Edge detection

Grad:in **first** ' its the **max** in '' **zero**,Img **not continuous** so fit surf, compute grad for surf.
Description:Normal:direction of the max intensity **direction:**perpendicular to the normal
Strength:speed of intensity variation across the edge (deritetive in normal direction)
Img grad:gradient magnitude unaffected by **orientation,**not directly usable in real-img
I = [∂I / ∂y , ∂I / ∂x] / G = sqrt([∂I / ∂y]^2 + [∂I / ∂x]^2) / ø = arctan[∂I / ∂y , ∂I / ∂x]
Noise: more noise **derivative** less detect edges
Fourier:Differentiating emphasizes high freq therefore noise!(**derivative Magnified Noise,** derivative of F is not recognisable) **-DFT** is discrete equivalent of 2D Fourier

$$F(\mu,\nu) = \frac{1}{\sqrt{M*N}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-2i\pi(\mu x/M+\nu y/N)}$$

$$f(x,y) = \frac{1}{\sqrt{M*N}}\sum_{\mu=0}^{M-1}\sum_{\nu=0}^{N-1} F(\mu,\nu)e^{+2i\pi(\mu x/M+\nu y/N)}$$

2D function f is written assum of sinusoids **-DFT** of f with g is product of g&fDFTs.
e(2πi(ux+vy)) -**Frequency = sqrt(u^2 +v^2)** larger=more lines
- **Removing high freq** for smooth result:
- μ2 + ν2 ≥ T depict high frequencies.

$$f(x,y) = \frac{1}{\sqrt{M*N}}\sum_{\mu^2+\nu^2<T} F(\mu,\nu)e^{+2i\pi(\mu x/M+\nu y/N)}$$ T is a hand-specified threshold.

Convolution: Faster because dg/dx precomputed (Kernel=gaussian ,mask)

$$m**f(x,y) = \sum_{i=0}^{w}\sum_{j=0}^{w} m(i,j)f(x-i,y-j)$$

$$\begin{bmatrix}-1&0&1\\-1&0&1\\-1&0&1\end{bmatrix} \text{ and } \begin{bmatrix}-1&-1&-1\\0&0&0\\1&1&1\end{bmatrix} \qquad \begin{bmatrix}-1&0&1\\-2&0&2\\-1&0&1\end{bmatrix} \text{ and } \begin{bmatrix}-1&-2&-1\\0&0&0\\1&2&1\end{bmatrix}$$

Prewitt operator Sobel operator

Gaussian:Fast:kernel **separable,small / integral=1 ,big sigma: broader,up robustness**,degrade **precision •sigma=0** approximate **Dirac** function **DFT of Gaussian:•DFT** of a Gaussian is a Gaussian.**•finite** support.• Its width is **inversely proportional** to the original Gaussian
Gaussians as Low-Pass Filters: The Fourier transform of a **convolution** is the product of their Fourier transforms: F(g * f) = F(g)F(f) • If **g is a Gaussian**, so is F(g).•if **g is broad**, the **support of F(g)** is small.**•support of F(g*f)**.• no high-frequencies in g * f.
Convolving with a Gaussian suppresses the high frequencies. **separable**(1D cov fast)

$$\frac{\partial}{\partial x}\int_u\int_v g_2(u,v)f(x-u,y-v)dudv = \int_u g_1'(u)\left(\int_v g_1(v)f(x-u,y-v)dv\right)du \qquad g_2(x,y) = \frac{1}{2\pi\sigma^2}\exp(-(x^2+y^2)/2\sigma^2)$$

Continuous Gaussian Derivatives: Img ' computed by convolving with the ' of Gaussian **Canny(Sig,T1,T2):1)Conv** Grad strength/direction ,**2)T 3)Non Maxima Suppression**:Find local max pixel on grad direction,require check interpolated pixels p&r.
4)Hysteresis Thresholding=T: -two T: high & low -pixel with edge strength **above high** T is edge,**below low** T is not,**above** the **low** threshold and **next** to edge is edge.
disadvantages:Choosing the right **scale(sigma)** is a difficult(**small**=more **details**)
advantages:Fast ,Not data training needs

