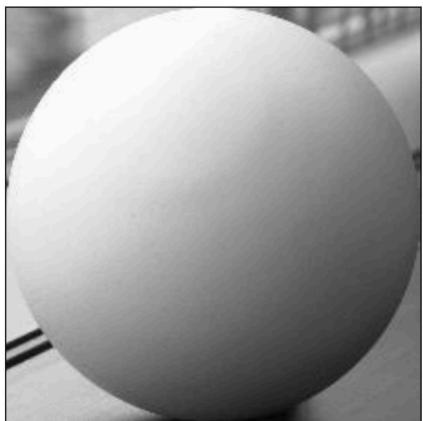


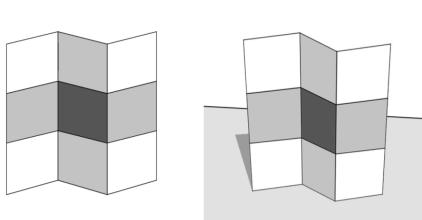
Lecture 8 - Shape from X

在之前的讲座中，学习了双目立体匹配的 3D 重建，以及多视角立体匹配的重建——本章节，主要讲授一下如何使用“Shading”线索

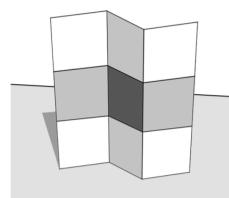
8.1 Shape-from-Shading



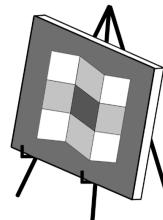
- 从这张图中，我们可以观察到 **intensity** (强度) 和 **shape** (形状) 的关系——从直觉上来说，我们可以从阴影的位置、深度，来判断物体所处的位置、物体的形状、光线的角度等信息
- 然而，如果我们想设计一个算法来提取这种信息，显然并非易事



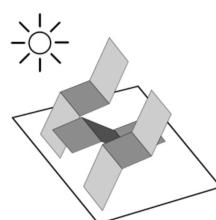
(a) an image



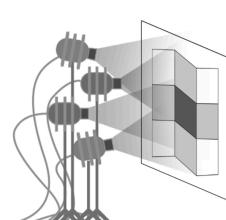
(b) a likely explanation



(c) painter's explanation

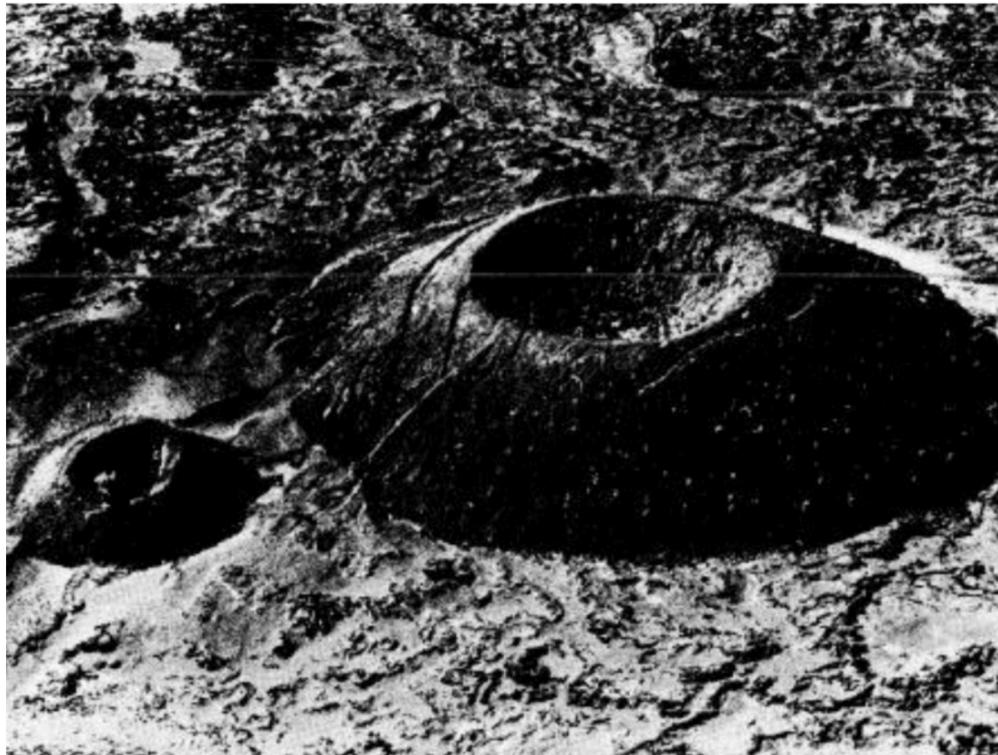


(d) sculptor's explanation

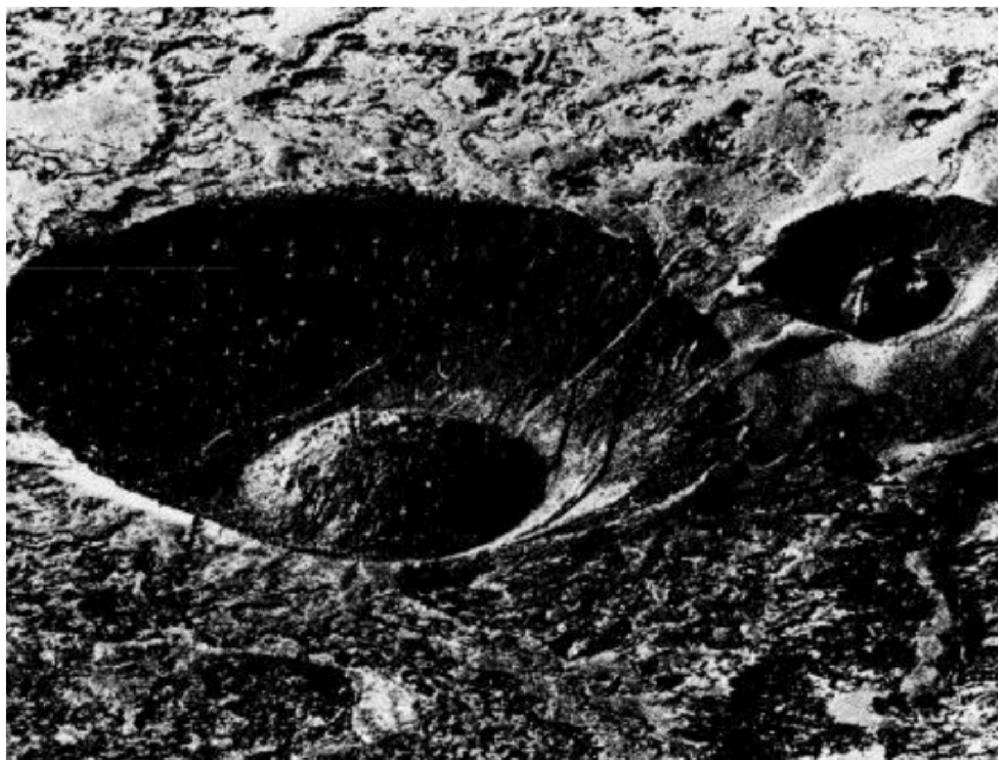


(e) gaffer's explanation

- 这是著名的来自 Adelson 和 Pentland 的工作室的一副图，现在有一幅图片如 a 所示，然后分别让一个普通人、画家、雕塑家、灯光师看一下，然后让他们描述这幅图——我们会发现，不同人的描述和解释方式不同
- 由此我们可以得知，从图像还原它的真实物体是很困难的——如果我们只观察图 a，它很有可能实际上是像图 b 所示的一张纸，但是它确实也可能是 c、d、e 所示的其他物品
 - 我们希望寻找一种简单的解释方式，但是必须融入某些**先验知识**
- 下面我们考虑一下，人类是如何从图像中感知物体的



- ■ 看到这张图片，大多数人可能会认为，图中所示的是一个小的火山，中间有一个凹陷的小火山口；那么如果将这张图颠倒过来，按照我们的直觉，看到的应该会是一个倒过来的小火山，但是请看下图



- ■ 这是这张图片实际倒过来的效果，我们再看上去，感觉它变成了一个大坑，在这个大坑的底部有一个小山丘——这似乎完全变成了另一幅场景，并不符合我们的预期
- 之所以我们会有这样的视觉“错觉”，是因为我们根据人生经验，对世界的场景有一种先入为主的假设——我们始终认为，光源应该是从上往下

的，因此产生了错觉

- 下面我们先回忆一下，第2章讲到的光线渲染方程 [Lecture 2 - Image Formation > ^dc6909](#)

- 令 $\mathbf{p} \in \mathbb{R}^3$ 表示 3D 表面的点， $\mathbf{v} \in \mathbb{R}^3$ 表示观察的方向， $\mathbf{s} \in \mathbb{R}^3$ 表示入射光线的方向；渲染方程描述了到达一个点 \mathbf{p} 的光线 L_{in} ，在方向 \mathbf{v} 上被反射的量 \$\$

$$\begin{aligned} L_{out}(\mathbf{p}, \mathbf{v}, \lambda) = & L_{emit} \\ & (\mathbf{p}, \mathbf{v}, \lambda) + \int \Omega \text{BRDF} \\ & (\mathbf{p}, \mathbf{s}, \mathbf{b}, \lambda) \cdot L_{in} \\ & (\mathbf{p}, \mathbf{s}, \lambda) \cdot (-\mathbf{n}^T \mathbf{s}) d\mathbf{s} \end{aligned}$$

- 符号说明 - $L_{out}(\mathbf{p}, \mathbf{v}, \lambda)$ - 点 \mathbf{t}

- !

[[lec_08_shape_from_x.pdf#page=13&rect=60,3,398,10
8|lec_08_shape_from_x, p.11|500]]

- 由于我们不考虑反射表面发光，所以将 L_{emit} 项去掉；然后将所有的 λ 和 \mathbf{p} 项的依赖都去除掉；由于我们假设**单点光源**，因此我们不需要在球面上对来自四面八方的光源积分，我们可以将积分项也去掉
- 然后我们假设一个纯粹的漫反射表面，因此它不再受到入射方向和出射方向的影响，假设反射率 (albedo) $\text{BRDF}(\mathbf{s}, \mathbf{v}) = \rho$ ，则我们可以用 ρ 代入公式 \$\$

$$L_{out} = \rho \cdot L_{in}(\mathbf{s}) \cdot (-\mathbf{n}^T \mathbf{s}) \cdot \mathbf{s}$$

- 此时我们发现， L_{out} 已经不受到 \mathbf{v} 的依赖，只受到 \mathbf{t}

$$L_{out} = \rho \cdot L_{in}(\mathbf{s}) \cdot \rho \cdot \mathbf{n}^T \mathbf{s} \cdot \mathbf{s}$$

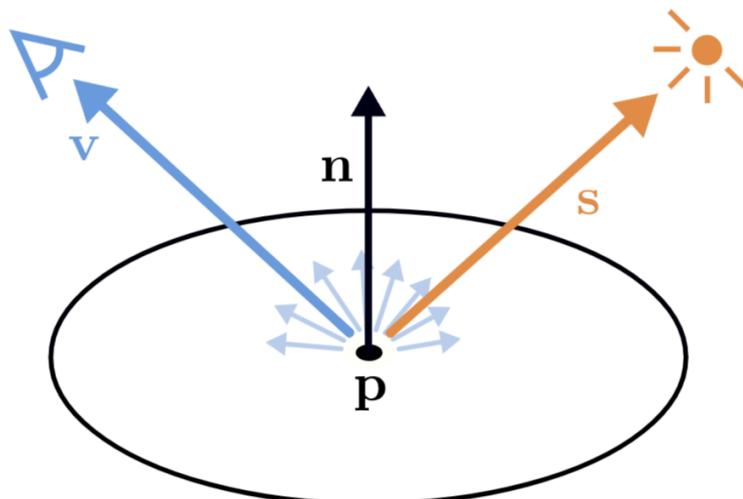
- 至此，我们在 Lambertian 假设、单点光源假设、纯漫反射假设的情况下，得到了这个非常简单的公式；由于单点光源被假设是已知的，因此我们发现实际上 L_{out} 变成了法线向量 \mathbf{n} 的函数 \$\$

$$L_{out} = \rho \cdot L_{in} \cdot \mathbf{dot}(\mathbf{n}^T, \mathbf{s}) = R(\mathbf{n})$$

