December 2011

UILU-ENG-11-2210
DC-253

SEEING THROUGH THE BLUR

Hossein Mobahi, Yi Ma, and Larry Zitnick

Coordinated Science Laboratory
1308 West Main Street, Urbana, IL 61801
University of Illinois at Urbana-Champaign

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 2011

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
Seeing Through the Blur

5. FUNDING NUMBERS
NSF IIS 11-16012

6. AUTHOR(S)

Hossein Mobahi, Yi Ma, and Larry Zitnick

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 West Main Street
Urbana, Illinois 61801-2307

8. PERFORMING RGANIZATION REPORT NUMBER
UILU-ENG-11-2210
DC-253

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

NSF, 4201 Wilson Blvd, Arlington, VA 22203

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official position, policy, or decision, unless so designated by other documentation

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This paper addresses the problem of image alignment using models such as affine and homography and by directly using pixel intensity values. Coarse-to-fine scheme has become a standard for direct intensity-based alignment. It is believed that such coarse-to-fine scale sampling (Gaussian blur) can improve region of convergence of the alignment optimization. Although, it has been proposed that such isotropic blur may not be optimal for some motion models, no rigorous derivation for such kernels has been known to date. In this work, we derive kernels for some of the common motion models such as affine and homography, which are able to smooth the alignment objective function. This is appealing because the smoothing process often removes poor local minima and thus reaches deeper solutions. Our derivation shows that these kernels coincide with Gaussian blur of the image only for displacement motion.

14. SUBJECT TERMS
1. Image alignment; 2. Homography; 3. Affine

15. NUMBER OF PAGES
21

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Seeing through the Blur

Hossein Mobahi[1], Yi Ma[2,3], and Larry Zitnick[4]

[1]CS Dept., University of Illinois at Urbana-Champaign, Urbana, IL
[2]ECE Dept., University of Illinois at Urbana-Champaign, Urbana, IL
[3]Visual Computing Group, Microsoft Research Asia, Beijing
[4]Interactive Visual Media Group, Microsoft Research, Redmond, WA
{hmobahi2,mayi}@illinois.edu, larryz@microsoft.com

Technical Report
Coordinated Science Laboratory (CSL)
University of Illinois at Urbana Champaign

December 18, 2011

Abstract

This paper addresses the problem of image alignment using models such as affine and homography and by directly using pixel intensity values. Coarse-to-fine scheme has become a standard for direct intensity-based alignment. It is believed that such coarse-to-fine scale sampling (Gaussian blur) can improve region of convergence of the alignment optimization. Although, it has been proposed that such isotropic blur may not be optimal for some motion models, no rigorous derivation for such kernels has been known to date. In this work, we derive kernels for some of the common motion models such as affine and homography, which are able to smooth the alignment objective function. This is appealing because the smoothing process often removes poor local minima and thus reaches deeper solutions. Our derivation shows that these kernels coincide with Gaussian blur of the image only for displacement motion.

1 Introduction

It is one of the most fundamental problems in computer vision to establish a correct alignment (or matching, or correspondence) of the same object between two (or multiple) images. This task is crucial for many important problems such as structure from motion, recognizing an object from different viewpoints, and tracking objects in videos. Roughly speaking, mainstream image alignment techniques can be categorized into "intensity-based" and "feature-based" methods. Intensity-based methods use dense pixel information (such as brightness pattern or correlation) integrated from image regions to estimate the geometric transformation [10]. In contrast, feature-based methods first extract a sparse set of local features from individual images, and then establish correspondence among them to infer the underlying transformation (for larger regions) [14].

In many applications intensity-based methods may seem appealing due to their direct access to richer information (i.e. to every single pixel) as well as other interesting properties [10]. However, their practical performance can be undermined by the associated optimization challenge [23]. Since the intensity pattern over a relatively large region can significantly change across images due to many nuisance factors such as domain transformation (i.e. due to change of viewpoint) and contrast (due to change of lighting), a cost function that directly compares image intensities usually contains many local minima. Thus, unless very good initialization is provided, plain direct alignment of image intensity may lead to poor results.

Lucas and Kanade made a major improvement to optimization for the direct intensity-based method by adopting a coarse-to-fine scheme [15]. For instance, to establish a matching between two images $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ differing by a displacement $\boldsymbol{\theta}$, one needs to minimize the function: $\int_{\mathcal{X}}(f_1(\boldsymbol{x}+\boldsymbol{\theta}) - f_2(\boldsymbol{x}))^2 d\boldsymbol{x}$. According to Lucas and Kanade, this can be more effectively done by approximating $f_1(\boldsymbol{x}+\boldsymbol{\theta})$ up to its first order term $f_1(\boldsymbol{x}+\boldsymbol{\theta}) \approx f_1(\boldsymbol{x}) + \nabla f_1(\boldsymbol{x})\delta\boldsymbol{\theta}$, and minimizing the resulted *convex* quadratic form. One of the key observations is that in order for such a linear approximation to be accurate enough at each iteration, two things help greatly: 1. run the iterations progressively with coarse-to-fine spatial scales to ensure the displacement $\boldsymbol{\theta}$ at each iteration is small; 2. smooth the image intensity functions f_1 and f_2 progressively with a coarse-to-fine (isotropic Gaussian) blur so that their higher-order terms (say Hessians) are small, hence negligible. So, the displacement $\hat{\boldsymbol{\theta}}$ estimated from the linearly approximated objective function at a coarse level gives a rough estimate of the actual displacement, and can be used as a good initialization for the next iteration at a finer scale.

In fact, it is later shown that such a coarse-to-fine scheme is guaranteed to recover the optimal *displacement* under some mild conditions (see [12] for a proof[1]). Although guarantee of correctness is only rigorously established for the translational motion, the notion of coarse-to-fine smoothing and linear approximation has been adopted in computer vision to matching with almost all parametric transformation models (for instance, the work of [1, 15, 21, 25]). In fact, the variants of the underlying concept (of

[1] Although the proof in [12] addresses noise free setting $\exists\boldsymbol{\theta}^*; f_1(\boldsymbol{x}+\boldsymbol{\theta}) = f_2(\boldsymbol{x})$, their presented analysis can be adapted to the noisy case as well.

local analysis combined with gradual smoothing) have been widely used in different disciplines for a long time, although often under different names such as graduated optimization [5], homotopy continuation [22], deterministic annealing [19], diffusion equation method [18], etc.

Despite its popularity, there are serious theoretical and practical issues with the Lucas-Kanade scheme when arbitrarily applied to non-displacement motions. For example, if the transformation is instead a scaling (say by $\theta : \boldsymbol{x} \to \theta \boldsymbol{x}$), then it is easy to show that in general the Hessian of the image function $f(\theta \boldsymbol{x})$ w.r.t. θ grows proportionally to $\|\boldsymbol{x}\|^2$ (as opposed to being constant in case of displacement). Its magnitude varies significantly depending on the location $\boldsymbol{x}$ and hence it is questionable whether a uniform and isotropic Gaussian blur remains a good choice when trying to confine the magnitude of Hessian across the image.

