Phase Shifting Structured Light - Camera Calibration

David Gu

Computer Science Department
Stony Brook University

gu@cs.stonybrook.edu

August 5, 2022

Camera Model

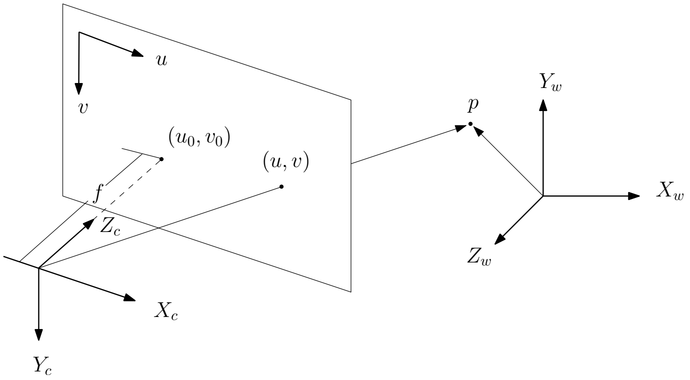


Figure: Model of a video camera.

Model of Projector and Camera

In practice, the mathematical model for camera and projector can be described using the following pipeline:

$$\begin{array}{ccccccccc}
(X_w, Y_w, Z_w) & \xrightarrow{\varphi_1} & (X_c, Y_c, Z_c) & \xrightarrow{\varphi_2} & (x_c, y_c) & \xrightarrow{\varphi_3} & (x_c^d, y_c^d) & \xrightarrow{\varphi_4} & (u_c, v_c) \\
\downarrow id & & & & & & & & \downarrow \psi \\
(X_w, Y_w, Z_w) & \xrightarrow{\varphi_1} & (X_p, Y_p, Z_p) & \xrightarrow{\varphi_2} & (x_p, y_p) & \xrightarrow{\varphi_3} & (x_p^d, y_p^d) & \xrightarrow{\varphi_4} & (u_p, v_p)
\end{array}$$

The top row shows the image formation process of the camera, the bottom row shows the image formation of the projector.

Pinhole Camera Model

1. The map $\varphi_1 : (X_w, Y_w, Z_w) \to (X_c, Y_c, Z_c)$ transforms from the *world coordinates* to the *camera coordinates*, which is a rotation and a translation, as shown in Eqn. (1);
2. $\varphi_2 : (X_c, Y_c, Z_c) \to (x_c, y_c)$ is the pinhole camera projection, maps from *camera coordinates* to the *camera projective coordinates*, as shown in Eqn. (2);
3. $\varphi_3 : (x_c, y_c) \to (x_c^d, y_c^d)$ is the camera distortion map in Eqn. (5), transforms from *camera projective coordinates* to the *distorted camera projective coordinates*, the distortion includes both *radial distortion* Eqn. (3) and *tangential distortion* Eqn. (4);

Pinhole Camera Model

1. $\varphi_4 : (x_c^d, y_c^d) \to (u_c, v_c)$ is the projective transformation in Eqn. (6), which maps from the *distorted camera projective coordinates* to the *camera image coordinates*.
2. The inverse of φ_3 maps from the *distorted camera projective coordinates* to the *camera projective coordinates*, $\varphi_3^{-1} : (x_c^d, y_c^d) \to (x_c, y_c)$, is Heikkila's formula in Eqn. (7).

Pinhole Camera Model

A point p in the world coordinate system is (X_w, Y_w, Z_w), in the camera coordinate system is (X_c, Y_c, Z_c), then

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T. \tag{1}$$

where R is the rotation matrix from the world coordinate system to the camera coordinate system, T is the translation vector.
The projection to the camera projective coordinates (without considering distortions) are given by:

$$\left\{ \begin{array}{lcl} x_c & = & X_c/Z_c \\ y_c & = & Y_c/Z_c \end{array} \right. \tag{2}$$

Distortion Model

In practice, the lense of the camera introduces distortions, the imaging is not ideal pinhole camera model, in calibration the distortions need to be considered. In general, the distortion include both radial distortion and tangential distortion. We use (x, y) to represent the projective coordinates on the image plane, such as (x_c, y_c). The radial distortion $(\delta_{xr}, \delta_{yr})$ are represented as

$$\begin{cases} \delta_{xr}(x,y) &= x(k_1 r^2 + k_2 r^4 + k_3 r^6 + \cdots), \\ \delta_{yr}(x,y) &= y(k_1 r^2 + k_2 r^4 + k_3 r^6 + \cdots), \end{cases} \tag{3}$$

where $r^2 = x^2 + y^2$, $k_1, k_2, k_3, \cdots$ are the radial distortion parameters. The tangential distortion $(\delta_{xt}, \delta_{yt})$ can be represented as

$$\begin{cases} \delta_{xt}(x,y) &= 2p_1 xy + p_2(r^2 + 2x^2), \\ \delta_{yt}(x,y) &= p_1(r^2 + 2y^2) + 2p_2 xy, \end{cases} \tag{4}$$

where p_1, p_2 are tangential distortion parameters.

Distortion Model

After considering the camera distortion, the distorted camera projective coordinates (x_d, y_d) of the point p can be represented as

$$\begin{cases} x_d &= x + \delta_{xr}(x,y) + \delta_{xt}(x,y) \\ y_d &= y + \delta_{yr}(x,y) + \delta_{yt}(x,y) \end{cases} \quad (5)$$

After projective transformation, the camera image coordinates of the point p can be represented as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = A \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \quad (6)$$

where f_u, f_v are the effective focal lengths along u and v directions respectively, s is the *slant parameter* of the coordinate axis, (u_0, v_0) are the coordinates of principle point, the intersection point between the optical axis of the camera and the image plane.

