Image Description: Histogram of Oriented Gradients

Alexandre Xavier Falcão

Institute of Computing - UNICAMP

afalcao@ic.unicamp.br

Histogram of Oriented Gradients (HoG)

- In object detection, for instance, each sample is a subimage (called window) around a candidate object.

Histogram of Oriented Gradients (HoG)

- In object detection, for instance, each sample is a subimage (called window) around a candidate object.
- The candidate objects reduce the number of windows for analysis and they can be obtained by segmentation and simple component analysis.

Histogram of Oriented Gradients (HoG)

- In object detection, for instance, each sample is a subimage (called window) around a candidate object.
- The candidate objects reduce the number of windows for analysis and they can be obtained by segmentation and simple component analysis.
- An example is the detection of car license plates in a grayscale image $\hat{I} = (D_I, I)$.

Histogram of Oriented Gradients (HoG)

- In object detection, for instance, each sample is a subimage (called window) around a candidate object.

- The candidate objects reduce the number of windows for analysis and they can be obtained by segmentation and simple component analysis.

- An example is the detection of car license plates in a grayscale image $\hat{I} = (D_I, I)$.

- The problem can be reduced to extract a HoG feature vector (or its concatenation with LBP) inside each window for pattern classification as car license plate or background.

Histogram of Oriented Gradients (HoG)

- In object detection, for instance, each sample is a subimage (called window) around a candidate object.
- The candidate objects reduce the number of windows for analysis and they can be obtained by segmentation and simple component analysis.
- An example is the detection of car license plates in a grayscale image $\hat{I} = (D_I, I)$.
- The problem can be reduced to extract a HoG feature vector (or its concatenation with LBP) inside each window for pattern classification as car license plate or background.
- The extension to color images can simply concatenate the HoG feature vectors of each band inside the window.

Agenda

The Histogram of Oriented Gradients (HoG) is a texture descriptor, which consists of the following steps.

Agenda

The Histogram of Oriented Gradients (HoG) is a texture descriptor, which consists of the following steps.

- Intensity normalization, gradient computation, and window definition.
- Cell definition.
- HoG computation per cell and pixel votes.
- Vote distribution.
- Coding – feature vector definition.

Intensity normalization and gradient computation

- As first step, the image intensities are normalized within an interval $[0 - L]$ (e.g., by gamma correction).

$$I'(p) \quad = \quad K \left[\frac{I(p)}{I_{\max}} \right]^{\gamma},$$

where $I_{\max} = \max_{\forall p \in D_I}\{I(p)\}$, $\gamma > 0$, and $K = 2^b - 1$.

Intensity normalization and gradient computation

- As first step, the image intensities are normalized within an interval $[0 - L]$ (e.g., by gamma correction).

$$I'(p) \quad = \quad K \left[\frac{I(p)}{I_{\max}} \right]^{\gamma},$$

where $I_{\max} = \max_{\forall p \in D_I}\{I(p)\}$, $\gamma > 0$, and $K = 2^b - 1$.

- Now, for each window of size $n_1 \times m_1$ pixels around a candidate object, the HoG feature vector requires the estimation of a gradient vector $\vec{g}(p)$ at each pixel p.

$$\vec{g}(p) \quad = \quad \sum_{\forall q \in \mathcal{A}_r(p)} [I(q) - I(p)] \exp\left(-\frac{\|q-p\|^2}{2\sigma^2} \right) \vec{pq},$$

where $\sigma = r/3$, $\vec{pq} = \frac{q-p}{\|q-p\|}$ and $r \geq 1$.

Intensity normalization and gradient computation

- As first step, the image intensities are normalized within an interval $[0 - L]$ (e.g., by gamma correction).

$$I'(p) \quad = \quad K \left[\frac{I(p)}{I_{\max}} \right]^{\gamma},$$

where $I_{\max} = \max_{\forall p \in D_I} \{I(p)\}$, $\gamma > 0$, and $K = 2^b - 1$.

- Now, for each window of size $n_1 \times m_1$ pixels around a candidate object, the HoG feature vector requires the estimation of a gradient vector $\vec{g}(p)$ at each pixel p.

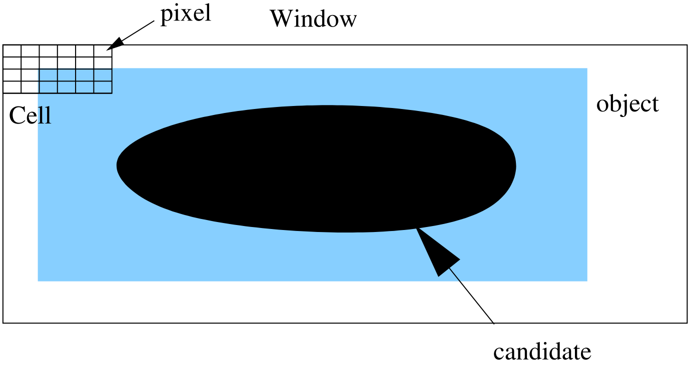
$$\vec{g}(p) \quad = \quad \sum_{\forall q \in \mathcal{A}_r(p)} [I(q) - I(p)] \exp\left(-\frac{\|q - p\|^2}{2\sigma^2}\right) \vec{pq},$$

where $\sigma = r/3$, $\vec{pq} = \frac{q-p}{\|q-p\|}$ and $r \geq 1$.