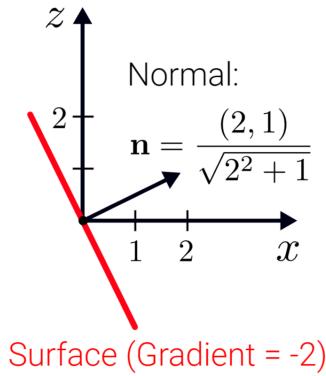
—在材质和光源固定的情况下，反射光强度 L_{out} 是法线向量 \mathbf{n} 的函数，这一

8.1.1 Shape-from-Shading (SfS)

- 首先，我们整理一下研究该技术的所有前提假设



- 1. 空间恒定反射率 ρ 的漫反射材质：针对材料表面的参数只有 1 个
 - 2. 已知单点光源位于无穷远处：在所有像素上，光线方向 s 不变，不受到几何形状或深度的影响
 - 3. 已知相机处于无穷远处 (orthography)：在所有像素上，观察方向 v 不变，不受到几何形状或深度的影响
- 首先，考虑一下我们应该如何参数化 n ？



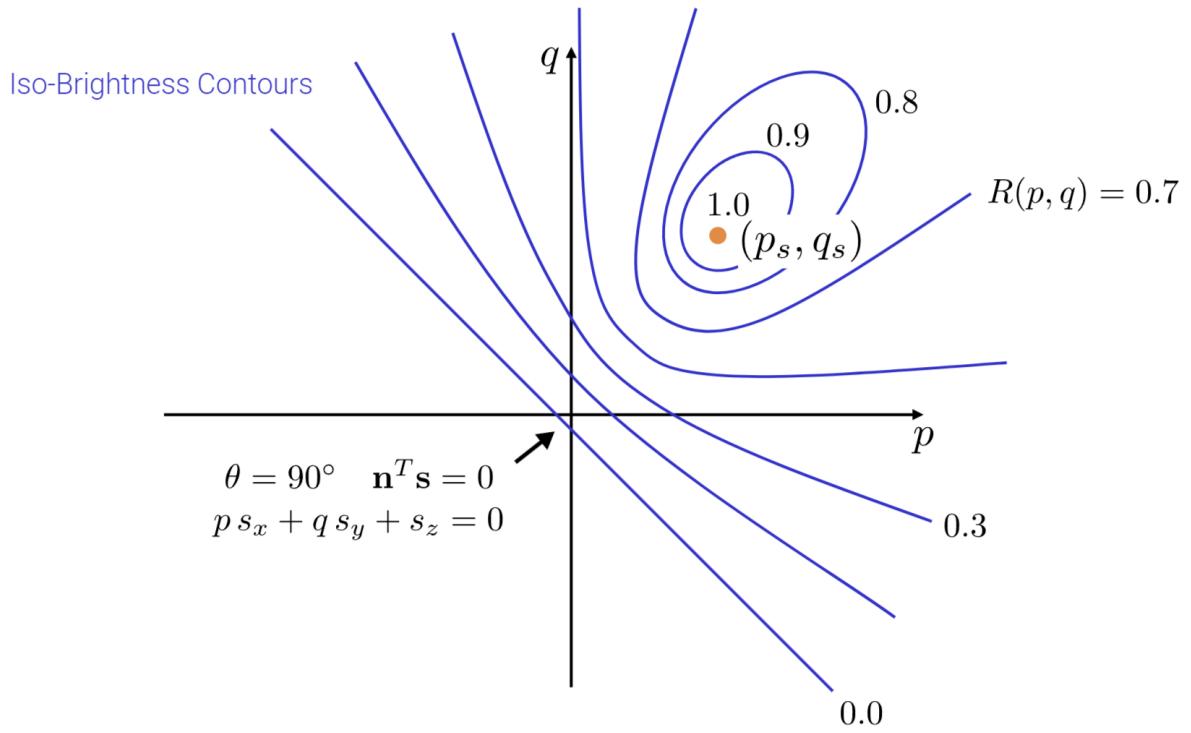
- 法线向量 $\mathbf{n} \in \mathbb{R}^3$, 但是实际上它只有 2 个自由度, 我们用 3 个分量来表示它是很不理想的
- 因此, 我们引入一种新的表示方法, Gradient Space Representation (梯度空间表示), 我们用深度图的负梯度来参数化法线向量 $\mathbf{n} \leftarrow \begin{pmatrix} -\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y} \end{pmatrix}$
- 我们想象有一个相机在上方, 沿着观察相机的主轴方向来测量物体的深度, {

- 上述公式实际上归一化了梯度向量, 确保 \mathbf{n} 的模长为 1
 - 我们假设 $\rho \cdot L_{in} = 1$, 则反射率映射可以写为 $R(p, q)$

$$R(\mathbf{n}) = \mathbf{n}^T \mathbf{s} = \frac{ps_x + qs_y + s_z}{\sqrt{p^2 + q^2 + 1}} = R(p, q)$$

- 其中, $\mathbf{s} = (s_x, s_y, s_z)$ - 现在, 我们将反射率函数改写

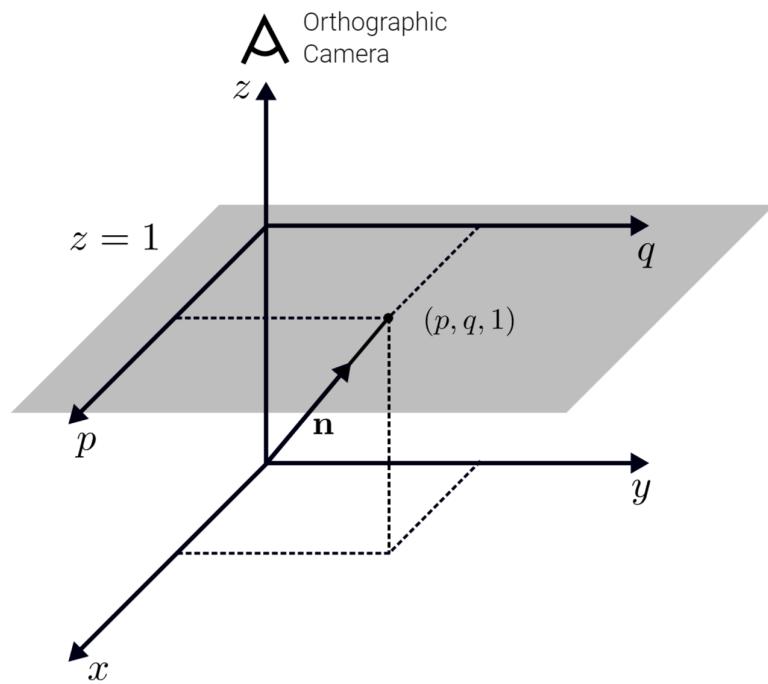
- Reflectance Map



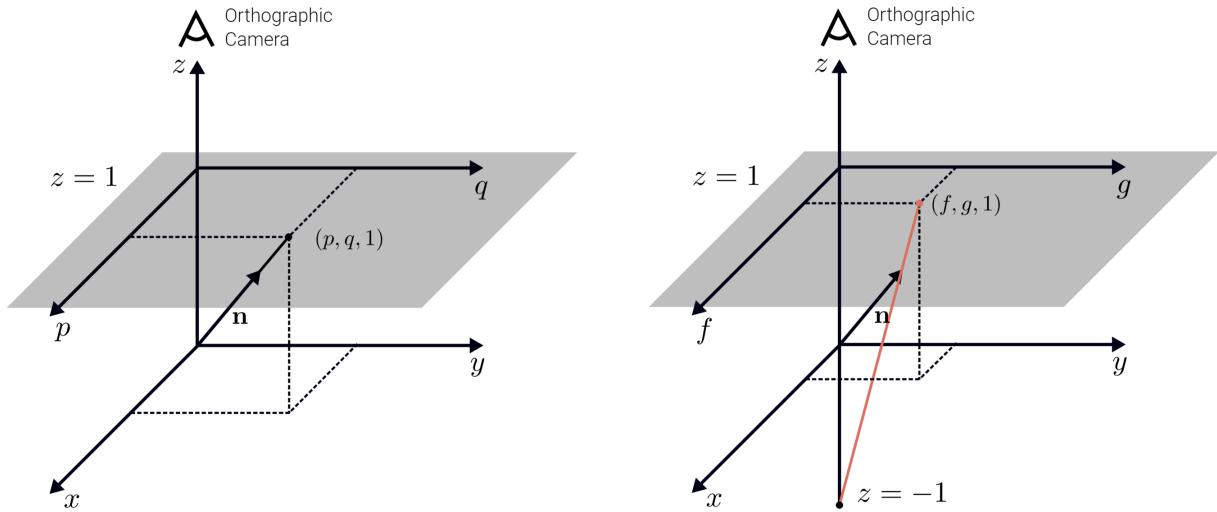
- 上图展示了 $R(p, q)$ 在梯度空间中的等值线（图中蓝色曲线），等值线表示相同反射率 R 的梯度组合 (p, q) ；其中 $R = 1.0$ 表示光线与法线向量完全对齐（正入射），而较低值（例如 $R = 0.7$ ）表示光线偏离法线向量更大的角度

- 问题：除了 $R(p, q) = 1.0$ 时比较特殊以外，对于任何给定的 R ，都可能有多个梯度 (p, q) 与之对应——这意味着，从单张图像中推导出唯一的法线向量是不可能的，需要进一步约束

- 解决方法：正则化 SfS，或者多视角观测来减少模糊性（被称为 Photometric Stereo 问题）



- 现在我们考虑留下的一个特殊问题——当法线向量方向与观察方向正交时，会怎么样？
- 如图中所示，我们无法找到一个交点——它位于无穷远处；不过，我们可以找到一个简单的方法来解决这个问题，它称为 **Stereographic Mapping**
-



- 为了解决这个问题，我们尝试通过改变一种表示方式，来限制表面梯度，使得新的坐标 (f, g) 模长小于 2
 - 左侧图为传统的 p, q 空间，我们采用一种新的映射方式，从 $z = -1$ 平面作为起点，也即从 $(0, 0, -1)$ 点开始，指向 $z = 1$ 平面——我们将其命名为 f, g 空间
 - 根据右图，我们可以得到 f, g 相对于 p, q 的表示法 \$\$
- ```
\begin{aligned} f &= \frac{2p}{1 + \sqrt{p^2 + q^2 + 1}} \\ g &= \frac{2q}{1 + \sqrt{p^2 + q^2 + 1}} \end{aligned}
```

—现在，反射率函数 $R$ 依赖于 $f, g$   $R(f(x, y), g(x, y))$

- !