One reason that Gaussian convolution has been so popular for creating blurred images is the rich results in *scale-space theory*[2] [11, 27]. A very interesting result of this theory asserts that Gaussian convolution is the "unique" linear operator that satisfies a minimal set of unbiased axioms [13]. In particular, these axioms require unbiasedness to location (i.e. shift-invariance), which constrains the linear operator to be a "convolution". Scale-space theorist in [8] state that axiom is "motivated by the notion that there is no a priori preferred point within the observable's field of view". Although looking consistent with intuition, it is not the way our eyes has evolved to process. In fact, the density of color receptors in the eye is highest at the center (fovea) and gradually decreases with distance from there. That gives the eye the highest acuity at the center of the view and makes its sight progressively more blurry toward the periphery [17].

The deficiency of isotropic Gaussian convolution for image blurring in the context of non-displacement motion has been noted before. Berg and Malik [3] introduced the notion of "geometric blur" of an image defined as $g(\boldsymbol{x}) = \int_{\Theta} w(\boldsymbol{\theta}) f(\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{\theta}))\, d\boldsymbol{\theta}$, where $\boldsymbol{\tau} : \mathcal{X} \times \Theta \to \mathcal{X}$ is some family of transformations parameterized by $\boldsymbol{\theta}$ and $w(\boldsymbol{\theta})$ is some weight function. Although this definition of geometric blur is very general and has some overlap with the goal of blurring for image alignment purpose, it is practically useful only when it can be written in the form of a kernel. In [3], a heuristically chosen family of kernels is suggested but without clear connection to common transformation models used in image alignment such as affine or homography.

In this paper, we provide a thorough analysis on how to smooth objective functions that arise in image alignment. In particular, we study the question that if the goal is to smooth the objective function against its unknown transformation parameters, what is the correct associated blurring we need to perform on the images. We call the kernel functions associated with such image blurring as the *transformation kernels*; and we derive closed-form solutions for such kernels associated with all popular transformation groups in image alignment, including translations, translation+scaling, affine, and homography. As we will see, all kernels are spatially varying as long as the transformation is not purely translational. In addition, the transformation blurring kernels derived in this paper do not fall into the class of kernels suggested by [3] or [24]. We will

[2] The idea of Gaussian smoothing in vision is even older than scale-space theory. For example, Marr and Hildreth [16] studied zero-crossings the Laplacian in images convolved with Gaussian kernels at different scales.

use the proposed kernels within a simple *deterministic* optimization algorithm. As the heuristics used in geometric blur [3] and DAISY [24] have been motivated largely for fast computation, we believe our results could open up new investigations on new class of fast geometric blurring kernels that can efficiently approximate the transformation kernels. Furthermore, the deterministic nature of the optimization algorithm provides a new opportunity for performance evaluation indices that are *repeatable*[3].

We emphasize that the goal of this work is to improve our understanding of the optimization problems associated with intensity-based image alignment, so as to improve its practical performance and effectiveness. We do not advocate that direct intensity-based alignment is better or worse than feature-based methods. In many practical situations, the feature-based and intensity-based methods can be easily engineered to complement each other.

2 Motivation

Here we present an illustrative example to explain the difference between Gaussian smoothing of the objective function versus Gaussian smoothing of the image itself. Consider a simple shape consisted of three concentric rings (figure 1-a). The goal is to align this image with a scaled version of it (figure 1-b). The only parameter subject to optimization is the scale factor. Formally, the goal is to seek θ such that $f_1(\theta \boldsymbol{x}) = f_2(\boldsymbol{x})$ holds.

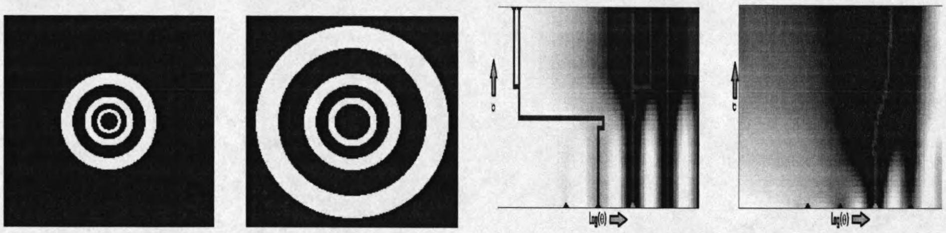
We cast this problem as optimization by trying to maximize the correlation $\int_{\mathcal{X}} f_1(\theta \boldsymbol{x}) f_2(\boldsymbol{x}) d\boldsymbol{x}$. We explore two ways of applying the smoothing procedure to this optimization. The first one, directly convolves the images with a Gaussian kernel whose bandwidth is σ. It then locally optimizes the objective function while gradually reducing the blur and tracing the solution. The other scheme is similar except that it blurs the objective function instead, i.e. convolves it with a Gaussian kernel in variable θ with bandwidth σ.

The associated optimization landscape is visualized (figure 1-c,d). There, the horizontal axis shows the choices of θ in log scale between -2 (left most) to 2 (rightmost). The vertical axis shows the amount of blur (parameter σ) where the top is the highest smoothing and bottom is the no smoothing. The value of the objective function is shown by intensity, black being the highest.

The local maxima at the highest and lowest blur are shown by blue spots. The global maximum at no blur (i.e. the desired solution θ^*) is marked by green. Basin of attractions originating from each local maximum at the highest blur are indicated by red. It can be observed that (figure 1-c) Gaussian blurring does not remove local maxima, even under extensive blurring[4] of the image. In this case, depending on the initial choice of θ at the highest blur phase, the process may trap into a basin of attractions that leads to a local maximum at the lowest blur side. However, when the objective

[3] In contrast, probabilistic schemes for robust fitting such as RANSAC [7] may produce a different answer in each run. The problem may persist even if aggregate a large batch of RANSAC solutions and chooses the bets among them.

[4] The highest level of image blur is created by repeatedly applying a 3×3 Gaussian convolution mask more than 400,000 times to the images



(a) (b) (c) (d)

Figure 1: Basin of attraction for the alignment of images in (a) and (b). The objective function when Gaussian blurring the image is shown in (c) and that of Gaussian blurring of the objective function in (d). Blue, green and red respectively indicate local maxima, global maximum and basin of attraction originating from local maxima of highest blur. Horizontal/Vertical axis show θ and σ in increasing order from left to right and bottom to top (top being the highest blur).

function is blurred, there is only one initial maximum whose basin of attraction leads to the global maximum of the non-blurred function (figure1-c). The work [20] also presents some empirical results on the role of (isotropic) *Gaussian blurring of images* (vs not blurring at all) in enlarging the basin of attractions within a motion estimation task. In contrast, in this work we focus on blurring the objective function instead of the image itself, which will lead to spatially varying and non-Gaussian kernels.

3 Transformation Kernel

3.1 Notation

The symbol $\triangleq$ is used for equality by definition. Also, we use x for scalars, $\boldsymbol{x}$ for vectors, $\boldsymbol{X}$ for matrices, and $\mathcal{X}$ for sets. In addition, $f(.)$ denotes a scalar valued function and $\boldsymbol{f}(.)$ a vector valued function. Unless stated otherwise, $\|\boldsymbol{x}\|$ means $\|\boldsymbol{x}\|_2$ and ∇ means $\nabla_{\boldsymbol{x}}$. Finally, $\star$ and $\circledast$ denote convolution operators in spaces Θ and $\mathcal{X}$ respectively.

