Template matching via densities on the roto-translation group
Bekkers, E.J.; Loog, M.; ter Haar Romenij, B.M.; Duits, R.

Published in:
IEEE Transactions on Pattern Analysis and Machine Intelligence

DOI:
10.1109/TPAMI.2017.2652452

Published: 01/01/2018

Document Version
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the author’s version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):
Abstract—We propose a template matching method for the detection of 2D image objects that are characterized by orientation patterns. Our method is based on data representations via orientation scores, which are functions on the space of positions and orientations, and which are obtained via a wavelet-type transform. This new representation allows us to detect orientation patterns in an intuitive and direct way, namely via cross-correlations. Additionally, we propose a generalized linear regression framework for the construction of suitable templates using smoothing splines. Here, it is important to recognize a curved geometry on the position-orientation domain, which we identify with the Lie group $SE(2)$: the roto-translation group. Templates are then optimized in a B-spline basis, and smoothness is defined with respect to the curved geometry. We achieve state-of-the-art results on three different applications: detection of the optic nerve head in the retina (99.83 percent success rate on 1,737 images), of the fovea in the retina (99.32 percent success rate on 1,616 images), and of the pupil in regular camera images (95.86 percent on 1,521 images). The high performance is due to inclusion of both intensity and orientation features with effective geometric priors in the template matching. Moreover, our method is fast due to a cross-correlation based matching approach.

Index Terms—Template matching, multi-orientation, invertible orientation scores, optic nerve head, fovea, retina

1 INTRODUCTION

We propose a cross-correlation based template matching scheme for the detection of objects characterized by orientation patterns. As one of the most basic forms of template matching, cross-correlation is intuitive, easy to implement, and due to the existence of optimization schemes for real-time processing a popular method to consider in computer vision tasks [1]. However, as intensity values alone provide little context, cross-correlation for the detection of objects has its limitations. More advanced data representations may be used, e.g., via wavelet transforms or feature descriptors [2], [3], [4], [5]. However, then standard cross-correlation can usually no longer be used and one typically resorts to classifiers, which take the new representations as input feature vectors. While in these generic approaches the detection performance often increases with the choice of a more complex representation, so does the computation time. In contrast, in this paper we stay in the framework of template matching via cross-correlation while working with a contextual representation of the image. To this end, we lift an image $f : \mathbb{R}^2 \to \mathbb{R}$ to an invertible orientation score $U_f : \mathbb{R}^2 \times S^1 \to \mathbb{C}$ via a wavelet-type transform using certain anisotropic filters [6], [7].

An orientation score is a complex valued function on the extended domain $\mathbb{R}^2 \times S^1 \equiv SE(2)$ of positions and orientations, and provides a comprehensive decomposition of an image based on local orientations, see Figs. 1 and 2. Cross-correlation based template matching is then defined via $L_2$ inner-products of a template $T \in L_2(SE(2))$ and an orientation score $U_f \in L_2(SE(2))$. In this paper, we learn templates $T$ by means of generalized linear regression.

In the $\mathbb{R}^2$-case (which we later extend to orientation scores, the $SE(2)$-case), we define templates $t \in L_2(\mathbb{R}^2)$ via the optimization of energy functionals of the form

$$ t^* = \arg\min_{t \in L_2(\mathbb{R}^2)} \{ E(t) := S(t) + R(t) \}, \tag{1} $$

where the energy functional $E(t)$ consists of a data term $S(t)$, and a regularization term $R(t)$. Since the templates optimized in this form are used in a linear cross-correlation based framework, we will use inner products in $S$, in which case (1) can be regarded as a generalized linear regression problem with a regularization term. For example, (1) becomes a regression problem generally known under the name ridge regression [8], when taking

$$ S(t) = \sum_{i=1}^{N} (t \cdot f_i)_{L_2(\mathbb{R}^2)} - y_i^2, \quad \text{and} \quad R(t) = \mu \| t \|_{L_2(\mathbb{R}^2)}^2, $$

Erik Johannes Bekkers, Marco Loog, Bart M. ter Haar Romeny, Senior Member, IEEE, and Remco Duits
where $f_i$ is one of $N$ image patches, $y_i \in \{0,1\}$ is the corresponding desired filter response, and where $\mu$ is a parameter weighting the regularization term. The regression is then from an input image patch $f_i$ to a desired response $y_i$, and the template $t$ can be regarded as the “set of weights” that are optimized in the regression problem. In this article we consider both quadratic (linear regression) and logistic (logistic regression) losses in $S$. For regularization we consider terms of the form

$$R(t) = \lambda \int_{\mathbb{R}^2} \|\nabla t(x)\|^2 dx + \mu \|t\|^2_{L^2(\mathbb{R}^2)},$$

and thus combine the classical ridge regression with a smoothing term (weighted by $\lambda$).

In our extension of smoothed regression to orientation scores we employ similar techniques. However, here we must recognize a curved geometry on the domain $\mathbb{R}^2 \times S^1$, which we identify with the group of roto-translations: the Lie group $SE(2)$ equipped with group product

$$g \cdot g' = (x, \theta) \cdot (x', \theta') = (R_{\theta}x + x, \theta + \theta').$$  \hspace{1cm} (2)

In this product the orientation $\theta$ influences the product on the spatial part. Therefore we write $\mathbb{R}^2 \times S^1$ instead of $\mathbb{R}^2 \times S^1$, as it is a semi-direct group product (and not a direct product). Accordingly, we must work with a rotating derivative frame (instead of axis aligned derivatives) that is aligned with the group elements $(x, \theta) \in SE(2)$, see e.g., the $(\partial_x, \partial_\theta, \partial_{\theta'})$-frames in Fig. 2. This derivative frame allows for (anisotropic) smoothing along oriented structures. As we will show in this article (Section 3.6), the proposed smoothing scheme has the probabilistic interpretation of time integrated Brownian motion on $SE(2)$ [9], [10].

**Regression and Group Theory.** Regularization in (generalized) linear regression generally leads to more robust classifiers/regressions, especially when a low number of training samples are available. Different types of regularizations in regression problems have been intensively studied in, e.g., [11], [12], [13], [14], [15], and the choice for regularization-type depends on the problem: E.g., $L_1$-type regularization is often used to sparsify the regression weights, whereas $L_2$-type regularization is more generally used to prevent over-fitting by penalizing outliers (e.g., in ridge regression [8]). Smoothing of regression coefficients by penalizing the $L_2$-norm of the derivative along the coefficients is less common, but it can have a significant effect on performance [13], [16].

We solve problem (1) in the context of smoothing splines: We discretize the problem by expanding the templates in a finite B-spline basis, and optimize over the spline coefficients. For d-dimensional euclidean spaces, smoothing splines have been well studied [17], [18], [19], [20]. In this paper, we extend the concept to the curved space $SE(2)$ and provide explicit forms of the discrete regularization matrices. Furthermore, we show that the extended framework can be used for time integrated Brownian motions on $SE(2)$, and show near to perfect comparisons to the exact solutions found in [9], [10].

In general, statistics and regression on Riemannian manifolds are powerful tools in medical imaging and computer vision [21], [22], [23], [24]. More specifically in pattern matching and registration problems Lie groups are often used to describe deformations. E.g., in [25] the authors learn a regression function $\mathbb{R}^m \rightarrow A(2)$ from a discrete $m$-dimensional feature vector to a deformation in the affine group $A(2)$. Their purpose is object tracking in video sequences. This work is however not concerned with deformation analysis, we instead learn a regression function $L^2(SE(2)) \rightarrow \mathbb{R}$ from continuous densities on the Lie group $SE(2)$ (obtained via an invertible orientation score transform) to a desired filter response. Our purpose is object detection in 2D images. In our regression we impose smoothed regression with a time-integrated hypo-elliptic Brownian motion prior and thereby extend least squares regression to smoothed regression on SE(2) involving first order variation in Sobolev-type of norms.

