A. Supplementary Appendix

In this appendix, we provide details on the derivation of the mathematics and the implementations used in the experiments.

A.1. Spatial and Fourier Domain Loss Functions

In the objective function of Eq. (2), it is easier to compute the shape regularizer loss $\mathcal{L}_c(\omega)$ in the Fourier domain, as it becomes a quadratic. Here we explain how we convert other loss functions to the Fourier domain as well for easy optimization.

In case of functions which are formed using inner products, the loss function can be easily converted from spatial domain to Fourier domain and vice versa. Due to Parseval's theorem, inner product is preserved between two spaces with proper normalisation. In other words,

$$\mathbf{w}_{nm}^\top \mathbf{x} = \mathbf{W}_{nm}^\top \mathbf{X}. \tag{9}$$

Applying this to Eq. (1) gives,

$$\mathbf{F}(\mathbf{x};\omega) = \sum_{n=1}^{N} \delta_n \max_{m=1}^{M} \mathbf{W}_{nm}^\top \mathbf{X} \quad , \tag{10}$$

Thus, we can easily obtain Eq. (3) and (8) in the Fourier domain by substitution.

After going to the Fourier domain, we do a simple feature mapping [2] to avoid the problem of dealing with complex numbers. This feature mapping again leverages the fact that the inner product is preserved between the spatial domain and the Fourier domain, and thus the inner product of two real signals in the Fourier domain are also real. The feature mapping is simply concatenating the real and imaginary components to form a real vector. If we denote this feature mapping as $\Phi(\mathbf{x})$

$$\Phi(\mathbf{x}) = \begin{bmatrix} \mathrm{Re}(\mathbf{x}) \\ \mathrm{Im}(\mathbf{x}) \end{bmatrix} \quad . \tag{11}$$

With this feature mapping any conventional solver designed for real numbers can be used.

A.2. Derivation of the Shape Regularizor Loss

Derivation of the shape regularizor loss in the Fourier domain (Eq. (6)) is straightforward. From the convolution theorem we have

$$\begin{aligned} \mathbf{w}_{n\eta_i(n)} * \mathbf{x}_i &= \mathbf{W}_{n\eta_i(n)} \circ \mathbf{X}_i &(12)\\ &= \mathbf{W}_{n\eta_i(n)}^\top \operatorname{diag}(\mathbf{X}_i) \quad , &(13) \end{aligned}$$

where $\circ$ denotes the Hadamard product. Thus the $\left\|\mathbf{w}_{n\eta_i(n)} * \mathbf{x}_i - \mathbf{w}_{n\eta_i(n)}^\top \mathbf{x}_i \mathbf{h}_i\right\|_2^2$ term in the spatial domain shape regularizor loss in Eq. (5) becomes

$$\begin{aligned} &\left\|\mathbf{w}_{n\eta_i(n)} * \mathbf{x}_i - \mathbf{w}_{n\eta_i(n)}^\top \mathbf{x}_i \mathbf{h}_i\right\|_2^2 &(14)\\ &= \left\|\mathbf{W}_{n\eta_i(n)}^\top \operatorname{diag}(\mathbf{X}_i) - \mathbf{W}_{n\eta_i(n)}^\top \mathbf{X}_i \mathbf{H}_i\right\|_2^2 &(15)\\ &= \left\|\mathbf{W}_{n\eta_i(n)}^\top \left(\operatorname{diag}(\mathbf{X}_i) - \mathbf{X}_i \mathbf{H}_i\right)\right\|_2^2 &(16)\\ &= \mathbf{W}_{n\eta_i(n)}^\top \mathbf{S}_i^\top \mathbf{S}_i \mathbf{W}_{n\eta_i(n)} \quad , &(17) \end{aligned}$$

where

$$\mathbf{S}_i = \left(\operatorname{diag}(\mathbf{X}_i) - \mathbf{X}_i \mathbf{H}_i\right)^\top . \tag{18}$$

In our case we deal with multichannel images and we sum the convolution results for each channel to get the final response. If we denote this summation as a matrix $\mathbf{E}$, we can simply substitute $\mathbf{W}_{n\eta_i(n)}^\top \operatorname{diag}(\mathbf{X}_i)$ to $\mathbf{W}_{n\eta_i(n)}^\top \operatorname{diag}(\mathbf{X}_i)\,\mathbf{E}$ and obtain the equation for multichannel image. This gives us

$$\mathbf{S}_i = \left(\operatorname{diag}(\mathbf{X}_i)\,\mathbf{E} - \mathbf{X}_i \mathbf{H}_i\right)^\top \quad . \tag{19}$$

Note that in Rodriguez *et al.*'s [26] work they restricted the desired shape $\mathbf{h}$ as the delta function. However, the restriction is not necessary as it does not affect any mathematical derivation. Furthermore the original derivation only considered single channel images.

A.3. Implementations of the Methods

Compared Methods For each method we sort the detected keypoints according to their respective response scores and keep the best keypoints. For MSER, we used the difference of Gaussian scores according to its scale as the sorting score.

Details for the implementations of the compared methods are as follows:

- SIFT: OpenCV library – http://opencv.org/downloads.html;
- SURF: OpenCV library – http://opencv.org/downloads.html;
- SFOP: Provided by the authors – http://www.ipb.uni-bonn.de/sfop/
- WADE: Provided by the authors – http://vision.deis.unibo.it/ssalti/?page_id=169
- MSER: Provided by the authors – http://www.robots.ox.ac.uk/œvgg/research/affine/
- FAST-9: Provided by the authors – http://www.edwardrosten.com/work/fast.html

Table 2: Shape parameters

	TILDE-GB	TILDE-CNN	TILDE-P
α	6	12	12
β	10.5	31.5	31.5
Patch Size	21×21	28×28	21×21

- LCF: Provided by the authors – http://chardson.com/papers/icra2013richardson.html
- SIFER: Publically available reference implementation – http://dev.ipol.im/~reyotero/comparing_20140906.tar.gz
- TaSK: Our implementation with SIFT keypoints as base keypoints.

Our Methods Table 2 shows the parameters used for the positive shapes in Eq. (4) and the patch size of the input image patch.

Details for the implementation of our methods are as follows:

- TILDE-CNN: We used the Python Theano library for implementation – http://deeplearning.net/software/theano/
- TILDE-GB: We used the same parameters as the work of Sironi *et al*. [31] and his implementation for the regressor. – http://cvlab.epfl.ch/software/centerline-detection
- TILDE-P: We implemented our method with MATLAB for learning and C for testing. We adapted the TRON solver [19] included in LibLinear library [9] to our problem. – http://www.csie.ntu.edu.tw/~cjlin/liblinear/

A.4. Number of Keypoints Used for Random 2%

The number of keypoints we used for obtaining 2% repeatability for all sequences are shown in Table 3. For both datasets we randomly sampled keypoint locations from images and increased the number of keypoints until we obtained 2% repeatability. Note that the numbers vary significantly for each sequence as the image sizes vary.

Table 3: Number of keypoints used for each sequence for evaluating Repeatability (*Random 2%*).

Oxford		*Webcam*	
Sequence	# Keypoints	Sequence	# Keypoints
Bark	50	*Mexico*	85
Bikes	174	*Chamonix*	122
Boat	65	*Courbevoie*	95
Graffiti	101	*Frankfurt*	204
Leuven	141	*Panorama*	161
Trees	175	*StLouis*	114
UBC	150		
Wall	175		