Introduction to 3D modeling and image acquisition

Augusto Sarti, Stefano Tubaro
Marco Marcon, Fabio Antonacci

Dipartimento di Elettronica,
Informazione e Bioingegneria
Politecnico di Milano

Image and Sound Processing Lab
Contents

A brief introduction to 3D data acquisition techniques
- Range Acquisition Taxonomy
- Optical range acquisition: stereo vision
- 3D scanning methods based on Active Triangulation

Photo-cameras and Pin-hole cameras
- Camera Model
- Solid State Sensors

Image Features in Computer Vision
- Line Features: Edges
- Point Features: Corners
- Video Sequence Analysis: Corner Tracking
Typical Data Types in 3D scanning

- Volumetric Data
  - Voxel grids
    - Occupancy
    - Density

- Surface Data
  - Triangle meshes (or Bézier curves, B-splines, NURBS surfaces) obtained from
    - 3D Point clouds
    - Range images (Depth maps)
  - In order to obtain a complete description of the considered scene it is often necessary to assemble several partial surfaces:
    - Alignment, registration
    - Global surface reconstruction, 3D scan merging
3D Scanning: Application Areas (1)

- **Medical applications**
  - Analysis of tissues and organs
    - 3D maps of: density, concentration of particular elements, …
  - Very important in modern medicine

- **Metrology**
  - Main goal: accuracy
    - Error minimization is more important than scanning speed and surface completeness
  - Applications
    - Industrial inspection
    - Quality control
    - Reverse engineering (from a real object to the CAD model)
3D Scanning : Application Areas (2)

- **Computer Graphics**
  - A complete model is often required
  - High quality images (low noise)
  - Geometric consistency is often more important than accuracy
  - Applications
    - automatic generation of virtual scenes/objects
    - animated CG characters

- **Robotics**
  - Action and motion planning, navigation
  - Scene understanding

- **Cultural heritage**
  - Automatic 3D relief
  - Monitoring of structural characteristics (deformation)
Range Acquisition Taxonomy

- **Contact**
  - Mechanical (CMM, jointed arm)
  - Inertial (gyroscope, accelerometer)
  - Ultrasonic trackers
  - Magnetic trackers

- **Transmissive**
  - Industrial CT
  - Ultrasound
  - MRI (Magnetic Resonance Imaging)

- **Reflective**
  - Non-optical
    - Radar
    - Sonar
  - Optical

**Range acquisition**
CMM (Contact Measurement Machine)

- very accurate (µm)
- very expensive
- requires the presence of an expert operator
Range Acquisition Taxonomy

Optical methods

Passive

Active

Shape from:
- stereo
- motion
- shading
- texture
- focus/defocus

Active variants of passive methods
- Stereo with projected texture
- Active depth from defocus
- Photometric stereo

Time of flight

Triangulation
Optical Range Scanning Methods

- **Advantages:**
  - Less invasive (no contact involved)
  - Safer (keep the distance)
  - Usually less expensive
  - Usually faster

- **Disadvantages:**
  - Sensitive to transparency
  - Confused by specularity and interreflection
  - Texture is an issue (helps some methods, hurts others)
Stereo (1)

- Use two cameras to view the object of interest from two different points of view
- Select a “feature” point in one image
- Search for a corresponding point in the other image along the epipolar line
- Use geometric triangulation to determine the 3D location of the selected feature
Stereo (2)

- Advantages:
  - Passive
  - Inexpensive hardware (2 cameras)
  - Easy to accommodate motion
  - Intuitive analogy with human vision

- Disadvantages:
  - 3D reconstruction is limited to matched image “features”
  - Sparse, relatively noisy data (finding correspondences is hard)
  - Poor reconstruction near object silhouettes
  - Confused by non-matte surfaces

- Multi-baseline stereo helps reduce the occurrence of ambiguities
Can be seen as a special case of multi-baseline stereo

Instead of feature matching, we have feature tracking in video sequences

For $n$ frames and $f$ features, have a data-set of $2 \cdot n \cdot f$ measurements (image coordinates), and $6 \cdot n + 3 \cdot f$ unknowns
Shape from Motion (2)

- Advantages:
  - Feature tracking in video sequences is easier than feature matching in sets of views with significant baseline
  - Temporal continuity in the camera parameters may be exploited for robust estimation
  - Differential Geometry results can be exploited

- Disadvantages:
  - Hard to accommodate object motion
  - Still problems in poorly textured areas, in non-matte regions, and near silhouettes
Shape from Shading (1)