- The magnitude $\|\vec{g}(p)\|$ and orientation $\theta(p)$ (angle between $\vec{g}(p)$ and x) are used as follows.

Cell definition

The window is further divided into an integer number of cells containing $n_2 \times m_2$ pixels each.



HoG computation per cell and pixel votes

- One histogram of gradient orientations per cell is obtained with n_b bins.

HoG computation per cell and pixel votes

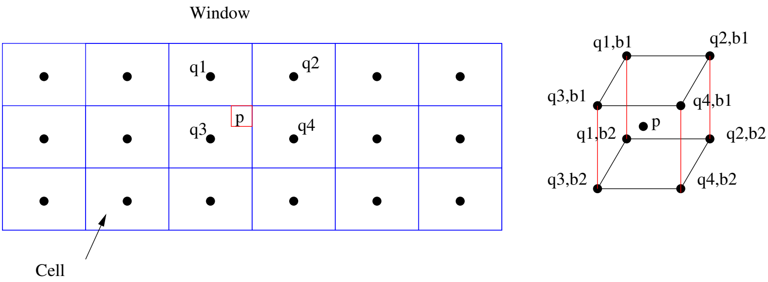
- One histogram of gradient orientations per cell is obtained with n_b bins.
- For $n_b = 9$ bins, for instance, the bin 0 may be used to accumulate votes from pixels whose $\|\vec{g}(p)\| = 0$ and the remaining bins store votes from pixels whose $\theta(p)$ falls within $[0-44], [45-89], \ldots, [315-359]$, respectively.

HoG computation per cell and pixel votes

- One histogram of gradient orientations per cell is obtained with n_b bins.
- For $n_b = 9$ bins, for instance, the bin 0 may be used to accumulate votes from pixels whose $\|\vec{g}(p)\| = 0$ and the remaining bins store votes from pixels whose $\theta(p)$ falls within $[0-44], [45-89], \ldots, [315-359]$, respectively.
- The orientation $\theta(p)$ for $h_x(p) = \frac{g_x(p)}{\|\vec{g}(p)\|}$ and $h_y(p) = \frac{g_y(p)}{\|\vec{g}(p)\|}$ is defined as

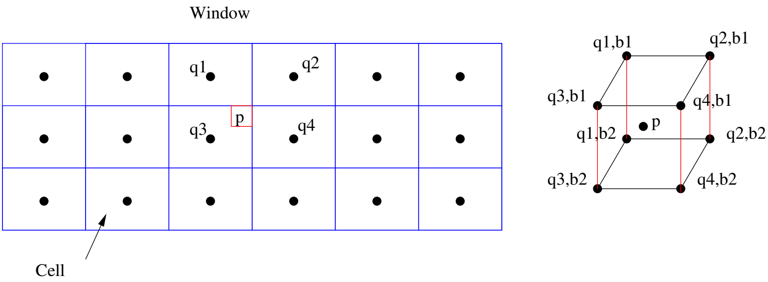
$$\theta(p) = \begin{cases} \frac{180}{\pi} \cos^{-1}(h_x(p)) & \text{if } h_y(p) \geq 0, \\ 360 - \frac{180}{\pi} \cos^{-1}(h_x(p)) & \text{if } h_y(p) < 0. \end{cases}$$

- Each pixel p distributes $\|\vec{g}(p)\|$ votes by trilinear interpolation between adjacent bins b_1 and b_2 of its four adjacent cells q_1, q_2, q_3, and q_4.



Vote distribution

- Each pixel p distributes $\|\vec{g}(p)\|$ votes by trilinear interpolation between adjacent bins b_1 and b_2 of its four adjacent cells $q_1, q_2, q_3,$ and q_4.



- For $\theta = 30$, for instance, $b_1 = 22$ and $b_2 = 67$, since the center of the 8 bins with non-zero gradient magnitude are represented by $22, 67, 112, 157, 202, 247, 292,$ and 337.

Vote distribution

- The distribution of votes aims to treat relevant pixels with high gradient magnitude that might fall in adjacent cells.