[[lec\_08\_shape\_from\_x.pdf#page=27&rect=49,37,400,2  
28|lec\_08\_shape\_from\_x, p.24|600]]

- 根据上图，我们可以直观的看到 **Stereographic Mapping** 的效果—由于观察方向和法线向量方向正交，因此它们位于图中球体的赤道线位置；如果我们在  $z=-1$  平面处取一个点，穿过赤道，可以在  $z=1$  平面画出一个圆；我们可以用其来替代原来的表示

- Shape-from-Shading Formulation

- 首先假设图像的辐照度（这里我们认为辐照度=强度  $\text{irradiance} = \text{intensity}$ ）等于反射率  $I(x,y) = R(f(x,y), g(x,y))$

- 基于上面的假设，我们可以对于  $SfS$  问题构建最小化优化式

$$E_{\text{image}}(f,g) = \iint |I(x,y) - R(f,g)|^2 dx dy$$

- 我们的目标是最小化图像强度和反射率之间的差距 - 然而，不难看出，这个

$$\begin{aligned} E_{\text{smooth}}(f, g) &= \iint \left( f_x^2 + f_y^2 + g_x^2 + g_y^2 \right) dx dy \\ \text{with gradients } f_x &= \frac{\partial f}{\partial x}, f_y = \frac{\partial f}{\partial y}, \dots \\ \end{aligned}$$

- 通过上式，我们惩罚梯度  $f, g$  的变化率，从而引入平滑度约束 -  $\text{Occl}$

$$E(f,g) = E_{\text{image}}(f,g) + \lambda E_{\text{smooth}}(f,g)$$

- 其中梯度场  $f(x, y)$  和  $g(x, y)$  受到 *Occluding Boundaries* 约束 - *Remarks* -

变分问题：数学和物理中一类重要的优化问题，其核心目标是通过寻找函数来使某种特定的量（通常是积分形式的量）达到极值（最小值或最大值）

## 8.1.2 Surface Integration

- 当我们拿到物体表面的梯度时（可以用 $p, q$ 空间表示，也可以用 $f, g$ 空间表示）  
\$\$

$$(p,q) = \left( -\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y} \right)$$

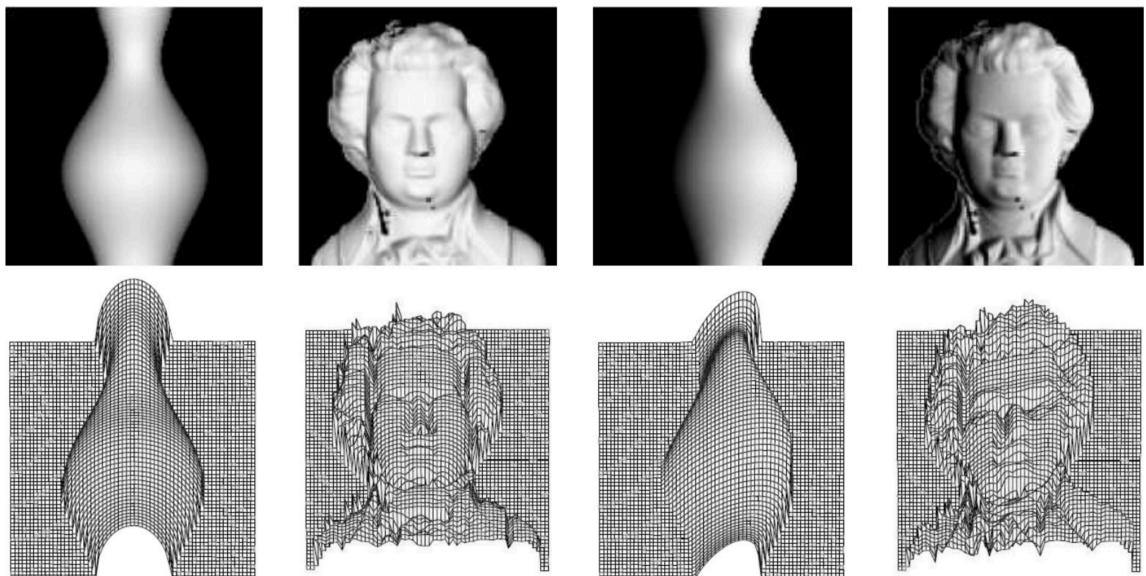
—我们该如何由此还原出3D表面？（或者深度图）

- 假设物体表面是平滑的，那么我们可以将其转化为下面这样一个变分问题  
\$\$

$$E(z) = \iint \left( \left( \frac{\partial z}{\partial x} + p \right)^2 + \left( \frac{\partial z}{\partial y} + q \right)^2 \right) dx dy$$

—我们可以用快速Fourier变换(FFT)来高效计算这一问题[Frankot and Chellappa, 1989]

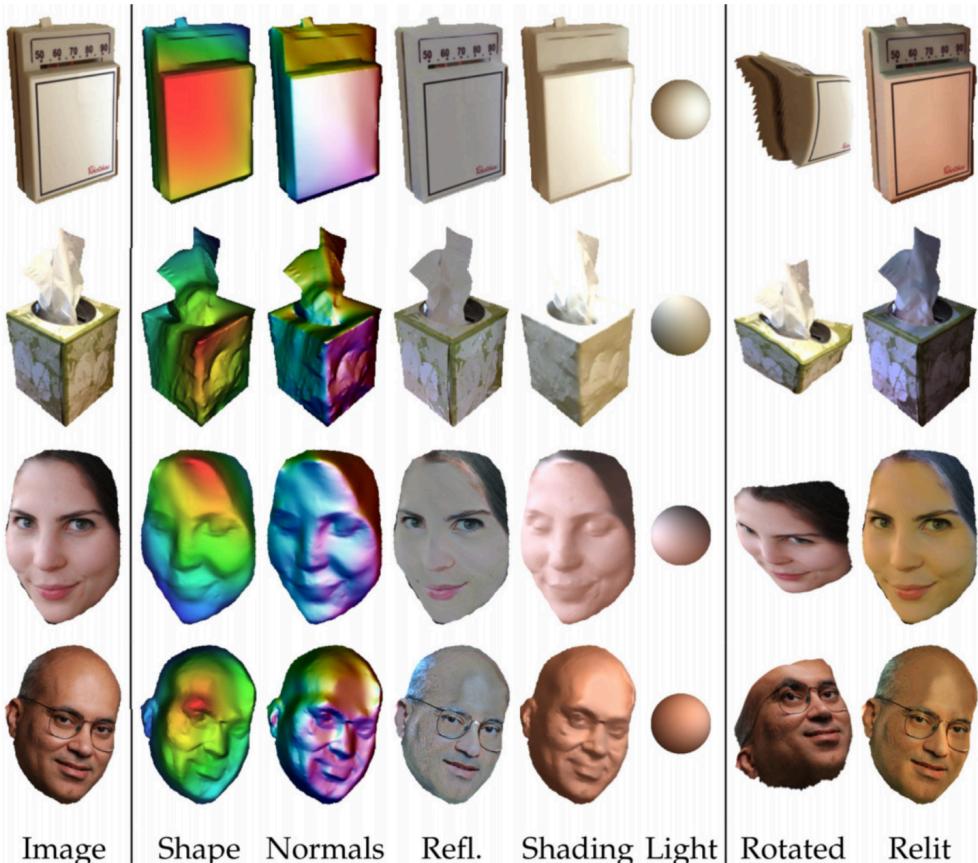
- 下面是一些应用结果



- 第一行图像为输入图像，第二行图像为从中还原出的梯度场或深度场结果——我们可以看出，从还原结果，可以得到物体的形状信息；但是当我们改变光源角度时，也会给这种方法带来误差——因此，这仍然是一项很艰巨的任务

- SIRFS: Shape, Illumination, and Reflectance from Shading

-

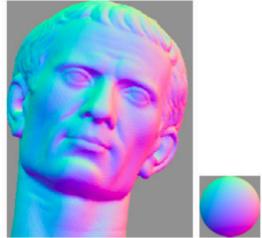
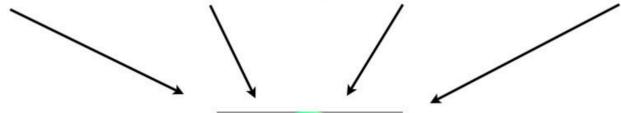
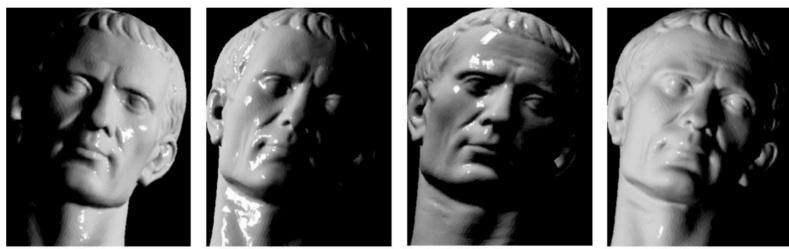


- SIRFS 是 SfS 领域的一项更现代化的工作
- 输入单张遮罩后的 RGB 图像，从中估计深度、法向量、反射率、阴影、亮度信息
- 取消假设已知点光源、取消假设恒定颜色，是一个比 SfS 更加困难的任务
- 尽管我们从图中仍然可以看出很多问题，但是考虑到基于这些放宽后的假设，这项工作仍然令人印象深刻

[Barron and Malik: Shape, Illumination, and Reflectance from Shading. TPAMI, 2015.](#)

## 8.2 Photometric Stereo

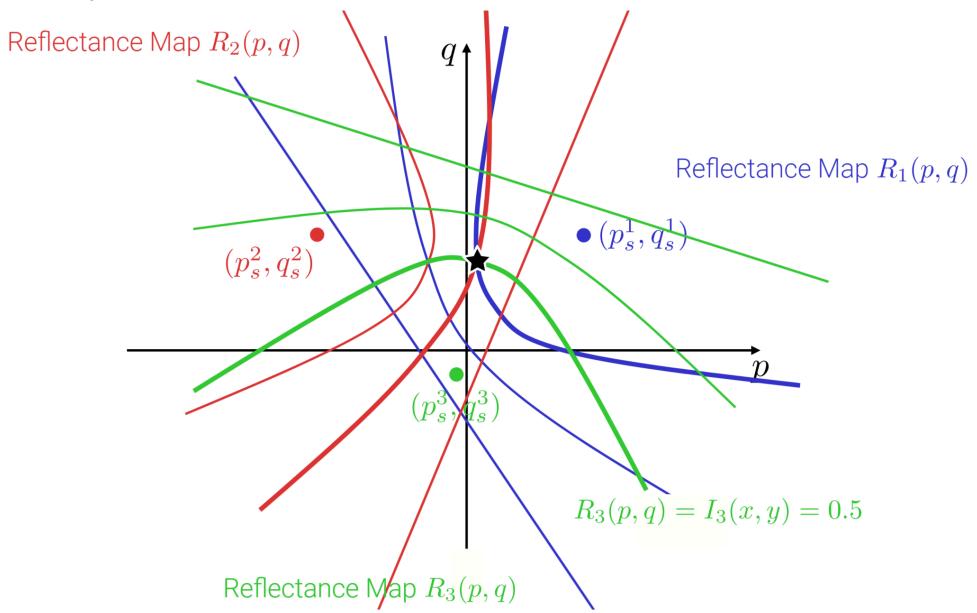
- 上一小节中，我们说明了引入正则化约束来减少模糊性，在这一节中我们将介绍另一种技术——光度立体 (Photometric Stereo)，来给每个像素提供更多的观察量



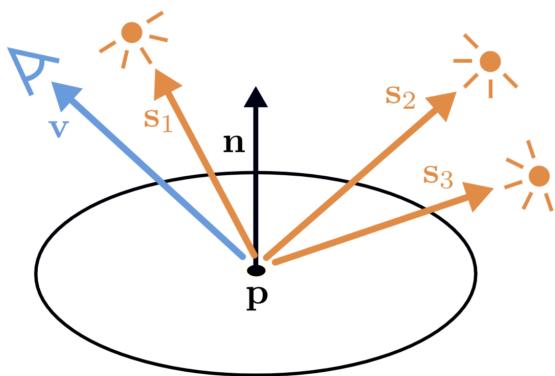
- 从同一个观察点拍摄  $K$  张图片，但是分别采用不同的已知点光源位置
- 基于此，从每个像素来评估法向量、反射率和材质
- 和上一小节一样，假设相机、光源距离物体无穷远处