3.2 Definitions

Definition [Domain Transformation] Given a function $f : \mathcal{X} \to \mathbb{R}$ and a vector field $\boldsymbol{\tau} : \mathcal{X} \times \Theta \to \mathcal{X}$, where $\mathcal{X} = \mathbb{R}^n$ and $\Theta = \mathbb{R}^m$. We refer to $\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{\theta})$ as the *domain transformation* parameterized by $\boldsymbol{\theta}$. Note that the parameter vector $\boldsymbol{\theta}$ is constructed by concatenation of all the parameters of a transformation. For example, if case of affine $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$ with $\boldsymbol{x} \in \mathbb{R}^2$, $\boldsymbol{\theta}$ is a 6 dimensional vectors containing the elements of $\boldsymbol{A}$ and $\boldsymbol{b}$.

Definition [Isotropic Gaussian]

$$k(\boldsymbol{x};\sigma^2) \triangleq \frac{1}{(\sqrt{2\pi}\sigma)^{\dim(\boldsymbol{x})}} e^{-\frac{\|\boldsymbol{x}\|^2}{2\sigma^2}}\,. \tag{1}$$

Definition [Anisotropic Gaussian]

$$K(\boldsymbol{x};\boldsymbol{\Sigma}) \triangleq \frac{1}{(\sqrt{2\pi})^{\dim(\boldsymbol{x})}\sqrt{\det(\boldsymbol{\Sigma})}} e^{-\frac{\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}}{2}}\,. \tag{2}$$

Definition [Fourier Transform]

We use the following convention for Fourier transform. The Fourier transform of a real valued function $f : \mathbb{R}^n \to \mathbb{R}$ is $\hat{f}(\boldsymbol{\omega}) = \int_{\mathbb{R}^n} f(\boldsymbol{x}) e^{-i\boldsymbol{\omega}^T\boldsymbol{x}} d\boldsymbol{x}$ and the inverse Fourier transform is $\hat{f}(\boldsymbol{x}) = (2\pi)^{-n}\int_{\mathbb{R}^n} f(\boldsymbol{\omega}) e^{i\boldsymbol{\omega}^T\boldsymbol{x}} d\boldsymbol{\omega}$.

Definition [Transformation Kernel]

Given a domain transformation $\boldsymbol{\tau} : \mathcal{X} \times \Theta \to \mathcal{X}$, where $\mathcal{X} = \mathbb{R}^n$ and $\Theta = \mathbb{R}^m$. We define a *transformation kernel* associated with $\boldsymbol{\tau}$ as $u_{\boldsymbol{\tau},\sigma} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that it satisfies the following *integral equation*,

$$\begin{aligned} &\forall f \quad : \\ &[f(\boldsymbol{\tau}(\boldsymbol{x},\cdot)) \star k(\cdot;\sigma^2)]\,(\boldsymbol{\theta}) = \int_{\mathcal{X}} f(\boldsymbol{y})\, u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{y}\,, \end{aligned} \tag{3}$$

where f is assumed to be a Schwartz function. Therefore, any transformation kernel that satisfies this equation allows the convolution of the transformed signal with the Gaussian kernel be equivalently written by the *integral transform* of the non-transformed signal with the kernel $u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y})$.

3.3 Derivation of Kernels by Fourier Transform

Proposition 1 *The following choice of u,*

$$\begin{aligned} &u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y}) \\ &= \frac{1}{(2\pi)^n}\int_{\Omega}\Big(\int_{\Theta} e^{i\boldsymbol{\omega}^T(\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{t})-\boldsymbol{y})} k(\boldsymbol{t}-\boldsymbol{\theta};\sigma^2)\,d\boldsymbol{t}\Big)\,d\boldsymbol{\omega} \end{aligned} \tag{4}$$

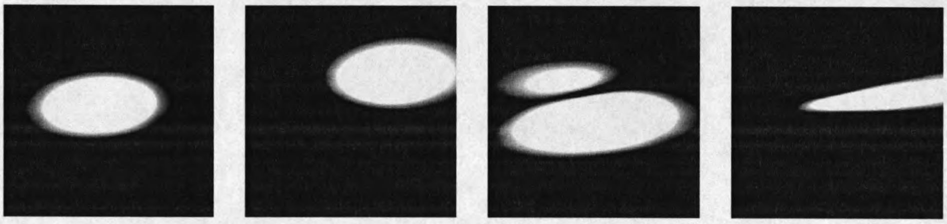
is a solution to the definition of kernel provided in (3). Here $\mathcal{X} = \Omega = \mathbb{R}^n$, and $k(\boldsymbol{t};\sigma^2)$ is some function $k(\,.\,;\sigma) : \mathcal{X} \to \mathbb{R}$ with some parameter σ, which in our case is simply an isotropic Gaussian with bandwidth σ.

The proof follows by using the Fourier representation $f(\boldsymbol{x}) = (2\pi)^{-n}\int_{\Omega}\hat{f}(\boldsymbol{\omega})e^{i\boldsymbol{\omega}^T\boldsymbol{x}}\,d\boldsymbol{\omega}$, where $\Omega = \mathbb{R}^n$ (similar to $\mathcal{X} = \mathbb{R}^n$), and then application of Parseval's theorem. See the appendix for details.

Now by plugging in the desired transformation $\boldsymbol{\tau}$ into the result of proposition 1, we can compute the integrals[5] and derive the corresponding kernel function as shown in Table 1. Note that the functions p and e, associated with the homography kernel,

Name	$\boldsymbol{\theta}$	$\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{\theta})$	$u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y})$
Translation	$\boldsymbol{d}_{n\times 1}$	$\boldsymbol{x}+\boldsymbol{d}$	$k(\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{\theta})-\boldsymbol{y};\sigma^2)$
Translation+Scale	$[\,\boldsymbol{a}_{n\times 1}\,,\,\boldsymbol{d}_{n\times 1}\,]$	$\boldsymbol{a}^T\boldsymbol{x}+\boldsymbol{d}$	$K(\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{\theta})-\boldsymbol{y};\sigma^2\,\mathrm{diag}([1+x_i^2]))$
Affine	$[\,vec(\boldsymbol{A}_{n\times n})\,,\,\boldsymbol{b}_{n\times 1}\,]$	$\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}$	$k(\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{\theta})-\boldsymbol{y};\sigma^2(1+\Vert\boldsymbol{x}\Vert^2))$
Homography	$[\,vec(\boldsymbol{A}_{n\times n})\,,\,\boldsymbol{b}_{n\times 1}\,,\,\boldsymbol{c}_{n\times 1}\,]$	$\frac{1}{1+\boldsymbol{c}^T\boldsymbol{x}}(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b})$	$n=2: p(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y},\sigma)\ e(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y},\sigma)$

Table 1: Kernels for some of the common transformations arising in vision.