- Given a view of a surface with known, constant reflectance, under known lighting conditions
- Estimate surface normals and integrate to find surfaces
- Problem: ambiguity
Shape from Shading (2)

- **Advantages:**
  - Works with a single image
  - No correspondence estimation
  - Similar mechanisms in human vision

- **Disadvantages:**
  - Mathematically unstable (ill conditioned)
  - No texturing allowed

- **Not very practical**
  - But see photometric stereo…
Shape from Focus and Defocus

- **Shape from focusing**: at which focus setting is a given image region the sharpest?
- **Shape from defocusing**: how out-of-focus is each image region?
  - Passive versions rarely used
  - Active depth from defocusing can be made of practical utility
Active Optical Methods

Advantages:
- Usually generate dense data
- Usually much more robust and accurate than passive techniques

Disadvantages:
- Require the introduction of artificial light into the scene (invasive, distracting, etc.)
- No equivalents in human vision
Active Variants of Passive Techniques

- Regular stereo with projected texture
  - Provides features for correspondence
- Active depth from defocusing
  - Knowing the pattern helps estimate defocusing
- Photometric stereo
  - Shape from shading with multiple known lights
Pulsed Light Beam: Time of Flight (1)

- Basic idea: send out a light pulse (usually laser), time how long it takes to return
  \[ r = \frac{1}{2} c \Delta t \]

- Amplitude Modulated light beam can be used: time delay becomes phase delay (possible ambiguities in phase detection)
Pulsed Light Beam: Time of Flight (2)

- **Advantages:**
  - Large working volume (up to 100 m.)

- **Disadvantages:**
  - Modest accuracy (at best ~5 mm.)
    - Requires timing as low as ~30 picoseconds
    - Does not scale down with working volume
  - Often used for scanning buildings, rooms, archeological sites, etc.
Zcam: a more complex system based on ToF analysis

- The system illuminates the scene with pulsed infrared light.
- A two-dimensional sensor array is used to estimate the scene depth (25 times per second)
3D scanner: Active Triangulation (1)
3D scanner: Active Triangulation (2)

To analyze significant portion of the object under analysis:
- Use a moving mirror to move the laser spot (flying spot)
- Project a light stripe
3D scanner: Active Triangulation (3)

In order to scan a whole object, illuminator and camera are rigidly moved.
Active Triangulation Performances

- Accuracy proportional to working volume (typically >1:1000)
- Scales down to small working vol. (e.g. 5 cm. working volume, 50 µm. accuracy)
- Does not scale up (baseline too large…)
- Two-line-of-sight problem (shadowing from either camera or laser)
- Triangulation angle: non-uniform resolution if too small, shadowing if too big (useful range: $15^\circ$-$30^\circ$)
Active Triangulation Scanner Issues

- Material properties (dark, specular)
- Subsurface scattering
- Laser speckle
- Edge curl
- Texture embossing
Multi-Stripe Triangulation

- To go faster, project multiple stripes
- But which stripe is which?
- Answer #1: assume surface continuity
Multi-Stripe Triangulation

- To go faster, project multiple stripes
- But which stripe is which?
- Answer #2: colored stripes (or dots)
Multi-Stripe Triangulation

- To go faster, project multiple stripes
- But which stripe is which?
- Answer #3: time-coded stripes
Time-Coded Light Patterns

- Assign to each stripe a unique illumination code over time (Binary Gray code)
- Allows the use of a multi-resolution approach in object 3D reconstruction
Time-Coded Light Patterns

An example

A brief introduction to image acquisition
(visible light)
The photographic camera

Image formation on the back-plate of a photographic camera
Pin-hole Camera, the Perspective Projection

The pinhole imaging model

Pinhole (Optical Center)

image plane

d
d
virtual image
We model the optical system as a simple thin lens with an aperture (diameter) $\Phi=d=2a$. Moreover we consider ideal lens, whose performance is only limited by diffraction effects (Fourier optics, non-coherent illumination). This type of optical system is normally indicated as a “diffraction-limited” system.
Real lenses: realization

Aberrations can be minimized by aligning several simple lenses with well-chosen shapes and refraction indices, separated by appropriate stops.
Solid State Sensor: the basic element

Si dioxide

+V

potential well

Charge packet generated by photoelectric effect

p-Si

Incoming light

Wave Length (nM)

Relative Response
Image sensors

Bi-dimensional array of elementary cells:

- Frame-Transfer CCD, Interline-Transfer CCD
- CMOS sensors
Image Acquisition Modeling

- $I_g(x,y,t)$: Irradiance (Intensity) field on the image plane (geometrical optics)
- $h_o(x,y)$: Takes into account diffraction effects due to the limited dimension of the camera lenses
- $h_s(x,y)$: Takes into account of the finite dimension of sensor cell
- $L(n,m,t)$: Acquired image

Linear systems

Ideal bidimensional sampling
Color acquisition: 3 sensors approach
Color acquisition: single sensor

The sensors pixels are covered by properly arranged colored filters (e.g. Bayer pattern)
Image Analysis: Feature Extraction
Image Features in Computer Vision

We search for entities, on the available images, that:

- Are not so difficult to detect/localize from the analysis of the image luminance profiles
- Have some “meaning” in the definition/interpretation of the imaged 3D scene
- Are relative, if possible, to 3D elements/structures, that are significantly present on the available images
Edges/Contours

Lines, curves along which abrupt luminance variations occur

3D scene

Captured Color Image

Texture Edge

Occlusion Edge (Horizon)

Sharp Edge

Camera

BW Image
Some of the possible 1-D Edge Profiles

Step Edge
2D Edges

- Transition Profile
- 2D Edge Shape
1D Edge: Signal and Derivatives Profiles

Without Noise

\[ f(x) \]
\[ f'(x) \]
\[ f''(x) \]

With Noise

\[ f(x) \]
\[ f'(x) \]
\[ f''(x) \]
Edge detection: Gradient based Method (1)

1D scheme

\[ f(x) \]  
\[ \frac{dQ}{dx} \]  
[\ldots]\  
[\text{threshold}]  
\[ > \text{threshold} \]  
\[ \text{Edge Thinning} \]  

\[ f(x) \]  
\[ f'(x) \]  
\[ \text{threshold} \]  
\[ \text{Edge} \]  
\[ \text{Suppression of pts that are not maxima} \]

\( d(x) \)

\( |\ldots| \)

\( > \text{threshold} \)

\( \text{Edge Thinning} \)

\( f(x) \)

\( f'(x) \)

\( \text{threshold} \)

\( \text{Edge} \)
Edge detection: Gradient based Method (2)

2D scheme

\[ \nabla f(x, y) = \frac{\partial f(x, y)}{\partial x} \hat{i}_x + \frac{\partial f(x, y)}{\partial y} \hat{i}_y \]

\[ |\nabla f(x, y)| = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2} \]

Non-maxima suppression
For the considered point, has \(|\nabla f(x, y)|\) a maximum in at least one direction? If Yes then the pt belongs to an edge.
Edge detection: Gradient based Method (3)

2D Discrete Domain implementation

**Row Gradient Generation**

\[ f(j,k) \rightarrow g_r(j,k) \]

**Column Gradient Generation**

\[ f(j,k) \rightarrow g_c(j,k) \]

**Gradient Combination**

\[ g^*(j,k) = \sqrt{g_r^2(j,k) + g_c^2(j,k)} \]

**Thresholding Postprocessing**

\[ g^*(j,k) = |g_r(j,k)| + |g_c(j,k)| \text{ to reduce computational load} \]

\[ g^*(j,k) = |g_r(j,k)| \text{ directional operator} \]

\[ g^*(j,k) = |g_c(j,k)| \text{ directional operator} \]

Edge map
Requirements for Row/Column Gradient Generator.

It should be an operator that is:

- linear
- zero-phase
- constitutes a good discretization of the derivative operator
- reduces the impact of additive noise
- does not require an excessive computational effort

Several solutions have been proposed in the literature

\[ \text{e.g. Sobel operator} \]

\[
\begin{align*}
g_r &= f ** h_r \\
g_c &= f ** h_c \\
h_r &= \frac{1}{8} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \\
h_c &= \frac{1}{8} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}
\end{align*}
\]

\[
g^* = \sqrt{g_c^2 + g_r^2}
\]
Edge detection: the Sobel Operator (1)

The output data are scaled in order to cover all the available image dynamics
The output data are scaled in order to enhance low gradient values.
The Prewitt and the Frei-Chen operators are similar to the Sobel one. They differ in the convolution kernel.