**Application Area of the Proposed Method.** The strength of our approach is demonstrated with the application to anatomical landmark detection in medical retinal images and pupil localization in regular camera images. In the retinal application we consider the problem of detecting the optic nerve head (ONH) and the fovea. Many image analysis applications require the robust, accurate and fast detection of these structures, see e.g., [26], [27], [28], [29]. In all three detection problems the objects of interest are characterized by (surrounding) curvilinear structures (blood vessels in the retina; eyebrow, eyelid, pupil and other contours for pupil detection), which are conveniently represented in invertible orientation scores. The invertibility condition implies that all image data is contained in the orientation score [7], [30]. With the proposed method we achieve state-of-the-art results both in terms of detection performance and speed: high detection performance is achieved by learning templates that make optimal use of the line patterns in

![Image](image-url)
2 Template Matching & Regression on $\mathbb{R}^2$

2.1 Object Detection via Cross-Correlation

We are considering the problem of finding the location of objects (with specific orientation patterns) in an image. While in principle an image may contain multiple objects of interest, the applications discussed in this paper only require the detection of one object per image. We search for the most likely location

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathbb{R}^2} P(\mathbf{x}),$$

with $P(\mathbf{x}) \in \mathbb{R}$ denoting the objective functional for finding the object of interest at location $\mathbf{x}$. We define $P$ based on inner products in a linear regression and logistic regression context, where we respectively define $P$ by

$$P(\mathbf{x}) = P_{\text{lin}}(\mathbf{x}) := (\mathbf{T}_x \mathbf{t}, \mathbf{f})_{L_2(\mathbb{R}^2)},$$

or

$$P(\mathbf{x}) = P_{\text{log}}(\mathbf{x}) := \sigma_n((\mathbf{T}_x \mathbf{t}, \mathbf{f})_{L_2(\mathbb{R}^2)}),$$

where $T_x$ denotes translation by $\mathbf{x}$ via

$$(T_x \mathbf{t})(\mathbf{x}) = t(\mathbf{x} - \mathbf{x}),$$

and where the $L_2(\mathbb{R}^2)$ inner product is given by

$$(\mathbf{t}, \mathbf{f})_{L_2(\mathbb{R}^2)} := \int_{\mathbb{R}^2} \overline{\mathbf{t}(\mathbf{x})} \mathbf{f}(\mathbf{x}) \, d\mathbf{x},$$

with associated norm $\| \cdot \|_{L_2(\mathbb{R}^2)} = \left( \int_{\mathbb{R}^2} |\mathbf{t}(\mathbf{x})|^2 \, d\mathbf{x} \right)^{1/2}$. Note that the inner-product based potentials $P(\mathbf{x})$ can be efficiently evaluated for each $\mathbf{x}$ using convolutions.

For a generalization of cross-correlation based template matching to normalized cross correlation, we refer the reader to the supplementary materials, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2017.2652452. For speed considerations we will however not use normalized cross correlation, but instead use a (fast) preprocessing step to locally normalize the images (cf. Section 4.2.1).

2.2 Optimizing $t$ Using Linear Regression

Our aim is to construct templates $\mathbf{t}$ that are “aligned” with image patches that contain the object of interest, and which are orthogonal to non-object patches. Hence, template $\mathbf{t}$ is found via the minimization of the following energy

$$E_{\text{lin}}(\mathbf{t}) = \sum_{i=1}^{N} \left( (\mathbf{t}, \mathbf{f}_i)_{L_2(\mathbb{R}^2)} - y_i \right)^2 + \lambda \int_{\mathbb{R}^2} \| \nabla \mathbf{t}(\mathbf{x}) \|^2 \, d\mathbf{x} + \mu \| \mathbf{t} \|^2_{L_2(\mathbb{R}^2)},$$

with $\mathbf{f}_i$ one of the $N$ training patches extracted from an image $\mathbf{f}_s$, and $y_i$ the corresponding label ($y_i = 1$ for objects and $y_i = 0$ for non-objects). In (7), the data-term (first term) aims for alignment of template $\mathbf{t}$ with object patches, in which case the inner product $(\mathbf{t}, \mathbf{f}_i)_{L_2(\mathbb{R}^2)}$ is ideally one, and indeed aims orthogonality to non-object patches (in which case the inner product is zero). The second term enforces spatial smoothness of the template by penalizing its gradient, controlled by $\lambda$. The third (ridge) term improves stability by dampening the $L_2$-norm of $\mathbf{t}$, controlled by $\mu$.

2.3 Optimizing $t$ Using Logistic Regression

In object detection we are essentially considering a two-class classification problem: the object is either present or it is not. In this respect, the quadratic loss term in (7) might not be the best choice as it penalizes any deviation from the desired response $y_i$, regardless of whether or not the response $(\mathbf{t}, \mathbf{f}_i)_{L_2(\mathbb{R}^2)}$ is on the correct side of a decision boundary. In other words, the aim is not necessarily to
construct a template that best maps an image patch $f_i$ to a response $y_i \in \{0, 1\}$, but rather the aim is to construct a template that best makes the separation between object and non-object patches. With this in mind we resort to the logistic regression model, in which case we interpret the non-linear objective function given in (5) as a probability, and define

$$p_i(f_i; t) = p(f_i; t),$$

$$p_0(f_i; t) = 1 - p(f_i; t),$$

with $p_i(f_i; t)$ and $p_0(f_i; t)$ denoting respectively the probabilities of a patch $f_i$ being an object or non-object patch. Our aim is now to maximize the likelihood (of each patch $f_i$ having maximum probability $p_i(f_i; t)$ for correct label $y_i$)

$$\ell(t) = \prod_{i=1}^N p_{y_i}(f_i; t) = \prod_{i=1}^N p(f_i; t)^{y_i} (1 - p(f_i; t))^{1-y_i}.$$  

(8)

We maximize the log-likelihood instead, which is given by

$$\ell_{\log}(t) := \log \left( \ell(t) \right) = \sum_{i=1}^N \log \left( p(f_i; t)^{y_i} (1 - p(f_i; t))^{1-y_i} \right) = \sum_{i=1}^N y_i (t, f_i)_{L_2([R^2])} - \log \left( 1 + e^{(t,f_i)_{L_2([R^2])}} \right).$$

(9)

Maximizing (10) is known as the problem of logistic regression. Similar to the linear regression case, we impose additional regularization and define the following regularized logistic regression energy, which we aim to maximize

$$E_{\log}^r(t) = \ell_{\log}(t) - \lambda \int_{\mathbb{R}^2} \| \nabla \ell(\tilde{x}) \|^2 d\tilde{x} - \mu \| t \|^2_{L_2([R^2])}.$$  

(11)

2.4 Template Optimization in a B-Spline Basis

Templates in a B-Spline Basis. In order to solve the optimizations (7) and (11), the template is described in a basis of direct products of $n$th order B-splines $B^n$

$$t(x, y) = \sum_{k=1}^{N_k} \sum_{l=1}^{N_l} c_{k,l} B^n \left( \frac{x}{s_k} - k \right) B^n \left( \frac{y}{s_l} - l \right),$$

(12)

with $B^n(x) = \left( 1 - \left[ \frac{2}{3} \right] ^n \right) \left[ \frac{2}{3} \right] ^n(x)$ a $n$th order B-spline obtained by $n$-fold convolution of the indicator function $1_{\left[ \frac{2}{3} \right]}$, and $c_{k,l}$ the coefficients belonging to the shifted B-splines. Here $s_k$ and $s_l$ scale the B-splines and typically depend on the number $N_k$ and $N_l$ of B-splines.