(a) (b) (c) (d)

Figure 2: Visualization of affine and homography kernels on the plane $\boldsymbol{x} \in [-1,1] \times [-1,1]$ with $\sigma = 0.5$. Both kernels are realized by $\boldsymbol{A}_0 = [2 \quad 0.2\,; -0.3 \quad 4], \boldsymbol{b}_0 = [0.15 \quad -0.25]$. In addition, for the homography we have $\boldsymbol{c}_0 = [1 \quad -5]$. The figure shows the contribution of the center and top right corner points, each shown separately. More precisely, it depicts (a) $u_{\text{affine},0.5}(\boldsymbol{\theta} = \boldsymbol{\theta}_0, \boldsymbol{x}, \boldsymbol{y} = (0,0))$ (b) $u_{\text{affine},0.5}(\boldsymbol{\theta} = \boldsymbol{\theta}_0, \boldsymbol{x}, \boldsymbol{y} = (1,1))$ (c) $u_{\text{homography},0.5}(\boldsymbol{\theta} = \boldsymbol{\theta}_0, \boldsymbol{x}, \boldsymbol{y} = (0,0))$ (d) $u_{\text{homography},0.5}(\boldsymbol{\theta} = \boldsymbol{\theta}_0, \boldsymbol{x}, \boldsymbol{y} = (1,1))$.

are respectively some polynomial (in $\boldsymbol{\theta}$) and exponential functions, whose complete expressions are provided in the appendix.

The derivation of the kernels are, although elementary, somewhat messy and long (specially for the homography). Rather than going through the complete derivation process, we provide a simple way to check the correctness of the kernels by the interested reader. This is achieved via the following two necessary conditions that must hold for the kernels.

1. **Heat Equation**: Consider the convolution $[f(\boldsymbol{\tau}(\boldsymbol{x},\,\cdot\,)) \overset{\ominus}{\circledast} k(\,\cdot\,;\sigma)](\boldsymbol{\theta})$. Since k is the Gaussian kernel, the convolution result necessarily obeys the *heat equation* [26].

$$\begin{aligned} & \sigma\Delta_{\boldsymbol{\theta}}[f(\boldsymbol{\tau}(\boldsymbol{x},\,\cdot\,)) \star k(\,\cdot\,;\sigma)](\boldsymbol{\theta}) \\ = \quad & (\partial/\partial\sigma)[f(\boldsymbol{\tau}(\boldsymbol{x},\,\cdot\,)) \star k(\,\cdot\,;\sigma)](\boldsymbol{\theta})\,. \end{aligned} \tag{5}$$

   Since we argue that $[f(\boldsymbol{\tau}(\boldsymbol{x},\,\cdot\,)) \star k(\,\cdot\,;\sigma)](\boldsymbol{\theta}) = \int_{\mathcal{X}} f(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y})\,d\boldsymbol{y}$, the following must hold.

[5] Although the integral in (4) does not necessarily have a "closed-form" for any arbitrary transformation $\boldsymbol{\tau}$, it does so for most of the transformations we care about in practice, as listed in Table 1.

$$
\begin{aligned}
& \sigma\Delta_{\boldsymbol{\theta}} \int_{\mathcal{X}} f(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \\
= \quad & \frac{\partial}{\partial\sigma} \int_{\mathcal{X}} f(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \qquad (6) \\
\equiv \quad & \int_{\mathcal{X}} f(\boldsymbol{y}) \sigma\Delta_{\boldsymbol{\theta}} u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \\
= \quad & \int_{\mathcal{X}} f(\boldsymbol{y}) \frac{\partial}{\partial\sigma} u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \qquad (7) \\
\Leftarrow \quad & \sigma\Delta_{\boldsymbol{\theta}} u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) \\
= \quad & \frac{\partial}{\partial\sigma} u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\,, \qquad (8)
\end{aligned}
$$

where $\Leftarrow$ in (8) means sufficient condition. Now it is much easier to check the identity (8) for the provided kernels. For example, in case of affine kernel $k(\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{\theta}) - \boldsymbol{y}; \sigma^2(1 + \|\boldsymbol{x}\|^2))$, both sides of the identity are equal to $(\frac{\|\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{\theta}) - \boldsymbol{y}\|^2}{\sigma^3(1+\|\boldsymbol{x}\|^2)} - \frac{n}{\sigma})\; k(\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{\theta}) - \boldsymbol{y}; \sigma^2(1 + \|\boldsymbol{x}\|^2))$.

2. **Limit Behavior**: When the amount of smoothing approaches zero, the integral transform must approach the original function that is transformed by $\boldsymbol{\tau}$. Formally, we want the following identity to hold.

$$
\lim_{\sigma \to 0^+} \int_{\mathcal{X}} f(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} = f(\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{\theta}))\,. \qquad (9)
$$

Sufficient condition for the above identity is that $\lim_{\sigma \to 0^+} u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) = \delta(\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{\theta}) - \boldsymbol{y})$, where δ is Dirac's delta function.

This is trivial for the kernels of affine and its subgroups; since the kernel itself is a Gaussian, $\lim_{\sigma \to 0^+}$ is equivalent to kernel's variance approaching to zero (for any bounded choice of $\|\boldsymbol{x}\|$). It is known that when the variance of the normal density function tends to zero it approaches Dirac's delta function.

Two interesting observations can be made about Table 1. The first is that, from a *purely objective* standpoint, the derived kernels exhibit "foveation", similar to that in the eye. It is known that the population of color receptors has the highest density at the center of our retina, i.e. the fovea. This number gradually decreases with distance from the fovea. Consequently, the eye has the highest acuity in the center and sees progressively more blurry toward the periphery [17]. Similarly, except translation, all the kernels are spatially varying with density decreasing in $\|\boldsymbol{x}\|$. This is very easy to check for translation+scale and affine kernels, where they are like a Gaussian distribution whose variance depends and increases in $\|\boldsymbol{x}\|$.

The second point is that the derived kernels are not necessarily rotation invariant. Therefore, the geometric blur kernel proposed by Berg and Malik [3], which requires rotational symmetry in $\boldsymbol{x}$ and $\boldsymbol{y}$ is unable to represent the basic transformations arising in vision, such as those in Table 1. In fact, to the best of our knowledge, this work is the first that derives exact kernels for such transformations.