Vote distribution

- The distribution of votes aims to treat relevant pixels with high gradient magnitude that might fall in adjacent cells.
- Let (x_p, y_p, z_p), $z_p = \theta(p)$, be the coordinate of p in a 3D space.

Vote distribution

- The distribution of votes aims to treat relevant pixels with high gradient magnitude that might fall in adjacent cells.
- Let (x_p, y_p, z_p), $z_p = \theta(p)$, be the coordinate of p in a 3D space.
- Let (x_i, y_i) be the center of the cell q_i, $i = 1, 2, 3, 4$ and (q_1, b_1), (q_2, b_1), (q_3, b_1), (q_4, b_1), (q_1, b_2), (q_2, b_2), (q_3, b_2), and (q_4, b_2) be the 8 vertices (x_i, y_i, z_i), $i = 1, 2, \ldots, 8$, around p.

Vote distribution

- The distribution of votes aims to treat relevant pixels with high gradient magnitude that might fall in adjacent cells.

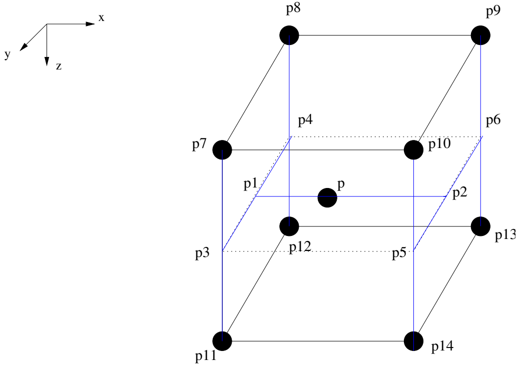
- Let (x_p, y_p, z_p), $z_p = \theta(p)$, be the coordinate of p in a 3D space.

- Let (x_i, y_i) be the center of the cell q_i, $i = 1, 2, 3, 4$ and (q_1, b_1), (q_2, b_1), (q_3, b_1), (q_4, b_1), (q_1, b_2), (q_2, b_2), (q_3, b_2), and (q_4, b_2) be the 8 vertices (x_i, y_i, z_i), $i = 1, 2, \ldots, 8$, around p.

- The gradient magnitude $w = \|\vec{g}(p)\|$ is a weight distributed among the 8 vertices by trilinear interpolation.

Vote distribution

The weight $w = \|\vec{g}(p)\|$ is first distributed between points p_1 and p_2 on opposite faces, then the weights on the faces are distributed among points p_3, p_4, p_5, p_6 of opposite edges, and finally the edge weights are distributed to the vertices p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, and p_{14} of the corresponding edges.



Vote distribution

The weights w_i of each point $p_i = (x_{p_i}, y_{p_i}, z_{p_i})$, $i = 1, 2, \ldots, 14$, are computed as

$$
\begin{aligned}
w_1 &= w \frac{(x_{p_2} - x_p)}{(x_{p_2} - x_{p_1})} \\
w_2 &= w \frac{(x_p - x_{p_1})}{(x_{p_2} - x_{p_1})} \\
w_3 &= w_1 \frac{(y_{p_1} - y_{p_4})}{(y_{p_3} - y_{p_4})} \\
w_4 &= w_1 \frac{(y_{p_3} - y_{p_1})}{(y_{p_3} - y_{p_4})}
\end{aligned}
$$

Vote distribution

$$
\begin{aligned}
w_5 &= w_2 \frac{(y_{p_2} - y_{p_6})}{(y_{p_5} - y_{p_6})} \\
w_6 &= w_2 \frac{(y_{p_5} - y_{p_2})}{(y_{p_5} - y_{p_6})} \\
w_7 &= w_3 \frac{(z_{p_{11}} - z_{p_3})}{(z_{p_{11}} - z_{p_7})} \\
w_{11} &= w_3 \frac{(z_{p_3} - z_{p_7})}{(y_{p_{11}} - z_{p_7})} \\
w_8 &= w_4 \frac{(z_{p_{12}} - z_{p_4})}{(z_{p_{12}} - z_{p_8})} \\
w_{12} &= w_4 \frac{(z_{p_4} - z_{p_8})}{(z_{p_{12}} - z_{p_8})}
\end{aligned}
$$

Vote distribution

$$
\begin{aligned}
w_{10} &= w_5 \frac{(z_{p_{14}} - z_{p_5})}{(z_{p_{14}} - z_{p_{10}})} \\
w_{14} &= w_5 \frac{(z_{p_5} - z_{p_{10}})}{(z_{p_{14}} - z_{p_{10}})} \\
w_9 &= w_6 \frac{(z_{p_{13}} - z_{p_6})}{(z_{p_{13}} - z_{p_9})} \\
w_{13} &= w_6 \frac{(z_{p_6} - z_{p_9})}{(z_{p_{13}} - z_{p_9})}
\end{aligned}
$$

