Fast Training of Pose Detectors in the Fourier Domain

João F. Henriques **Pedro Martins** **Rui Caseiro** **Jorge Batista**
Institute of Systems and Robotics
University of Coimbra
{henriques,pedromartins,ruicaseiro,batista}@isr.uc.pt

B MATLAB code and additional figures

This appendix contains code and additional figures that were not included in the main paper to meet length requirements.

Algorithm 1 MATLAB code for fast Fourier training of several pose detectors. X contains sample groups, with s transformed samples each (e.g. rotations of the same object). Note that the transformation should be cyclic. The samples within a group must also have the same Euclidean norm.

Inputs:
- X ($m \times n \times s$ data matrix with m features, n sample groups and s transformations)
- Y ($n \times s$ labels matrix)
- λ (scalar, regularization parameter)
- `regression` (a dual complex-valued regression algorithm – Alg. 2 or Alg. 3)

Output:
- W ($m \times s$ weights matrix – one pose detector per column)

```
g = zeros(n, n, s);
for p = 1:s
  g(:,:,p) = X(:,:,1).' * X(:,:,p);                    %Equation 13
end
X = fft(X, [], 3);
g = fft(g, [], 3);
Y = conj(fft(Y, [], 2));
for f = 1:s                                            %Equation 12
  W(:,f) = X(:,:,f) * regression(g(:,:,f), Y(:,f), lambda);
end
W = real(ifft(W, [], 2));
```

Algorithm 2 Complex-valued Dual Ridge Regression (used with Alg. 1).

```
function alphas = regression(G, y, lambda)
  alphas = (G + lambda * eye(size(G,1))) \ y;          %Equation 4
end
```

Algorithm 3 Complex-valued Dual Support Vector Regression (can replace `regression` in Alg. 1). The `real_svr` function can be any off-the-shelf SVR solver that accepts a "custom kernel matrix" (Gram matrix), e.g. `libsvm`.[1]

```
function alphas = svr(G, y, lambda)
  G = [real(G), imag(G).'; imag(G), real(G)];        %Equation A.29
  y = [real(y); imag(y)];
  alphas = real_svr(G, y, lambda);
  alphas = alphas(1:end/2) + 1i * alphas(end/2+1:end);
end
```