Linear Regression. By substitution of (12) in (7), the energy functional can be expressed in matrix-vector form (see Section 2 of the supplementary materials, available online)

$$E_{\log}^B(c) = \|Sc - y\|^2 + \lambda c^T Rc + \mu c^T I c.$$  

(13)

Regarding our notations we note that for spatial template $t$ given by (12) we have $E_{\log}(t) = E_{\log}^B(c)$, and label ‘B’ indicates finite expansion in the B-spline basis. The minimizer of (13) is given by

$$E_{\log}^B(c) = (S^T S + \lambda R + \mu I) c = S^T y,$$  

(14)

with $\dagger$ denoting the conjugate transpose, and $I$ denoting the identity matrix. Here $S$ is a $[N \times N_k N_l]$ matrix given by

$$S = \left( s_{1,1} \ldots s_{1,N_l}, s_{2,1} \ldots s_{2,N_l}, \ldots s_{N_k,1} \ldots s_{N_k,N_l} \right)_{i=1}^N,$$  

(15)

with $B_{k,l}(x, y) = B_{k}^n(x / s_k) B_{l}^n(y / s_l)$, for all $(x, y)$ on the discrete spatial grid on which the input image $f_D : \{1, N_y \} \times \{1, N_y \} \rightarrow \mathbb{R}$ is defined. Here $N_k$ and $N_l$ denote the number of splines in resp. $x$ and $y$ direction, and $s_k = \frac{N_y}{N_k}$ and $s_l = \frac{N_y}{N_l}$ are the corresponding resolution parameters. The $[N_k N_l \times 1]$ column vector $c$ contains the B-spline coefficients, and the $[N \times 1]$ column vector $y$ contains the labels, stored in the following form

$$c = (c_{1,1}, \ldots, c_{1,N_l}, c_{2,1}, \ldots, c_{2,N_l}, \ldots, c_{N_k,1}, \ldots, c_{N_k,N_l})^T,$$

$$y = (y_1, y_2, \ldots, y_N)^T.$$  

The $[N_k N_l \times N_k N_l]$ regularization matrix $R$ is given by

$$R = R_x^x \otimes R_x^y + R_y^y \otimes R_y^y,$$  

(17)

where $\otimes$ denotes the Kronecker product, and with

$$R_x^x(k, k') = -\frac{1}{s_k} \frac{\partial^2 B_{2n+1}}{\partial x^2} (k' - k),$$

$$R_y^y(l, l') = s_l B_{2n+1}(l' - l),$$

$$R_x^y(k, k') = s_k B_{2n+1}(k' - k),$$

$$R_y^x(l, l') = -\frac{1}{s_l} \frac{\partial^2 B_{2n+1}}{\partial y^2} (l' - l),$$

with $k, k' = 1, 2, \ldots, N_k$ and $l, l' = 1, 2, \ldots, N_l$. The coefficients $c$ can then be computed by solving (14) directly, or via linear system solvers such as conjugate gradient descent. For a derivation of the regularization matrix $R$ we refer to supplementary materials, Section 2, available online.

Logistic Regression. The logistic regression log-likelihood functional (11) can be expressed in matrix-vector notations as follows:

$$E_{\log}^B(c) = \left[ y^T Sc - 1_N^T \log \left( 1_N + \exp(Sc) \right) \right]$$

$$\quad - \lambda c^T Rc - \mu c^T I c,$$  

(19)

where $1_N = \{1, 1, \ldots, 1\}^T \in \mathbb{R}^{N \times 1}$, and where the exponential and logarithm are evaluated element-wise. We follow a standard approach for the optimization of (19), see e.g., [11], and find the minimizer by settings the derivative to $c$ to zero

$$\nabla_c E_{\log}^B(c) = S^T (y - p) - \lambda Rc - \mu I c = 0,$$  

(20)

with $p = (p_1, \ldots, p_N)^T \in \mathbb{R}^{N \times 1}$, with $p_i = \sigma((Sc)_i)$. To solve (20), we use a Newton-Raphson optimization scheme. This requires computation of the Hessian matrix, given by

$$\nabla^2 E_{\log}^B(c) = - (S^T WS + \lambda R + \mu I),$$  

(21)
with diagonal matrix \( W = \text{diag} \{ p_i(1 - p_i) \} \). The Newton-Raphson update rule is then given by

\[
c^{\text{new}} = c^{\text{old}} - \mathcal{H}(E^{\log}_D)^{-1} (\nabla E^{\log}_D (c)) = (STWS + \lambda R + \mu I)^{-1} STWz.
\]

with \( z = SE^{\text{old}} + W^{-1}(y - p) \), see e.g., [11, ch. 4.4]. Optimal coefficients found at convergence are denoted with \( c^* \).

Summarizing, we obtain the solution of (3) by substituting the optimized B-spline coefficients \( c^* \) into (12), and the resulting \( t \) enters (4) or (5). The most likely object location \( x^* \) is then found via (3).

### 3 Template Matching & Regression on \( SE(2) \)

This section starts with details on the representation of image data in the form of orientation scores (Section 3.1). Then, we repeat the sections from Section 2 in Sections 3.2, 3.3, 3.4, and 3.5, but now in the context of the extended domain \( SE(2) \).

#### 3.1 Orientation Scores on \( SE(2) \)

**Transformation.** An orientation score, constructed from image \( f : \mathbb{R}^2 \rightarrow \mathbb{R} \), is defined as a function \( U_f : \mathbb{R}^2 \times S^1 \rightarrow \mathbb{C} \) and depends on two variables \((x, \theta)\), where \( x = (x, y) \in \mathbb{R}^2 \) denotes position and \( \theta \in [0, 2\pi) \) denotes the orientation variable. An orientation score \( U_f \) of image \( f \) can be constructed by means of correlation with some anisotropic wavelet \( \psi \) via

\[
U_f(x, \theta) = (W_\psi f)(x, \theta) = \int_{\mathbb{R}^2} \psi(R_\theta^{-1}(x - z), y) f(z) dz,
\]

where \( \psi \in L_2(\mathbb{R}^2) \) is the correlation kernel, aligned with the \( x \)-axis, where \( W_\psi \) denotes the transformation between image \( f \) and orientation score \( U_f \), \( \psi_\theta(x) = \psi(R_\theta^{-1}x) \), and \( R_\theta \) is a counter clockwise rotation over angle \( \theta \).

In this work we choose cake wavelets [6], [7] for \( \psi \). While in general any kind of anisotropic wavelet could be used to lift the image to \( SE(2) \), cake wavelets ensure that no data-evidence is lost during the transformation: By design the set of all rotated wavelets uniformly cover the full Fourier domain of disk-limited functions with zero mean, and have thereby the advantage over other oriented wavelets (s.a. Gabor wavelets for specific scales) that they capture all scales and allow for a stable inverse transformation \( W_\psi^* \) from the score back to the image [6], [10].

**Left-Invariant Derivatives.** The domain of an orientation score is essentially the classical euclidean motion group \( SE(2) \) of planar translations and rotations, and is equipped with group product \( g \cdot g' = (x, \theta) \cdot (x', \theta') = (Rx + x, \theta + \theta') \). Here, we can recognize a curved geometry (cf. Fig. 2), and it is therefore useful to work in rotating frame of reference. As such, we use the left invariant derivative frame [9], [10]

\[
\{ \partial_\xi := \cos \theta \partial_x + \sin \theta \partial_y, \partial_\eta := -\sin \theta \partial_x + \cos \theta \partial_y, \partial_\theta \}.
\]

Using this derivative frame we will construct in Section 3.3 a regularization term in which we can control the amount of (anisotropic) smoothness along line structures.

