Appendix: Dynamic Steerable Blocks in Deep Residual Networks

Jörn-Henrik Jacobsen\textsuperscript{1}  
j.jacobsen@uva.nl
Bert de Brabandere\textsuperscript{2}  
bert.debrabandere@esat.kuleuven.be
Arnold W.M. Smeulders\textsuperscript{1}  
A.W.M.Smeulders@uva.nl

\textsuperscript{1} Informatics Institute  
University of Amsterdam  
Amsterdam, NL
\textsuperscript{2} ESAT-PSI  
KU Leuven,  
Leuven, BE

1 Equivariance Proof & Steering Equation Derivation

To prove that a frame is equivariant with respect to the action of a group transformation, determined by its generator $L_i$, we simply have to show that:

$$L_i \Phi(x, y) = B_i \Phi(x, y),$$

(1)

Where $B_i$ is some $n \times n$ matrix.

In our experiments we use Gaussian derivatives as the steerable frame and given that they are equivalent to the Hermite polynomials multiplied with a Gaussian aperture, it is sufficient to derive steering equations for rotation for the Hermite polynomials (here considered up to second order):

$$\Phi(x, y) = \{1, x, y, x^2 - 1, xy, y^2 - 1\}.$$  

(2)

To verify that these functions span an equivariant function space with respect to rotation, we apply the generator of rotations to the frame and verify that equation 1 holds. The generator of rotations in the plane is given by $L_r = -x \frac{d}{dy} + y \frac{d}{dx}$, applied to each frame element, we get:

$$L_r \Phi(x, y) = \begin{bmatrix} 0 & y \\ -x & 2xy \\ -x^2 + y^2 \\ -2xy \end{bmatrix} = B_r \begin{bmatrix} 1 \\ x \\ x^2 - 1 \\ xy \\ y^2 - 1 \end{bmatrix}.$$  

(3)

It is straightforward to solve this linear system and obtain the $6 \times 6$ matrix $B_r$:

$$B_r = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & -2 & 0 \end{bmatrix}.$$
Thus, we have proven that the function space is closed under the action of the group and the Hermite polynomials constitute an equivariant function space with respect to rotation. Subsequently, the exponential map directly yields the steering equations collected in the interpolation matrix $A^\theta$, that can rotate the whole frame by $\theta$:

$$\Phi^\theta(x,y) = e^{\theta B_r} \Phi(x,y) = A^\theta \Phi(x,y),$$

$$\Phi^\theta(x,y) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 & 0 & 0 \\ 0 & -\sin \theta & \cos \theta & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} + \frac{1}{2} \cos 2\theta & \sin 2\theta & \frac{1}{2} - \frac{1}{2} \cos 2\theta \\ 0 & 0 & 0 & -\frac{1}{2} \sin 2\theta & \cos 2\theta & \frac{1}{2} \sin 2\theta \\ 0 & 0 & 0 & \frac{1}{2} - \frac{1}{2} \cos 2\theta & -\sin 2\theta & \frac{1}{2} + \frac{1}{2} \cos 2\theta \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ x^2 - 1 \\ xy \\ y^2 - 1 \end{bmatrix}.$$

$\Phi^\theta(x,y)$ is the frame rotated by some angle $\theta$ and we use $A^\theta$ as activation functions on the output of the pose estimating network $F(\Phi)$. Given that every filter is learned as a linear combination over a frame, this gives us the possibility to rotate any learned feature by arbitrary and continuous angles $\theta$. The whole procedure is completely analogous for any other Lie group transformation. Further, k-parameter transformation groups can be composed from smaller groups. Here an example of the general linear group of rotation, anisotropic scalings and skew:

$$\Phi^{\{\theta_1, s_x, s_y, \theta_2\}}(x,y) = A^{\{\theta_1, s_x, s_y, \theta_2\}} \Phi(x,y) = e^{\theta_2 B_r} \cdot e^{s_x B_{sx}} \cdot e^{s_y B_{sy}} \cdot e^{\theta_1 B_r} \Phi(x,y).$$

(4)
2 Example of Dynamic Steerable Block for Boundary Experiments

\[ A(\theta) = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \]

Figure 1: Example of Dynamic Steerable Block with two gradient filters. \((G_x, G_y)\) can be replaced by any steerable frame. 1) Change from input to frame space by convolving with \(\Phi(x) = (G_x, G_y)\), 2) The interpolator network \(F(\Phi(x))\) estimates local pose from the invariant subspace, outputting a set of pose variables \(\theta\) for each location in the image and for each input/output channel (this can be changed depending on application by broadcasting pose variables across multiple channels). For the interpolator network \(F\) we used a small network with 8-16 units and three layers with tanh nonlinearities as they seemed suitable to approximate trigonometric steering equation solutions. For scalings, we found softplus and relu nonlinearities to work well. Optional: 3) the steering functions derived above are applied to the pose variable maps \(A(\theta)\) and effectively act as nonlinear pose-parametrized activation functions that regularize the interpolation network to output an explicitly interpretable pose space. 4) A 1x1 convolution layer, depicted as \([w_1, w_2]^T\) is applied to the already transformed frame outputs, this convolution represents the weights \(w^j_i\) of each feature governing the canonical appearance of the \(i_{\text{th}}\) feature map in the \(j_{\text{th}}\) layer. Again, to share weights across multiple feature maps, the weights can be broadcasted as desired across channels.