Deformable Part Models with Individual Part Scaling

Charles Dubout
charles.dubout@idiap.ch
François Fleuret
francois.fleuret@idiap.ch

Computer Vision and Learning Group
Idiap Research Institute
Martigny, Switzerland

École Polytechnique Fédérale de
Lausanne, Switzerland

Current deformable part models such as the ones introduced by Felzenszwalb *et al.* [2] let the parts deform only at a fixed predetermined scale relative to that of the root of the models (typically at twice the resolution). They do so because it enables them to find the optimal placement of each part efficiently, using a fast 2D distance transform algorithm.

We demonstrate in our paper that if one settles for approximately optimal placements, it is possible to efficiently deform the parts across scales as well, by reusing the original convolutions and distance transforms. Allowing parts to move in 3D increases the expressivity of the models, permitting them to compensate for a wider class of deformations, and might approximate an increase in the scanning resolution.

Let H be a feature pyramid and $\mathbf{p} = (x, y, z)$ specify a 2D position (x, y) in the z-th level of the pyramid. Let $\phi(\mathbf{p})$ denote the vector obtained by concatenating the feature vectors in the sub-window of H centered at $\mathbf{p}$, and $\phi_d(\mathbf{p})$ be the deformation features.

A model for an object with n parts is composed of a root filter $\mathbf{w}_0$ and n pairs $(\mathbf{w}_i, \mathbf{d}_i)$, where $\mathbf{w}_i$ is the filter of the i-th part and $\mathbf{d}_i$ is a vector specifying the deformation cost of the part placement.

An object hypothesis $\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_n$ specifies the location of the center of each filter in a feature pyramid. Its score is given by the score of each filter at its respective location, minus a deformation cost that depends on the location of each part with respect to the root position,

$$S(\mathbf{p}_0, \ldots, \mathbf{p}_n) = \mathbf{w}_0^\mathsf{T} \phi(\mathbf{p}_0) + \sum_{i=1}^{n} \mathbf{w}_i^\mathsf{T} \phi(\mathbf{p}_i) - \mathbf{d}_i^\mathsf{T} \phi_d(\mathbf{p}_i - \mathbf{p}_0). \quad (1)$$

In our algorithm, we allow the parts to also move across scales, and extend the usual deformation features to include the z component of the disparity between root and parts positions,

$$\phi_d(\mathbf{p}) = (1, x, y, z, x^2, y^2, z^2). \quad (2)$$

Ideally, one would like to now find the optimal location of each part in this way,

$$\mathbf{p}_i^\star = \underset{\mathbf{p} \in \mathbb{Z}^3}{\operatorname{argmax}} \, \mathbf{w}_i^\mathsf{T} \phi(\mathbf{p}) - \mathbf{d}_i^\mathsf{T} \phi_d(\mathbf{p} - \mathbf{p}_0(z_i)), \quad (3)$$

where $\mathbf{p}_0(z_i) = \left(\lambda^{z_i - z_0} x_0, \lambda^{z_i - z_0} y_0, z_0\right)$ are the coordinates of the root position in the i-th part's level z_i, λ being the scaling factor between two successive levels of the feature pyramid.

Unfortunately, $\mathbf{p}_0(z_i)$ is likely to be non-integral, and the generalized distance transform [1] thus cannot be used directly anymore. We thus approximate the root position at the scale of the i-th part by the closest integer one,

$$\tilde{\mathbf{p}}_0(z_i) = \underset{\mathbf{p} \in \mathbb{Z}^2 \times \{z_i\}}{\operatorname{argmin}} \, ||\mathbf{p} - \mathbf{p}_0(z_i)||. \quad (4)$$

Using this approximate root position, we can now again use the generalized distance transform in order to find the optimal part location,

$$\tilde{\mathbf{p}}_i^\star(\mathbf{p}_0) = \underset{\mathbf{p} \in \mathbb{Z}^3}{\operatorname{argmax}} \, \mathbf{w}_i^\mathsf{T} \phi(\mathbf{p}) - \mathbf{d}_i^\mathsf{T} \phi_d(\mathbf{p} - \tilde{\mathbf{p}}_0(z_i)). \quad (5)$$