Texture:LIMIT non-local /**•Non** trivial to measure/• Subject to deformations/Hard to characterize
Note:used in conjunction with effective Machine Learning technique.
/**scale** dependent phenomenon /**Assign** pixels whose texture is similar the same values to textural image.

$$f(x,y) = \frac{1}{\sqrt{M*N}}\sum_{\mu=0}^{M-1}\sum_{\nu=0}^{N-1} F(\mu,\nu)e^{+2i\pi(\mu x/M+\nu y/N)}$$

texel represents the smallest graphical element repeat on texture,**Microstructures** define reflectance properties **quantities/statistics** of texture can computed from gray/ colors **statistical** less intuitive, more effective. **Creating Textural: Each pixel**, compute a feature vector using image patch or filters set **Run** classification to assign texture value to each pixel.**Textural Metrics: Spectral metrics:Fourier transform.**Angular and radial bins in the Fourier domain capture the directionality and fluctuation speed of an image texture
LINIT: DFT on **small patches** create bad boundary effects. /**Only applicable** for uniform over large areas/ **can** improved by using wavelets, but only up to a point./ **More** local metrics required.
Statistical Metrics: intensity and color in a region /
First order gray-level statistics: • **Statistics** of single pixels in terms of histograms. • **Insensitive** to neighborhood relationships./**orientation** of the underlying **plane**. **Edge Density and Direction:**• Edge detection is first step .• **number of edge** pixels in a fixed-size region tell how busy that region.• **directions** of the edges also help characterize the texture Edgeness per unit area:• **{ p : gradient_magnitude(p) ≥ threshold}| / N** ,N=unit area or region. **Edge magnitude/direction** histograms: • {HG,Ht} **Second Order Measures:**Histogram of the co-occurrence of particular intensity values in the image.• **Specified** in terms of geometric relationships between pixel pairs:• **Distance• Orientation/ Contrast /Dissimilarity / Homogeneity / Energy /Uniformly /Angular second moment /• P(i,j,d,θ) Frequency/ Entropy** with which a pixel with value j occurs at distance d and orientation θ from a pixel with value i. **Parameters choosing:** • window size,• direction of offset,• offset distance,• what channels to use, • what measures to use. • Can be addressed using Machine Learning.
Filter Based Measures: Represent textures using responses of filters collection. **appropriate** filter bank extract useful information such as spots and edges. **Traditionally** one or two spot filters and several oriented bar filters./ **Gaussian:** respond to horizontal and vertical edges. can be used to compute higher-order image derivatives. **For** every pixel, compute responses vector to each filter /**small** scale =local details/ **large** scale = larger details. **Gabor Filters** : products of a Gaussian filter with **oriented sinusoids**, **pairs** of a **symmetric** filter and an **anti-symmetric** filter/ **filter bank** is formed by varying the frequency, the scale, and the filter orientation $G_{\text{asym}}(x,y) = \sin(k_x x + k_y y)\exp(-\frac{x^2+y^2}{2\sigma^2})$ $G_{\text{sym}}(x,y) = \cos(k_x x + k_y y)\exp(-\frac{x^2+y^2}{2\sigma^2})$
where kx and ky determine the spatial frequency and the orientation of the filter and σ determines the scale. **Perform classification** on output of Gabor: **Decision Forests** ,**ConvNets**(for every pixel a feature vector containing output of intermediate layers.),**Feature Maps**(look very Gabor like,requires a large training), **U-Net**

Shape from shading:

I. Reflectance Maps A. Lambertian Reflectance represent the relationship between intensity and the dot product of the light direction and the normal vectors. **Lambertian Reflectance model,** this relationshiplinear, perfectly diffuse reflecting surface.
II. Variational Methods inverse 3D modeling, including the recovery of surface normals and the 3D surface itself.
A. Inverse Variational omprises data fidelity, smoothness, integrability terms. It recover normals and recover the 3D surface.
B. Direct Recovery direct recover 3D instead first recovering normals. necessitates solving 2-order partial differential equation.
Both need certain boundary conditions to function properly.
III.Variational to Deep: Normal Map Stream allowed generation of depth and normal maps/ data-driven approach provides a means to handle complex behaviors and non-linearities more effectively.
Ambiguities in 3D Modeling 3D modeling, while powerful, comes with its share of challenges and ambiguities.
I. Bas-Relief Ambiguity Transform normals, not affect image , result in normals being non-integrable.good for Monge surfaces,
II.Convex/Concave Ambig : by known light ,**3D shape ambiguous.** convex/concave produce identical imag, ambiguity surface.
Making the Shape-from-Shading (SfS): SfS made well-posed by leveraging perspective projection /radiance models /deep
I. Perspective Projection and Radiance
By using a perspective projection model and considering the **distance to the light** , the ambiguities in the SfS problem can be **mitigated.** makes the problem well-posed, albeit more computationally complex .also makes assumption of constant albedo.
II. Leveraging Deep Learning potential to overcome the limitations posed by the variational approach and the assumptions about albedo. This technology uses understand and learn from the data, offering a more robust solution to the SfS problem.