#### 3.2 Object Detection via Cross-Correlation

As in Section 2, we search for the most likely object location \( x^* \) via (3), but now we define functional \( P \) respectively for the linear and logistic regression case in \( SE(2) \) by

\[
P(x) = P'_\log(x) := (T \times T, [U_j])_{L^2(SE(2))},
\]

or

\[
P(x) = P''_\log(x) := \sigma((T \times T, [U_j])_{L^2(SE(2))}),
\]

with \( (T \times T)(x, \theta) = T(x - x, \theta) \). The \( L_2(S(E(2))) \)-inner product is defined by

\[
(T, [U_j])_{L^2(SE(2))} := \int_{\mathbb{R}^2} \int_0^{2\pi} \overline{T(x, \theta)} U_j(x, \theta) dx d\theta,
\]

with norm \( \| \cdot \|_{L^2(SE(2))} = \sqrt{(\cdot, \cdot)_{L^2(SE(2))}} \).

#### 3.3 Optimizing \( T \) Using Linear Regression

Following the same reasoning as in Section 2.2 we search for the template that minimizes

\[
E_{\text{lin}}(T) = \sum_{i=1}^N \left( (T, [U_j])_{L^2(SE(2))} - y_i \right)^2 + \lambda \int_{\mathbb{R}^2} \int_0^{2\pi} \| \nabla T(x, \theta) \|^2 dx d\theta + \mu \| T \|_{L^2(SE(2))}^2,
\]

with smoothing term

\[
\| \nabla T(x) \|^2_D = D_{\xi, \eta} \frac{\partial^2 T}{\partial \xi^2}(x, \theta) + D_{\eta} \frac{\partial^2 T}{\partial \eta^2}(x, \theta) + D_{\theta} \frac{\partial^2 T}{\partial \theta^2}(x, \theta).
\]

Here, \( \nabla T = (\frac{\partial T}{\partial \xi}, \frac{\partial T}{\partial \eta}, \frac{\partial T}{\partial \theta})^T \) denotes the left-invariant gradient. Note that \( \partial_\xi \) gives the spatial derivative in the direction aligned with the orientation score kernel used at layer \( \theta \), recall Fig. 2. The parameters \( D_{\xi, \eta}, D_\eta \) and \( D_\theta \geq 0 \) are then used to balance the regularization in the three directions. Similar to this problem, first order Tikhonov-regularization on \( SE(2) \) is related, via temporal Laplace transforms, to left-invariant diffusions on the group \( SE(2) \) (Section 3.6), in which case \( D_{\xi, \eta}, D_\eta \) and \( D_\theta \) denote the diffusion constants in \( \xi, \eta \) and \( \theta \) direction. Here we set \( D_{\xi, \eta} = 1, D_\eta = 0 \), and thereby we get Laplace transforms of hypo-elliptic diffusion processes [10], [33]. Parameter \( D_\theta \) can be used to tune between isotropic (large \( D_\theta \)) and anisotropic (low \( D_\theta \)) diffusion (see e.g., [32, Fig. 3]). Note that anisotropic diffusion, via a low \( D_\theta \), is preferred as we want to maintain line structures in orientation scores.

#### 3.4 Optimizing \( T \) Using Logistic Regression

Similarly to what is done in Section 2.3 we can change the quadratic loss of (28) to a logistic loss, yielding the following energy functional

\[
E_{\text{log}}(T) = \mathcal{L}_{\text{log}}(T) - \lambda \int_{\mathbb{R}^2} \int_0^{2\pi} \| \nabla T(x, \theta) \|^2 dx d\theta - \mu \| T \|_{L^2(SE(2))}^2.
\]

1. Since both the inner product and the construction of orientation scores \( U_j \) from images \( f \) are linear, template matching might as well be performed directly on the 2D images (likewise (4) and (5)). Hence, here we take the modulus of the score as a non-linear intermediate step [32].
with log-likelihood (akin to (10) for the \( \mathbb{R}^2 \) case)
\[
\mathcal{L}(T) = \sum_{i=1}^{N} y_i(T, \{ U_{f_i} \})_{L_2(\mathcal{SE}(2))} - \log \left( 1 + e^{(T)U_{f_i}}_{L_2(\mathcal{SE}(2))} \right).
\]

The optimization of (28) and (30) follows quite closely the procedure as described in Section 2 for the 2D case. In fact, when \( T \) is expanded in a B-spline basis, the exact same matrix-vector formulation can be used.

### 3.5 Template Optimization in a B-Spline Basis

**Templates in a B-Spline Basis.** The template \( T \) is expanded in a B-spline basis as follows:
\[
T(x, y, \theta) = \sum_{k=1}^{N_k} \sum_{l=1}^{N_l} \sum_{m=1}^{N_m} c_{k,l,m} B^p_{s_k}(x - k) B^p_{s_l}(y - l) B^p(\theta \text{ mod } 2\pi - m),
\]
with \( N_k, N_l \) and \( N_m \) the number of B-splines in respectively the \( x, y \) and \( \theta \) direction, \( c_{k,l,m} \) the corresponding basis coefficients, and with angular resolution parameter \( s_m = 2\pi/N_m \).

**Linear Regression.** The shape of the minimizer of energy functional \( E_{\text{lin}}(T) \) in the \( \mathcal{SE}(2) \) case is the same as for \( E_{\text{lin}}(t) \) in the \( \mathbb{R}^2 \) case, and is again of the form given in (13). However, now the definitions of \( S, R \) and \( c \) are different. Now, \( S \) is a \([N \times N_k N_l N_m]\) matrix given by
\[
S = \left\{ (s_{1,1,1}, \ldots, s_{1,1,N_l}, \ldots, s_{1,N_k,N_l}, \ldots, s_{N_k,N_l,N_m}) \right\}^N_{k=1},
\]
\[
s_{k,l,m} = (B^p_{s_k}(x - k) U_{f_i}^T)(k, l, m),
\]
with \( B^p_{s_k}(x, y, \theta) = B^p(x) B^p(y) B^p(\theta \text{ mod } 2\pi) \). Vector \( c \) is a \([N_k N_l N_m \times 1]\) column vector containing the B-spline coefficients and is stored as follows:
\[
c = (c_{1,1,1}, \ldots, c_{1,1,N_l}, \ldots, c_{1,N_k,N_l}, \ldots, c_{N_k,N_l,N_m})^T.
\]

The explicit expression and the derivation of \([N_k N_l N_m \times N_k N_l N_m]\) matrix \( R \), which encodes the left invariant derivatives, can be found in the supplementary materials Section 2, available online.

**Logistic Regression.** Also for the logistic regression case we optimize energy functional (30) in the same form as (11) in the \( \mathbb{R}^2 \) case, by using the corresponding expressions for \( S, R \), and \( c \) in Eq. (19). These expressions can be inserted in the functional (19) and again the same techniques (as presented in Section 2.4) can be used to minimize this cost on \( \mathcal{SE}(2) \).