For Prewitt:

\[
h_r = \frac{1}{6} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}
\]

For Frei-Chen:

\[
h_r = \frac{1}{4 + 2\sqrt{2}} \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}
\]

\[
h_c = \frac{1}{4 + 2\sqrt{2}} \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}
\]

Such operators differ in the degree of isotropy of their global response.
Edge detection: Threshold Selection

- For noise-free images the threshold can be chosen so that all significant amplitude discontinuities will be detected as edges.
- For noisy images, the threshold should be determined while taking into account both noise and image characteristics.
  - In particular, it is necessary to have an a-priori statistical model of edge spatial distribution and the conditional (edge or non-edge point) densities of gradient values.
  - All this statistical information is difficult to obtain and is only valid for specific image classes.
- The threshold is usually selected with very simple methods based on noise level estimation (through the analysis of uniform image areas) and on contrast step estimation of the edge profiles.
- Post-processing after thresholding (edge thinning, edge linking) is used for improving edge detection quality.
Sobel Edge Detector (1)

Thick edges
Sobel Edge Detector (2)

After simple edge thinning
Postprocessing: edge linking-clearing

- A continuous edge is created by “filling” the gaps between short edges that “seem” part of an unique structure.

- Criteria used to identify similarities between separated edge points:
  - **Gradient Magnitude Similarity**
    \[
    \| \nabla f(j_1, k_1) - \nabla f(j_2, k_2) \| < T_M
    \]
  - **Gradient Orientation Similarity**
    \[
    \angle [\nabla f(j_1, k_1)] - \angle [\nabla f(j_2, k_2)] < T_O
    \]

- In case of high similarity, the two edge portions are linked together promoting gap points to an “edge status”.

- Edge portions that are too short may be deleted (cleared).
Postprocessing: subpixel edge detection

1D case

\[ f(x) \]

Pixel values

\[ f'(x) \]

threshold

Polynomial interpolation from pixel values

Edge position (sub-pixel resolution)

Edge position (pixel resolution)

2D case

The location of the gradient maximum (sub-pixel resolution) is searched by interpolating available data in the normal direction to the local edge orientation. To reduce the complexity, only horizontal and vertical directions may be considered.
Edge detection: 2nd Derivative Method

1D case

\[ f(x) \]
\[ f'(x) \]
\[ f''(x) \]
Edge detection: Laplacian based Method (1)

2D case

Second order derivative operator = Laplacian operator

\[
\nabla^2 f(x, y) = \nabla \cdot (\nabla f(x, y)) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}
\]

\[
\frac{\partial^2 f(x, y)}{\partial x^2} \Rightarrow f_x(j, k) - f_x(j - 1, k) = f(j + 1, k) - 2f(j, k) + f(j - 1, k)
\]

\[
\frac{\partial^2 f(x, y)}{\partial y^2} \Rightarrow f_y(j, k) - f_y(j, k - 1) = f(j, k + 1) - 2f(j, k) + f(j, k - 1)
\]

\[
\nabla^2 f(j, k) = f(j, k) * h(j, k)
\]

\[
h_1(j, k) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad h_2(j, k) = \frac{1}{4} \begin{bmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{bmatrix} + \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 2 & 0 & 2 \\ 0 & -8 & 0 \\ 2 & 0 & 2 \end{bmatrix}
\]

\[
h_3(j, k) = -\frac{1}{4} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}
\]
Edge detection: Laplacian based Method (2)

\[
F \nabla^2(\bullet) \iff -4\pi^2(f_x^2 + f_y^2)
\]
Edge detection: Laplacian based Method (3)

The Laplacian Operator presents a high noise sensitivity

Black points identify Laplacian sign changes in at least one direction
Edge detection: Laplacian based Method (4)

A possible strategy to reduce “false edges”.

\[
\nabla^2(f(j,k)) \rightarrow |\nabla(\cdot)| \rightarrow |\nabla[f(j,k)]| \\
\]

Yes                          Yes

Yes

Sign Change?