Algorithm 1 Alignment by Gaussian Continuation.

```
Input: $f_1 : \mathcal{X} \to \mathbb{R}$, $f_2 : \mathcal{X} \to \mathbb{R}$, $\boldsymbol{\theta}_0 \in \Theta$, $\{\sigma_k\}$ for $k = 1, \dots, K$ s.t. $0 < \sigma_{k+1} < \sigma_k$
for $k = 1 \to K$ do
    $\boldsymbol{\theta}_k =$ Local maximizer of $z(\boldsymbol{\theta}; \sigma_k)$, initialized at $\boldsymbol{\theta}_{k-1}$
end for
Output: $\boldsymbol{\theta}_K$
```

4 Smoothed Alignment Objective

We use the inner product between the transformed signal f_1 and the reference signal f_2 as the alignment objective function. Note that f_1 and f_2 are the input to the alignment algorithm, and could possibly be different from the original images. For example, they could have been mean subtracted or normalized by their ℓ_2 norm. The alignment objective function is denoted by $h(\boldsymbol{\theta})$ and defined as follows.

$$h(\boldsymbol{\theta}) \triangleq \int_{\mathcal{X}} f_1(\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{\theta}))\, f_2(\boldsymbol{x})\, d\boldsymbol{x}\,. \tag{10}$$

The optimal alignment $\boldsymbol{\theta}^*$ is then the choice of $\boldsymbol{\theta}$ which maximizes the alignment objective function. Instead of directly solving the optimization on h, we resort to a coarse-to-fine smoothing scheme. Thus, we introduce the "smoothed" alignment objective z as the following.

$$\begin{aligned}
& z(\boldsymbol{\theta}, \sigma) && (11)\\
\triangleq\ & [h \star k(\,\cdot\,, \sigma^2)](\boldsymbol{\theta}) && (12)\\
=\ & \int_{\mathcal{X}} \big(f_2(\boldsymbol{x})[f_1(\boldsymbol{\tau}(\boldsymbol{x}, .)) \star k(\,\cdot\,, \sigma^2)](\boldsymbol{\theta})\big)\, d\boldsymbol{x} && (13)\\
=\ & \int_{\mathcal{X}} \Big(f_2(\boldsymbol{x}) \Big(\int_{\mathcal{X}} f(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y}\Big)\Big)\, d\boldsymbol{x}\,, && (14)
\end{aligned}$$

where integral transform in (14) uses the definition of kernel provided in (3). We can now develop a simple iterative algorithm for coarse-to-fine alignment (algorithm 1) relying on (14) for computation of $z(\boldsymbol{\theta}, \sigma)$. Detailed numerical procedure for computing the integral transform in (14) will be provided in section 5.

4.1 Image Blurring vs. Objective Blurring

Plugging the translation kernel from Table 1 into smoothed objective function (14) yields the following.

$$\begin{aligned}
&z(\boldsymbol{\theta}, \sigma) && (15)\\
&= \int_{\mathcal{X}} \left(f_2(\boldsymbol{x}) \int_{\mathcal{X}} f_1(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \right) d\boldsymbol{x} && (16)\\
&= \int_{\mathcal{X}} \left(f_2(\boldsymbol{x}) \int_{\mathcal{X}} f_1(\boldsymbol{y}) k(\boldsymbol{\theta} + \boldsymbol{x} - \boldsymbol{y}; \sigma^2)\, d\boldsymbol{y} \right) d\boldsymbol{x} && (17)\\
&= \int_{\mathcal{X}} \big(f_2(\boldsymbol{x})\ [f_1(\,\cdot\,) \circledast k(\,\cdot\,; \sigma^2)]\, (\boldsymbol{\theta} + \boldsymbol{x}) \big)\, d\boldsymbol{x}\,. && (18)
\end{aligned}$$

It can be observed that, *for "translation transformation", Gaussian convolution of the alignment objective with respect to the optimization variables is equivalent to the "Gaussian blurring" of the signal and then shifting it by $\boldsymbol{\theta}$.*

However, this argument does not hold for other transformations. For example, the smoothed f_1 using the affine kernel is the below.

$$\begin{aligned}
&[f_1(\boldsymbol{\tau}(\boldsymbol{x}, \cdot)) \star k(\,\cdot\,, \sigma^2)](\boldsymbol{\theta}) && (19)\\
&= \int_{\mathcal{X}} f_1(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} && (20)\\
&= \frac{1}{(\sigma\sqrt{2\pi(1 + \|\boldsymbol{x}\|^2)})^n} \int_{\mathcal{X}} f_1(\boldsymbol{y}) e^{-\frac{\|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} - \boldsymbol{y}\|^2}{2\sigma^2(1 + \|\boldsymbol{x}\|^2)}}\, d\boldsymbol{y}\,. && (21)
\end{aligned}$$

Therefore, *for "affine transformation", Gaussian convolution of the alignment objective with respect to the optimization variables is equivalent to an "integral transform" (which is not a convolution) of the signal.*

5 Computation of the Integral Transform

There are two computational advantages when using kernels to compute the smoothed objective function (11).

1. Observe that the kernel u for affine and its subgroups (table 1) are Gaussian forms in variable $\boldsymbol{y}$. In this case, expressing f_1 by some Gaussian Basis Functions[6] will lead to a closed form for the integral transform. More details are provided in section 5.1.

2. For kernels of affine and its subgroups (Table 1), one may also choose to represent f_1 by piecewise constant or piecewise polynomial forms. Again, since the kernels are Gaussian forms in variable $\boldsymbol{y}$, derivation of a closed form for the integral is possible. See section 5.2 for details.

[6] A GBF is a function of form $\Phi(\boldsymbol{x}; \boldsymbol{x}_0, \boldsymbol{\Delta}_0) = e^{-\frac{(\boldsymbol{x}-\boldsymbol{x}_0)^T \boldsymbol{\Delta}_0{}^{-1} (\boldsymbol{x}-\boldsymbol{x}_0)}{2}}$, where the matrix $\boldsymbol{\Delta}$ must be positive definite. It is known that Gaussian RBFs $\phi(\boldsymbol{x}; \boldsymbol{x}_0, \delta_0) = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x}_0\|^2}{2\delta_0^2}}$, which are a special case of GBFs, are *general function approximators*.

3. If the kernel u is not Gaussian in $\boldsymbol{y}$ (such as in homography), the derivation of a closed form for the integral transform may not be possible. However, numerical integration is done much more efficiently using the kernelized form (14) compared to the original form (11). More precisely, when $n = 2$, integration in the original form is over $\boldsymbol{\theta}$ and for homography $\dim(\boldsymbol{\theta}) = 8$. However, the equivalent integral transform is over $\boldsymbol{y}$, where $\dim(\boldsymbol{y}) = 2$.

5.1 Gaussian RBF Representation of f_1

Using GBFs to represent f_1, whenever the transformation kernel $u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$ is Gaussian in $\boldsymbol{y}$, then the integral transform (14) has closed form. By a kernel being Gaussian in $\boldsymbol{y}$, we mean it can be written as $u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) = k(\boldsymbol{\tau}(\boldsymbol{\theta}, \boldsymbol{x}) - \boldsymbol{y}\,;\, s^2(\boldsymbol{\theta}, \boldsymbol{x}))$, $s : \Theta \times \mathcal{X} \to \mathbb{R}_+$ is an arbitrary maps. Notice that the maps $\boldsymbol{\tau}$ and s must be *independent* of $\boldsymbol{y}$. For brevity, we drop the arguments of the maps $\boldsymbol{\tau}$ and $\boldsymbol{s}$ in the following derivation.

Here we obtain the closed form of the integral transform when f_1 is represented by Gaussian Radial Basis Functions (GRBFs) $\phi(\boldsymbol{x}; \boldsymbol{x}_0, \delta_0) = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x}_0\|^2}{2\delta_0^2}}$; the more general case of GBFs can be obtained in a similar fashion.

