Introduction to HOG3D

- HOG3D: Histogram of Oriented Gradients in 3D
- Extension of 2D HOG for volumetric data

3D Gradient Computation

- Computes gradients in x, y, and z directions
- Each gradient represents the rate of change of intensity in that direction
- Smoothing (e.g., Gaussian filter) is often applied first to reduce noise

```
smoothed_data = gaussian_filter(ct_data, sigma=1)
gx, gy, gz = np.gradient(smoothed_data)
```

Magnitude Calculation

- Magnitude represents the strength of the gradient
- Higher magnitude indicates stronger edges or intensity changes
- Crucial for weighting the importance of gradients in later steps

```
magnitude = np.sqrt(gx**2 + gy**2 + gz**2)
```

Orientation Calculation - Phi (Elevation)

- Phi represents the elevation angle in spherical coordinates
- Measures the angle between the gradient vector and the z-axis
- Range: [0, π]
- Captures the "up-down" orientation of the gradient

```
phi = np.arctan2(np.sqrt(gx**2 + gy**2), gz)
```

Orientation Calculation - Theta (Azimuth)

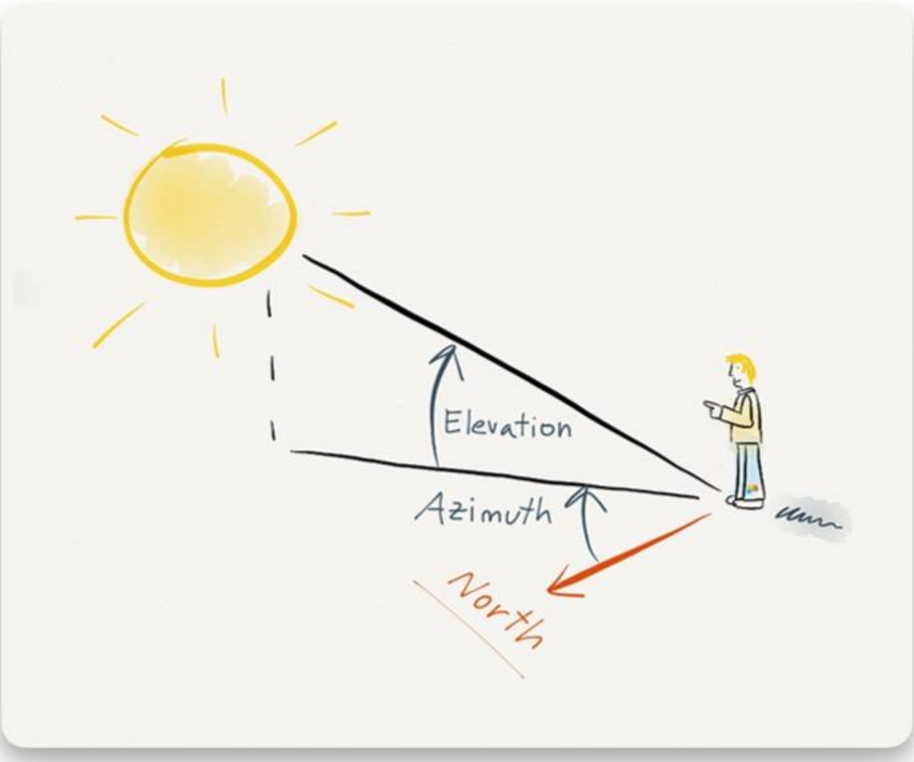
- Theta represents the azimuth angle in spherical coordinates
- Measures the angle between the x-axis and the projection of the gradient vector onto the xy-plane
- Range: $[-\pi, \pi]$
- Captures the "left-right" orientation of the gradient

```
theta = np.arctan2(gy, gx)
```

Understanding Spherical Coordinates

- **Phi (Elevation):**
    - 0° = pointing straight up (+z direction)
    - 90° = horizontal
    - 180° = pointing straight down (-z direction)