### 3.6 Probabilistic Interpretation of the \( \mathcal{SE}(2) \) Smoothing Prior

In this section we only provide a brief introduction to the probabilistic interpretation of the \( \mathcal{SE}(2) \) smoothing prior, and refer the interested reader to the supplementary materials for full details, available online. Consider the classic approach to noise suppression in images via diffusion regularizations with PDE’s of the form
\[
\begin{cases}
\frac{\partial}{\partial t}u = \Delta u, \\
u|_{t=0} = u_0,
\end{cases}
\]
where \( \Delta \) denotes the Laplace operator. Solving (35) for any diffusion time \( \tau > 0 \) gives a smoothed version of the input \( u_0 \). The time-resolvent process of the PDE is defined by the Laplace transform with respect to \( \tau \) time \( \tau \) is integrated out using a memoryless negative exponential distribution \( P(T = \tau) = e^{-\alpha \tau} \). Then, the time integrated solutions
\[
t(x) = \alpha \int_0^\infty u(x, \tau) e^{-\alpha \tau} d\tau,
\]
with decay parameter \( \alpha \), are in fact the solutions [34]
\[
t = \arg\min_{t \in \mathbb{R}_+} \left[ ||t - t_0||_{L_2(\mathbb{R}^2)} + \lambda \int_{\mathbb{R}^2} ||\nabla t(\bar{x})||^2 d\bar{x} \right],
\]
with \( \lambda = \alpha^{-1} \). Such time integrated diffusions (Eq. (36)) can also be obtained by optimization of the linear regression functionals given by Eqs. (7) and (25) for the \( \mathbb{R}^2 \) and \( \mathcal{SE}(2) \) case respectively.

In the supplementary materials, available online, we establish this connection for the \( \mathcal{SE}(2) \) case, and show how the smoothing regularizer in (28) and (30) relates to Laplace transforms of hypo-elliptic diffusions on the group \( \mathcal{SE}(2) \) [9], [10]. More precisely, we formulate a special case of our problem (the single patch problem) which involves only a single training sample \( U_{f_i} \), and show in a formal theorem that the solution is up to scalar multiplication the same as the resolvent hypo-elliptic diffusion kernel. The underlying probabilistic interpretation is that of Brownian motions on \( \mathcal{SE}(2) \), where the resolvent hypo-elliptic diffusion kernel gives a probability density of finding a random brush stroke at location \( x \) with orientation \( \theta \), given that a ‘drunkman’s pencil’ starts at the origin at time zero.

In the supplementary materials, available online, we demonstrate the high accuracy of our discrete numeric regression method using B-spline expansions with near to perfect comparisons to the continuous exact solutions of the single patch problem. In fact, we have established an efficient B-spline finite element implementation of hypo-elliptic Brownian motions on \( \mathcal{SE}(2) \), in addition to other numerical approaches in [9].

### 4 Applications

Our applications of interest are in retinal image analysis. In this section we establish and validate an algorithm pipeline for the detection of the optic nerve head (Section 4.2) and fovea (Section 4.3) in retinal images, and the pupil (Section 4.4) in regular camera images. Before we proceed to the application sections, we first describe the experimental set-up (Section 4.1). All experiments discussed in this section are reproducible; the data (with annotations) as well as the full code (Wolfram Mathematica notebooks) used in the experiments are made available at: http://erikbekkers.bitbucket.org/TMSE2.html. In the upcoming sections we only report the most relevant experimental results. More details on each application (examples of training samples, implementation details, a discussion on parameter settings, computation times, and examples of successful/failed detections) are provided in the supplementary materials, available online.

#### 4.1 The Experimental Set-Up

**Templates.** In our experiments we compare the performance of different template types, which we label as follows:
A: Templates obtained by taking the average of all positive patches ($y_i = 1$) in the training set, then normalized to zero mean and unit standard deviation.
B: Templates optimized without any regularization.
C: Templates optimized with an optimal $\mu$, and with $\lambda = 0$.
D: Templates optimized with an optimal $\lambda$ and with $\mu = 0$.
E: Templates optimized with optimal $\mu$ and $\lambda$.

The trained templates (B-E) are obtained either via linear or logistic regression in the $\mathbb{R}^2$ setting (see Sections 2.4 and 2.4.1, or in the $SE(2)$ setting (see Sections 3.5 and 3.5). In both the $\mathbb{R}^2$ and $SE(2)$ case, linear regression based templates are indicated with subscript $lin$, and logistic regression based templates with $log$. Optimality of parameter values is defined using generalized cross validation (GCV), which we soon explain in this section. We generally found that (via optimization using GCV) the optimal settings for template $E$ were $\mu \approx 0.5 \mu^*$, and $\lambda \approx 0.5 \lambda^*$, with $\mu^*$ and $\lambda^*$ respectively the optimal parameters for template $C$ and $D$.

Matching with Multiple Templates. When performing template matching, we use Eqs. (4) and (25) for respectively the $\mathbb{R}^2$ and $SE(2)$ case for templates obtained via linear regression and and for template $A$. For templates obtained via logistic regression we use respectively Eqs. (5) and (26). When we combine multiple templates we simply add the objective functionals. E.g., when combining template $C_{lin;\mathbb{R}^2}$ and $D_{log;SE(2)}$ we solve the problem

$$ \mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathbb{R}^2} P_{C_{lin;\mathbb{R}^2}}(\mathbf{x}) + P_{D_{log;SE(2)}}(\mathbf{x}), $$

where $P_{C_{lin;\mathbb{R}^2}}(\mathbf{x})$ is the objective function (see Eq. (4)) obtained with template $C_{lin;\mathbb{R}^2}$, and $P_{D_{log;SE(2)}}(\mathbf{x})$ (see Eq. (26)) is obtained with template $D_{log;SE(2)}$.

Rotation and Scale Invariance. The proposed template matching scheme can adapted for rotation-scale invariant matching, this is discussed in Section 5 of the supplementary materials, available online. For a generic object recognition task, however, global rotation or scale invariance are not necessarily desired properties. Datasets often contain objects in a human environment context, in which some objects tend to appear in specific orientations (e.g., eyebrows are often horizontal above the eye and vascular trees in the retina depart the ONH typically along a vertical axis). Discarding such knowledge by introducing rotation/scale invariance is likely to have an adversary effect on the performance, while increasing computational load. In Section 5 of the supplementary materials, available online, we tested a rotation/scale invariant adaptation of our method and show that in the three discussed applications this did indeed not lead to improved results, but in fact worsened the results slightly.

Automatic Parameter Selection via Generalized Cross Validation. An ideal template generalizes well to new data samples, meaning that it has low prediction error on independent data samples. One method to predict how well the system generalizes to new data is via generalized cross validation (GCV), which is essentially an approximation of leave-one-out cross validation [35]. The vector containing all predictions is given by $\mathbf{y} = SC_{\mu, \lambda}$, in which we can substitute the solution for $c_{\mu, \lambda}$ (from Eq. (14)) to obtain

$$ \mathbf{y} = A_{\mu, \lambda}\mathbf{y}, $$

with

$$ A_{\mu, \lambda} = (S^T S + \lambda R + \mu I)^{-1} S^T, $$

where $A_{\mu, \lambda}$ is the so-called smoother matrix. Then the generalized cross validation value [35] is defined as

$$ GCV(\mu, \lambda) = \frac{1}{N} \left( \frac{1}{(1 - \text{trace}(A_{\mu, \lambda})/N)^2} \right). $$

In the retinal imaging applications we set $\Omega = I$. In the pupil detection application we set $\Omega = \text{diag} \{y_i\}$. As such, we do not penalize errors on negative samples as here the diversity of negative patches is too large for parameter optimization via GCV. Parameter settings are considered optimal when they minimize the GCV value.

In literature various extensions of GCV are proposed for generalized linear models [36], [37], [38]. For logistic regression we use the approach by O’Sullivan et al. [36]: we iterate the Newton-Raphson algorithm until convergence, then, at the final iteration we compute the GCV value on the quadratic approximation (Eq. (22)).

Success Rates. Performance of the templates is evaluated using success rates. The success rate of a template is the percentage of images in which the target object was successfully localized. In both optic nerve head (Section 4.2) and fovea (Section 4.3) detection experiments, a successful detection is defined as such if the detected location $\mathbf{x}^*$ (Eq. (3)) lies within one optic disk radius distance to the actual location. For pupil detection both the left and right eye need to be detected and we therefore use the following normalized error metric

$$ e = \frac{\max(d_{left}, d_{right})}{w}, $$

in which $w$ is the (ground truth) distance between the left and right eye, and $d_{left}$ and $d_{right}$ are respectively the distances of detection locations to the left and right eye.