Camera Calibration

camera calibration aims at find all the parameters of the camera, including

- Extrinsic parameters: rotation R, translation T;
- Intrinsic parameters: effective focal lengths f_u, f_v; slant parameter s, principle center (u_0, v_0);
- Distortion parameters: radial distortion parameters k_1, k_2, k_3; tangential distortion parameters p_1, p_2.

In practice, intrinsic parameters also include distortion parameters. Generally, k_3 and s are small enough, and usually treated as 0's. We denote all the extrinsic and intrinsic parameters as

$$\mu = (R_c, T_c, f_u, f_v, s, u_0, v_0),$$

and all the distortion parameters as

$$\lambda = (k_1, k_2, k_3, p_1, p_2).$$

Calibration Board

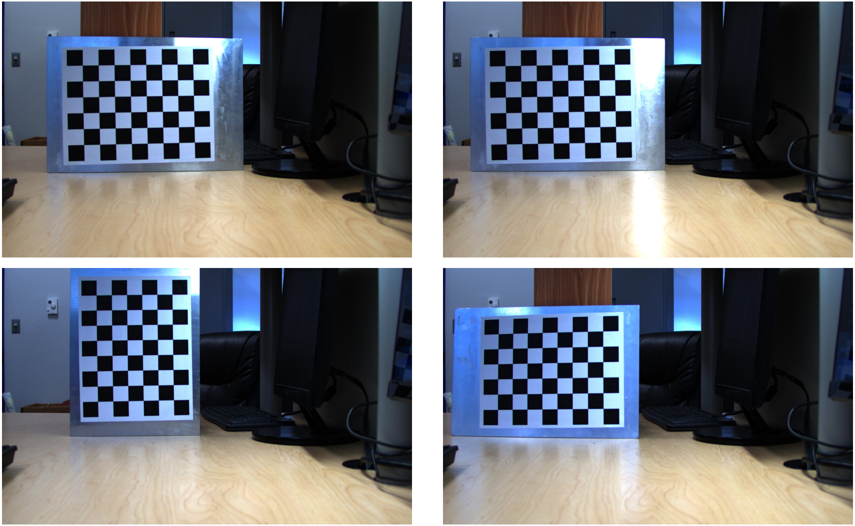


Figure: calibration target board.

Target Board

During the calibration process, each time we fix the position of the target board plane π, the local coordinates system of the target plane is treated as the world coordinates system, the plane equation is $Z_w = 0$, the centers of every star center is known, denoted as

$$\{(X_w^1, Y_w^1), (X_w^2, Y_w^2), \cdots, (X_w^n, Y_w^n)\},$$

the image coordinates of each star center is captured

$$\{(u_1, v_1), (u_2, v_2), \cdots, (u_n, v_n)\}.$$

From the mapping $\{(X_w^i, Y_w^i)\} \to \{(u_i, v_i)\}$, by optimization, we can estimate the extrinsic and intrinsic parameters μ.

Intrinsic and Extrinsic Parameters Estimation

The image formation mapping, also called the *forward projection*, depends on the extrinsic and the intrinsic parameters,

$$\varphi_{\mu,\lambda} : (X_w, Y_w, Z_w) \to (u, v).$$

The calibration problem is formulated as an optimization problem:

$$\min_{\lambda,\mu} E(\lambda, \mu) = \min_{\lambda,\mu} \sum_{i=1}^{n} \|\varphi_{\lambda,\mu}(X_w^i, Y_w^i) - (u_i, v_i)\|^2.$$

By alternating optimizations, we can reach the optimum

$$(\lambda^*, \mu^*) = argmin_{\lambda,\mu} E(\lambda, \mu).$$

The optimization can be carried out using gradient descend algorithm:

$$\frac{\nabla E}{\partial \lambda} = \left[\frac{\partial E}{\partial k_1}, \frac{\partial E}{\partial k_2}, \frac{\partial E}{\partial k_3}, \frac{\partial E}{\partial p_1}, \frac{\partial E}{\partial p_2}\right]^T.$$

Back Projection

The inverse of the forward projection $\varphi_{\lambda,\mu}$ is called the back projection. Because the radial distortion Eqn. (3) and the tangential distortion Eqn. (4) are nonlinear, the transformation from (x, y) to (x_d, y_d) in Eqn. (5) can not be directly inverted. One needs to use iterative method or polynomial approximation method to invert Eqn. (5).
Heikkilä use the following polynomial approximation to compute the inverse transformation:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{G} \begin{bmatrix} x_d(1 + a_1 r_d^2 + a_2 r_d^4) + 2a_3 x_d y_d + a_4(r_d^2 + 2x_d^2) \\ y_d(1 + a_1 r_d^2 + a_2 r_d^4) + a_3(x_d^2 + 2y_d^2) + 2a_4 x_d y_d \end{bmatrix}, \tag{7}$$

where

$$G = (a_5 r_d^2 + a_6 x_d + a_7 y_d + a_8) r_d^2 + 1, \tag{8}$$

and $r_d^2 = x_d^2 + y_d^2$, $a_1, a_2, \cdots, a_8$ are back projection distortion parameters.