Proposition 2 *Suppose $f_1 = \sum_{k=1}^{p} a_k \phi(\boldsymbol{y}; \boldsymbol{x}_k, \delta_k)$, where $\phi(\boldsymbol{x}; \boldsymbol{x}_k, \delta_k) = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x}_k\|^2}{2\delta_k^2}}$. Assume that $u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$ is Gaussian in variable $\boldsymbol{y}$. Then the following identity holds.*

$$\int_{\mathcal{X}} f_1(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \tag{22}$$

$$= \sum_{i=1}^{p} a_i \Big(\frac{\delta_i}{\sqrt{\delta_i^2 + s^2}}\Big)^n e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{\tau}\|^2}{2(\delta_i^2 + s^2)}} \,. \tag{23}$$

See the appendix for a proof.

5.2 Piecewise Constant Representation of f_1

Here we provide a result for when the covariance of the Gaussian is diagonal, which includes isotropic Gaussians k. However, extension to non-diagonal covariances is possible.

Proposition 3 *Suppose $f_1(\boldsymbol{x}) = c$ on a rectangular piece $\boldsymbol{x} \in \mathcal{X}^\dagger \triangleq \Pi_{k=1}^{n} [\underline{x_k}, \overline{x_k}]$. Assume that $u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$ is Gaussian in variable $\boldsymbol{y}$ with diagonal covariance structure, i.e. $u = K(\boldsymbol{\tau} - \boldsymbol{y}; \mathrm{diag}(s_1^2, \cdots, s_n^2))$. Then the following identity holds.*

$$\int_{\mathcal{X}^\dagger} f_1(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \tag{24}$$

$$= \prod_{k=1}^{n} \frac{1}{2}\Big(\mathrm{erf}\,\big(\frac{x_k - \overline{x_k}}{\sqrt{2} s_k}\big) - \mathrm{erf}\,\big(\frac{x_k - \underline{s_k}}{\sqrt{2} s_k}\big)\Big). \tag{25}$$

The proof is elementary and uses separability of integrals for diagonal K.

6 Regularization

Excessive smoothing of the objective function brings the risk of numerical instability. Regularization can be used to improve numerical stability of the method. Additionally, regularization *improves well-posedness* of the task. By this we mean if in the non-regularized task, there are multiple transformations (e.g. when image content has *symmetries*) that lead to equally good alignments, the regularization prefers the closest to some given $\boldsymbol{\theta}_0$, making existence of a *unique global optimum* more presumable. We achieve this goal by replacing f_1 with the following regularized version of it.

$$\tilde{f}_1(\boldsymbol{\tau}(\,\cdot\,,\,\cdot\,),\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\theta}_0,r) \triangleq k(\boldsymbol{\theta}-\boldsymbol{\theta}_0;r^2)f_1(\boldsymbol{\tau}(\boldsymbol{x};\boldsymbol{\theta})). \tag{26}$$

This regularization shrinks the signal f_1 at peculiar transformations [7], i.e. those with very large $\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|$, where $\boldsymbol{\theta}_0$ is, on *average*, the most common transformation (typically the identity transformation $\boldsymbol{\tau}(\boldsymbol{x};\boldsymbol{\theta})=\boldsymbol{x}$). In principle, we do not lose much by attenuating the objective function for very large $\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|$, because these transformations are physically unrealizable anyway.

6.1 Regularized Objective Function

Applying transformation and spatial regularization to the objective function yields the following regularized objective function.

$$\begin{aligned}
\tilde{h}(\boldsymbol{\theta};\boldsymbol{\theta}_0,r) &\triangleq \int_{\mathcal{X}} \Big(\tilde{f}_1(\boldsymbol{\tau},\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\theta}_0,r)f_2(\boldsymbol{x})\Big)\,d\boldsymbol{x} \\
&= \int_{\mathcal{X}} k(\boldsymbol{\theta}-\boldsymbol{\theta}_0;r^2)f_1(\boldsymbol{\tau}(\boldsymbol{x};\boldsymbol{\theta}))f_2(\boldsymbol{x})\,d\boldsymbol{x}\,.
\end{aligned}$$

6.2 Smoothed Regularized Objective

We define the smoothed *regularized* objective as the following.

$$\tilde{z}(\boldsymbol{\theta},\boldsymbol{\theta}_0,r,\sigma) \triangleq [\tilde{h}(\,\cdot\,,\boldsymbol{\theta}_0,r)\star k(\,\cdot\,;\sigma^2)](\boldsymbol{\theta}). \tag{27}$$

This form is still amenable to kernel computation using the following proposition.

Proposition 4 *The regularized objective function $\tilde{z}(\boldsymbol{\theta},\boldsymbol{\theta}_0,r,\sigma)$ can be written using transformation kernels as follows.*

[7] One may ask if we could shrink the objective function h, instead of shirking the signal f_1, for peculiar transformations. Although the former may seem more intuitive, it conflicts with the optimization goals. More precisely, since we seek for the maximizer of the objective function, shirking h near the infinite ball may significantly lower the value of the objective function (relative to the objective value in the rest of the domain). At the extreme, this may even move the global maximizer to the infinite ball, no matter how bad the alignment is at that place.

(a) (b) (c) (d)

Figure 3: Representative views –the rectified views– from the dataset provided in [6] (a) colors (b) grace (c) posters (d) underground.

$$
\begin{aligned}
& \tilde{z}(\boldsymbol{\theta}, \boldsymbol{\theta}_0, r, \sigma) && (28)\\
= \; & [\tilde{h}(\,\cdot\,, \boldsymbol{\theta}_0, r) \overset{\ominus}{\circledast} k(\,\cdot\,; \sigma^2)](\boldsymbol{\theta}) && (29)\\
= \; & \int_{\mathcal{X}} \Big(k(\boldsymbol{\theta} - \boldsymbol{\theta}_0; r^2 + \sigma^2) f_2(\boldsymbol{x}) ... && (30)\\
& \cdot \int_{\mathcal{X}} \Big(f_1(\boldsymbol{y}) u_{\boldsymbol{\tau}, \frac{r\sigma}{\sqrt{r^2+\sigma^2}}} \big(\frac{r^2 \boldsymbol{\theta} + \sigma^2 \boldsymbol{\theta}_0}{r^2 + \sigma^2}, \boldsymbol{x}, \boldsymbol{y} \big) \Big) \, d\boldsymbol{y} \Big) d\boldsymbol{x} \, .
\end{aligned}
$$

See the appendix for the proof.

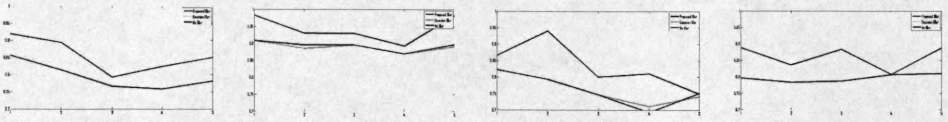
7 Experiments

In this section, we evaluate how the proposed scheme for smoothing the alignment objective function compares against traditional Gaussian blurring and also no blurring at all. We use the dataset provided in [6]. The dataset consists of 5 scenes taken in 6 different views, with progressively more drastic homography effect. We exclude one of the scenes which consists of two planes and not suitable for homography fitting. The rest of the scenes are very close to a single plane, which are used in our evaluation (see figure 3).