$k$-Fold Cross Validation. For correct unbiased evaluation, none of the test images are used for training of the templates, nor are they used for parameter optimization. We perform $k$-fold cross validation: The complete dataset is randomly partitioned into $k$ subsets. Training (patch extraction, parameter optimization and template construction) is done using the data from $k-1$ subsets. Template matching is then performed on the remaining subset. This is done for all $k$ configurations with $k-1$ training subsets and one test subset, allowing us to compute the average performance (success rate) and standard deviation. We set $k = 5$.

### 4.2 Optic Nerve Head Detection in Retinal Images

Our first application to retinal images is optic nerve head detection. The ONH is one of the key anatomical landmarks in the retina, and its location is often used as a reference point to define regions of interest for the analysis of the retina. The detection hereof is therefore an essential step in many automated retinal image analysis pipelines.

The ONH has two main characteristics: 1) it often appears as a bright disk-like structure on color fundus (CF) images (dark on SLO images), and 2) it is the place from which blood vessels leave the retina. Traditionally, methods...
The maximum intensity projections are defined as $\mu$. In our experiments we made use of both publicly and our own databases. The images in the databases contain a mix of good quality healthy images, and challenging diabetic retinopathy cases. Especially MESSIDOR and STARE contain high-quality healthy images, and challenging diabetic retinopathy cases. Especially MESSIDOR and STARE contain challenging images.

4.2.1 Processing Pipeline & Data

Processing Pipeline. First, the images are rescaled to a working resolution of $40 \, \mu m/\text{pix}$. In our experiments the average resolution per dataset was determined using the average optic disk diameter (which is on average $1.84 \, \text{mm}$). The images are normalized for contrast and illumination variations using the method from [47]. Finally, in order to put more emphasis on contextual/shape information, rather than pixel intensities, we apply a soft binarization to the locally normalized image $f$ via the mapping $\text{erf}(8f)$.

For the orientation score transform we use $N_o = 12$ uniformly sampled orientations from 0 to $\pi$ and lift the image using cake wavelets [6], [7]. For phase-invariant, nonlinear, left-invariant [10], and contractive [48] processing on $SE(2)$, we work with the modulus of the complex valued orientation scores rather than with the complex-valued scores themselves (taking the modulus of quadrature filter responses is an effective technique for line detection, see e.g., Freeman et al. [49]).

Due to differences in image characteristics, training and matching is done separately for the SLO and the color fundus images. For SLO images we use the near infrared channel, for RGB fundus images we use the green channel.

Positive training samples $f_i$ are defined as $N_x \times N_y$ patches, with $N_x = N_y = 251$, centered around true ONH location in each image. For every image, a negative sample is defined as an image patch centered around random location in the image that does not lie within one optic disk radius distance to the true ONH location. An exemplary ONH patch is given in Fig. 1. For the B-spline expansion of the templates we set $N_k = N_l = 51$ and $N_m = 12$.

Data. In our experiments we made use of both publicly available data, and a private database. The private database consists of 208 SLO images taken with an EasyScan fundus camera (i-Optics B.V., the Netherlands) and 208 CF images taken with a Topcon NW200 (Topcon Corp., Japan). Both cameras were used to image both eyes of the same patient, taking an ONH centered image, and a fovea centered image per eye. The two sets of images are labeled as “ES” and “TC” respectively. The following (widely used) public databases are also used: MESSIDOR (http://messidor.crihan.fr/index-en.php), DRIVE (http://www.isi.uu.nl/Research/Databases/DRIVE) and STARE (http://www.ces.clemson.edu/~ahover/stare), consisting of 1,200, 40 and 81 images respectively. For each image, the circumference of the ONH was annotated, and parameterized by an ellipse. The annotations for the MESSIDOR dataset were kindly made available by the authors of [50] (http://www.uhu.es/retinopathy). The ONH contour in the remaining images were manually outlined by ourselves. The annotations are made available on our website. The images in the databases contain a mix of good quality healthy images, and challenging diabetic retinopathy cases. Especially MESSIDOR and STARE contain challenging images.

4.2.2 Results and Discussion

The Templates. The different templates for ONH detection are visualized in Fig. 3. The $SE(2)$ templates are visualized using maximum intensity projections over $\theta$. In this figure we have also shown template responses to an example image. Visually one can clearly recognize the typical disk shape in the $\mathbb{R}^2$.
templates, whereas the SE(2) templates also seem to capture the typical pattern of outward radiating blood vessels (compare, e.g., $A_{\text{SE}(2)}$ with $A_{\text{SE}(2)}$). Indeed, when applied to a retinal image, where we took an example with an optic disk like pathology, we see that the $\mathbb{R}^2$ templates respond well to the disk shape, but also (more strongly) to the pathology. In contrast, the $SE(2)$ templates respond mainly to vessel pattern and ignore the pathology. We also see, as expected, a smoothing effect of gradient based regularization ($D$ and $E$) in comparison to standard $L_2$-norm regularization ($C$) and no regularization ($B$). Finally, in comparison to linear regression templates, the logistic regression templates have a more binary response due to the logistic sigmoid mapping.

Detection Results. Table 1 gives a breakdown of the quantitative results for the different databases used in the experiments. The templates are grouped in $\mathbb{R}^2$ templates, $SE(2)$ templates, and combination of templates. Within these groups, they are further divided in average, linear regression, and logistic regression templates. The best overall performance within each group is highlighted in gray.

Overall, we see that the $SE(2)$ templates out-perform their $\mathbb{R}^2$ equivalents, and that combinations of the two types of templates give best results. The two types are nicely complementary to each other due to disk-like sensitivity of the $\mathbb{R}^2$ templates and the vessel pattern sensitivity of the $SE(2)$ templates. If one of the two ONH characteristics is less obvious (as is, e.g., for the disk-shape in Fig. 3), the other can still be detected. Also, the failures of $\mathbb{R}^2$ templates are mainly due to either distracting pathologies in the retina, or poor contrast of the optic disk. As reflected by the increased performance of $SE(2)$ templates over $\mathbb{R}^2$ templates, a more stable pattern seems to be the vessel pattern.

From Table 1 we also deduce that the individual performances of the linear regression templates outperform the logistic regression templates. Moreover, the average templates give best individual performance, which indicates that with our effective template matching framework good performance can already be achieved with basic templates. However, we also see that low performing individual templates can prove useful when combining templates. In fact, we see that combinations with all linear $\mathbb{R}^2$ templates are highly ranked, and for the $SE(2)$ templates it is mainly the logistic regression templates. This can be explained by the binary nature of the logistic templates: even when the maximum response of the templates is at an incorrect location, the difference with the correct location is often small.
The $R^2$ template then adds to the sensitivity and precision. The best results obtained with untrained templates was a 99.19 percent success rate (14 fails), and with the overall best template combination we obtained a 99.83 percent success rate (3 fails).

State of the Art. In Table 2 we compare our results on the publicly available benchmark databases MESSIDOR, DRIVE and STARE, with the most recent methods for ONH detection (sorted from oldest to newest from top to bottom). In this comparison, our best performing method ($A_{R^2} + C_{log(SE(2))}$) performs better than or equally well as the best methods from literature. We have also listed the computation times, and see that our method is also ranked as one of the fastest methods for ONH detection. The average computation time, using our experimental implementation in Wolfram Mathematica 10.4, was 0.5 seconds per image on a computer with an Intel Core i703612QM CPU and 8 GB memory. A full breakdown of timings of the processing pipeline is given in the supplementary materials Section 4, available online.