Light Field Camera Model

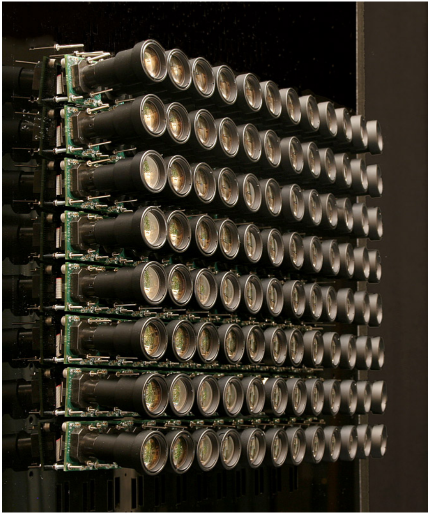


Figure: Stanford light field Camera.

Definition (light field)

All the rays in $\mathbb{R}^3$ form a 4 dimensional space. Each ray is associated with a color.

Light field camera has been overdued in vision and graphics. Each pin-hole camera collects a 2 dimensional family of rays. The camera array is 2 dimensional.

Lytro camera

First shoot, then focus !

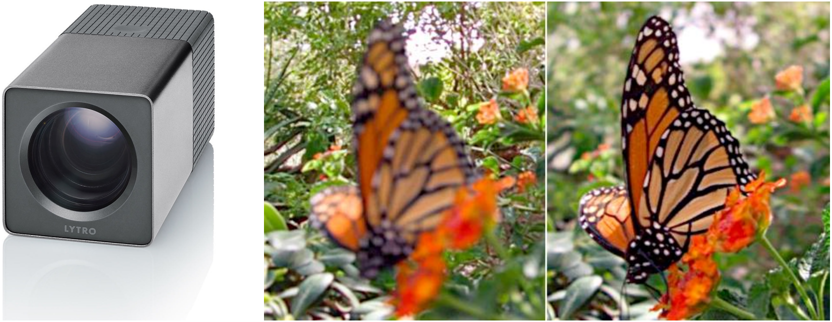


Figure: Lytro Camera and light field image.

Light Field Camera Model and Calibration

In the light field camera model, each pixel is associated with a ray, the rays are independent. The light field camera model is much more general, and much more accurate than the conventional pinhole model.

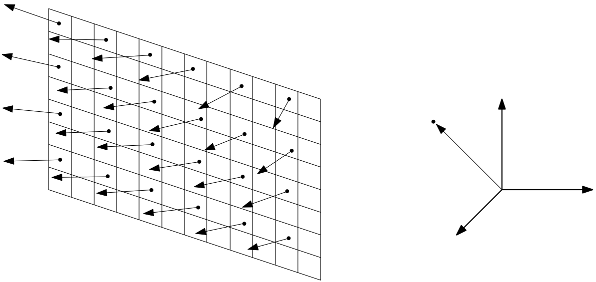
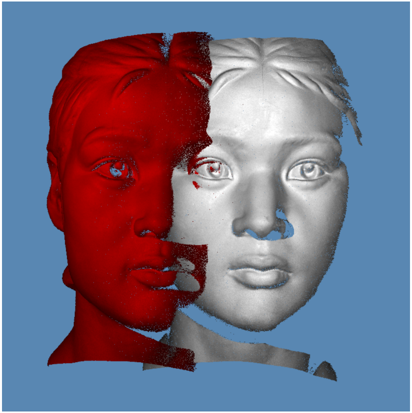


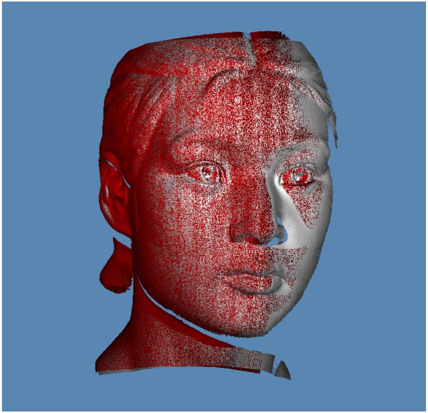
Figure: Light Field Model

Point Cloud Fusion

Point Cloud Fusion



(a). before fusion



(b). after fusion

Figure: Point cloud fusion.

Normal Estimation

(a). merged point clouds

(b). with estimated normal

Figure: Normal estimation.

Point Cloud Fusion

One of the fundamental problems in SLAM (Simultaneous localization and mapping) is to fuse point clouds with global consistency.

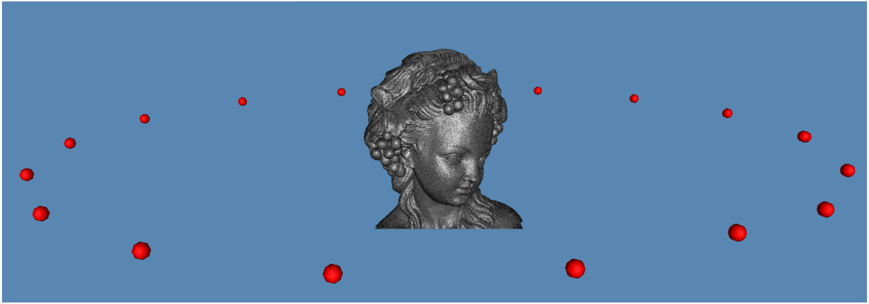


Figure: point cloud fusion with global consistency.

Loop Close Problem

Definition (View Graph)

The view graph $G = (V, E)$ is a graph, where each node represents a point cloud, each edge represents two overlapping point clouds.

