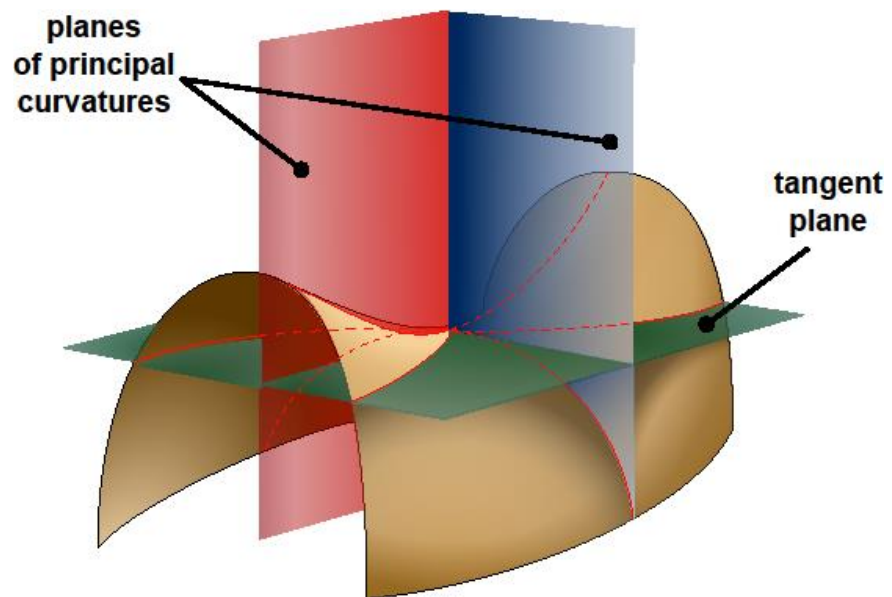
分段光滑

曲率正则项

曲面可以被曲率描述



分段光滑



$$\text{平均曲率} = \frac{k_1 + k_2}{2}$$

$$\text{高斯曲率} = k_1 * k_2$$

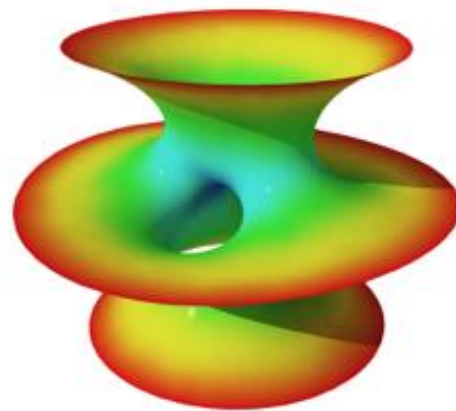
曲率正则项



分段光滑

传统的几何流

平均曲率流



极小曲面

高斯曲率流



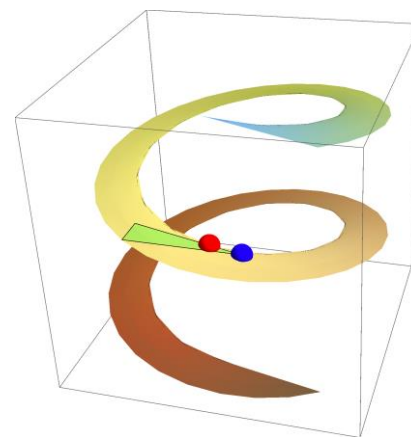
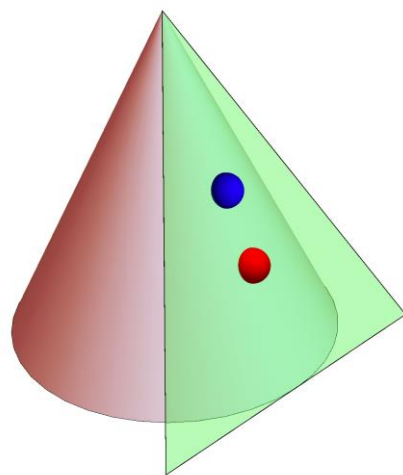
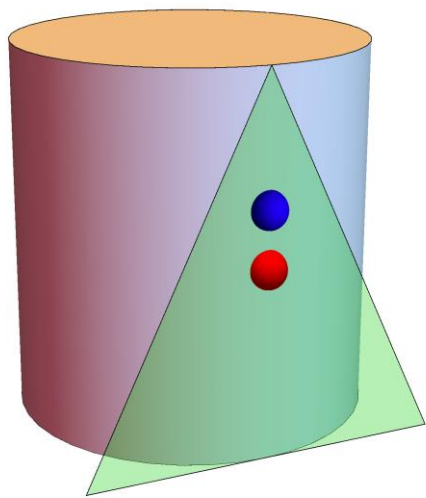
可展曲面