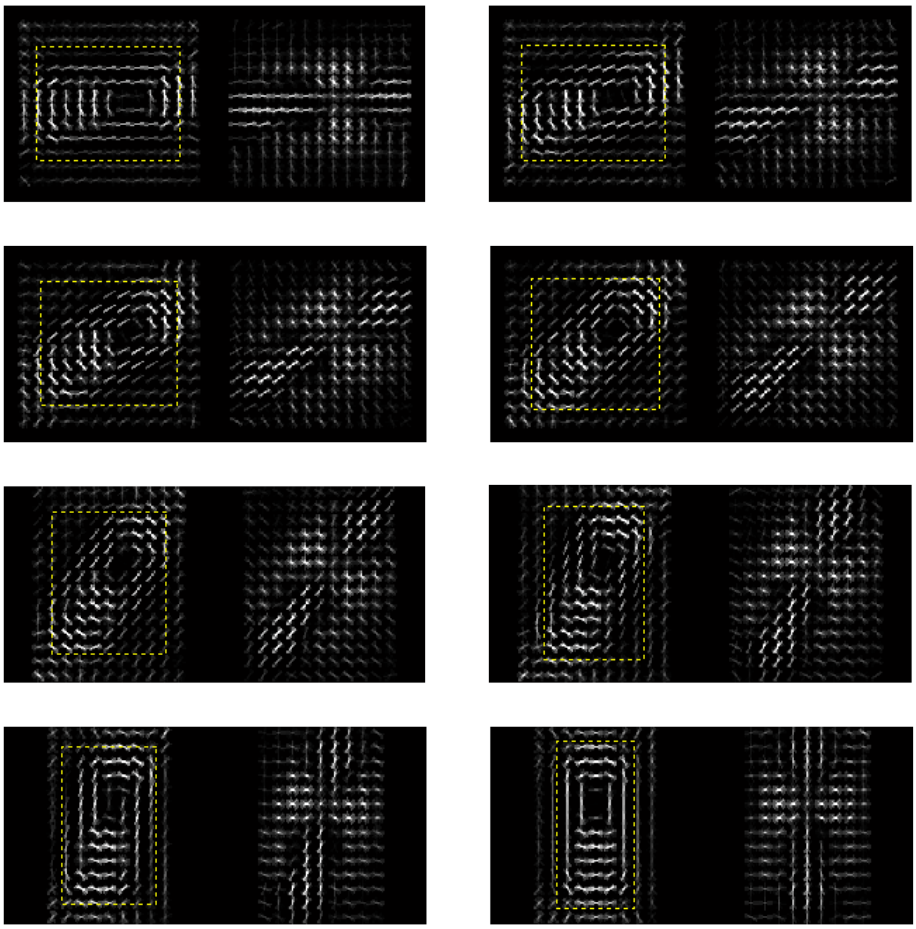

Figure 1: Visualization of HOG templates, obtained with Fourier training and RR, for the Satellite dataset. Each template represents a distinct pose across planar rotations. Only the templates from 0º to 90º are shown. Positive weights are shown on the left, negative weights on the right. The tight-fitting bounding box is also displayed as a yellow line. Note that rotated HOG templates cannot be obtained from a single template simply by applying a rotation to the cells – the gradient bins must also adjust their orientation correctly. For more complicated transformations and features, this cannot be done in closed form. The implicit transformation model allows us to bypass this issue, and obtain correct gradient orientations.