Problem (Loop Close)

Given a view graph $G = (V, E)$, for each oriented edge $[n_i, n_j]$, find a rigid motion (a rotation and translation) from n_i to n_j, T_{ij}, such that, for each loop γ with ordered nodes $n_0, n_1, \cdots, n_{k-1}$, the composition

$$T_{k-1,0} \circ T_{k-2,k-1} \circ \cdots \circ T_{1,2} \circ T_{0,1} = Id.$$

Point Cloud Normal Estimation

Figure: Normal estimation for merged point clouds.

Point Cloud Fusion

Problem

Given two corresponding point clouds:

$$P = \{p_1, p_2, \cdots, p_n\}, \quad Q = \{q_1, q_2, \cdots, q_n\}$$

p_k corresponds to q_k, find the optimal translation T and rotation R to minimizes the registration error:

$$E(R, T) := \frac{1}{n}\sum_{i=1}^{n} \|p_i - (Rq_i + T)\|^2.$$

Point Cloud Fusion

Center of Mass

Compute the centers of mass of P and Q,

$$\mu_p = \frac{1}{N_p}\sum_{i=1}^{N_p} p_i, \quad \mu_q = \frac{1}{N_q}\sum_{j=1}^{N_q} q_j.$$

Covariance Matrix

The covariance matrix is given by

$$M = \sum_{i=1}^{N_p} (p_i - \mu_p)(q_i - \mu_q)^T$$

Point Cloud Fusion

Covariance Matrix

Denote the singular value decomposition (SVD) of the covariance matrix by:

$$M = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} V^T$$

where $U, V \in GL(\mathbb{R}, 3)$ are unitary, and $\sigma_1 \geq \sigma_2 \geq \sigma_3$ are the singular values of M.

Point Cloud Fusion

Theorem

If the covariance matrix is full rank, then the optimal solution of $E(R, T)$ is unique and given by:

$$R = UV^T$$
$$T = \mu_p - R\mu_q$$

The error $E(R, T)$ is given by

$$E(R, T) = \sum_{i=1}^{N_p} (\|p_i - \mu_p\|^2 + \|q_i - \mu_q\|^2) - 2(\sigma_1 + \sigma_2 + \sigma_3).$$

Proof.

By $\frac{\partial E}{\partial T} = 0$, we obtain

$$\frac{\partial E}{\partial T} = \frac{2}{N_p} \sum_{i=1}^{N_p} (Rq_i + T - p_i) = 2(R\mu_q + T - \mu_p) = 0,$$

hence $T = \mu_p - R\mu_q$. Plug into $E(R, T)$,

$$E(R, T) = \sum_{i=1}^{N_p} \|p_i - (Rq_i + T)\|^2 = \sum_{i=1}^{N_p} \|(p_i - \mu_p) - R(q_i - \mu_q)\|^2$$

$$= \sum_{i=1}^{N_p} \bar{p}_i^T \bar{p}_i - \bar{q}_i^T R^T \bar{p}_i - \bar{p}_i^T R\bar{q}_i + \bar{q}_i^T q_i.$$

where $\bar{p}_i = p_i - \mu_p$ and $\bar{q}_i - \mu_q$. □

Point Cloud Fusion

Proof.

$$
\begin{aligned}
2\sum_{i=1}^{N_p} \bar{q}_i^T R^T \bar{p}_i &= 2\sum_{i=1}^{N_p} Tr(\bar{q}_i^T R^T \bar{p}_i) = 2\sum_{i=1}^{N_p} Tr(R^T \bar{p}_i \bar{q}_i^T) \\
&= 2Tr\left(R^T \sum_{i=1}^{N_p} \bar{p}_i \bar{q}_i^T\right) = 2Tr(R^T M) \\
&= 2Tr(R^T U \Sigma V^T) = 2Tr(V^T R U \Sigma)
\end{aligned}
$$

where $\Sigma = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$, $V^T R^T U$ is a rotation matrix. So the above terms reaches maximum if and only if $V^T R^T U$ is the identity matrix, $R = UV^T$, and

$$
2Tr(V^T R U \Sigma) \leq \sigma_1 + \sigma_2 + \sigma_3.
$$

□

Iterative Closest Point

If the correct correspondences are not known, it is generally impossible to determine the optimal relative rotation and translation in one step.

1. Initialize registration parameters (R, T) and registration error $E(R, T)$;
2. For each point in the scene shape, find the corresponding closest point in the model shape;
3. Calculate registration parameters given point correspondences obtained from step 2.
4. Apply the alignment to the scence shape;
5. Calculate the registration error between the currently aligned scene shape and the model shape;
6. If the error is greater than threshold, return to step 2, else return with new sence shape.

ICP Variants

1. Point subsets from one or both point sets
2. Weighting the correspondences
3. Data association
4. Rejecting outlier point pairs

Selecting Source Points

- Use all the points
- Uniform sub-sampling
- Random sampling
- Feature based sampling
- Normal space sampling: ensure the samples have normals distributed as uniformly as possible

Data Association

- Greatest effect on convergence and speed
- Closest point
- Normal shooting
- Closest compatible point
- Projection
- Using kd-trees or oc-trees

Rejecting Outlier Point Pairs

- Sorting all correspondences with respect to their error and deleting the worst $k\%$ pairs
- k is to estimate with respect to the overlap

Surface Reconstruction

Point Cloud Normal Estimation

Figure: Normal estimation for merged point clouds.