目标

- ▶ 减小高斯或者平均曲率，而且不需要计算它（隐式优化）
- ▶ 比几何流更加高效
- ▶ 计算简单，容易实现和并行

以高斯曲率为例

►根据微分几何，只有三种可展曲面：柱面，锥面，切线面



►对于这三种曲面上的任何一点（红色球），它一定位于它附近某点（蓝色点）的切面（绿色三角形）上。

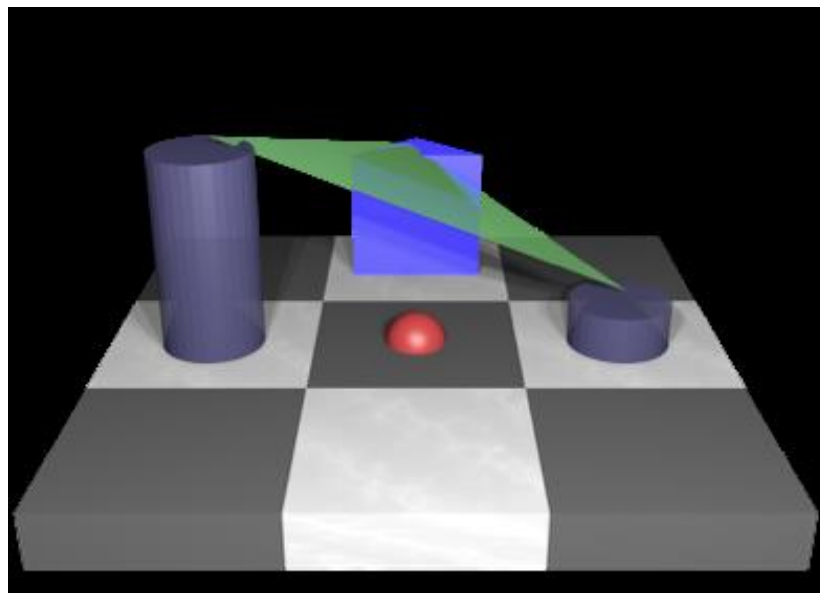
这就是必要性定理

高斯曲率滤波

- ▶ 高斯曲率为 $k1 * k2$
- ▶ 根据必要性定理，如果高斯曲率为零， $k1$ 或者 $k2$ 中必然有一个为零。
- ▶ 我们只需要使其中一个为零即可：不需要计算高斯曲率！

我小学没毕业的时候都知道这个啊！

高斯曲率滤波



切面表示

柱体高度表示图像灰度值，该绿色三角形是蓝色三面柱的切面

高斯曲率滤波

Minimal Projection Operator:

$$d_1 = (U(i-1, j) + U(i+1, j))/2 - U(i, j)$$

$$d_2 = (U(i, j-1) + U(i, j+1))/2 - U(i, j)$$

$$d_3 = (U(i-1, j-1) + U(i+1, j+1))/2 - U(i, j)$$

$$d_4 = (U(i-1, j+1) + U(i+1, j-1))/2 - U(i, j)$$

$$d_5 = U(i-1, j) + U(i, j-1) - U(i-1, j-1) - U(i, j)$$

$$d_6 = U(i-1, j) + U(i, j+1) - U(i-1, j+1) - U(i, j)$$

$$d_7 = U(i, j-1) + U(i+1, j) - U(i+1, j-1) - U(i, j)$$

$$d_8 = U(i, j+1) + U(i+1, j) - U(i+1, j+1) - U(i, j)$$

$$\text{find } d_m, \text{ such that } |d_m| = \min\{|d_i|, i = 1, \dots, 8\}$$

$$\hat{U}(i, j) = U(i, j) + d_m$$

我寻寻觅觅的最小距离啊！

only 25 (+, -, /) ops. per pixel !!!

寻找最小距离

平均曲率滤波，整体变分滤波，这些跟
高斯曲率滤波十分相似。

曲率滤波为啥重要？

- 高效性：比传统方法快**一百到一千**倍
- 通用性：可以解**任意复杂**的噪声模型
- 理论保证：基于经典微分几何理论
- 容易实现和并行：**Matlab 40行，C++ 100行**

快！

通过！

简单！

曲率滤波

图像处理中的瑞士军刀！

source code at <https://github.com/YuanhaoGong>