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

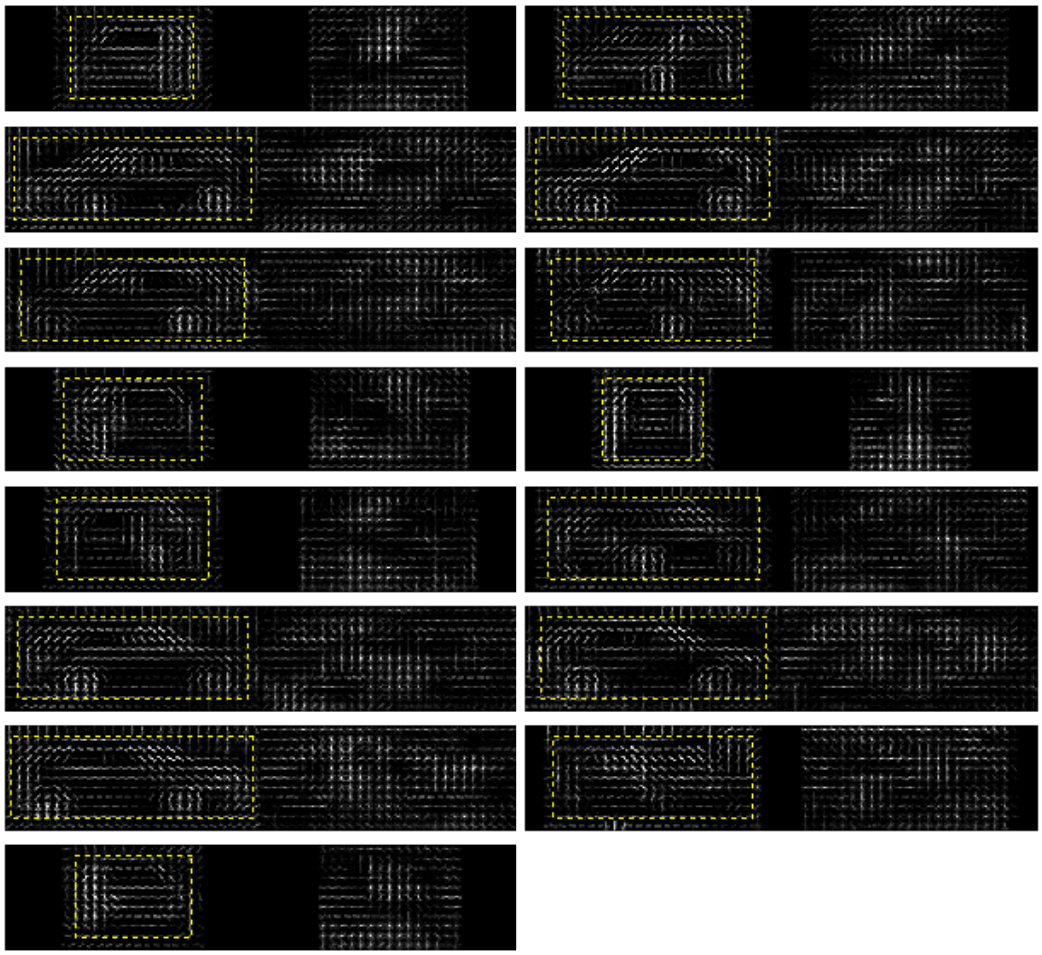

Figure 2: Visualization of HOG templates, obtained with Fourier training and RR, for the KITTI dataset. Each template represents a distinct pose across out-of-plane rotations (along the vertical axis). Positive weights are shown on the left, negative weights on the right. The tight-fitting bounding box is also displayed as a yellow line.

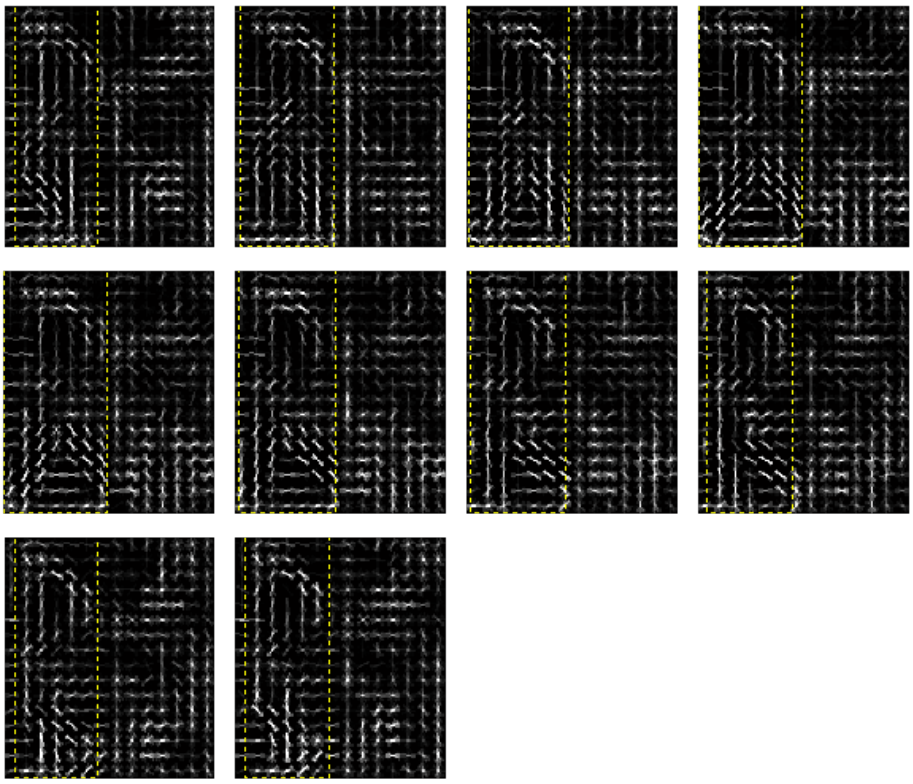

Figure 3: Visualization of HOG templates, obtained with Fourier training and RR, for the TUD-Campus/TUD-Crossing dataset. Each template represents a distinct pose across a pedestrian's walk cycle. Positive weights are shown on the left, negative weights on the right. The tight-fitting bounding box is also displayed as a yellow line.