Poisson Mesh Reconstruction

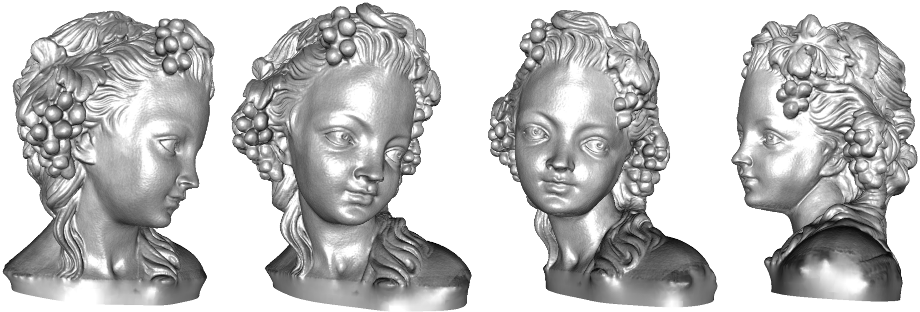


Figure: Poisson mesh reconstruction.

Poisson Mesh Reconstruction

Figure: 3D printed model and the original sculpture.

Reconstruction

Problem (Surface Reconstruction)

Given a set of points $P = \{p_1, p_2, \cdots, p_n\}$ *with* $p_i \in \mathbb{R}^3$*, find a manifold surface* $S \subset \mathbb{R}^3$ *which approximates* P*.*

Approaches

1. Explicit: local surface connectivity estimation, point interpolation. Ball pivoting algorithm, Delaunay triangulation, Alpha shapes, Zippering, image space triangulation;
2. Implicit: signed distance function estimation, mesh approximation; SDF estimation via RBF

Implicit Surface Reconstruction

Implicit Surface Reconstruction

- Generate an implicit surface description from the point cloud
- generate surface from this using marching cubes

Marching Cube Algorithm

Algorithm for $f(x, y) = 0$ level set

1. Sample function on uniform grid
2. Check for each cell whether it intersects the iso-line
    - Compute for cell faces the intersection with the $f(x, y) = 0$
    - connect intersections
3. Repeat for all cells
4. Care for ambiguous configuration

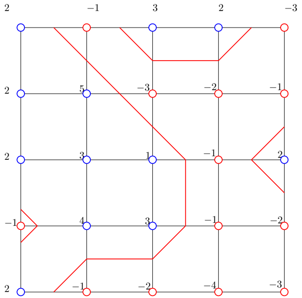


Figure: Marching cube: evaluation on grid points, interpolate the zero level set.

Marching Cube Algorithm

Algorithm for iso-surface

1. Sample function on uniform grid
2. For each cell in grid
   - Mark corners whether they are smaller or larger than iso-value
   - Cell has 8 vertices, therefore there are 256 different +/- configurations (due to symmetry, cases may be reduced to 15)
   - Determine correct case, use lookup table to find triangulation
   - Adjust vertex positions according to linear interpolation

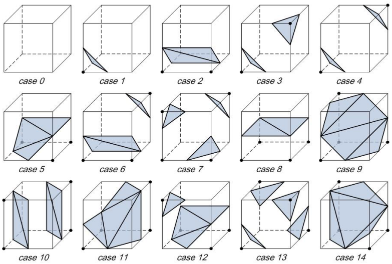


Figure: Marching cube: evaluation on grid points, interpolate the zero level set.

Implicit Surface Reconstruction

Hoppe's Method

- Evaluate the signed distance function on uniform grid (=volume)
- When evaluating the signed distance function at p
  1. Find closest point q with normal n
  2. Compute distance as the distance to the tangent plane at q

$$f(p) = (p - q) \cdot n$$

- Run marching cubes on volume to extract $f(x, y, z) = 0$

Implicit Surface Reconstruction

Hoppe's Method

- The signed distance function is $f(p) = (p - q_i) \cdot n_i$, q_i is the closest point to p;
- $f(p)$ is piecewise linear, defined on the Voronoi diagram of the input points;
- discontinuous along Voronoi edges;
- Marching cubes makes it manifold again;

Partition Unity Implicit

Improvement: To evaluate the implicit function at some point p

- Look for the k nearest samples q_i
- Compute their distance just as before: $d_i = (q_i - p) \cdot n_i$
- Blend these, e.g., based on their distance to p:

$$f(p) = \frac{\sum_{i=1}^{k} w(\|q_i - p\|) d_i}{\sum_{i=1}^{k} w(\|q_i - p\|)}$$

This leads to smoother signed distance function.

Poisson Reconstruction

Definition (Indicator Function)

Suppose M is a volumetric domain in $\mathbb{R}^3$, its indicator function

$$\chi_M(p) := \begin{cases} 1 & \textit{if } \ p \in M \\ 0 & \textit{if } \ p \notin M \end{cases}$$

The gradient of the indicator function $\nabla\chi_M$ equals to the inverse normal field of the boundary surface ∂M.