- 这是一个典型的灯光配置，我们从中可以实现刚才的说法——从一个观察点拍摄  $K$  张图片，但是分别采用不同的光源配置



- 我们分别在三种不同的光源配置下，拍摄三张图像，其  $p, q$  空间如图所示，分别为蓝色、红色、绿色三种曲线
- 当我们只有蓝色曲线时，可以将法向量确定在一条曲线上；当我们同时有蓝色、红色曲线时，可以看到存在 2 个交点，也即 2 个解；当我们同时有蓝色、红色、绿色曲线时，可以看到我们确定了唯一的 1 个交点，也即确定了 1 个解
- 下面我们考虑一下数学化的光度立体



- 我们假设 Lambertian 表面，为简化问题而假设  $L_{in} = 1$ ，则图像的强度（反射光）  

$$I_{out} = \rho \cdot \mathbf{n}^T \mathbf{s}_1 = \rho \cdot \mathbf{s}_1^T \mathbf{n}$$

$$I_{out} = \rho \cdot \mathbf{n}^T \mathbf{s}_2 = \rho \cdot \mathbf{s}_2^T \mathbf{n}$$

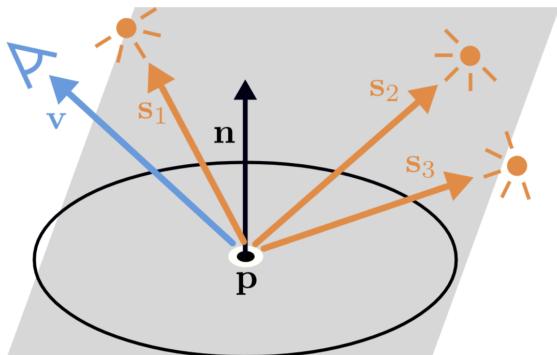
$$I_{out} = \rho \cdot \mathbf{n}^T \mathbf{s}_3 = \rho \cdot \mathbf{s}_3^T \mathbf{n}$$

- 给出 3 个观察量（相同  $\mathbf{v}$ ，不同  $\mathbf{s}$ ），我们可以将其压缩为矩阵形式  

$$\begin{pmatrix} I_1 \\ I_2 \\ I_3 \end{pmatrix} = \underbrace{\rho}_{\mathbf{I}} \cdot \mathbf{n}^T \begin{pmatrix} \mathbf{s}_1^T \\ \mathbf{s}_2^T \\ \mathbf{s}_3^T \end{pmatrix}$$

- 按照上述公式压缩矩阵后，我们可以得到  
 $\text{I} = \text{s} \tilde{\text{n}}$ ，由此可以得到解  
 $\tilde{\text{n}} = \text{s}^{-1} \text{I}$ ，则  
 $\rho = \|\tilde{\text{n}}\|_2$ ，  
 $\text{n} = \frac{\tilde{\text{n}}}{\rho}$

- 到目前为止，似乎感觉光度立体都非常好用——那么，它的局限性在什么时候体现出来呢？
  - 我们要求  $\text{s}$  的 3 个分量必须提供 3 个观察量——也即， $\text{s}$  向量的秩为 3（或者说， $\text{s}$  向量的 3 个分量必须是线性无关的）



- 当  $\text{s}$  的 3 个分量中存在线性依赖，则解将不再是唯一的
- 如果拍摄更多的图像，我们还可以获得更好的结果（均衡噪声）  
 $\begin{aligned} & \underbrace{\begin{pmatrix} 1 \\ \vdots \\ K \end{pmatrix}}_{\text{I}} \\ & \&= \\ & \underbrace{\begin{pmatrix} \mathbf{s}_1^\top \\ \vdots \\ \mathbf{s}_K^\top \end{pmatrix}}_{\text{S}} \end{aligned}$

```

\begin{aligned}
& \underbrace{\begin{pmatrix} 1 \\ \vdots \\ K \end{pmatrix}}_{\text{I}} \\
& \&= \\
& \underbrace{\begin{pmatrix} \mathbf{s}_1^\top \\ \vdots \\ \mathbf{s}_K^\top \end{pmatrix}}_{\text{S}}
\end{aligned}

```

$\underbrace{\rho \mathbf{n}}_{\tilde{\mathbf{n}}} \end{aligned}$

$$\mathbf{S}^\top \mathbf{I} = \mathbf{S}$$

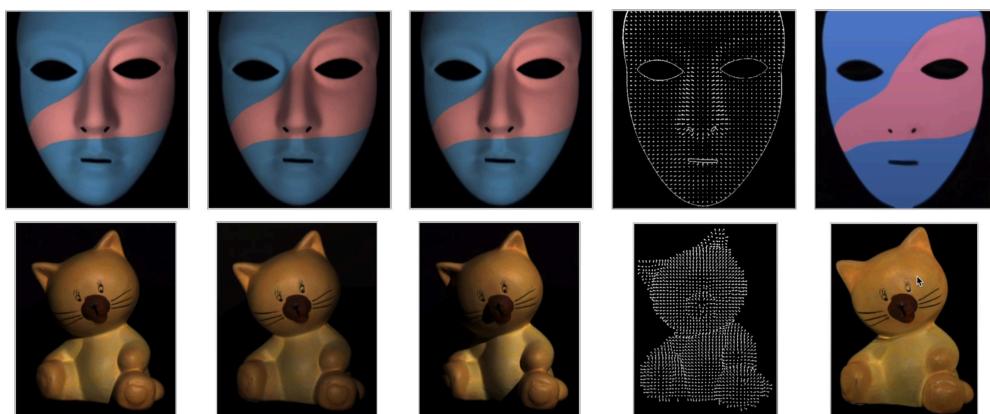
如上，例如我们有  $K$  个图像，则我们可以使用最小二乘解得到  $\tilde{\mathbf{n}}$

- 然后反射率  $\rho$  和法向量  $\mathbf{n}$  可以用与之前同样的方法得到

- 可视化结果

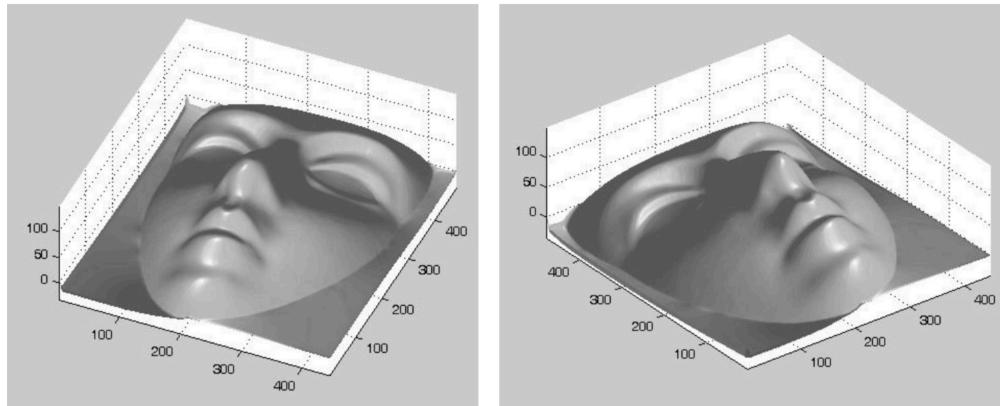


- 我们从一个符合 Lambertian 假设的佛像中观察一下结果
  - 第 2 幅图、第 3 幅图分别为法向量图和反射率图
    - 我们首先从输入图像分别计算每个像素的表面法向量和反射率
    - 然后利用上一小节提到的变分公式，整合法向量得到深度图
    - 然后我们可以通过选择统一的反射率，来重新照亮整个场景



- 对于前面三种采样输入，我们应用 Photometric Stereo，如图中面具样例所示，我们可以很好的估计出原图的反射率，可以清晰地看到鼻子的形状

- 但是对于不满足 Lambertian 假设的情况，我们可以看到 PS 算法并不是那么有效——如第二个玩具样例所示，在还原结果中存在一些伪影



- 这是上一张图中面具的法向量 3D 重建效果，可以看到，符合 Lambertian 假设的情况下，我们应用 PS 算法已经得到了非常详细的 3D 重建结果
- 我们也可以将 PS 算法应用在户外场景下：

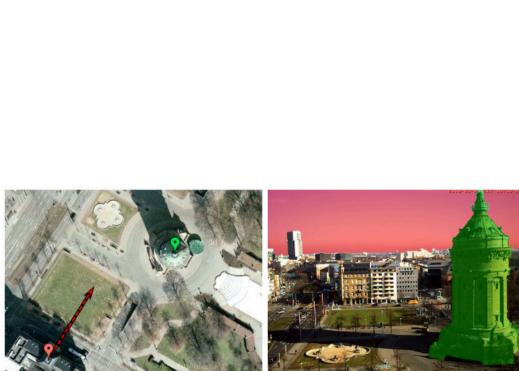


Figure 1. The geographic context of the scene with camera (red marker) and target object (green marker) on the left, and an example image from the camera with sky mask (red region) and object mask (green region) on the right. Sat image © by Google Inc.



Figure 6. One input image, detected shadow regions, selected points for intensity estimation and the recovered object albedo.

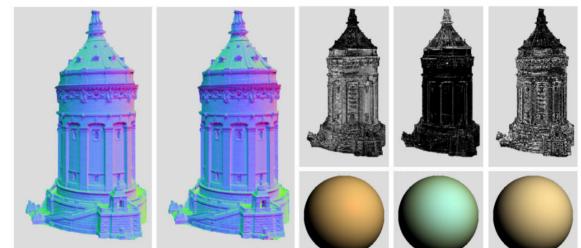


Figure 7. The initial normal map, the final normal map, and the four recovered BRDFs with corresponding material map.

- 由于太阳近似是一个无限远处的完美点光源，因此我们可以在同一位置拍摄一天中不同时间的户外场景，并据此应用 PS 算法

Ackermann, Langguth, Fuhrmann and Goesele: Photometric stereo for outdoor webcams. CVPR, 2012.