For the proposed method, we used the homography kernel due to the nature of the test data. The goal by either of the three methods is to maximize the correlation between the two images, by transforming one to the other. The local maximization in Algorithm 1 as well as in the Gaussian blur and no blur schemes is achieved by gradient ascent method. The integral transform in Algorithm 1 is computed by the rectangle rule of integral approximation, evaluated on a 25×25 grid.

Coordinate of images were normalized to be in range $[-1, 1]$. The intensity of the input images f_1 and f_2 to these algorithms was subtracted by their joint mean (i.e. $(\bar{f}_1 + \bar{f}_2)/2$, where $\bar{f}_1$ is the average intensity value of image f_1). The sequence of decreasing σ used in our experiments (for both the proposed kernel and Gaussian kernel) starts from $\sigma = 2$, and is multiplied by $2/3$ in each iteration of algorithm until it goes below 0.01. The regularization constant r was set to 1.

The performance of these methods is summarized in figure 4. Each plot corresponds to one of the scenes in the dataset. For each scene, there is one rectified view



(a) (b) (c) (d)

Figure 4: NCC value after alignment. Horizontal axis is the view index (increasing in complexity) of the scene. Four views are used for each scene, each one being as f_2 and compared against $\tilde{f}_1$, which is a rectified vew, in the dataset [6] : (a) colors (b) grace (c) posters (d) underground.

that is used as f_1. The rest of 5 views, indexed from 1 to 5, in increasing order of complexity[8] are used as f_2. The vertical axis in the plots indicates the normalized correlation coefficient (NCC) between f_2 and transformed $\tilde{f}_1$. It can clearly be observed that, while Gaussian blur sometimes does a little bit better than no blur, the proposed smoothing scheme leads to much higher NCC value.

8 Conclusion

In this paper we studied the problem image smoothing for the purpose of alignment by direct intensity-based method. We argued that the traditional Gaussian blurring of the image, which is mainly inspired due to the work of Lucas and Kanade [15], may not be suitable for non-displacement alignment tasks. Instead, we suggested directly smoothing the alignment objective function, which is a common practice in the optimization literature [18]. We formally showed that, smoothing the objective function is equivalent to Gaussian blurring of the image only when the transformation is a displacement. We then rigorously derived exact form of spatially varying kernels required for smoothing the objective function of common model-based alignment tasks including affine and homography models. Although the idea of using spatially varying kernels has been used before [3, 24], the derivation has been based on heuristics and intuition. The kernel derived in [3, 24] are, however, very limited to what was shown to be theoretically correct choice. Specifically, the derived kernels are not rotationally symmetric like those in [3, 24]. Yet, such kernels [3, 24] have been successfully utilized in various applications such as face detection [4] and object recognition [2, 9]. This work is the first to objectively derive suitable kernels for some of the common transformation models. Thus, it may open up new research avenues on how such exact kernels can be used and efficiently implemented for vision applications.

[8] Here the complexity of the view is referred to how drastic the homography transformation is, in order to bring it to the rectified view.

Acknowledgement

We would like to thank John Wright of Columbia University and Vadim Zharnitsky of UIUC for their comments and helpful discussions. First author is supported by the generous CSE PhD fellowship of UIUC.

References

[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework: Part 1: The quantity approximated, the warp update rule, and the gradient descent approximation. *International Journal of Computer Vision*, 56:221–255, 2004. 2

[2] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proceedings of CVPR 2005.*, volume 1, pages 26–33, 2005. 14

[3] A. C. Berg and J. Malik. Geometric blur for template matching. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), with CD-ROM, 8-14 December 2001, Kauai, HI, USA*, pages 607–614. IEEE Computer Society, 2001. 3, 4, 8, 14

[4] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y.-W. Teh, E. Learned-Miller, and D. A. Forsyth. Names and faces in the news. In *Proceedings of CVPR 2004.*, volume 2, pages 848–854. IEEE, 2004. 14

[5] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987. 3

[6] K. Cordes, B. Rosenhahn, and J. Ostermann. Increasing the accuracy of feature evaluation benchmarks using differential evolution. In *IEEE Symposium Series on Computational Intelligence (SSCI) - IEEE Symposium on Differential Evolution (SDE)*, apr 2011. 13, 14

[7] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Comm. of the ACM*, 24(8):381–395, 1989. 4

[8] L. Florack, B. M. Haar Romeny, J. J. Koenderink, and M. A. Viergever. On the multiple-minima problem in the conformational analysis of molecules: deformation of the potential energy hypersurface by the diffusion equation method. *Journal of Mathematical Imaging and Vision*, 4(4):325–351, 1994. 3

[9] A. Frome, F. Sha, Y. Singer, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *In ICCV*, 2007. 14

[10] M. Irani and P. Anandan. All about direct methods. 1999. 2

[11] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50(5):363–370–370, Aug. 1984. 3

[12] M. Lefebure and L. D. Cohen. Image registration, optical flow and local rigidity. *Journal of Mathematical Imaging and Vision*, 14:131–147, March 2001. 2

[13] T. Lindeberg. On the Axiomatic Foundations of Linear Scale-Space: Combining Semi-Group Structure with Causality vs. Scale Invariance. In *Gaussian Scale-Space Theory: Proc. PhD School on Scale-Space Theory*. Kluwer Academic Publishers, 1994. 3

[14] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, ICCV '99, pages 1150–1157. IEEE Computer Society, 1999. 2

[15] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, April 1981. 2, 14

[16] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London Series B*, 207:187–217, 1980. 3

[17] G. Osterberg. *Topography of the layer of rods and cones in the human retina*. Acta ophthalmologica: Supplementum. Levin & Munksgaard, 1935. 3, 8

[18] L. Piela, J. Kostrowicki, and H. A. Scheraga. On the multiple-minima problem in the conformational analysis of molecules: deformation of the potential energy hypersurface by the diffusion equation method. *Journal of Physical Chemistry*, 93(8):3339–3346, 1989. 3, 14

[19] K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. In *Proceedings of the IEEE*, pages 2210–2239, 1998. 3

[20] L. Sevilla and E. Learned-Miller. Distribution fields. Technical Report UM-CS-2011-027, Dept. of Computer Science, University of Massachusetts Amherst, 2011. 5

[21] P. Simard, Y. LeCun, and J. S. Denker. Efficient pattern recognition using a new transformation distance. In *Advances in Neural Information Processing Systems 5, [NIPS Conference]*, pages 50–58, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc. 2

[22] A. J. Sommese and C. W. Wampler. *The numerical solution of systems of polynomials - arising in engineering and science.* World Scientific, 2005. 3

[23] R. Szeliski. Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.*, 2:1–104, January 2006. 2

[24] E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. In *Conference on Computer Vision and Pattern Recognition*, Alaska, USA, 2008. 3, 4, 14

[25] N. Vasconcelos and A. Lippman. Multiresolution tangent distance for affine-invariant classification. *Advances in Neural Information Processing Systems*, pages 843–849, 1998. 2

[26] D. V. Widder. *The Heat Equation*. Academic Press, 1975. 7

[27] A. P. Witkin. Scale-Space Filtering. In *8th Int. Joint Conf. Artificial Intelligence*, volume 2, pages 1019–1022, Karlsruhe, Aug. 1983. 3

Appendix

8.1 Notation

The symbol $\triangleq$ is used for equality by definition. Also, we use x for scalars, $\boldsymbol{x}$ for vectors, $\boldsymbol{X}$ for matrices, and $\mathcal{X}$ for sets. In addition, $f(.)$ denotes a scalar valued function and $\boldsymbol{f}(.)$ a vector valued function. Unless stated otherwise, $\|\boldsymbol{x}\|$ means $\|\boldsymbol{x}\|_2$ and ∇ means $\nabla_{\boldsymbol{x}}$. Finally, $\star$ and $\circledast$ denote convolution operators in spaces Θ and $\mathcal{X}$ respectively.