Poisson Reconstruction

Find the function χ whose gradient best approximate a vector field $\mathbf{v}$,

$$\min_{\chi} \|\nabla\chi - \mathbf{v}\|$$

By Hodge decomposition

$$\mathbf{v} = \nabla\chi + \nabla \times \mathbf{w} + \mathbf{h}$$

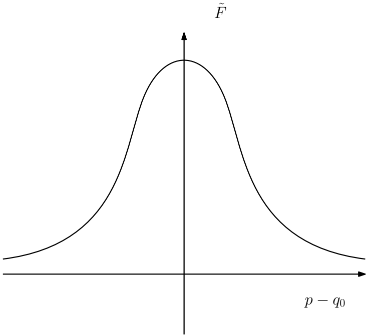
Because $\mathbb{R}^3$ is topologically trivial, the harmonic component $\mathbf{h}$ is zero.

$$\nabla \cdot \mathbf{v} = \nabla \cdot \nabla\chi + \nabla \cdot \nabla \times \mathbf{w} = \Delta\chi$$

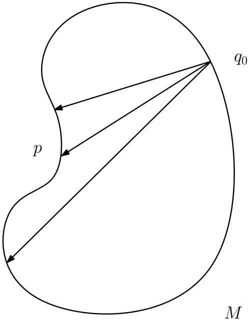
Smoothing the indicator function

$$(\chi * \tilde{F})(q_0) = \int_M \tilde{F}(p - q_0)\chi(p)dp$$

$$\nabla(\chi * \tilde{F})(q_0) = \int_M \tilde{F}(p - q_0)\mathbf{N}(p)dp$$



filter function



integration over all the domain.

Poisson Reconstruction

Given point cloud with normals

1. Set Octree to partition $\mathbb{R}^3$
2. Compute the vector field
3. Compute indicator function
4. Extract iso-surface

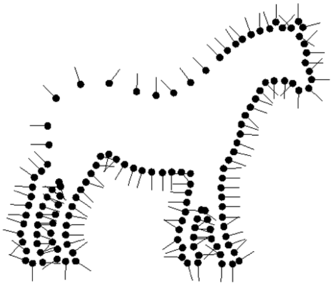


Figure: Input point cloud

Poisson Reconstruction

Given point cloud with normals

1. Set Octree $\mathcal{Q}$ to partition $\mathbb{R}^3$ with prescribed depth D, each sampling point must lie inside a depth D cell.
2. Compute the vector field
3. Compute indicator function
4. Extract iso-surface

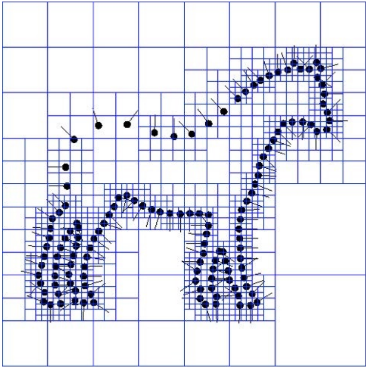
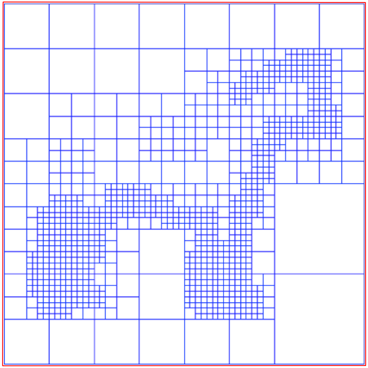


Figure: The Octree $\mathcal{O}$.

Poisson Reconstruction

Given point cloud with normals

1. Set Octree to partition $\mathbb{R}^3$
2. Compute the vector field
    1. Define function space
    2. Splat samples
3. Compute indicator function
4. Extract iso-surface



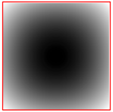


Figure: Base function associated with the root node.

- For every cell $o \in \mathcal{O}$, a basis function is defined as

$$F_o(p) = \frac{1}{o.w^3}\phi\left(\frac{p - o.c}{o.w}\right)$$

- ϕ is a tri-quadratic function approximating a Gaussian with unit variance
- $F_o(p)$ will be used as the filter function too, i.e.

$$\tilde{F}(p - o.c) = F_o(p).$$

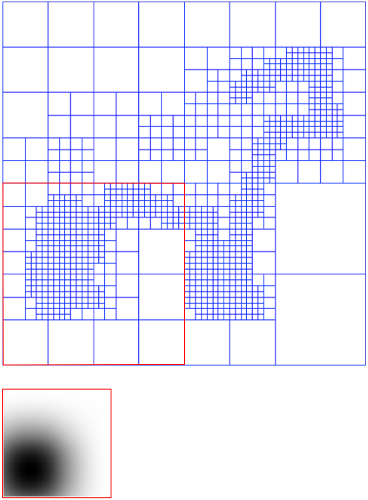


Figure: Base function associated with the current node.

Poisson Reconstruction

Given point cloud with normals

1. Set Octree to partition $\mathbb{R}^3$
2. Compute the vector field
    1. Define function space
    2. Splat samples
3. Compute indicator function
4. Extract iso-surface

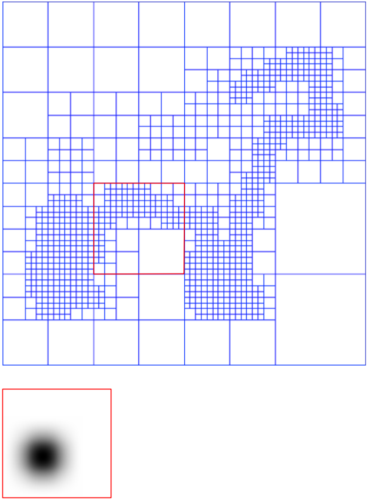


Figure: Base function associated with the current node.