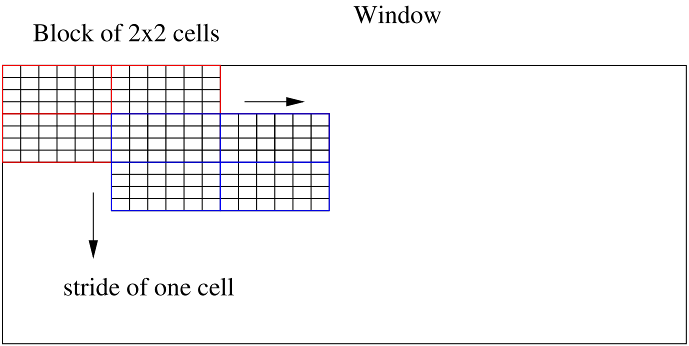
Finally the weights w_i are accumulated in the corresponding bin of the cell represented by p_i, $i = 7, 8, 9, 10, 11, 12, 13, 14$.

Coding: feature vector definition

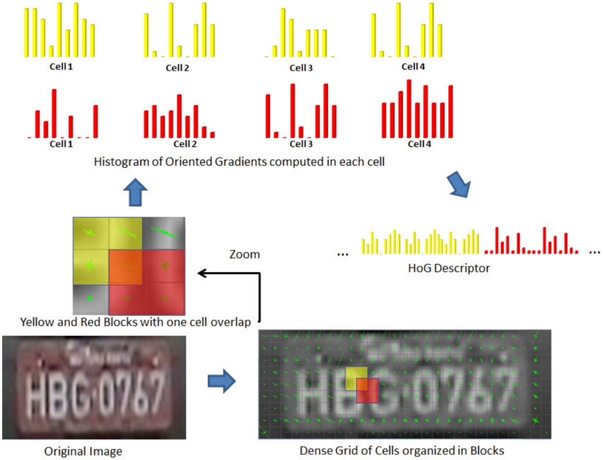
- Now, each group of $n_3 \times m_3$ cells constitutes a block.

Coding: feature vector definition

- Now, each group of $n_3 \times m_3$ cells constitutes a block.
- Adjacent blocks are defined by stride (displacement in x and y).



The cell histograms in each block are concatenated from left to right, top to bottom, and normalized, to treat contrast variations. Similarly, the block feature vectors are concatenated to output a HoG feature vector for the window.



Coding: feature vector definition

- Let $h_k(i)$, $i = 0, 1, \ldots, n_b - 1$ and $k = 1, 2, \ldots, n_3 \times m_3$, be the cell histograms in a block with $n_3 \times m_3$ cells.

Coding: feature vector definition

- Let $h_k(i)$, $i = 0, 1, \ldots, n_b - 1$ and $k = 1, 2, \ldots, n_3 \times m_3$, be the cell histograms in a block with $n_3 \times m_3$ cells.
- Their concatenation from left to right, top to bottom, generates a vector with features v_j, $j = 1, 2, \ldots, n_b \times n_3 \times m_3$.

Coding: feature vector definition

- Let $h_k(i)$, $i = 0, 1, \ldots, n_b - 1$ and $k = 1, 2, \ldots, n_3 \times m_3$, be the cell histograms in a block with $n_3 \times m_3$ cells.
- Their concatenation from left to right, top to bottom, generates a vector with features v_j, $j = 1, 2, \ldots, n_b \times n_3 \times m_3$.
- These features are normalized as

$$v_j = \frac{v_j}{\sqrt{\sum_{j=1}^{n_b \times n_3 \times m_3} v_j v_j} + \epsilon}$$

where ϵ is a small number.

Example

- For instance, for a window with 126×36 pixels and cells with 6×6 pixels, each window contains 21×6 cells.

Example

- For instance, for a window with 126×36 pixels and cells with 6×6 pixels, each window contains 21×6 cells.
- If each block is defined by 2×2 cells and the stride is 1 cell in x and y, each window generates 20×5 blocks.

Example

- For instance, for a window with 126×36 pixels and cells with 6×6 pixels, each window contains 21×6 cells.
- If each block is defined by 2×2 cells and the stride is 1 cell in x and y, each window generates 20×5 blocks.
- The four cell histograms of 9 bins in each block are concatenated and normalized to compose a vector of 36 features per block.

Example

- For instance, for a window with 126×36 pixels and cells with 6×6 pixels, each window contains 21×6 cells.
- If each block is defined by 2×2 cells and the stride is 1 cell in x and y, each window generates 20×5 blocks.
- The four cell histograms of 9 bins in each block are concatenated and normalized to compose a vector of 36 features per block.
- The feature vectors of the blocks are then concatenated to form a HoG vector with $20 \times 5 \times 36$ features.