4.3 Fovea Detection in Retinal Images

Our second application to retinal images is for the detection of the fovea. The fovea is the location in the retina which is responsible for sharp central vision. It is characterized by a small depression in thickness of the retina, and on healthy retinal images it often appears as a darkened area. Since the foveal area is responsible for detailed vision, this area is weighted most heavily in grading schemes that describe the severity of a disease. Therefore, correct localization of the fovea is essential in automatic grading systems [52].

Methods for the detection of the fovea heavily rely on contextual features in the retina [28], [45], [53], [54], [55], and take into account the prior knowledge that 1) the fovea is located approximately 2.5 optic disk diameters lateral to the ONH center, that 2) it lies within an avascular zone, and that 3) it is surrounded by the main vessel arcades. All of these methods restrict their search region for the fovea location to a region relative to the (automatically detected) ONH location. To the best of our knowledge, the proposed detection pipeline is the first that is completely independent of vessel segmentations and ONH detection. This is made possible due to the fact that anatomical reference patterns, in particular the vessel structures, are generically incorporated in the learned templates via data representations in orientation scores.

4.3.1 Processing Pipeline & Data

Processing Pipeline. The proposed fovea detection pipeline is the same as for ONH detection, however, now the positive training samples $f_i$ are centered around the fovea.

Data. The proposed fovea detection method is validated on our (annotated) databases “ES” and “TC”, each consisting of 208 SLO and 208 color fundus images respectively (cf. Section 4.2.1). We further test our method on the most used publicly available benchmark dataset MESSIDOR (1,200 images). Success rates were computed based on the fovea annotations kindly made available by the authors of [28].

4.3.2 Results and Discussion

The $R^2$ templates seem to be more tuned towards the dark

![Fig. 4](image-url)
TABLE 3
Comparison to State of the Art: Fovea Detection Success Rates, the Number of Fails (in Parentheses), and Computation Times

<table>
<thead>
<tr>
<th>Method</th>
<th>MESSIDOR</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Niemeijer et al. [28], [55]</td>
<td>97.9% (25)</td>
<td>7.6 \cite{5}</td>
</tr>
<tr>
<td>Yu et al. [54]</td>
<td>95.0%* \cite{60}</td>
<td>3.9 \cite{5}</td>
</tr>
<tr>
<td>Gegendes-Arias et al. [28]</td>
<td>96.9% (37)</td>
<td>0.9</td>
</tr>
<tr>
<td>Giachetti et al. [45]</td>
<td>99.1% (11)</td>
<td>5.0 \cite{5}</td>
</tr>
<tr>
<td>Aquino [53]</td>
<td>98.2% (21)</td>
<td>10.9 \cite{5}</td>
</tr>
<tr>
<td>Proposed</td>
<td>99.7% (3)</td>
<td>0.5</td>
</tr>
</tbody>
</table>

\*Timing success-criterion based on half optic radius. 
\*\*Timing includes ONH detection.

(isotropic) blob like appearance of the fovea, whereas in the \( SE(2) \) templates one can also recognize the pattern of vessels surrounding the fovea (compare \( A_{SE(2)} \) with \( A_{SE(2)} \)). To illustrate the difference between these type of templates, we selected an image in which the fovea location is occluded with bright lesions. In this case the method has to rely on contextual information (e.g., the blood vessels). Indeed, we see that the \( R^2 \) templates fail due to the absence of a clear foveal blob shape, and that the \( SE(2) \) templates correctly identify the fovea location. The effect of regularization is also clearly visible; no regularization \( (B) \) results in noisy templates, standard \( L_2 \) regularization \( (C) \) results in more stable templates, and smoothed regularization \( (D \) and \( E) \) results in smooth templates. In templates \( D_{SE(2)} \) and \( E_{SE(2)} \) we see that more emphasis is put on line structures.

Detection Results. A full overview of individual and combined template performance is discussed in the supplementary materials, available online, here we only provide a summary. Again there is an improvement using \( SE(2) \) templates over \( R^2 \) templates, although the difference is smaller than in the ONH application. Apparently both the dark blob-like appearance \( (R^2 \) templates) and vessel patterns \( (SE(2) \) templates) are equally reliable features of the fovea. A combination of templates leads to improved results and we conclude that the templates are again complementary to each other. Furthermore, again linear regression performs better than logistic regression. In fovea detection we do observe a large improvement of template training over basic averaging; 1,529 of 1,616 (94.6 percent) successful detections with \( C_{Lin.SE(2)} \) versus 1,488 (92.1 percent) with \( A_{SE(2)} \). The best performing \( R^2 \) template was \( A_{R^2} \) (65.6 percent), the best \( SE(2) \) template was \( C_{Lin.SE(2)} \) (94.6 percent). The best combination of templates was \( C_{Lin.R^2} + C_{log.SE(2)} \) with 1,605 (99.3 percent) detections. When using non-optimized templates 1,588 (98.3 percent) successful detections were achieved (with \( A_{R^2} + A_{SE(2)} \)).

State of the Art. In Table 3 we compared our results on the publicly available benchmark database MESSIDOR with the most recent methods for fovea detection (sorted from oldest to newest from top to bottom). In this comparison, our best performing method \( (C_{Lin.R^2} + C_{log.SE(2)}) \) quite significantly outperforms the best methods from literature. Furthermore, our detection pipeline is also the most efficient one; the computation time for fovea detection is the same as for ONH detection, which is 0.5 seconds.

4.4 Pupil Detection

Our third application is that of pupil localization in regular camera images, which is relevant in many applications as they provide important visual cues for face detection, face recognition, and understanding of facial expressions. In particular in gaze estimation the accurate localization of the pupil is essential. Eye detection and tracking is however challenging due to, amongst others: occlusion by the eyelids and variability in size, shape, reflectivity or head pose.

Many pupil localization algorithms are designed to work on periocular images, these are close-up views of the eyes. Such images can be acquired by dedicated eye imaging devices, or by means of cropping a full facial image (see Fig. 5a). We will consider both the problem of detection pupils in periocular images and the more difficult problem of detection in full images.

We compare our method against the seven most recent pupil detection methods from literature, for a full overview see \cite{56} and \cite{57}. A method similar to our \( R^2 \) approach in the sense that it is also based on 2D linear filtering is the method by Kroon et al. \cite{58}. In their paper templates are obtained via linear discriminant analysis of pupil images. Asteriada et al. \cite{59} detect the pupil by matching templates using features that are based on distances to the nearest strong (facial) edges in the image. Campadelli et al. \cite{60} use a supervised approach with a SVM classifier and Haar wavelet features. The method by Timm et al. \cite{61} is based on searching for gradient fields with a circular symmetry. Valenti et al. \cite{62} use a similar approach but additionally include information of isophote curvature, with supervised refinement. Markus et al. \cite{57} employ a supervised approach using an ensemble of randomized regression trees. Leo et al. \cite{56} employ a completely unsupervised approach similar to those in \cite{61}, \cite{62}, but additionally include analysis of self-similarity.

A relevant remark is that all of the above mentioned methods rely on prior face detection, and restrict their search region to periocular images. Our method works completely stand alone, and can be used on full images.

4.4.1 Processing Pipeline & Data

Processing Pipeline. Interestingly, we could again employ the same processing pipeline (including local normalization via \cite{47}) which was used for ONH and fovea detection. In our experiments we train templates for the left and right eye separately.

Data. We validated our pupil detection approach on the publicly available BioID database (http://www.biod.com), which is generally considered as one of the most challenging and realistic databases for pupil detection in facial images. The database consists of 1,521 frontal face grayscale images with significant variation in illumination, scale and pose.