Yes

|\nabla[f(j,k)]| > threshold

No

No

Yes

Yes
Edge detection: Laplacian based Method (5)
An image smoothing (low-pass filtering) can improve the performance of a Laplacian-based edge detector. The smoothing can be obtained by convolving the image with a two-dimensional gaussian kernel.

\[
  h(x, y) = \exp\left[-\frac{x^2 + y^2}{2\sigma^2}\right]
\]

\[
  H(\omega_x, \omega_y) = \sqrt{2\pi\sigma^2} \exp\left[-\pi^2 \sigma^2 (\omega_x^2 + \omega_y^2)\right]
\]
Gaussian Filtering

\[ h(x, y) = \exp\left[-\frac{x^2 + y^2}{2\sigma^2}\right] \]

\[ H(\omega_x, \omega_y) = \sqrt{2\pi\sigma^2} \exp\left[-2\pi^2 \sigma^2 (\omega_x^2 + \omega_y^2)\right] \]

- The bandwidth of the filter is determined by \( \sigma \)
- The impulse response can be discretized with minimum aliasing
- The extension of the discretized kernel is strictly related to \( \sigma \): \( \exp(-\alpha^2) \bigg|_{\alpha=2.7} < 10^{-3} \)
- The kernel is separable:
  \[ h(x, y) = h_m(x)h_m(y) \]
- An iterated gaussian filtering can be used to obtain image representations at different resolution scales
The LoG (Laplacian of Gaussian) operator

\[ g(x, y) = \nabla^2 \left( f(x, y) * h(x, y) \right) \]

- Both operators (prefiltering and Laplacian) are linear and can be interchanged

\[ g(x, y) = f(x, y) * \left[ \nabla^2 h(x, y) \right] \]

\[ \nabla^2 h(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \exp \left[ -\frac{x^2 + y^2}{2\sigma^2} \right] \]
The LoG impulse response

$$-\nabla^2 h(x, y)$$
Edge detection: Laplacian of Gaussian (LOG)
Canny Edge detector (1)

The image is “enhanced” by filtering with an appropriate impulse response. The points of maximum on the filtered image are edge points.

The filter is chosen in order to:

- Enhance the detection probability of “real edges”
- Minimize the localization error
- Reduce the detection of “false edges” and the creation of “ghost edges” parallel to real ones.

This approach leads, through a complex optimization, to have impulse response that reduce noise (low-pass filtering), allows good localization, has smooth profile, and depends to the profile of the considered edge (a-priori hypothesis).
Canny Edge detector (2)

For a 1D step edge model, the impulse response of the optimal filter is very similar to a first derivative of a gaussian function. Therefore the edge location is obtained as the maximum of the first derivative (zero crossing of the second derivative) of the luminance function convolved with a gaussian function (smoothing function).
Canny Edge detector: localization error

\[ G(x) \quad \text{edge profile (edge transition in } x = 0) \]
\[ G(x) + n(x) \quad \text{observed data} \]
\[ \text{original profile + gaussian white noise} \]
\[ (n^2 \text{ mean square noise amplitude}) \]
\[ f(x) \quad \text{impulse response of the filter} \]
\[ \text{(extension - w, w)} \]

\[ H_G(x) = G(x) * f(x) \]
\[ H_G(0) = \int_{-w}^{w} G(-x)f(x)dx \]
\[ H'_G(0) = 0 \quad \text{edge transition in } x = 0 \]

\[ H_n(x) = n(x) * f(x) \]
\[ [H'_G(x) + H'_n(x)]_{x = x_0} = 0 \quad \text{noise moves the edge} \]

\[ H'_G(x_0) = H'_G(0) + H''_G(0)x_0 + ... \]
\[ H'_G(x_0) \approx H''_G(0)x_0 \approx -H'_n(x_0) \]
\[ x_0 \approx -\frac{H'_n(x_0)}{H''_G(0)} \]

\[ E[H'_n(x_0)^2] = n^2 \int_{-w}^{w} f'^2(x)dx \]

\[ E[x_0^2] \approx \frac{n^2}{W} \int_{-W}^{W} f'^2(x)dx \]
\[ \left[ \int_{-W}^{W} G'(-x)f'(x)dx \right]^2 \]
Canny Edge detector: implementation

- $\sigma$ controls the smoothing (normally $\sigma=1/2$)
- For each point we see if the gradient has a maximum in at least one direction (it is equivalent to work with oriented operators)
- An hysteretic mechanism is used for the threshold in order to implement “edge following”
- It is possible to implement subpixel detection
Canny Edge detector: results
Corner Point: definition

What is a corner?