Poisson Reconstruction

Given point cloud with normals

1. Set Octree to partition $\mathbb{R}^3$
2. Compute the vector field
   1. Define function space
   2. Splat samples
3. Compute indicator function
4. Extract iso-surface

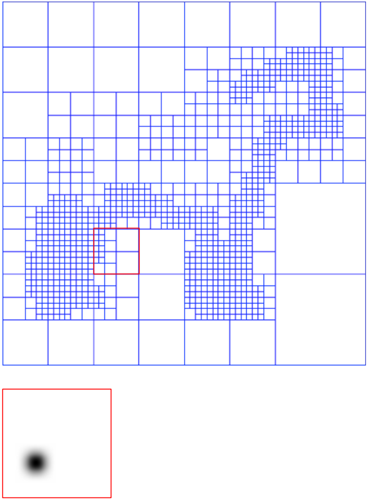


Figure: Base function associated with the current node.

Poisson Reconstruction

Given point cloud with normals

1. Set Octree to partition $\mathbb{R}^3$
2. Compute the vector field
    1. Define function space
    2. Splat samples
3. Compute indicator function
4. Extract iso-surface

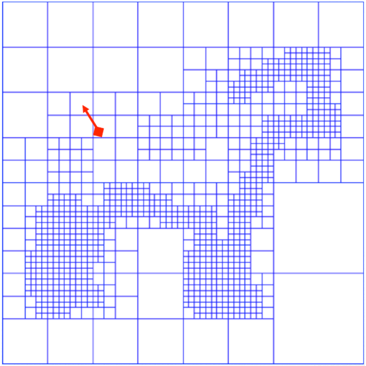


Figure: Splat the samples.

Poisson Reconstruction

Given point cloud with normals

1. Set Octree to partition $\mathbb{R}^3$
2. Compute the vector field
   1. Define function space
   2. Splat samples
3. Compute indicator function
4. Extract iso-surface

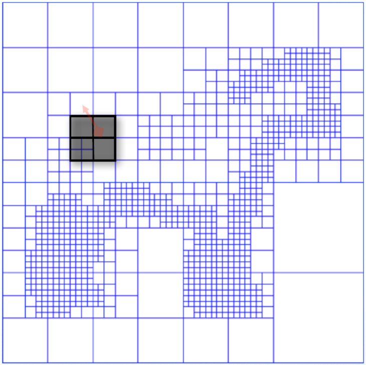


Figure: Splat the samples.

Given point cloud with normals

1. Set Octree to partition $\mathbb{R}^3$
2. Compute the vector field
    1. Define function space
    2. Splat samples
3. Compute indicator function
4. Extract iso-surface

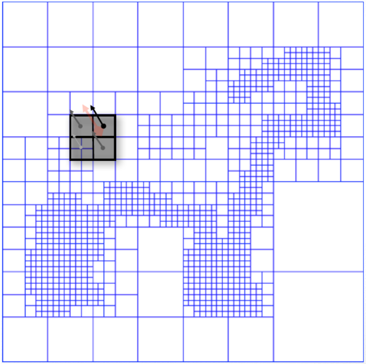


Figure: Splat the samples.

Poisson Reconstruction

Given point cloud with normals

1. Set Octree to partition $\mathbb{R}^3$
2. Compute the vector field
    1. Define function space
    2. Splat samples
3. Compute indicator function
4. Extract iso-surface

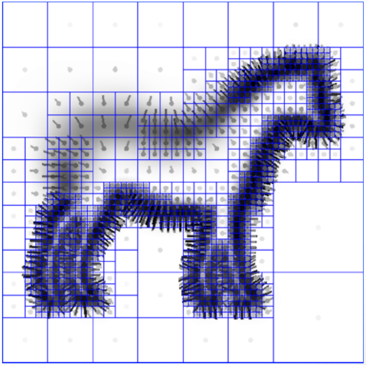


Figure: Splat the samples.

Original Definition:

$$\vec{V}(q_0) = \int_M \tilde{F}(p - q_0)\vec{N}(p)dp$$

Numerical Approximation ($\mathcal{S}$ is the set of sample points):

$$\vec{V}(q_0) = \sum_{s\in\mathcal{S}} \tilde{F}(s - q_0)s.\vec{N}\mathcal{P}_s$$

Ignoring the constant surface area and using trilinear interpolation

$$\vec{V}(q_0) = \sum_{s\in\mathcal{S}} \sum_{o\in\mathcal{N}_D(s)} \alpha_{o,s}F_o(q_0)s.\vec{N}$$

$\tilde{F}(s - q_o) = \sum_{o\in\mathcal{N}_D(s)} \alpha_{o,s}F_o(q_0).$

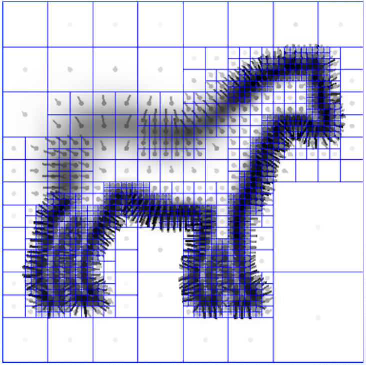


Figure: Splat the samples.