4.4.2 Results and Discussion

The Templates. Figs. 5b and 5c show respectively the trained \( R^2 \) and \( SE(2) \) templates for pupil detection of the right eye, and their filtering response to the input image in Fig. 5a. Here the trained \( R^2 \) templates seemed to capture the pupil as a small blob in the center of the template, but apart from that no real structure can be observed. In the average template we do however clearly see structure in the form of an “average face”. The \( SE(2) \) templates reveal structures that resemble the eyelids in nearly all templates. The linear regression templates look sharper and seem to contain more
detail than the average template, and the logistic regression templates seem to take a good compromise between course features and details.

**Detection Results.** We again refer to the supplementary materials, available online, for a full benchmarking analysis, in summary we observed the following. In terms of success rates we see a similar pattern as with the ONH and fovea application, however, here we see that the learned templates ($C; D$ and $E$) significantly outperform the average templates, and that logistic regression leads to better templates than using linear regression (94.0 percent success rate for $C_{\log \cdot \text{SE}(2)}$ versus 87.2 percent for $D_{\text{lin} \cdot \text{SE}(2)}$). Overall, the $\text{SE}(2)$ templates outperform the $\mathbb{R}^2$ templates, linear regression templates outperform the average template, and logistic regression templates outperform linear regression templates. The best $\mathbb{R}^2$ template was $D_{\text{lin} \cdot \mathbb{R}^2}$ with 1,151 of 1,521 detections (75.7%), the best $\text{SE}(2)$ template was $C_{\log \cdot \text{SE}(2)}$ (94.0%).

The best combination of templates was $D_{\text{lin} \cdot \mathbb{R}^2}$ with $E_{\text{lin} \cdot \text{SE}(2)}$ (95.6 percent). Without template training (i.e., using average templates $A$) the performance was only 68.2 percent. Success rates using the best template combination are given in Figs. 5d and 5e. The processing time for detecting both pupils simultaneously was on average 0.4 seconds per image.

**State of the Art.** In Fig. 5d we compared our approach to the two most recent pupil detection methods from literature for several normalized error thresholds. Here we see that with allowed errors of 0.1 (blue circles Fig. 5a) and higher our method competes very well with the state of the art, despite the fact that our generic method is not adapted to the application. Further application specific tuning and preprocessing could be applied to improve precision (for $e \leq 0.1$), but this is beyond the scope of this article. Moreover, we see that our method can be used on full images instead of the periocular images without much loss in performance. The fact that our method is still very accurate on full image processing shows that it can be used as a preprocessing step for other applications.

If Fig. 5e we compared our approach to the seven most recent methods from literature (sorted from old to new). Here we see that the only method outperforming our method, at standard accuracy requirements ($e \leq 0.1$), is the method by Markus et al. [57]. Even when considering processing of the full images the only other method that outperforms ours is the method by Timm et al. [61], whose performance is measured using periocular images.

### 4.5 General Observations

The application of our method to the three problems (ONH, fovea and pupil detection) showed the following:
1) State-of-the-art performance was achieved on three different applications, using a single (generic) detection framework and without application specific parameter adaptations.

2) Cross correlation based template matching via data representations on SE(2) improves results over standard $\mathbb{R}^2$ filtering.

3) Trained templates, obtained using energy functionals of the form (1), often perform better than basic average templates. In particular in pupil detection the optimization of templates proved to be essential.

4) Our newly introduced logistic regression approach leads to improved results in pupil detection via single templates. When combining templates we observe only a small improvement of choosing logistic regression (instead of linear regression) for the application of ONH and fovea detection.

5) Regularization in both linear and logistic regression is important. Here both ridge and smoothing regularization priors have complementary benefits.

6) Our method does not rely on any other detection systems (such as ONH detection in the fovea application, or face detection in the pupil detection), and still performs well compared to methods that do.

7) Our method is fast and parallelizable as it is based on inner products, as such it could be efficiently implemented using convolutions.

5 Conclusion

In this paper we have presented an efficient cross-correlation based template matching scheme for the detection of combined orientation and blob patterns. Furthermore, we have provided a generalized regression framework for the construction of templates. The method relies on data representations in orientation scores, which are functions on the Lie group SE(2), and we have provided the tools for proper smoothing priors via resolvent hypo-elliptic diffusion processes on SE(2) (solving time-integrated hypo-elliptic Brownian motions on SE(2)). The strength of the method was demonstrated with two applications in retinal image analysis (the detection of the optic nerve head, and the detection of the fovea) and additional experiments for pupil detection in regular camera images. In the retinal applications we achieved state-of-the-art results with an average detection rate of 99.83 percent on 1,737 images for ONH detection, and 99.32 percent on 1,616 images for fovea detection. Also on pupil detection we obtained state-of-the-art performance with a 95.86 percent success rate on 1,521 images. We showed that the success of the method is due to the inclusion of both intensity and orientation features in template matching. The method is also computationally efficient as it is entirely based on a sequence of convolutions (which can be efficiently done using fast Fourier transforms). These convolutions are parallelizable, which can further speed up our already fast experimental Mathematica implementations that are publicly available at http://erikbekkers.bitbucket.org/TMSE2.html. In future work we plan to investigate the applicability of smoothing on SE(2) in variational settings, as this could also be used in (sparse) line enhancement and segmentation problems.

ACKNOWLEDGMENTS

The authors would like to thank: the groups that kindly made available the benchmark datasets and annotations; Gonzalo Sanguinetti (TU/e) for fruitful discussions and feedback on this manuscript; and the anonymous reviewers for their very valuable suggestions and comments. The research leading to the results of this article has received funding from the European Research Council under the European Community’s 7th Framework Programme (FP7/2007-2014)/ERC grant agreement No. 335555. This work is also part of the Hé Programme of Innovation Cooperation, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

REFERENCES


Marco Loog received the MSc degree in mathematics from Utrecht University and the PhD degree from the Image Sciences Institute, The Netherlands. After this latter, joyful event, he moved to Copenhagen where he acted as assistant and, eventually, associate professor, next to which he worked as a research scientist with Nordic Bioscience. Following several splendid years in Denmark, he moved to Delft University of Technology where he now works as an assistant professor in the Pattern Recognition Laboratory. Currently, he is also honorary professor in pattern recognition with the University of Copenhagen and chairman of Technical Committee 1 of the IAPR. His principal research interests is with supervised pattern recognition in all sorts of shapes and sizes, but with a mild bias towards learning problems coming from image processing and analysis.

Bart M. ter Haar Romeny received the MSc degree in applied physics from Delft University of Technology and the PhD degree from Utrecht University. He is professor in biomedical image analysis in the Department of Biomedical Engineering, Eindhoven University of Technology, The Netherlands, and distinguished professor with Northeastern University, Shenyang, China. His main interest is in brain-inspired medical image analysis, and models of visual perception. He is author of more than 200 refereed journal and conference papers, 12 books and book chapters, and holds two patents. He is project leader of the Sino-Dutch RetinaCheck project. He is a senior member of the IEEE, associate member of the Chinese Brainnec tome consortium, EAMBES Fellow, Governing Board member of IAPR, and chairman of the Dutch Society for Pattern Recognition and Image Processing.

Remco Duits received the MSc (cum laude) degree in mathematics from the TU/e, The Netherlands, in 2001 and the PhD (cum laude) degree from the Department of Biomedical Engineering, TU/e. Now, he is associate professor in the Department of Applied Mathematics & Computer Science and affiliated part-time in the Biomedical Image Analysis Group, Dept. of Biomedical Engineering. He has received an ERC-StG 2013 personal grant (Lie analysis, no. 335555). His research interests subtends group theory, functional analysis, differential geometry, PDEs, harmonic analysis, geometric control and their applications to biomedical imaging and vision.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.