- 也有工作尝试输入大量遮罩后的图像，以监督方式进行立体光度

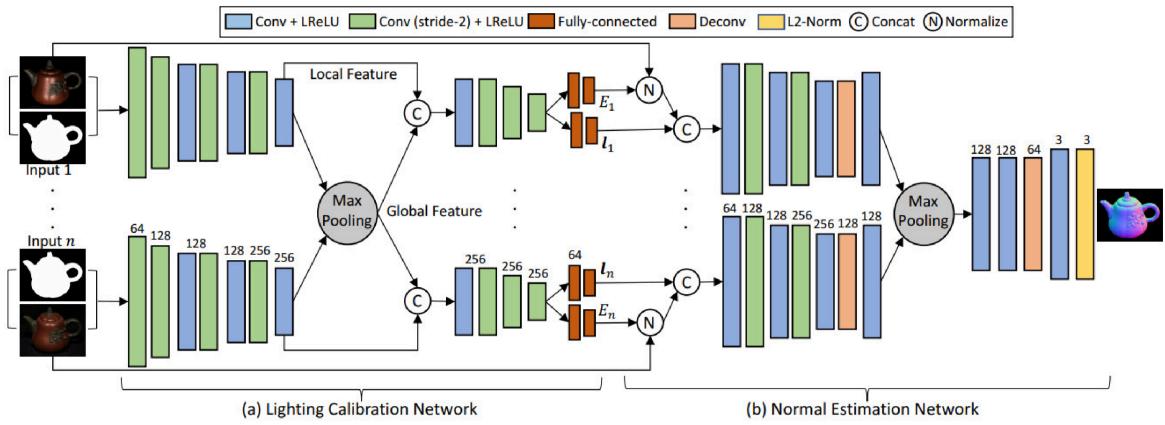
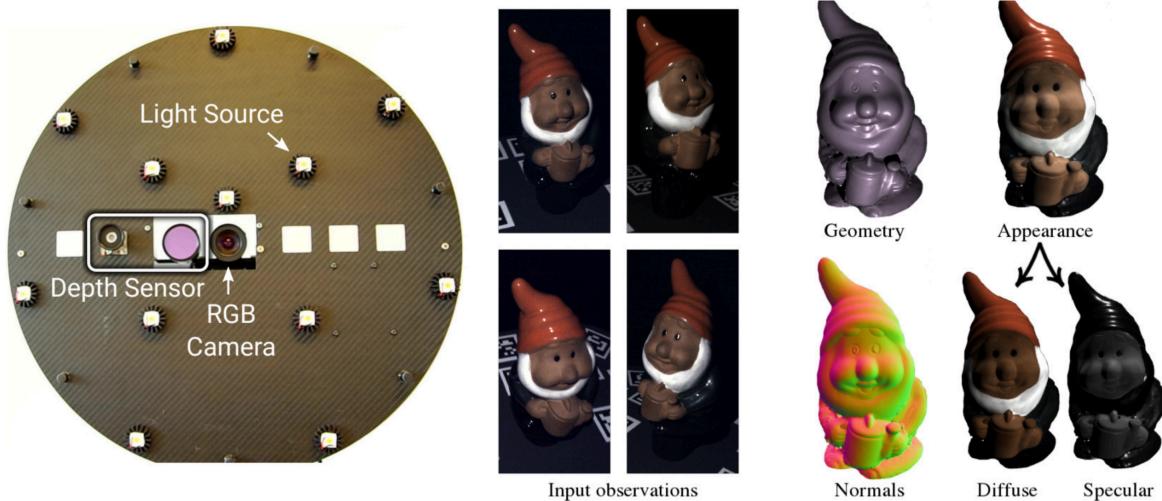


Figure 1. The network architecture of SDPS-Net is composed of (a) Lighting Calibration Network and (b) Normal Estimation Network. Kernel sizes for all convolutional layers are  $3 \times 3$ , and values above the layers indicate the number of feature channels.

[Chen, Han, Shi, Matsushita and Wong: Self-Calibrating Deep Photometric Stereo Networks. CVPR, 2019.](#)

- 在此基础上，我们也可以尝试引入更多的传感器，探寻更加复杂的BRDF情况



[Schmitt, Donne, Riegler, Koltun and Geiger: On Joint Estimation of Pose, Geometry and svBRDF from a Handheld Scanner. CVPR, 2020.](#)

- 我们也可以据此，通过多视角来实现体积估计

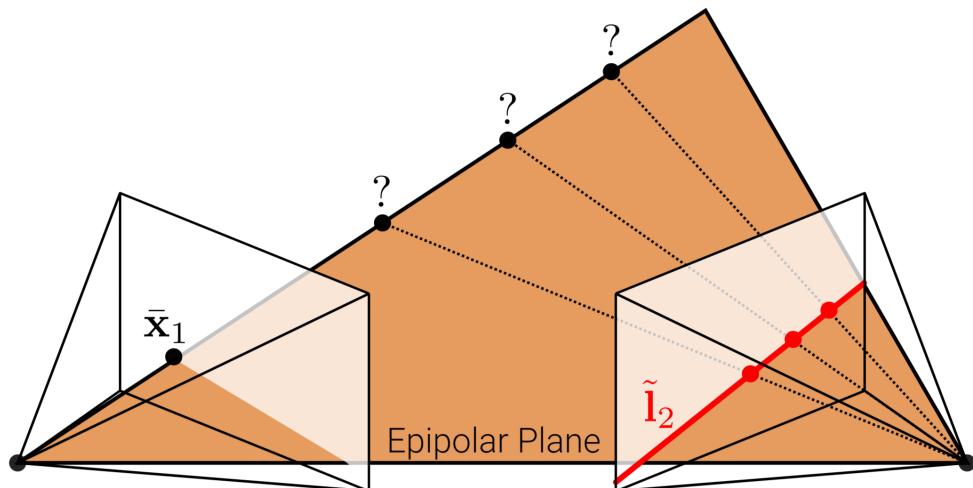


[Logothetis, Mecca and Cipolla: A differential volumetric approach to multi-view photometric stereo. ICCV, 2019](#)

## 8.3 Shape-from-X

在这一单元中，我们将快速浏览一下目前所有学到和未学到的 Shape-from-X 技术

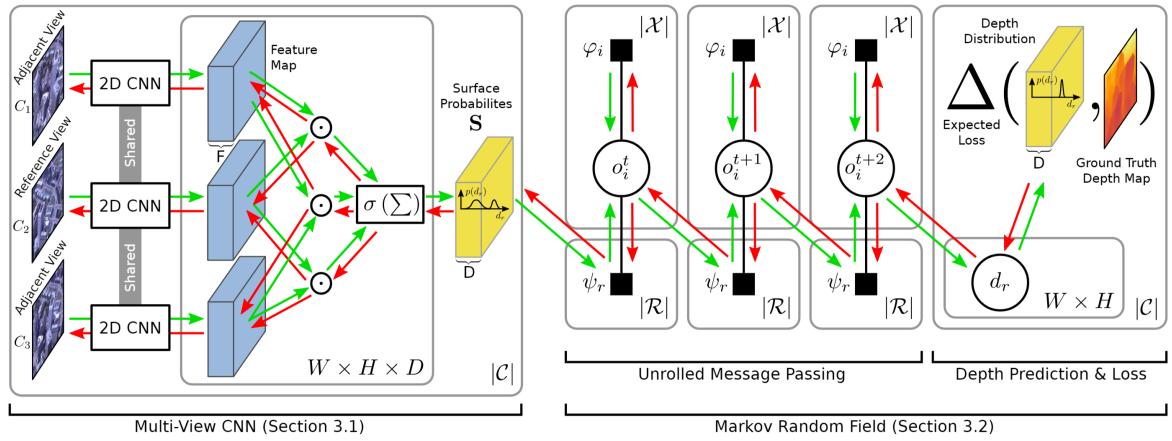
- Binocular Stereo



- 利用 Epipolar 几何学，从 2 个视角的图像中寻找对应关系匹配

- Multi-View Stereo

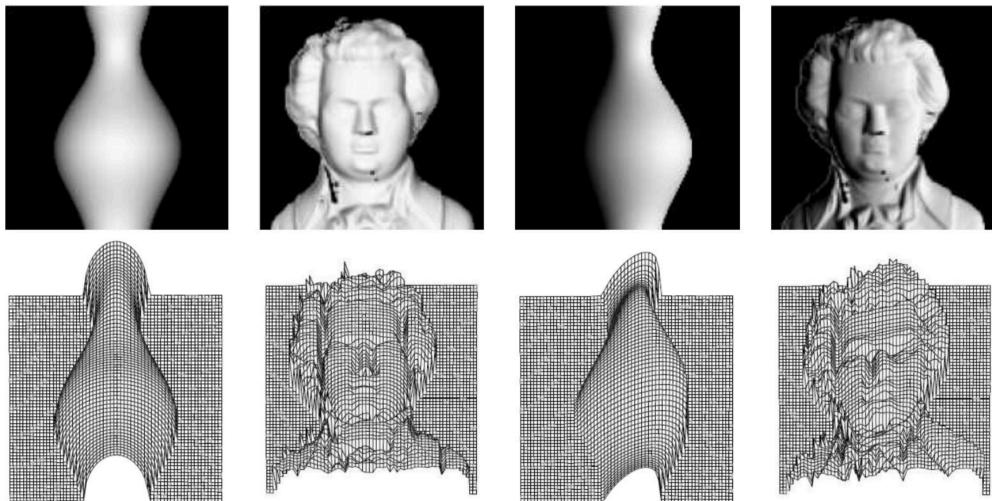
-



- 查找多个视角图像中的对应关系匹配，或者评估体积的模型

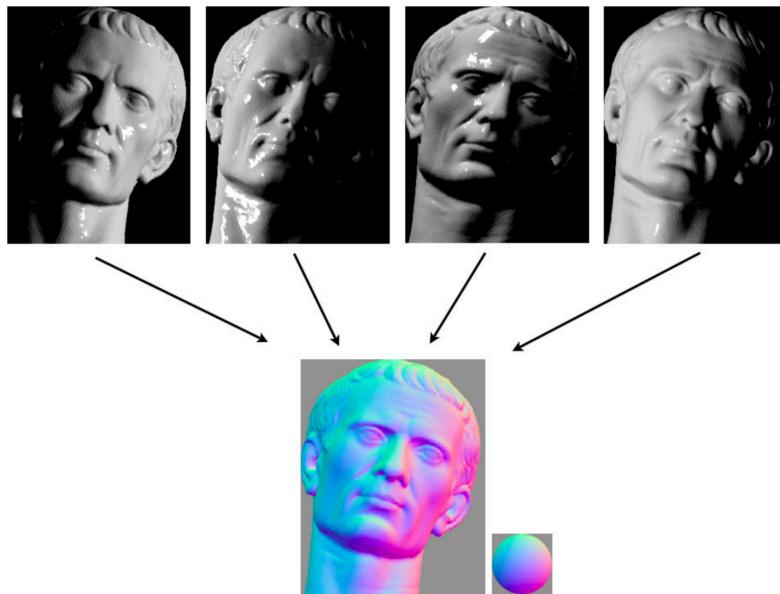
Paschalidou, Ulusoy, Schmitt, van Gool and Geiger: RayNet: Learning Volumetric 3D Reconstruction with Ray Potentials. CVPR, 2018.

- Shape-from-Shading



- ■ 如 8.1 所讲，从单张图像的阴影中估计形状（平滑、Lambertian 的假设）

- Photometric Stereo



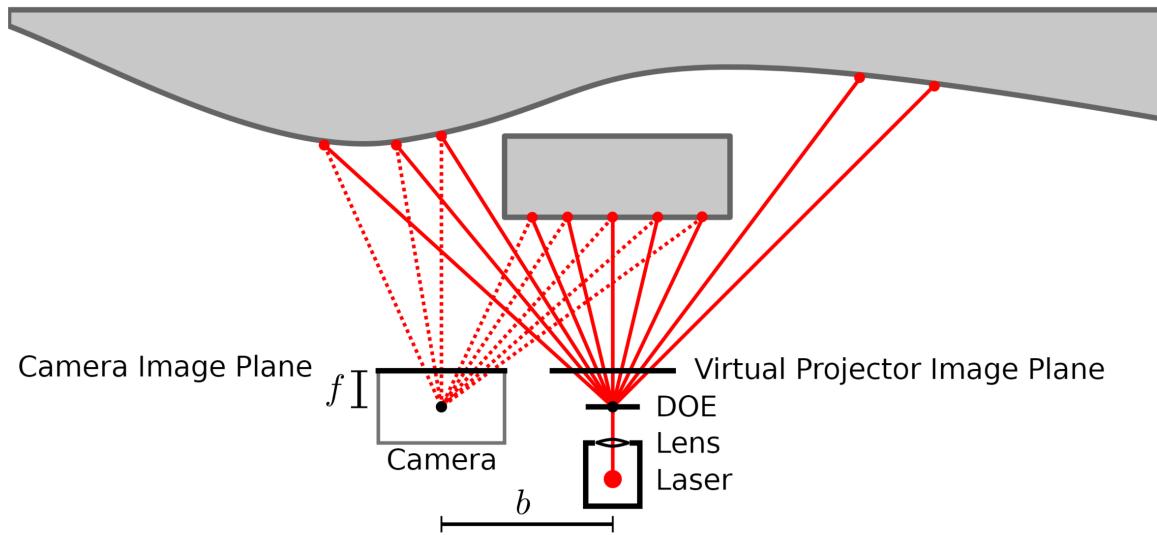
- ■ 如 8.2 所讲，同一位置不同光照配置的多幅图像中评估形状、反射率
- Shape-from-Texture