- A 2D point that corresponds to a well-defined 3D feature point
- An image point where an abrupt luminance variation (in more than one direction) occurs
- A point that is easy to track along an image sequence
Corner Point: a specific characteristic

A corner point is characterized by an abrupt variation of the image gradient direction
Corner Point: how to detect (1)

A possible approach:
- Find image edges
- Corner points are edge points with high curvature (of the edge line)

A modified approach:
- Find image edges
- Approximate edge lines through straight segments
- Corners points are the intersections of straight segments (intersection angles < threshold)
Corner Point: how to detect (2)
Corner Point: how to detect (3)

How to proceed using only local operators (Kitchen, Rosenfeld):

\[ I(x, y) \] Luminance Image
\[ I_x(x, y), I_y(x, y) \] Image gradients along x and y direction
\[ \theta(x, y) = \arctan\left( \frac{I_y}{I_x} \right) \] Gradient direction

\[ \theta_x(x, y) = \frac{I_{xx}I_y - I_{xy}I_y}{I_x^2} \], \[ \theta_y(x, y) = \frac{I_{yy}I_x - I_{xy}I_y}{I_y^2} \] Derivatives of the Gradient Orientation along x and y directions

\( (I_x, I_y) \) Gradient Vector

\( (-I_y, I_x) \) Vector that identify direction of minimum luminance variation (direction of the edge)

\[ k = (\theta_x, \theta_y) \cdot (-I_y, I_x) = \frac{I_{xx}I_y^2 + I_{yy}I_x^2 - 2I_{xy}I_xI_y}{I_x^2 + I_y^2} \]

This quantity measures, for a point, the rate of change of luminance gradient along the direction of minimal luminance variation, multiplied by gradient magnitude.

It can be seen as a “cornerness” measure.

if \( k(\bar{x}, \bar{y}) > \text{threshold and is a maximum} \) \[ \Rightarrow (\bar{x}, \bar{y}) \] is a Corner Point
The Noble/Harris Corner detector (1)

Sum of the square differences between an original image region (centered in \([x, y]\)) and a displaced \([u, v]\) version of the region itself:

\[
E(u, v) = \sum_{x, y} w(x, y) \left[ I(x + u, y + v) - I(x, y) \right]^2
\]

Window function

Shifted intensity

Intensity

Window function \(w(x, y) = \)

1 in window, 0 outside

or

Gaussian
The Noble/Harris Corner detector (2)

For small shifts $[u,v]$ we can use the following approximation:

$$E(u,v) \approx \sum_{x,y} w(x,y)[(I_x(x,y) + I_x(x,y)u + I_y(x,y)v) - I_x(x,y)]^2 =$$

$$= \sum_{x,y} w(x,y)[I_x(x,y)u + I_y(x,y)v]^2 =$$

$$= [u, v] \begin{bmatrix} \sum_{x,y} w(x,y)[I_x(x,y)]^2 & \sum_{x,y} w(x,y)I_x(x,y)I_y(x,y) \\ \sum_{x,y} w(x,y)I_x(x,y)I_y(x,y) & \sum_{x,y} w(x,y)[I_y(x,y)]^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$
The Noble/Harris Corner detector (3)

Therefore we have:

\[ E(u, v) \approx [u, v] \begin{bmatrix} u \\ v \end{bmatrix} \]

where \( M \) is a 2×2 matrix computed from image derivatives:

\[ M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \]
The Noble/Harris Corner detector (4)

\[ E(u, v) \cong [u, v] \begin{bmatrix} u \\ v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \]

\( \lambda_1, \lambda_2 \) eigenvalues of \( M \)

Ellipse \( \rightarrow \) \( E(u,v) = \text{const} \)
The Noble/Harris Corner detector (5)

\[ E(u, v) \cong \begin{bmatrix} u, v \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \begin{bmatrix} \lambda_1, \lambda_2 \end{bmatrix} \text{ eigenvalues of } M \]

Classification of image points using eigenvalues of \( M \):

- **“Corner”**: \( \lambda_1 \) and \( \lambda_2 \) are large, \( \lambda_1 \approx \lambda_2 \); \( E \) increases in all directions.
- **“Edge”**: \( \lambda_1 \gg \lambda_2 \)
- **“Flat”** region
The Noble/Harris Corner detector (6)

Measure of corner response:

\[ R_H = \det M - k (\text{trace} M)^2 \]  

\( (k \text{ – empirical constant, } k = 0.04-0.06) \)

\[ \frac{1}{R_N} = \frac{\det M}{\text{trace} M} \]  

The point (x,y) is corner if the considered quantity is > threshold and for (x,y) there is a relative maximum of the considered function.

It is possible to implement a super-resolution search interpolating the corner response function outside of the pixel grid.