Delineation: h(x1,x2..) = -∑g(x_k) + ∑r(x_k,x_k+1) / r(x_k, x+k+1) = diff(ø(x_k), ø(x_k+1))
Live Wire:1)define start/end points / Dynamic programing(N^2) / **1-D:1)** Find min h(x1,x2..) where : h(x1,x2..) = r(s,x1) + ∑(r(xi) +r(xn+g)
2) f_1(x2)=min(r(s,x1)+r(x1,x2), f_2(x3)=min(r(x2,x3)+f(x2) **3) min h(x1,x2…) = min (r(xn+g) + fn-1(xn)**
2-D: s(start point) , L (List active nodes) ,C(u,v) (local cost u->v) ,d(v) (Total cost s->v): **1)** Init: d(s)<-0 and d(u) <- inf, for u≠s , T=ø ,v=s
2) Loop: T<- TU{v} ,for all v->u edges such u not in T ,if d(v)+c(u,v) < d(u) then(d(u) <- d(u) + c(u,v) end) end) ,v= argmin_w_notinT(d(w)
improve Live Wire:sort ->O(n(log(n)) **-Local** cost c(uv) = 255 -1/2(g(u) + g(v)) **-Add** const cost each edge **-Mult** non diagonal edges *5/7
-Can be integrate for user correct mistakes ,**Limits: optimal** path not the best, **hard** impose global constraints ,**cost** grows exp with dim
Must often look for local as apposed to global optimum using gradient decent tools
Snake:Max the grad along curve, **Min** Energy ->**can generalizes** to handle sophisticated models/ **Embed curve in viscous mid to solve**

Energy: Weighting coefficient, Average gradient, Average sum of squared 2nd derivatives

$$E = -\frac{\lambda}{N+1}\sum_{i=0}^{N} G(x_i,y_i) + \frac{1}{N}\sum_{i=1}^{N}\left((2x_i - x_{i-1} - x_{i+1})^2 + (2y_i - y_{i-1} - y_{i+1})^2\right)$$

(x_{i-1}, y_{i-1}) (x_i, y_i)

$$E = E_G + 1/2X^tKX + 1/2Y^tKY \qquad X = [x_1,\dots,x_N]^t \qquad Y = [y_1,\dots,y_N]^t$$

$$K = \begin{bmatrix} . & . & . & . & . & . & . \\ . & . & 1 & -4 & 6 & -4 & 1 \\ . & . & . & 1 & -4 & 6 & -4 \\ . & . & . & . & 1 & -4 & 6 \end{bmatrix}$$