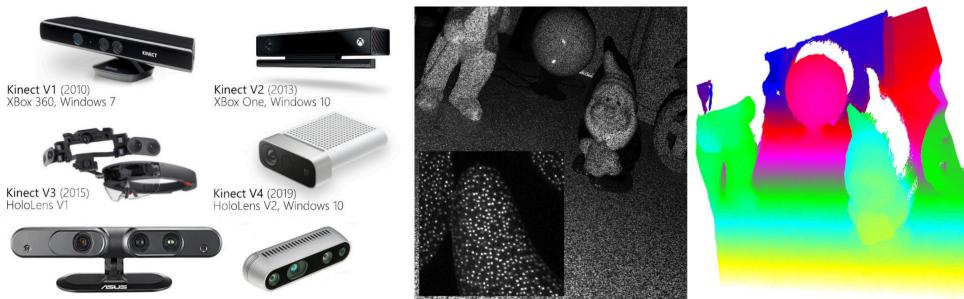
- 这是一种不太常见的评估方式，因为对于 texture 的研究和特征提取也是一个专门的领域——从本地纹理特征中评估形状

**Verbin, E. Zickler: Toward a Universal Model for Shape From Texture. CVPR, 2020.**

- Structured Light



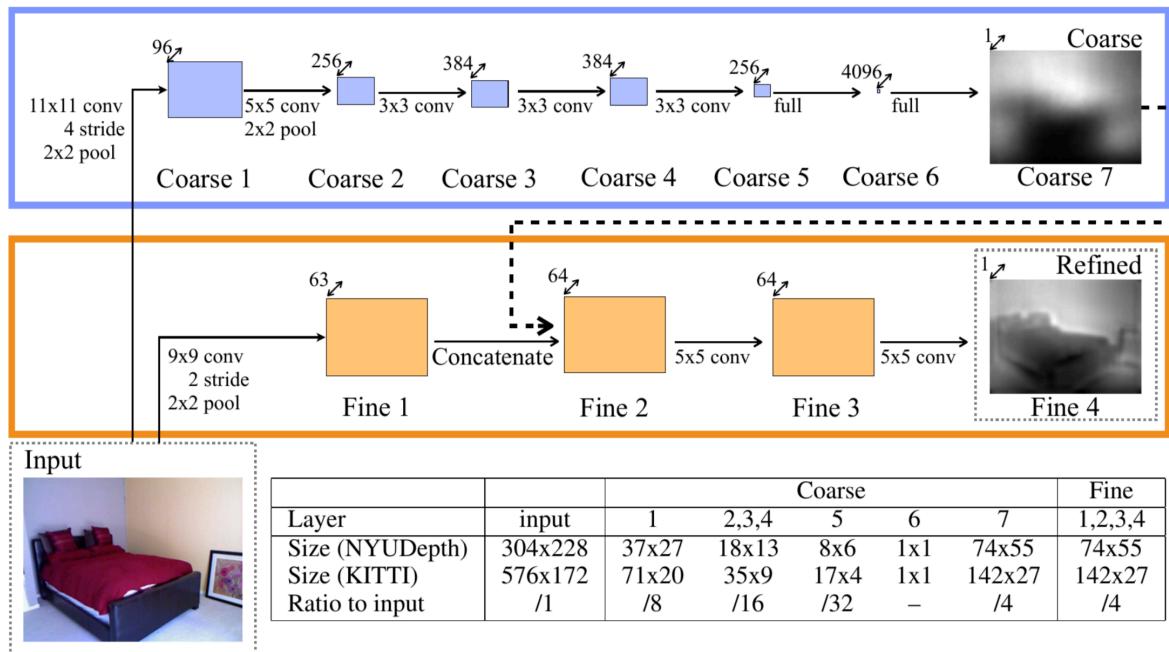
- 还有一种很常见的技术，称为结构光结束——采用模式化的投影仪来照亮场景；如果该模式化的投影设备的校准参数是已知的，那么一个相机就够了



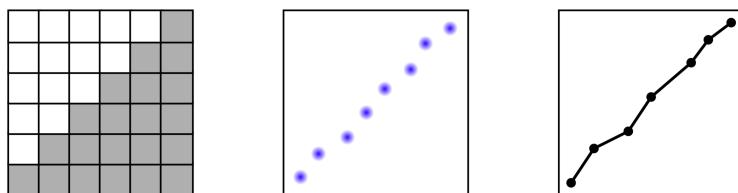
- 结构光技术在很多方面都有应用，最早的平民化设备是在 2010 年由微软提出的 Kinect V1 设备，它最早是为了游戏行业开发的，但是却在机器人、计算机视觉等领域有了很大的应用
- 结构光设备一般来说必须在室内应用，因为它的投影仪发出的光照非常微弱，如果有阳光等强光的干扰，则无法正常接收到合适的图像

Tosi, Liao, Schmitt and Geiger: SMD-Nets: Stereo Mixture Density Networks. CVPR, 2021.

- Monocular Depth Estimation



- 训练模型从大型的标注 RGB 数据集中，学习深度估计的预测



Voxels



Points



Meshes

[Maturana et al., IROS 2015]

[Fan et al., CVPR 2017]

[Groueix et al., CVPR 2018]

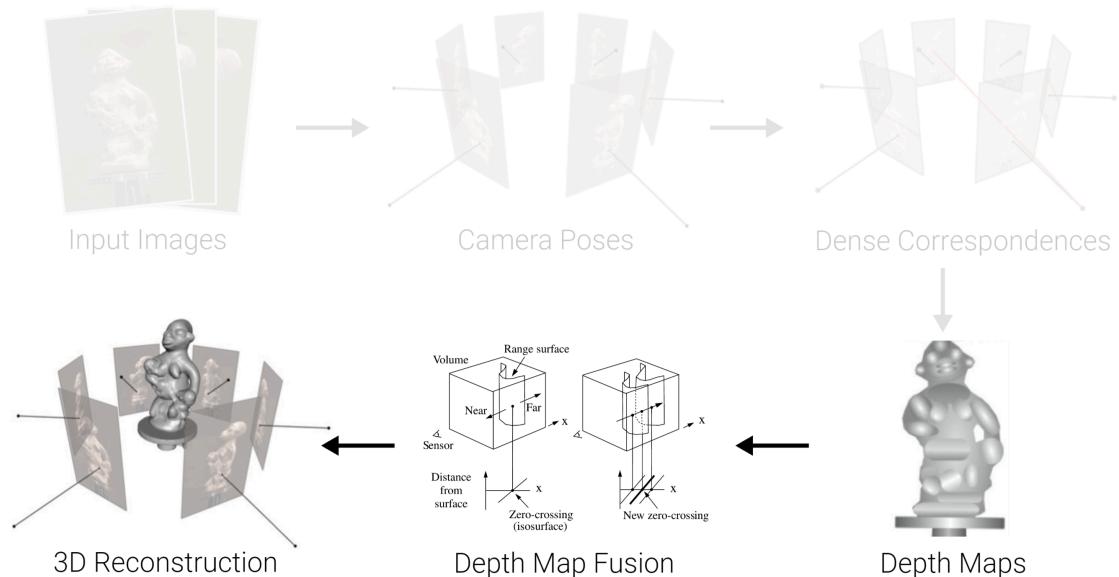
- 单目深度估计也是一个专门的领域，它已经有了非常多的应用，也有非常多优秀的工作，包括从单个图像中估计体素、点云、网格等方式重建 3D 模型结构

## 8.4 Volumetric Fusion

到目前为止，我们研究的都是还原出单个物体或场景的立体结构，但是对于 3D 还原问题而言，我们希望还原出多个物体，或者说整个场景（包含多个对象）的立体结构

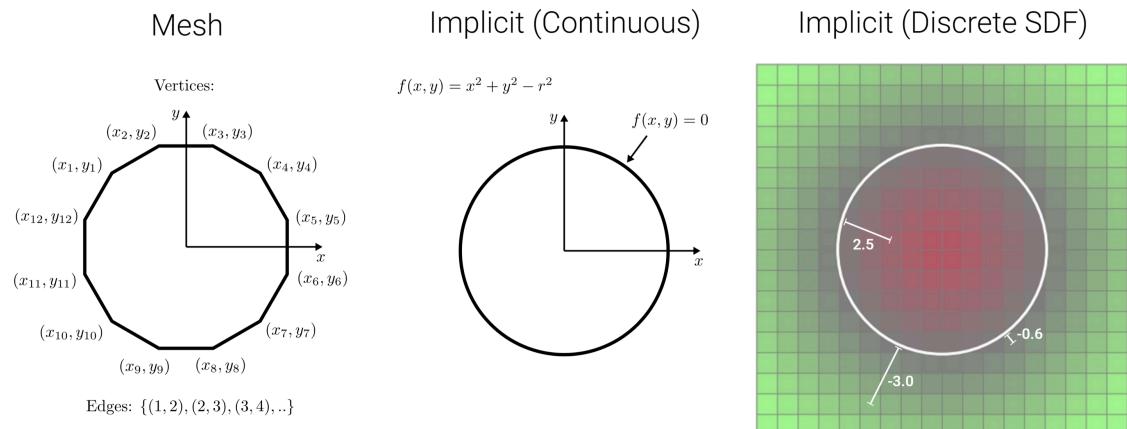
为此，我们通常会使用一种称为 Volumetric Fusion 的技术

- 传统的 3D 重建流程如下



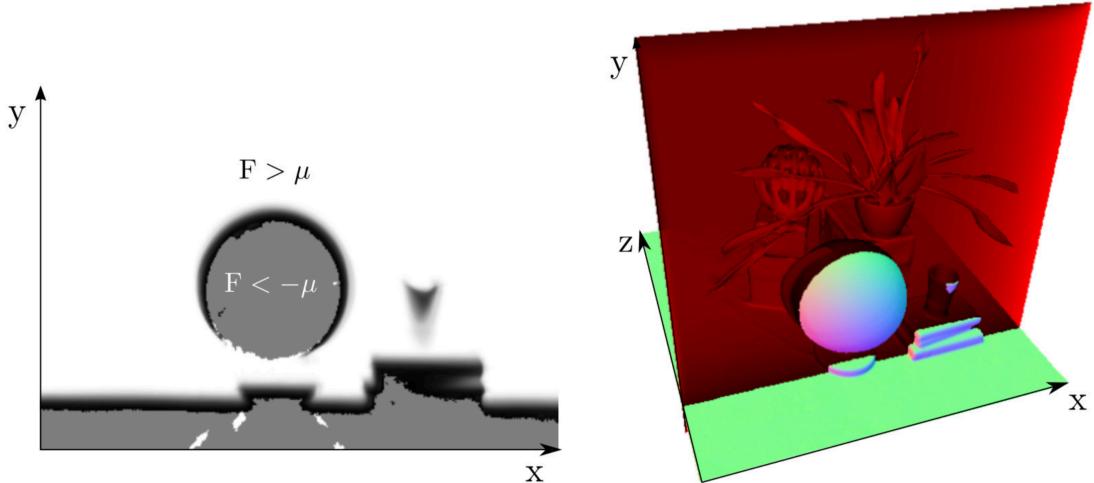
- ■ 输入图像，估计每一个视角的相机姿态，然后进行稠密相关性估计；再然后计算一种中间信息，例如深度图——然后进行深度图融合，最后将它们重建为 3D 模型

- 表示方式



- ■ Mesh 表示 (离散)

- 网格化表示，用多个点和多条边来表示
- 隐式表示（连续）
  - 我们也可以用连续域中的表达式来表示
- Signed Distance Function (SDF) 隐式表示
  - SDF 模型存储每一个体素到最近的表面的带符号距离
  - 因此，表面被隐式地表示为 0 值线
- 隐式地表示方式有一个好处，在于我们可以比较容易的改变其拓扑结构