Problem: Change of the smoothing function ⇒ Change of the estimated corner coordinates
The approach starts from two considerations:

- **Corner points are characterized by the fact that the Laplacian of their luminance profile is 0 (this does not depend on how filtered/smoothed is the image)**
- **The Baudet operator:**

\[
DET = \det \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} = I_{xx}I_{yy} - I_{xy}^2
\]

has a relative maximum (in all directions) in the proximity of the corner and, when applied to a set of progressively more filtered version of the image, the maxima can be shown to lie on a line (corner bisector)
The Giraudon-Deriche Corner detector (2)

Steps:

- Corner locations are roughly estimated (Harris, DET maxima, etc.)
- For different levels of resolution, near the estimated corner locations, the positions of DET maxima are determined with subpixel accuracy
- The bisector of the corner is computed through linear regression of the coordinates of the maxima
- The zero-crossing of the laplacian is searched along the line of maxima. This is the corner location (subpixel accuracy).
The Giraudon-Deriche Corner detector (3)

An example on real data
The KLT Corner (Feature) detector (1)

Assumption: Corners (Features) are image structures (a small set of neighboring pixels) easy to track along a sequence

The displacements of a image window from time n to time n+1 can be estimated (pure translational model) as:

$$D_n(x, y) = \arg\min_{\Delta x, \Delta y} \left\{ \sum_{u, v} |I(x + \Delta x + u, x + \Delta y + v, n + 1) - I(x + u, x + v, n)|^2 \right\}$$

To minimize the sum of the dissimilarity energies we differentiate it with respect to the unknowns and set the result to zero. Moreover we linearize the system by considering:

$$I(x + \Delta x, y + \Delta y) = I(x) + I_x \Delta x + I_y \Delta y$$

This yields the following system of equations:

$$M \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = -A;$$

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_{xy} \\ \sum I_{xy} & \sum I_y^2 \end{bmatrix}; \quad A = \begin{bmatrix} \sum I_t I_x \\ \sum I_t I_y \end{bmatrix};$$

$$I_t(x, y) = I(x, y, n + 1) - I(x, y, n)$$

KLT $\Rightarrow$ Kanade, Lucas, Tomasi
The KLT Corner (Feature) detector (2)

\[
M \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = -A; \quad M = \begin{bmatrix} \sum I_x^2 & \sum I_{xy} \\ \sum I_{xy} & \sum I_y^2 \end{bmatrix}; \quad A = \begin{bmatrix} \sum I_t I_x \\ \sum I_t I_y \end{bmatrix}
\]

A simple 1D example to justify the previous equations:

\[\Delta x = -\frac{I_x}{I_t}\]

Both the eigenvalues of M must be sufficiently large to ensure a good solution of the system. This is the condition to select the features.
Corner detection on real data
Corner/Feature tracking (1)

Use of correlation techniques:

- Select a set of Corners/Features on the first image
- Define a patch/window that surrounds each selected points
- For each point, search on the other images the position of the window more similar to the one that surrounds the Corner/Feature:

\[
\mathbf{p}_n = \arg \max_{\mathbf{p}} \left\{ \frac{\int_{\Omega} [I(\mathbf{x}, \mathbf{l}) \cdot I(\mathbf{T}(\mathbf{x}, \mathbf{p}), n)] d\mathbf{x}}{\sqrt{\int_{\Omega} [I(\mathbf{x}, \mathbf{l})]^2 d\mathbf{x}} \cdot \sqrt{\int_{\Omega} [I(\mathbf{T}(\mathbf{x}, \mathbf{p}), n)]^2 d\mathbf{x}}} \right\}
\]

\(\mathbf{T}(\mathbf{x}, \mathbf{p})\) Linear transformation mapping coordinates on the reference image into coordinates in the current image
Corner/Feature tracking (2)

A refinement:

\[
\begin{align*}
p_n &= \arg \max_p \left\{ \frac{\int [\bar{I}(x,1) \cdot \bar{I}(T(x,p),n)]^2 \, dx}{\sqrt{\int \bar{I}(x,1)^2 \, dx} \cdot \sqrt{\int [\bar{I}(T(x,p),n)]^2 \, dx}} \right\} \\
\bar{I}(x) &= I(x) - \frac{\int I(x) \, dx}{\int \, dx} \\
T(x, p) &\text{ Linear transformation mapping coordinates on the reference image into coordinates in the current image}
\end{align*}
\]