We expect this location to coincide most of the time with the optimal one for the real root position, the difference between the real and the approximate one being at most 0.5 along the x and y axes. However, the optimal score returned by the transform will generally not match the score of any real root and part configuration so we recompute it in constant time using this time the real root position $\mathbf{p}_0(z_i)$,

$$\tilde{S}(\mathbf{p}_0) = \mathbf{w}_0^\mathsf{T} \phi(\mathbf{p}_0) + \sum_{i=1}^{n} \mathbf{w}_i^\mathsf{T} \phi(\tilde{\mathbf{p}}_i^\star(\mathbf{p}_0)) - \mathbf{d}_i^\mathsf{T} \phi_d(\tilde{\mathbf{p}}_i^\star(\mathbf{p}_0) - \mathbf{p}_0(z_i)), \quad (6)$$

in order to obtain a lower bound on the true optimal score.

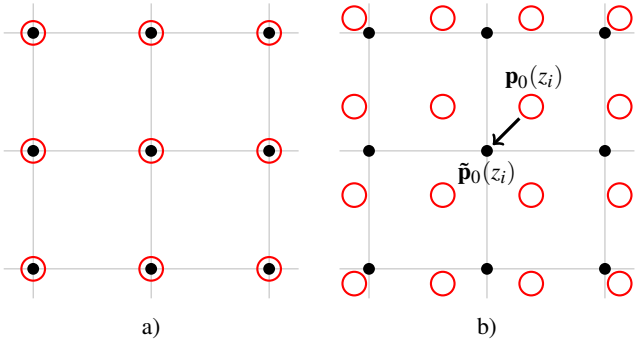

Figure 1: Lattices of part locations (in black) in a particular pyramid level. The red circles indicate root positions. In a) the part and root positions are at the same scale, as is always the case with standard models. In b) there is a mismatch between the scales of the two, and we show how we approximate a root position $\mathbf{p}_0(z_i)$ by rounding it to the closest integral position $\tilde{\mathbf{p}}_0(z_i)$ when looking for its optimal part placement.

	aero	bike	bird	boat	bottle	bus	car
voc-release4 (AP)	28.9	60.2	1.7	8.3	20.6	53.5	51.3
2D DPM (AP)	30.3	57.7	4.4	11.4	25.2	55.0	53.3
3D DPM (AP)	33.5	59.4	6.9	13.1	28.7	59.0	52.9
Rel. gain (%)	10.6	2.8	55.4	15.1	13.9	7.2	-0.8

	cat	chair	cow	table	dog	horse	mbike
voc-release4 (AP)	6.9	3.3	54.5	47.6	18.7	20.1	13.8
2D DPM (AP)	11.1	5.0	59.2	48.5	19.2	22.5	24.4
3D DPM (AP)	19.5	6.5	61.2	48.9	20.6	26.8	25.9
Rel. gain (%)	75.9	29.2	3.5	0.8	7.3	19.4	6.3

	person	plant	sheep	sofa	train	tv	mean
voc-release4 (AP)	38.8	5.8	14.3	28.1	37.3	39.0	**27.6**
2D (AP)	35.7	8.6	18.8	28.4	42.3	42.2	**30.2**
3D (AP)	32.9	11.1	21.5	31.7	44.5	43.8	**32.4**
Rel. gain (%)	-7.7	29.5	14.3	11.6	5.3	3.9	**15.2**

Table 1: Pascal VOC 2007 challenge Average Precision comparison for the models of [2] as well as our 2D and 3D models.

	Brute-force	Approx. DT
mean (AP)	45.0	44.9

Table 2: Comparison between an exact method as well as our approximation to the generalized distance transform.

The idea motivating our work is to make full use of all the convolutions between pyramid levels and part filters evaluated in DPMs, reusing them to deform parts across multiple scales. The extension we presented increases on average the detection accuracy of the models by 15% for a moderate augmentation of its total computational cost, the number of convolutions and distance transforms remaining constant. Despite relying on an approximation to the generalized distance transform, our approach obtains scores virtually equal to its exact but much slower counterpart.

[1] P. F. Felzenszwalb and D. P. Huttenlocher. Distance Transforms of Sampled Functions. Technical report, Cornell Computing and Information Science, 2004.

[2] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32 (9):1627–1645, 2010.