Ribbon Snake: the wipe don't change quickly $E = E_G + 1/2X^tKX + 1/2Y^tKY + 1/2W^tK_WW$
Level Set:define T, z=f(x,y) in<Z<out ,**Consider** the curve as the zero level set of a surface. **β(κ)** is speed at surface deforms
0=f(x,y) +β(κ)|f'| properties:Converges towards circles. **Total** curvature decreases. **Number** of curvature extrema and zeros of curvature decreases. **Relationship with Gaussian smoothing: Analogous** to Gaussian smoothing of boundary over the short run, but does not cause self-intersections or overemphasize elongated parts. **Can** be implemented by Gaussian smoothing the function of a region.
Hugo: Input: Canny edge points.Gradient magnitude and orientation. **Given** a parametric model of a curve: **Map** each contour point onto the set of parameter values for which the curves passes through it. **Find** the intersection for all parameter sets thus mapped.
Algorithm: Canny ,R-Table ,possible reference points initialized to zero.**For each** edge point,Compute possible centers: for each table entry, compute x=x_e + r*cos(ø(ß)) and Increment the accumulator array / **R-table**:Set of potential displacement vectors r,a given the boundary orientation f. **Notes: can** perform least squares / **Instead** of indexing displacements by gradient orientation, index by **"visual codeword" / Limitations: Computational** cost grows exponentially with parameters number / works Only on **shape** that can defined by small parameter number / **Approach** is robust but lacks flexibility.
Graph: Generate min distance graph **O(N log(N),works** in different situations, **Node**: Pixel/ **Edges** between to pixels /**Min Span Tree**
Deep: U-Net/ AlexNet / Use **same network** to progressively refine the results keeping the number of parameters constant
Delineation Steps 1.Compute a probability map.2. Sample and connect the samples.3. Assign weight to paths.4. Retain the best paths.
Human annotations Correcting :jointly train the network and adjust the annotations while preserving their topology.
Short: Edge is **noisy** / combination of **graph- based techniques**, **machine learning**, and **semi-automated tools is required**.

Segmentation :Same statistical properties -> same object / Help recognition ,tracking ,compression
Hard:basic image operations not suffice/ basic image operations do not suffice /not a Single Answer
Homogeneity:Uniform gray-level/RGB/TEX/motion/depth statistics.•parametric surface can fitted.
Region growing:Algorithm:Pre: Labeled/ Unabled/ T , Given a set of regions A1, ... , An,
let T = {x ∉ (∪ni=1Ai), N(x) ∩ (∪ni=1Ai) ≠ ∅}, the set of unlabeled pixels that are neighbors of already labeled ones ,d be a metric, such as δ(x) = |g(x) − meany∈Ai(x)[g(y)]|.Until all pixel labeled:
1. Represent T as a sorted list, the SSL **2**. Label the first point in T.**3.** Add the neighbors to the SSL.
Limit: **1.**Result depends in **which pixels** taken **2.**hmogeneity is **noise** sensitive
Histogram:similar pixels appear as bumps/ Split histogram at local-min / Label pixels according which bump
Recursive Splitting: Compute histogram/Smooth histogram(less peaks)/find peaks separated by deep valleys/ Group pixels into connected regions/Smooth these regions/Iterate(until no clear mins) **Limit: T** hard to choose **T Not** always can be find in the histogram
Local Histograms:•Compute local histograms on a coarse grid. •Use them instead of a global one.
limit:not **neighborhood relation**./ **T=?**./ gray-l both side belong to same peak so **boundaries not found**
Saturation His: RGB,HSV, CIE LAB Color Space: three coordinates, one "brightness" two chromaticity.
Alg: Compute multiple his for each segment. • Use one to split each segment. • Repeat on result smaller segs.
Segmentation <- Clustering:Each pixel has 2 spatial coordinates and 1 gray level or 3 color components.
- 1D space (G);• 3D space (x,y,G), (H,S,V), (L,a,b);• 5D space (x,y,L,a,b).• each cluster compact as possible.

Given a set of input samples:**•Group** the samples into K clusters.**•K** is assumed to be known/given.
any cluster has centroid(μk)=mean of points in that cluster. **min** sum squared distances to each point to centroid.
ALG: Randomly initialize centroids.**Repeat until convergence:Assign** points to nearest. **Update** the centroids.
Note: Works in any dimensional space. **Sensitive** to initial conditions; multiple initializations are advised.
Application in Image Processing: GRAY-LEVEL ONLY (1D): **Clustering** based on pixel intensity.
Color Only (3D): **Clustering** based on pixel colors/ XY + Color (5D): **Spatial** and color information combined to form superpixels(.**Superpixels:** Homogeneous segments of an image. /**Useful** for reducing img complexity)
assigned probability: statistics each superpixel.**Train** classifier to determine probability of superpixel
Use probabilities to generate segmentations using graph-based techniques or other methods.
Graph: Images -> graphs (G(V,E)) with pixels = vertices (V) / edges (E) = weighted based pixel similarity.
Algorithm: **Graph** Representation
Parse img into a pixel value matrix. **Treat** pixel as a node in a graph. **Draw** edges between adjacent nodes.
Graph-Cut: total edge weight needed to cut graph into two parts.**Min-Cut:** cut of minimum weight
-**Algorithm: Min-Cut/Max-Flow: Initialize** flow to 0. **Select** two regions in the img define: source (S) & sink (T).
Treat each pixel as node, and connect them to S and T. **Find** a path from the source to the sink.
Add the path flow to the total flow. **Repeat** until no path from the source to the sink remains in the residual graph.
-**Note The** set of edges whose removal disconnects the source from the sink constitutes the Min-Cut.
This segmentation divides the image into two distinct regions based on pixel similarity.
-Trivial Cut: This approach favors shorter cuts, potentially leading to bad solutions. must therefore be controlled.
-Interactive ST Min-Cut: guessing initial linkages between S and T.