- **Theta (Azimuth):**
    - 0° = pointing along +x axis
    - 90° = pointing along +y axis
    - 180° or -180° = pointing along -x axis
    - -90° = pointing along -y axis



Cells in HOG3D

- Cells are small 3D regions of the input volume
- Typical size: 4x4x4 or 8x8x8 voxels
- Purpose: Capture local gradient information
- Smaller than the object of interest, allowing for some spatial invariance

```
cell_phi = phi[x+i:x+i+cell_size, y+j:y+j+cell_size, z+k:z+k+cell_size]
cell_theta = theta[x+i:x+i+cell_size, y+j:y+j+cell_size, z+k:z+k+cell_size]
cell_magnitude = magnitude[x+i:x+i+cell_size, y+j:y+j+cell_size, z+k:z+k+cell_size]
```

Cell Histograms

- Compute histograms of gradient orientations for each cell
- Separate histograms for phi and theta
- Weighted by gradient magnitude
- Captures the distribution of edge directions within the cell

```
hist_phi, _ = np.histogram(
    cell_phi.ravel(),
    bins=num_bins,
    range=(-np.pi, np.pi),
    weights=cell_magnitude.ravel()
)
```

```
hist_theta, _ = np.histogram(
    cell_theta.ravel(),
    bins=num_bins,
    range=(-np.pi, np.pi),
    weights=cell_magnitude.ravel()
)
```

Blocks in HOG3D

- Blocks are larger 3D regions composed of multiple cells
- Typical size: 2x2x2 or 3x3x3 cells
- Purpose: Provide a broader context and allow for normalization

```
block_hist[:, cell_index] = np.concatenate([hist_phi, hist_theta])
```

Block Formation

- Iterate over cells within a block
- Concatenate phi and theta histograms for each cell
- Results in a descriptor for the entire block

```
for i in range(0, block_size, cell_size):
    for j in range(0, block_size, cell_size):
        for k in range(0, block_size, cell_size):
            # ... compute cell histograms ...
            block_hist[:, cell_index] = np.concatenate([hist_phi, hist_theta])
            cell_index += 1
```

Block Normalization

- Normalize the combined histograms within each block
- Improves robustness to illumination and contrast variations
- Helps in comparing features across different parts of the volume

```
block_hist = block_hist.ravel()
block_hist /= np.linalg.norm(block_hist) + 1e-5
```

Parameters

- **Downsample Factor**: Reduces input data size (e.g., 4 means 1/4 size in each dimension)
- **Block Size**: Size of each 3D block (e.g., 4 means 4x4x4 voxels)
- **Cell Size**: Size of each cell within a block (e.g., 2 means 2x2x2 voxels)
- **Stride**: Step size between blocks (controls overlap)
- **Num Bins**: Number of orientation bins for histograms

```
block_size = 4
cell_size = 2
stride = 2
num_bins = 18
```

Final HOG3D Descriptor

- Concatenate normalized block histograms from across the volume
- Results in a high-dimensional feature vector
- Captures both local and semi-local 3D gradient information

```
hog3d_features.append(block_hist)
hog3d_positions.append([x + block_size // 2, y + block_size // 2, z + block_size // 2])
hog3d_orientations.append([np.mean(cell_phi), np.mean(cell_theta)])
```

Breakdown of Features

Breakdown of 288 Features

- The number 288 in the HOG3D feature vector represents the total number of features per block.

Here's how it's calculated:

- Number of orientation bins: 18 for phi + 18 for theta **= 36 bins total**
- Number of cells in a block: (4 // 2)^3 = 2^3 = **8 cells**
- Features per block: 36 * 8 = **288**

Cell	Phi Bins	Theta Bins
Cell 1	Features 1-18	Features 19-36
Cell 2	Features 37-54	Features 55-72
Cell 3	Features 73-90	Features 91-108
Cell 4	Features 109-126	Features 127-144
Cell 5	Features 145-162	Features 163-180
Cell 6	Features 181-198	Features 199-216
Cell 7	Features 217-234	Features 235-252
Cell 8	Features 253-270	Features 271-288

Visualizing Features with respect to Positions