Simple recognition of an object at a given position using video images

Jan Behrens

2014-06-02

This paper shall define a simple but effective algorithm for detecting a single object at a given position using video images. While the algorithm is comparatively easy to describe, it provides only minimal robustness against a change of lighting conditions and is thus only suitable for use in controlled environments.

The algorithm consists of two parts: the learning part and the recognition part. Input to the learning part is the following data:

- the dimensions D_X and D_Y of all video images (i. e. image resolution in x-axis and y-axis),
- a set E of neutral images $(E_1, E_2, \ldots, E_{N_E})$ (e. g. of an empty room), where the pixel values at coordinates x, y, and color channel c are given as $E_{n,x,y,c} \in \mathbb{R}$,
- a number $P \geq 1$ of positions to distinguish,
- for every position $p \in \{1, \ldots, P\}$ a set T_p of training images $(T_{p,1}, T_{p,2}, \ldots, T_{p,N_{T_p}})$, where the pixel values for x, y, c are given as $T_{p,n,x,y,c} \in \mathbb{R}$,
- optionally a set F of images $(F_1, F_2, \ldots, F_{N_F})$ that show the object in an invalid state, where the pixel values for x, y, c are given as $F_{n,x,y,c} \in \mathbb{R}$.

with $x \in \{1, \ldots, D_X\}$, $y \in \{1, \ldots, D_Y\}$, $c \in \{\text{red}, \text{green}, \text{blue}\}$

The learning part of the algorithm will create $(P + 1) \cdot D_X \cdot D_Y \cdot 3$ values as output: $\overline{E}_{x,y,c} \in \mathbb{R}$ and $C_{p,x,y,c} \in \mathbb{R}$ with $p \in \{1, \ldots, P\}$. To create these values, we proceed as explained on the following two pages.

For every pixel, we calculate the average neutral value:

$$\overline{E}_{x,y,c} := \frac{1}{N_E} \sum_{n=1}^{N_E} E_{n,x,y,c}$$

We then substract these average neutral pixel values from the other input images:

$$T'_{p,n,x,y,c} := T_{p,n,x,y,c} - \overline{E}_{x,y,c}$$
$$F'_{n,x,y,c} := F_{n,x,y,c} - \overline{E}_{x,y,c}$$

We calculate the average pixel value differences for every position $p \in \{1, \ldots, P\}$ per pixel:

$$\overline{T}_{p,x,y,c} := \frac{1}{N_{T_p}} \sum_{n=1}^{N_{T_p}} T'_{p,n,x,y,c}$$

Now we remove the constant component of every averaged image $\overline{T}_p$ per color channel c:

$$\widetilde{\overline{T}}_{p,x,y,c} := \overline{T}_{p,x,y,c} - \frac{1}{D_X D_Y} \sum_{i=1}^{D_X} \sum_{j=1}^{D_Y} \overline{T}_{p,i,j,c}$$

We do the same for the pixel value differences $T'_{p,n}$ and F'_n of all training (and invalid) images:

$$\widetilde{T}_{p,n,x,y,c} := T'_{p,n,x,y,c} - \frac{1}{D_X D_Y} \sum_{i=1}^{D_X} \sum_{j=1}^{D_Y} T'_{p,n,i,j,c}$$
$$\widetilde{F}_{n,x,y,c} := F'_{n,x,y,c} - \frac{1}{D_X D_Y} \sum_{i=1}^{D_X} \sum_{j=1}^{D_Y} F'_{n,i,j,c}$$

Now, for every position $p \in \{1, \ldots, P\}$, we try to find values $W_{p,x,y,c} \in \mathbb{R}^+$ such that the following values V_p get reasonably small ($V_p \approx 0$):

$$V_p := \frac{1}{N_{T_p}} \sum_{n=1}^{N_{T_p}} \left[\left(\max \left(0, 1 - \frac{\sum_{i=1}^{D_X} \sum_{j=1}^{D_Y} \sum_{c}^{3} \widetilde{T}_{p,n,i,j,c} \, \overline{T}_{p,i,j,c} \, W_{p,i,j,c}}{\sum_{i=1}^{D_X} \sum_{j=1}^{D_Y} \sum_{c}^{3} \widetilde{\overline{T}}_{p,i,j,c} \, \overline{T}_{p,i,j,c} \, W_{p,i,j,c}} \right) \right)^2 \right]$$

$$+ \frac{1}{P-1} \sum_{q \in \{1,\ldots,P\} \setminus \{p\}} \frac{1}{N_{T_q}} \sum_{n=1}^{N_{T_q}} \left[\left(\max \left(0, \frac{\sum_{i=1}^{D_X} \sum_{j=1}^{D_Y} \sum_{c}^{3} \widetilde{T}_{q,n,i,j,c} \, \overline{T}_{p,i,j,c} \, W_{p,i,j,c}}{\sum_{i=1}^{D_X} \sum_{j=1}^{D_Y} \sum_{c}^{3} \widetilde{\overline{T}}_{p,i,j,c} \, \overline{T}_{p,i,j,c} \, W_{p,i,j,c}} - \frac{1}{2} \right) \right)^2 \right]$$

$$+ \frac{1}{N_F} \sum_{n=1}^{N_F} \left[\left(\max \left(0, \frac{\sum_{i=1}^{D_X} \sum_{j=1}^{D_Y} \sum_{c}^{3} \widetilde{F}_{n,i,j,c} \, \overline{T}_{p,i,j,c} \, W_{p,i,j,c}}{\sum_{i=1}^{D_X} \sum_{j=1}^{D_Y} \sum_{c}^{3} \widetilde{\overline{T}}_{p,i,j,c} \, \overline{T}_{p,i,j,c} \, W_{p,i,j,c}} - \frac{1}{2} \right) \right)^2 \right]$$

$$\max(a, b) = \begin{cases} a & \text{if } a \geq b \\ b & \text{if } a < b \end{cases}$$

$$\max(0, x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$$

If $P = 1$, we omit the second summand (replace it with 0), and if $N_F = 0$, we omit the third summand (replace it with 0), avoiding the division by zero.

Once we found proper $W_{p,x,y,c}$, we can calculate the coefficients $C_{p,x,y,c}$:

$$C_{p,x,y,c} := \frac{\overline{T}_{p,x,y,c} \, W_{p,x,y,c} - \frac{1}{D_X D_Y} \sum_{i=1}^{D_X} \sum_{j=1}^{D_Y} \overline{T}_{p,i,j,c} \, W_{p,i,j,c}}{\sum_{i=1}^{D_X} \sum_{j=1}^{D_Y} \sum_{c'}^{3} \widetilde{\overline{T}}_{p,i,j,c'} \, \overline{T}_{p,i,j,c'} \, W_{p,i,j,c'}}$$

While the calculation of the weights $W_{p,x,y,c}$ during learning may be rather difficult, the recognition whether an image I (with the pixel values $I_{x,y,c} \in \mathbb{R}$) is showing the object in position p is rather easy. We assume a match if:

$$\sum_{x=1}^{D_X} \sum_{y=1}^{D_Y} \sum_{c}^{3} \left[\left(I_{x,y,c} - \overline{E}_{x,y,c} \right) \cdot C_{p,x,y,c} \right] \ \geq \ \frac{3}{4}$$