$$E(y|x,\lambda) = \sum_i \underbrace{\psi(y_i|x_i)}_{\text{unary term}} + \lambda \sum_{(i,j)\in\mathcal{E}} \underbrace{\phi(y_i,y_j|x_i,x_j)}_{\text{pairwise term}}$$

-Object Classification and Loss Function Minimization
given set of supervoxels [y1,...,yn], E= negative log-likelihood of the labels being correct.
Supervoxel Classification:strong edges from supervoxels to: source high probabilities / weak edges low prob
Classifier Training: Develop classifiers to weight edges according to supervoxel class similarities.
Loss Function Minimization: Reduce negative log-likelihood for correct labels in a supervoxel set.
Tri-class Modeling: Achieve global optimum by categorizing objects as 'Inside', 'Boundary', or 'Everything else'.
Graph-Cut on Standard Images: Utilize 'Graph-Cut' for insights in regular images.
Interactive Foreground Extraction: Use K-means for color distributions and Graph cuts for iterative segmentation of foreground and background.

I. Photometric Stereo:technique utilized for estimating the shape of a surface by observing it under known lighting conditions. **II. Influence of Light Source** Quantity in Photometric Stereo
III count of light sources significantly affects the process of surface shape determination:
A. Single Light Source many potential normals for each image point, leading to ambiguities.
B. Two Light Sources: presence may decrease ,**not** completely resolve ambiguities.
C. Three Light Sources: good eliminate ambiguities, even when the albedo (reflectivity) unknown.
III. Mathematical Formulation in Photometric Stereo Lambertian model: represents intensity (I) as the product of albedo (α) and the dot product of light source direction (L) and normal vectors (N).
3-vector(M) estimated by solving a 3x3 linear system, transform this to over-constraint problem,
adding more light sources can make the solution more robust against noise.
IV. Considerations for Shadows/Specularities: A. Handling Shadows ,Shadowed pixels for a specific light bad for Lambertian model and reducing contributions because lower intensities.
B. Accommodating Specularities Specularities,mirror-like, reflectance map accommodating specularities more complex due to incorporation of both diffuse(scattered) &specular(mirror-like).
V. Inferring Shape from Specularities utilized to infer surface normals. effective for highly specular or shiny objects ,specularities do not fully constrain all the degrees of freedom.can used to additional info/alternatively remove noise.
assumptions: constant albedo/ absence of interreflections/ shadows/ specularities/ In single image shape-from-shading most pronounced when it's used alongside other info sources.
VII. Key Angles in Photometric Stereo
Angle of Incidence (i): angle between the surface normal and the direction of the incident light ray.
Angle of Emittance (e): angle between the emitted light ray and the surface normal.
Phase Angle (g): Denotes the angle between the incident and emitted light rays.
More than three light increasing robustness against noise,each light providing further info solution
IX. Lambertian + Specular Reflectance Map: surfaces that exhibit both Lambertian and specular reflectance map is combination of Lambertian and specular, representing a complex scenario.
S-Specular if object has glossy finish/mirrored surface, specularities give infer normals help info.