Examples of linear coordinate transformations

\[
\begin{align*}
T(x_1, p) &\Rightarrow \begin{cases} 
x_n = x_1 + p_1 \\
y_n = y_1 + p_2
\end{cases} \quad \text{simple translational model} \\
T(x_1, p) &\Rightarrow \begin{cases} 
x_n = p_1 x_1 + p_2 y_1 + p_3 \\
y_n = p_4 x_1 + p_5 y_1 + p_6
\end{cases} \quad \text{affine model}
\end{align*}
\]
Corner/Feature tracking(3)

Remembering that \((A-B)^2 = A^2 + B^2 - 2AB\) we can track features also as:

\[
p_n = \arg \min_p \left\{ \int_\Omega \left[ \tilde{I}(x,1) - \tilde{I}(T(x,p),n) \right]^2 dx \right\}
\]

\[
\tilde{I}(x) = I(x) - \frac{\int_\Omega I(x) dx}{\int_\Omega dx}
\]

\(T(x,p)\) Linear transformation mapping coordinates on the reference image into coordinates in the current image

To reduce the computational power some further:

\[
p_n = \arg \min_p \left\{ \int_\Omega |I(x,1) - I(T(x,p),n)| dx \right\}
\]
When temporal feature displacements are small (compared to the spatial luminance variations), or when we need to estimate just a refinement of previously available displacements, it is possible to use a differential approach (e.g. KLT tracking). Both translational and affine motion models can be considered.

Differential tracking algorithms are much faster than correlation-based tracking methods.
Corner/Feature tracking (5)

To reduce noise in the estimation of the Corner/Feature 2D trajectory it is possible to use the data obtained with one of the previously-described methods to perform curve fitting.

This can be justified by the fact that the Corner/Feature represents a real 3D point whose motion seems to have a certain momentum, therefore its 2D image projection is expected to follow a “regular” trajectory.

Possible trajectory fitting:
- Polynomial fitting
- B-spline fitting
- ........
How to track in time Corners/Features (6)

Another way to force a “regular” trajectory:
Use Kalman filtering in Corner/Feature motion estimation.
Example: we assume that the point moves (in 2D) with a constant acceleration, noise takes into account for changes in the acceleration. The input data is obtained by a corner tracker based on a correlation technique.

\[ \mathbf{a} = (x, x', x'')^T \] state vector
\[ \mathbf{a}_i = \phi \mathbf{a}_{i-1} + \mathbf{w}_{i-1} \] state model
\[ \phi = \begin{bmatrix} I_2 & I_2 \Delta t & \frac{1}{2} I_2 \Delta t^2 \\ 0_2 & I_2 & I_2 \Delta t \\ 0_2 & 0_2 & I_2 \end{bmatrix} \]
\[ \hat{x}_i = \mathbf{H} \mathbf{a}_i + \mathbf{v}_i = [I_2, 0_2, 0_2] \mathbf{a}_i + \mathbf{v}_i \] measur. eq.
(\( \bar{x}_k \) available data)
\[ \mathbf{w}_i = \mathbf{N}(0, \mathbf{Q}_i); \mathbf{v}_i = \mathbf{N}(0, \mathbf{R}_i) \]
\( \hat{a}_i \) estimated state vector; \( \mathbf{e}_i = \hat{a}_i - \mathbf{a}_i \)

\[ E[\mathbf{a}_0] = \hat{\mathbf{a}}_0 \]
\[ E[\mathbf{e}_0 \mathbf{e}_0^T] = \mathbf{P}_0 \]
\[ \hat{\mathbf{a}}_i (-) = \phi \hat{\mathbf{a}}_{i-1} (+) \]
\[ \mathbf{P}_i (-) = \phi \mathbf{P}_{i-1} (+) \phi^T + \mathbf{Q}_{i-1} \]
\[ \hat{\mathbf{a}}_i (+) = \hat{\mathbf{a}}_i (-) + \mathbf{K}_i (\tilde{x}_i - \hat{\mathbf{a}}_i (-)) \]
\[ \mathbf{P}_i (+) = (I - \mathbf{K}_i \mathbf{H}) \mathbf{P}_i (-) \]
\[ \mathbf{K}_i = \mathbf{P}_i (-) \mathbf{H}^T \left( \mathbf{H} \mathbf{P}_i (-) \mathbf{H}^T + \mathbf{R}_i \right)^{-1} \]

Filter equations
Some specific references

- L. Kitchen, A. Rosenfeld, Gray-level corner detection, pattern Recognition Letters, N. 1, 1982, pp. 95-102