Poisson Reconstruction:Vector field $\vec{V}$

Original Definition:

$$\vec{V}(q_0) = \int_M \tilde{F}(p - q_0)\vec{N}(p)dp$$

Numerical Approximation ($\mathcal{S}$ is the set of sample points):

$$\vec{V}(q_0) = \sum_{s\in\mathcal{S}} \tilde{F}(s - q_0)s.\vec{N}\mathcal{P}_s$$

Ignoring the constant surface area and using trilinear interpolation

$$\vec{V}(q_0) = \sum_{s\in\mathcal{S}} \sum_{o\in\mathcal{N}_D(s)} \alpha_{o,s}F_o(q_0)s.\vec{N}$$

$\tilde{F}(s - q_o) = \sum_{o\in\mathcal{N}_D(s)} \alpha_{o,s}F_o(q_0)$.

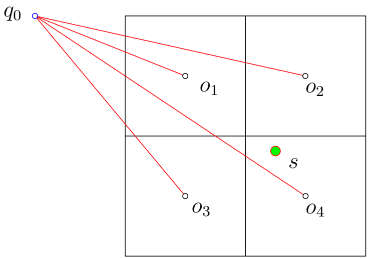


Figure: Trilinear interpolation to obtain all the coefficients $\alpha_{o,s}$'s.

Poisson Reconstruction

Given point cloud with normals

1. Set Octree to partition $\mathbb{R}^3$
2. Compute the vector field
3. Compute indicator function
    1. Compute Divergence
    2. Solve Poisson Equation
4. Extract iso-surface

Figure: Compute the divergence.

The numerical solution is defined as

$$\chi(p) = \sum_{o \in \mathcal{Q}} \chi_o F_o(p)$$

the goal is to find the values of χ_o for every octree cell by minimizing

$$\sum_{o \in \mathcal{O}} |\langle \Delta\chi - \nabla \cdot \vec{V}, F_o \rangle|^2$$

$$= \sum_{o \in \mathcal{O}} |\langle \Delta\chi, F_o \rangle - \langle \nabla \cdot \vec{V}, F_o \rangle|^2$$

Figure: Compute the divergence.

Poisson Reconstruction - Solving Poisson Equation

which reduces to minimizing the following norm:

$$\|L\chi - V\|_2$$

where

$$v_o = \int_M F_o(q)\nabla \cdot \vec{V} dq$$

$$L_{oo'} = \int_M \Delta F_o(q) F_{o'}(q) dq$$

Figure: Compute the divergence.

which reduces to minimizing the following norm:

$$\|L\chi - V\|_2$$

where

$$v_o = \int_M F_o(q)\nabla \cdot \vec{V} dq$$

$$L_{oo'} = \int_M \Delta F_o(q) F_{o'}(q) dq$$

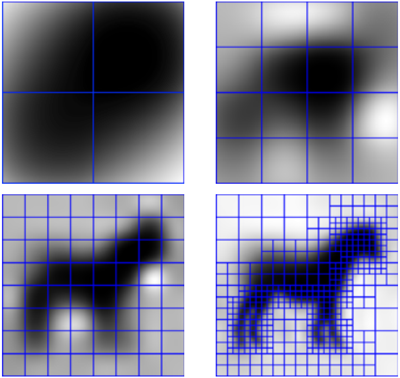


Figure: Multi-grid method.

Poisson Reconstruction

Given point cloud with normals

1. Set Octree to partition $\mathbb{R}^3$
2. Compute the vector field
3. Compute indicator function
    1. Compute Divergence
    2. Solve Poisson Equation
4. Extract iso-surface

Figure: Solve Poisson equation.

Poisson Reconstruction

Given point cloud with normals

1. Set Octree to partition $\mathbb{R}^3$
2. Compute the vector field
3. Compute indicator function
   1. Compute Divergence
   2. Solve Poisson Equation
4. Extract iso-surface

$$\partial \tilde{M} := \{p \in \mathbb{R}^3 : \tilde{\chi}(p) = \gamma\}$$

$$\gamma = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} \tilde{\chi}(s.p)$$

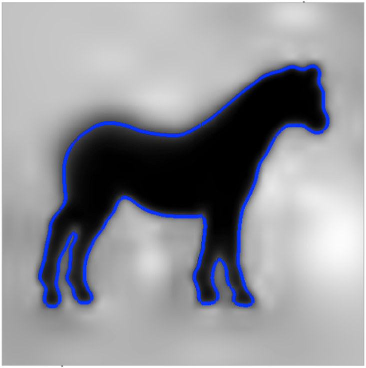


Figure: Extract iso-surface.

Poisson Mesh Reconstruction

Figure: Poisson mesh reconstruction.

Poisson Mesh Reconstruction

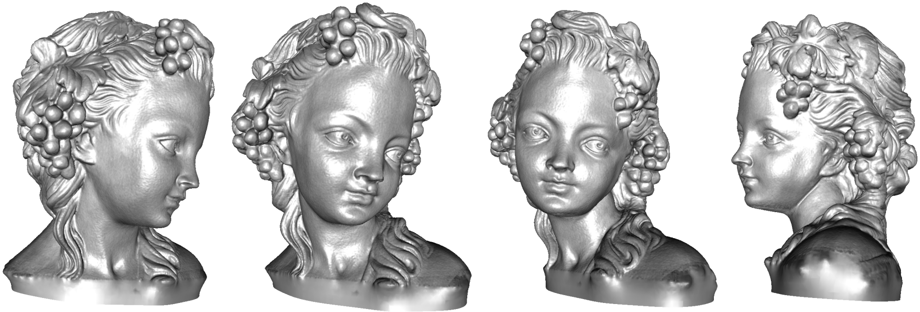


Figure: Poisson mesh reconstruction.