Shape from Shading Steps: Image: Use an image of a scene with known lighting conditions. **Model:** Assume a reflectance model for the objects in the scene.**Estimate:** Derive depth and surface orientation information from the intensity of the pixels.
Reflectance Map: **Create**: Generate a map relating surface orientations to intensities.
Use: Use map to estimate orientations in the scene.
Specular Model: Assumption: Surfaces reflect light in mirror-like manner. **Challenge:** Hard to estimate surface information due to high variability.
Lambertian Model:Assumption: Surfaces scatter light uniformly in all directions.
Benefit: Simplifies the estimation of surface information.
Body Reflection: Concept: Reflection depends on surface orientation and viewing angle. **Challenge:** Requires complex modelling for accurate surface estimation.
Variational Method: Approach: Optimizes a cost function relating intensity and surface orientation. **Benefit:** Can handle more complex scenarios and lighting conditions.**Advantages** of Shape from Shading: **Single Image:** Can infer 3D information from a single image. **Detail**: Can capture fine details not visible in stereo or motion-based methods. **Disadvantages** of Shape from Shading:Assumptions: **Makes** strong assumptions about light and surface properties. **Illumination:** Sensitive to changes in illumination and reflectance properties.

$$\int\int \left([I(u,v) - Ref(p,q)]^2 + \lambda\left[\left(\frac{\delta p}{\delta u}\right)^2 + \left(\frac{\delta p}{\delta v}\right)^2 + \left(\frac{\delta q}{\delta u}\right)^2 + \left(\frac{\delta q}{\delta v}\right)^2\right] + \mu\left(\frac{\delta p}{\delta v} - \frac{\delta q}{\delta u}\right)^2\right) \epsilon$$

Data term Smoothness term Integrability term

Shape from Texture
Shape from X: Deals with surface orientation or shape recovery from image texture, based on texture deformation due to surface curvature.
Structural Shape Recovery: Assumes texture resides on surface with no thickness,computation Perspective, Paraperspective, Orthographic projections.
Perspective Projection: Involves pinhole geometry without image, distortion is anisotropic, depends on depth and surface orientation.
Foreshortening: Depth versus orientation principle, with object scaling and foreshortening as special cases.
Orthographic Projection: Includes Tilt and Slant derived from image direction and compression extent.
Paraperspective Projection: A generalization of orthographic projection involving parallel projection followed by scaling.
Texture Gradient: Central to statistical shape recovery process.
Statistical Shape Recovery: Measures texture density or the number of textural primitives per unit surface, assumes homogeneous texture.
Machine Learning: Involves training a regressor for depth prediction, which may result in noisy predictions.
Markov Random Field (MRF): Graph-based technique to enforce consistency.
Deep Learning with MRF: Combines deep learning with MRF for consistency.
Enforcing Task Consistency: Training networks to predict multiple things to increase robustness.
Diverse Training Database: Very diverse training databases improve results.
Transformer Architecture: Good at modeling long-range relationships, but flattening patches loses some information.
Strengths and Limitations: Emulates an important human ability with limitations including regular texture requirement, strong assumptions, though deep learning can help weaken them.

Ambiguities: Repetitive patterns/ textureless/ occlusions cause problems.
Occlusions: Some points are only visible in one image. They not matched.
Ignoring Occluded Pixels: pixels haven't corresponding pixel in other img Left right consistency test.**Disparity Map**: Black pixels indicate no disparity.
Combine Disparity Maps: Merge disparity maps and smoothing result map.
Solving Variational Problem: Discretize integral and solve linear problem.
Real-T Implementation: duplicated computations can implemented (faster)
window size: Speed is independent to w.s /**Small windows:**•Good precision Sensitive to noise /**Large windows:**• Diminished precision• Increased noise
Replace Normalized Cross Correlation by Siamese nets designed to return a similarity score for potentially matching patches.
Comparative Results: Improved performance on test data but how it generalizes to unseen imgs,its necessity for this application(not clear yet)
3D Point Cloud: Disparity map transform each triplet (u,v,d) into 3D point (x,y,z), resulting in a 3D point cloud.
Merge the 3D point clouds: dense models when have enough imgs.
Scale-Space Revisited: Gaussian pyramid, Difference of Gaussians.
Fronto-Parallel Assumption:disparity assumed be same over correlation window, equivalent to constant depth.**Multi-view reconstruction setup**: Adjust correlation window shapes to handle orientation.
Scene Flow: Correspondences across cameras and across time.
Refining using Shape From Shading: Shape-from-shading can used to refine shape and provide high-frequency details.
Precision vs Baseline: Beyond a certain depth stereo stops being useful. Precision is proportional to baseline length.
Short baseline: good matches /few occlusions /poor precision
Long baseline: harder match /more occlusions /better precision.
Multi-Camera: **3 cam:** robustness/precision.**4 cam:** more redundancy.
Structured Light:IR pattern and measures depth from its distortion.
Faces LowResolution Videos:No calibration data/little texture/difficult light.
Model Based Bundle Adjustment: Adjusting PCA to minimize objective func yield accurate face reconstruction from lowResolution images(also **Old Movie)**
Energy Minimization: Minimize the disparity continuity using graph-based seg.
Graph Cut for Stereo: Stereo is a labeling problem, use graph cut.
Assigning Edge Weights: Assign a weight that is proportional |I2(x+1,y)-I1(x,y)|
Minimizing the Objective Function: Graph cut algorithm, Guarantees absolute minimum only when there are only two possible disparities.
α-Expansion: Nodes having label different than α can either keep or switch to α.
Graph Cut Algorithm: Start with an arbitrary labeling, Find the α-Expansion that minimizes the function, Update graph by adding ,erasing edges, Quit when no expansion improves the cost, Induce pixel labels.
NCC vs Graph-Cut: Normalized correlation Graph Cut.
NCC vs Graph Cut: left image true disparities, Normalized correlation Graph Cut
NCC vs Graph Cut - Strengths and Limitations: method for depth recovery (real-time ,Requires multiple views, Only to reasonably textured objects).