- ■ 例如右图中，用一个红色面切割物体，横截面得到左侧的表示  
——当我们整体改变 SDF 数值时，物体的边界也在发生变化  
——例如，我们可以通过将所有的值都增加直至变成正值，从而“删除”物体

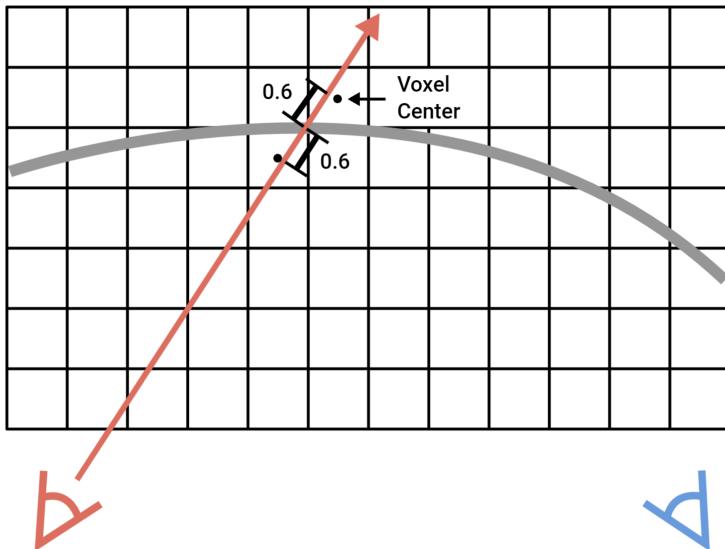
**拓扑学：**研究空间的形状、结构以及不同空间之间连续变换性质的一门学科，通俗地说，拓扑关心的是物体在连续变化下的“骨架”特征，而不在意形状的细节

例如，一个甜甜圈和一个咖啡杯在拓扑学中是“等价的”，因为可以通过连续变形将一个变成另一个（甜甜圈的“洞”变成了杯子的“手柄”）。

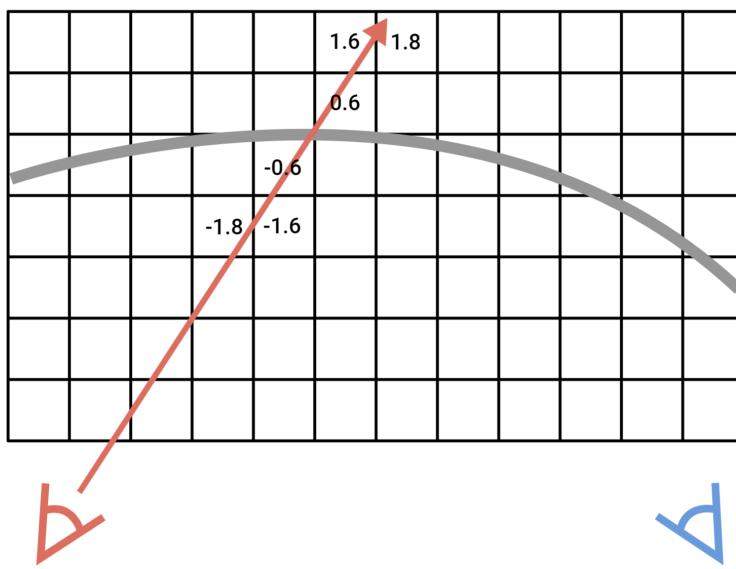
- 接下来，我们可以正式考虑 Volumetric Fusion 的过程，具体可以分为 3 个步骤
  1. Depth-to-SDF Conversion
  2. Volumetric Fusion
  3. Mesh Extraction

### 8.4.1 Depth-to-SDF Conversion

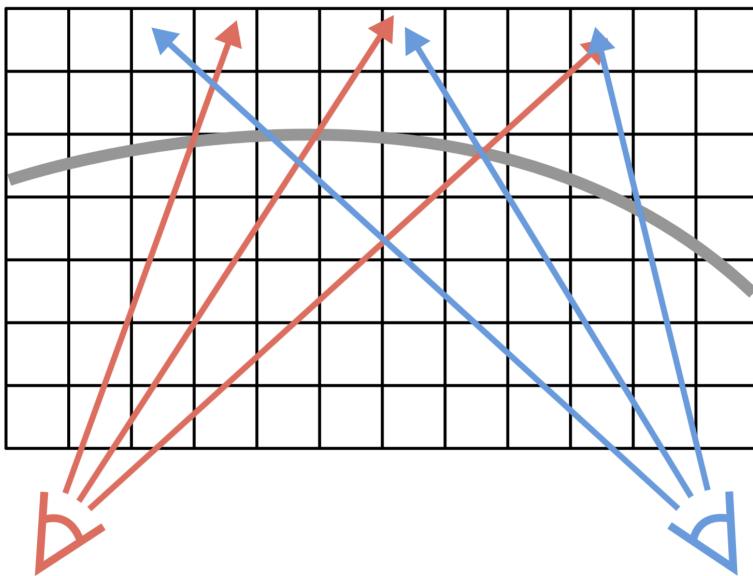
- 首先，我们将 Depth 表示转化为 SDF 表示，但实际上这个过程并没有那么容易，因为我们并不知道每个体素距离物体表面的实际距离



- 由于不知道实际值，我们计算沿着光线（图中红色箭头）的距离作为近似值，利用射线上的“近似距离”来更新体素的 SDF 值

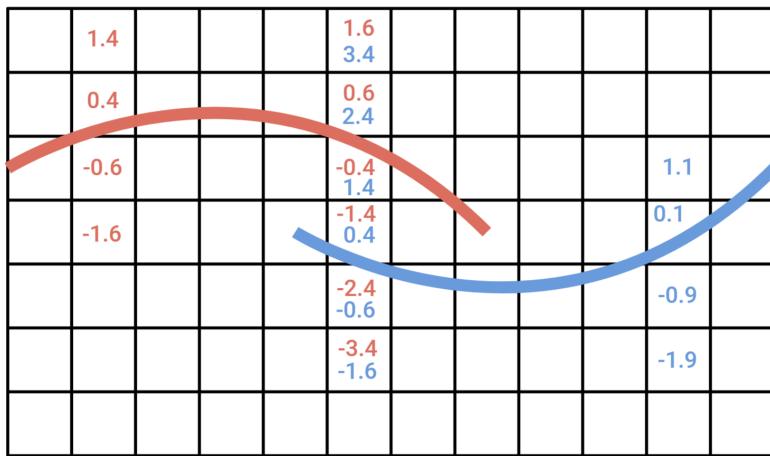


- 但是这种估计方法只在物体表面附近是有效的，因为射线的误差会随着距离增加而扩大，因此对于远离物体表面的区域，该估计方法可能产生较大的误差

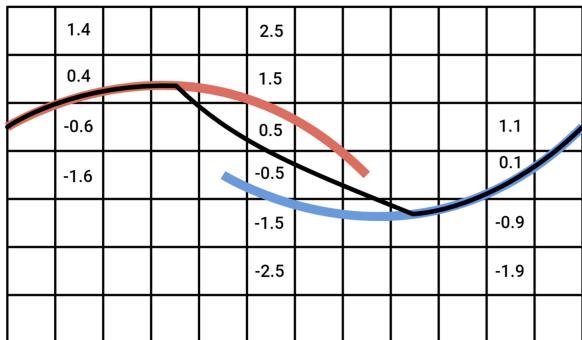


- 对于多个相机光线相交的情况下，有一些体素没有任何光线与其相交，我们可能需要进行插值运算；然而，在实际应用中，我们通常会将体素还原到对应的图像的像素，然后估计其深度，再进行计算

#### 8.4.2 Volumetric Fusion



- 首先，为了简化问题，我们考虑正交投影——也即，观察光线都是平行直射的
  - 我们可以看到，这里有红色、蓝色两个相机实例，以及对应各自不同的零值线，各自的梯度场
  - 对于 Volumetric Fusion 问题，我们在这里对所有的实例取平均值，如下图所示



- 图中黑色的部分就是红色、蓝色相机的实例融合的结果——可以看出，融合的结果并不具有很好的连续性——因为有一部分黑色仅来自红色相机的贡献，有一部分黑色同时有红色、蓝色相机的贡献
- 下面让我们数学化这个过程，SDF Fusion 计算每个体素的加权平均值 \$\$
\begin{aligned}
D(\mathbf{x}) &= \frac{\sum\_i w\_i(\mathbf{x}) d\_i(\mathbf{x})}{\sum\_i w\_i(\mathbf{x})} \\
w(\mathbf{x}) &= \sum\_i w\_i(\mathbf{x})
\end{aligned}
\$\$

$w_i(\mathbf{x}), d_i(\mathbf{x})$  : 相机  $i$  沿着光线到达体素  $\mathbf{x}$  的权重和距离 –  $W(\mathbf{x}), D(\mathbf{x})$

- 为什么我们要使用不同权重的加权平均呢？应该采取什么样的权重分配策略？
  - !

`[[lec_08_shape_from_x.pdf#page=77&rect=32,72,421,231|lec_08_shape_from_x, p.66|600]]`

- 对于物体表面以外的空间，我们有真是的观测值——而对于物体以内的值，并没有很高的置信度——因为它无法在图像中直接展示出来；因此，我们利用权重来降低物体表面后的 SDF 值

- 同时，当我们有多个相机时，我们可以降低一些效果不太好的相机的 SDF 权重——由于观察角度问题、观察距离问题，导致这些相机的 SDF 误差较大，我们可以通过使用权重的方式来平衡这一问题

- 使用加权平均的策略，不仅可以平衡测量值，使其更加可靠；并且也是因为它符合加权最小二乘的数学优化目标 \$\$

$$D(\mathbf{x}) = \frac{\sum_i w_i d_i(\mathbf{x})}{\sum_i w_i}$$

- 它也是下面加权最小二乘问题的解

$$D^* = \underset{D}{\arg \min} \sum_i w_i (d_i - D)^2$$

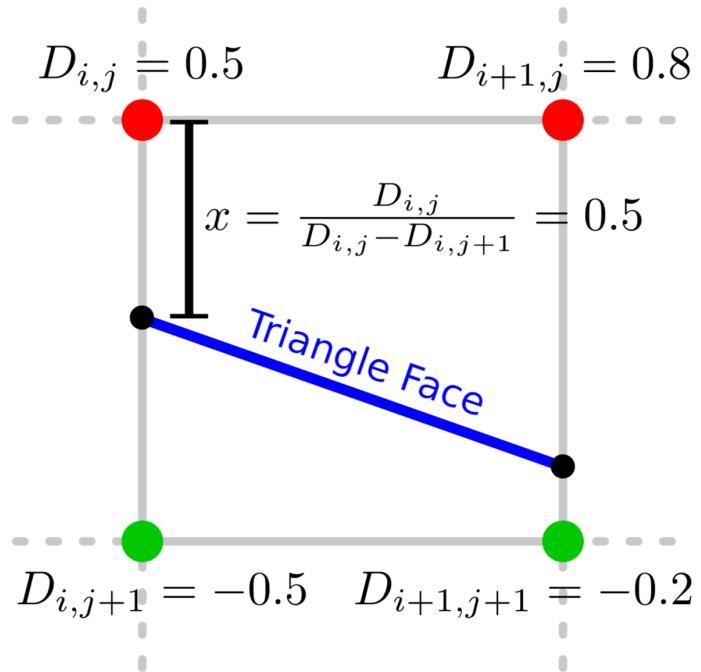
- 换句话说，在每个体素  $\mathbf{x}$  的位置，加权平均计算出的  $D(\mathbf{x})$  也是加权最小二乘问题的解