Shape from Motion Steps:
- **Capture**: Take multiple images of a scene from different viewpoints.
- **Detect Features**: Identify key features in each image.
- **Match Features**: Match features across images to track movement.
- **Estimate Structure**: Use matched features and camera movement to estimate the 3D

1. **Advantages:Flexible**: Works with unstructured image sets.
   - **Automatic**: Estimates camera parameters autonomously.
   - **Cost-Effective**: Cheaper 3D mapping option.
2. **Disadvs: Dependent**:Quality relies on identifiable image features.
   - **Scale Ambiguity**: Absolute scale usually unknown.
3. **Additional Points:**
   - **Photogrammetry Type**: Constructs 3D from 2D images.
   - **Movement**: Uses object or/and camera motion.
   - **Applications**: Useful in archaeology, architecture, robotics.
   - **Bundle Adjustment**: Optimization method for refining 3D positions.

3D Deformable Model Steps:
- **Initialize**: Use a visual hull.
- **Maximize**: Aim for color and silhouette consistency, and smoothness.

Advantages:
- **Adaptable**: Fits various shapes.
- **Detailed**: Captures intricate forms.
- **Consistency**: Offers color and silhouette uniformity.

Disadvantages:
- **Resource-Intensive**: Requires more computational
- **Initialization Dependency**: Needs a good initial esti for proper convergence.

Additional Points:
- **Physics-Based**: Often used for realistic deformation
- **Wide Use**: Applied in medical imaging, graphics, ma
- **Smoothness**: Helps create realistic models.
- **Control Points**: Direct deformation.
- **Constraint Challenges**: Like object volume preserv
- **Interactivity**: Allows user manipulation.
- **Iterative Refinement**: Model refined over iterations

Depth Prediction:
- **Capture: Get two images, stereo cameras. (Known: camera parameters)**
Match: Identify same features in both images (Known: image features)
Calculate Disparity: Positional difference for features. (Known: feature positions)
Compute Depth: Disparity and camera parameters gives depth.
Advantages:
3D Perception: Understand depth in scenes
Matching Difficulty: Identifying same feature across images is challenging.
Occlusions:Some areas appear in only one