### 8.4.3 Mesh Extraction

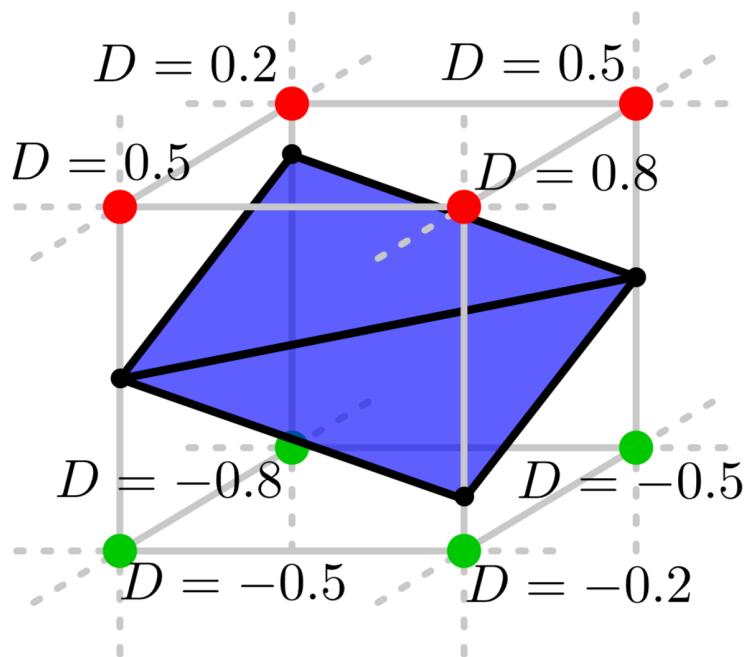
- 那么，我们如何可以从一个 SDF 结果中提取网格呢？这里使用到的就是 **Marching Cubes**（行进立方）算法

[Curless and Levoy: A Volumetric Method for Building Complex Models from Range Images. SIGGRAPH, 1996.](#)

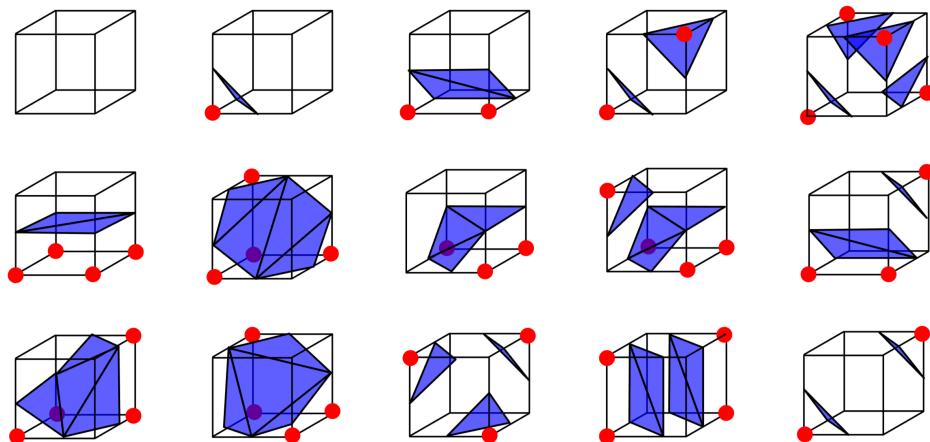
- 我们如何从 SDF 中提取网格？



- ■ 首先我们看一个 2D 的例子，图中是 2D 的 SDF 的一个像素，红色点为正值点，绿色点为负值点
- 根据 SDF 结果，我们知道物体的表面一定是位于正负值交界处的，因此我们遍历所有的体素/像素，检测符号变化的地方，然后插入一个三角形
- 在每个网格单元中，顶点  $D$  的符号定义单元的拓扑类型，在二维情况下，共有  $2^4 = 16$  种拓扑类型；在三维情况下，共有  $2^8 = 256$  种拓扑类型
- 确定拓扑类型后，我们怎么确定三角形具体的顶点位置呢？  
——分别采用线性插值的方式，如图中所示，找到正负点之间的边的线性零值点，作为顶点



- 上面是一个三维的例子，方法本质和二维的情况相同



- 在三维情况下， $2^8 = 256$  种拓扑结构可以分类为上图的 15 种类别  
(在合并旋转、对称相同的情况下)

#### 8.4.4 Application

- KinectFusion

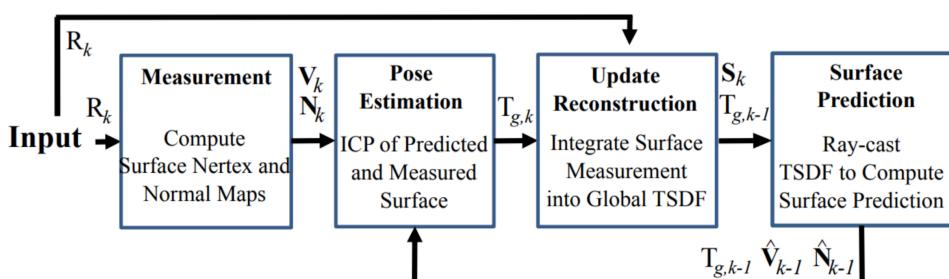
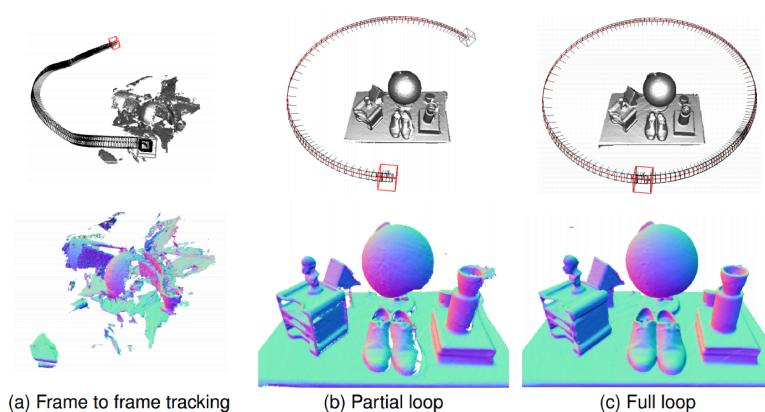


Figure 3: Overall system workflow.

- KinectFusion利用深度摄像头采集的连续深度图，通过Volumetric Fusion，将深度图转换为三维场景的高精度表面表示
- 提出了一个基于 ICP (Iterative Closest Point) 的位姿估计方法，通过连续深度帧对比精确跟踪摄像头的 6 自由度位姿 (位置与方向)

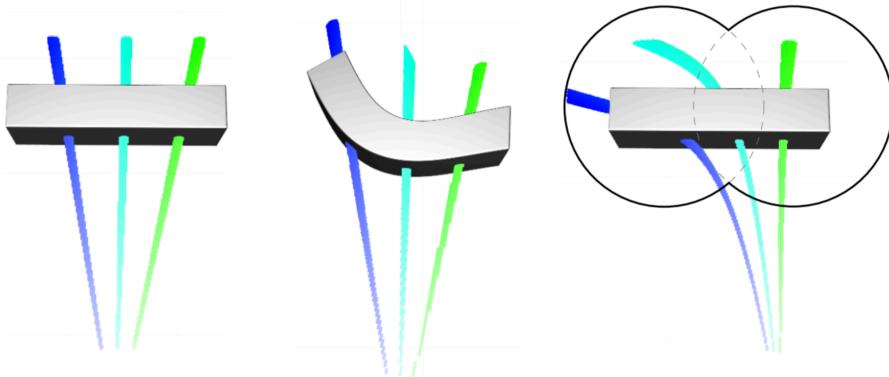


- 在室内场景可以重建细节丰富的三维模型
- 能够在现代 GPU 的支持下实时运行
- 在多种复杂场景下表现出良好的稳定性和重建效果，尤其是在摄像头快速移动和动态变化场景中

[Newcombe et al.: KinectFusion: Real-time dense surface mapping and tracking. ISMAR, 2011.](#)

- DynamicFusion

-



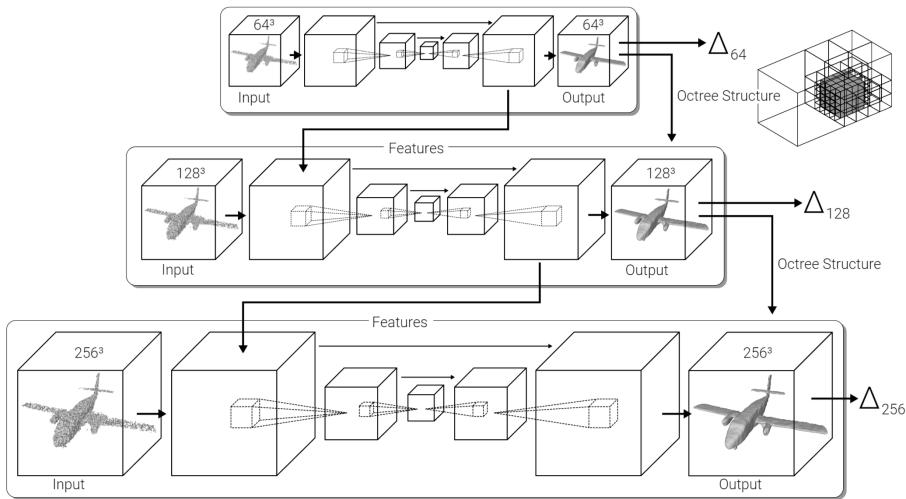
(a) Live frame  $t = 0$  (b) Live Frame  $t = 1$  (c) Canonical  $\mapsto$  Live

- 能够在非刚性场景下实现实时三维重建和跟踪的方法，系统不需要额外的先验模型（如骨骼模型或模板），完全基于深度数据在线推断物体的形变
- 通过将非刚性场景的深度图输入，联合优化摄像头位姿和一个基于位姿场的形变模型，实时融合生成三维表面的动态重建

[Newcombe, Fox, Seitz: DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. CVPR, 2015.](#)

- OctNetFusion

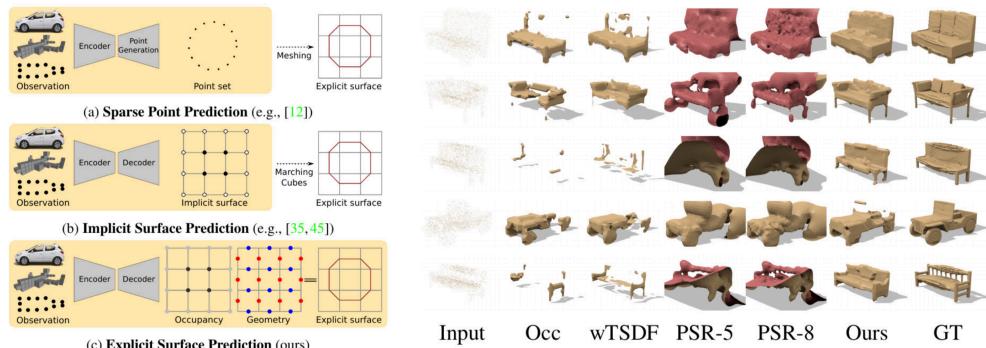
-



- 一种基于数据驱动的深度融合方法，名为 OctNetFusion，结合了深度学习和稀疏八叉树表示，用于实时的三维重建和深度图数据融合

[Riegler, Ulusoy, Bischof and Geiger: OctNetFusion: Learning Depth Fusion from Data. 3DV, 2017.](#)

- Deep Marching Cubes



- ■ 一种深度学习方法，用于从三维点云或体素数据中学习显式表面表示。其核心是将传统的 Marching Cubes 算法与深度学习结合，通过端到端学习的方式生成高质量的三维表面

[Liao, Donn é and Geiger: Deep Marching Cubes: Learning Explicit Surface Representations. CVPR, 2018.](#)