1. **Triangulation**: A process of determining the location of a point by measuring angles to it from known points at either end of a fixed baseline.
2. **Geometric Stereo**: A technique used in computer vision to estimate depth information from two or more images taken from different points of view.
3. **Epipolar Geometry**: Refers to the geometric constraints between two views of a 3D scene. It defines the term epipolar line, which is the line on which the corresponding point must lie, and the epipole, which is the intersection of line joining optical centers and image plane.
4. **Calibration Grid**: A tool used to infer projection matrices and compute epipolar lines, often taking pictures of a grid with each camera to achieve this.
5. **Rectification**: A process that involves reprojecting images onto a common image plane, typically used to simplify the process of finding matching points between images.
6. **Disparity**:**Disparity refers to the difference in the horizontal position of a point in two images, i.e., the left image and the right image. This difference is due to the slightly different viewing angles of the two cameras.** Disparity proportional to depth →
   - It can be represented mathematically as d = x - x', where x and x' are the locations of the same point in the left and right images, respectively.
7. **Depth**: **Depth, or depth map, is the distance from the camera to each pixel (point) in the image. It gives a sense of the third dimension (z-direction) in 2D images. In other words, depth provides information about how far an object is from the viewer or the camera.**
   - The depth of a point can be calculated from its disparity, given the baseline distance (B) between the two cameras and the focal length (f) of the camera. The formula is Z = (B * f) / d.
8. **Correlation and Normalized Cross Correlation**: Measure of similarity between two signals, in this case, image patterns. Normalized cross correlation involves normalization to make the measure more robust to variations in intensity.
9. **Occlusions**: Areas in images that become hidden in another view. These regions pose a challenge as they cannot be matched in stereo vision.
10. **Window-Based Approach to Establishing Correspondences**: Involves comparing pixel windows between two images and selecting the best matching window based on a cost function.
11. **Variational Approach**: This is used for solving an optimization problem where the aim is to find the disparity map that minimizes a certain energy function.
12. **3D Point Cloud**: Collection of data points in 3D space, which are typically produced by stereo imaging or lidar.
13. **Multi-view Reconstruction**: An approach that uses multiple images of a scene to reconstruct 3D information. It adjusts correlation window shapes to handle orientation.
14. **Short vs Long Baseline**: Short baselines offer good matches with few occlusions but poor precision, whereas long baselines are harder to match with more occlusions but offer better precision.
15. **Kinect: Structured Light**: A technique used by Kinect to determine depth information. It projects an infrared pattern and measures depth based on its distortion.
16. **Energy Minimization**: Method used to find the disparity map that results in the best match between images while maintaining smoothness. It often involves Graph Cut or other optimization techniques.
17. **Face Reconstruction**: Techniques used to construct a 3D model of a face from images, typically involving fitting a model to the image data.

Sum of Squared Differences (SSD):
1/Calculate SSD: SSD = Σ (I1(x) - I2(x))^2. //2 Used for similarity measure in image//3 Lower SSD = Higher similarity. // 4 Minimize SSD over local window for disparity.
Knowns:Two images: I1, I2. SSD formula. // **Unknowns**: Pixel location similarity. //**Goal**: Find image similarity via SSD. Minimize SSD locally for disparity map.

Fundamental Matrix F

Epipolar constraint: $\begin{bmatrix} x_l & y_l & z_l \end{bmatrix} \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = 0$

Rewriting in terms of image coordinates:

$[u_l \; v_l \; 1] \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = $

Fundamental Matrix F

$\boxed{E = K_l^T F K_r}$ $\boxed{E = T_\times R}$

Projection Matrix P

Camera to Pixel

$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$

$\tilde{u} = M_{int} \tilde{x}_c$

World to Camera

$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

$\tilde{x}_c = M_{ext} \tilde{x}_w$

Combining the above two equations, we get the full projection matrix P: $\boxed{\tilde{u} = M_{int} M_{ext} \tilde{x}_w = P \tilde{x}_w}$

Left Camera

1.Assume Camera Matrix K is known for each cam
2 Find a few Reliable Corresponding Points
3.Find Relative Camera Position t and Orientation
4. Find Dense Correspondence
5. Compute Depth using Triangulation

Repetitive patterns, textureless occlusions can cause problems

$X' = RX + T$ so $X' \cdot (T \times RX) = 0$.
Essential Matrix: $E = T_\times R$, rank=2, free=5.
Epipolar constrain: $p'^T E p = 0$ or $X'^T E X = 0$.
Fundamental Matrix: $F = K^{-T} E K^{-1}$, rank=2, free=7 and we have $\bar{p}'^T F \bar{p} = 0$.
Alg. for finding F: collect 8+ constrains, normalize, solve using SVD, rearrange and find a rank 2 approximation. A constrain is:
$$[u' \;\; v' \;\; 1] F [u \;\; v \;\; 1]^T = 0$$