8.2 Definitions

Definition [Domain Transformation] Given a function $f : \mathcal{X} \to \mathbb{R}$ and a vector field $\boldsymbol{\tau} : \mathcal{X} \times \Theta \to \mathcal{X}$, where $\mathcal{X} = \mathbb{R}^n$ and $\Theta = \mathbb{R}^m$. We refer to $\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{\theta})$ as the *domain transformation* parameterized by $\boldsymbol{\theta}$. Note that the parameter vector $\boldsymbol{\theta}$ is constructed by concatenation of all the parameters of a transformation. For example, if case of affine $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$ with $\boldsymbol{x} \in \mathbb{R}^2$, $\boldsymbol{\theta}$ is a 6 dimensional vectors containing the elements of $\boldsymbol{A}$ and $\boldsymbol{b}$.

Definition [Isotropic Gaussian]

$$k(\boldsymbol{x}; \sigma^2) \triangleq \frac{1}{(\sqrt{2\pi}\sigma)^{\dim(\boldsymbol{x})}} e^{-\frac{\|\boldsymbol{x}\|^2}{2\sigma^2}} \,. \tag{31}$$

Definition [Anisotropic Gaussian]

$$K(\boldsymbol{x}; \boldsymbol{\Sigma}) \triangleq \frac{1}{(\sqrt{2\pi})^{\dim(\boldsymbol{x})} \sqrt{\det(\boldsymbol{\Sigma})}} e^{-\frac{\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}}{2}} \,. \tag{32}$$

Definition [Fourier Transform]

We use the following convention for Fourier transform. The Fourier transform of a real valued function $f : \mathbb{R}^n \to \mathbb{R}$ is $\hat{f}(\boldsymbol{\omega}) = \int_{\mathbb{R}^n} f(\boldsymbol{x}) e^{-i\boldsymbol{\omega}^T \boldsymbol{x}} d\boldsymbol{x}$ and the inverse Fourier transform is $\hat{f}(\boldsymbol{x}) = (2\pi)^{-n} \int_{\mathbb{R}^n} f(\boldsymbol{\omega}) e^{i\boldsymbol{\omega}^T \boldsymbol{x}} d\boldsymbol{\omega}$.

Definition [Transformation Kernel]

Given a domain transformation $\boldsymbol{\tau} : \mathcal{X} \times \Theta \to \mathcal{X}$, where $\mathcal{X} = \mathbb{R}^n$ and $\Theta = \mathbb{R}^m$. We define a *transformation kernel* associated with $\boldsymbol{\tau}$ as $u_{\boldsymbol{\tau},\sigma} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that it satisfies the following *integral equation*,

$$\begin{aligned} &\forall f \quad : \\ &\left[f(\boldsymbol{\tau}(\boldsymbol{x}, \cdot)) \star k(\cdot; \sigma^2) \right](\boldsymbol{\theta}) = \int_{\mathcal{X}} f(\boldsymbol{y}) \, u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{y} \,, \end{aligned} \tag{33}$$

where f is assumed to be a Schwartz function. Therefore, any transformation kernel that satisfies this equation allows the convolution of the transformed signal with the Gaussian kernel be equivalently written by the *integral transform* of the non-transformed signal with the kernel $u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$.

Definition [Smoothed Regularized Objective]
We define the smoothed *regularized* objective as the following.

$$\tilde{z}(\boldsymbol{\theta}, \boldsymbol{\theta}_0, r, \sigma) \triangleq [\tilde{h}(\,\cdot\,, \boldsymbol{\theta}_0, r) \star k(\,\cdot\,; \sigma^2)](\boldsymbol{\theta}). \tag{34}$$

8.3 Exact Expression for Homography Kernel

The exact expressions for functions $p(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}, \sigma)$ and $e(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}, \sigma)$ are as the following.

$$\begin{aligned}
e &\triangleq \exp\Big(- \frac{\|\boldsymbol{c}\|^2}{2\sigma^2} - \frac{\|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} - \boldsymbol{y}\|^2}{2\sigma^2(1 + \|\boldsymbol{x}\|^2)} + \frac{1}{2\sigma^2(1 + \|\boldsymbol{x}\|^2(1 + \|\boldsymbol{y}\|^2))} && (35)\\
&\qquad \Big(c_2^2(1 + x_2^2 + x_1^2(1 + \|\boldsymbol{y}\|^2)) + c_1^2(1 + x_1^2 + x_2^2(1 + \|\boldsymbol{y}\|^2)) && (36)\\
&\qquad + 2\boldsymbol{c}^T \boldsymbol{x} \boldsymbol{y}^T (\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} - \boldsymbol{y}) - 2c_1c_2x_1x_2\|\boldsymbol{y}\|^2 && (37)\\
&\qquad + \frac{\|\boldsymbol{x}\|^2}{1 + \|\boldsymbol{x}\|^2}(\boldsymbol{y}^T (\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} - \boldsymbol{y}))^2 \quad \Big)\Big) && (38)\\
p &\triangleq t_6 \quad \Big(t_3^4 + 4t_2^2t_4^2x_1^2 + 4t_2t_3t_4x_1(t_5x_1 + t_4x_2) - 2t_3^3(t_5x_1 + (t_4 + x_1)x_2) && (39)\\
&\qquad + 4t_1^2(4t_2^2 + t_5^2x_2^2 + 2t_2x_2(2t_5 + x_2)) && (40)\\
&\qquad + t_3^2(-2t_2x_1(2t_4 + x_1) + (t_5x_1 + t_4x_2)^2) && (41)\\
&\qquad + 2t_1 \quad \Big(4t_2^2x_1(2t_4 + x_1) + 4t_2(-t_3^2 + t_3t_5x_1 + t_4t_5x_1x_2 + t_3(t_4 + x_1)x_2) && (42)\\
&\qquad\qquad + t_3x_2(-t_3(2t_5 + x_2) + 2t_5(t_5x_1 + t_4x_2)) \quad \Big) \quad \Big) && (43)\\
t_0 &\triangleq \frac{1}{2\sigma^2(1 + \|\boldsymbol{x}\|^2)} && (44)\\
t_1 &\triangleq t_0(1 + x_2^2 + x_1^2(1 + \|\boldsymbol{y}\|^2)) && (45)\\
t_2 &\triangleq t_0(1 + x_1^2 + x_2^2(1 + \|\boldsymbol{y}\|^2)) && (46)\\
t_3 &\triangleq -2t_0x_1x_2\|\boldsymbol{y}\|^2 && (47)\\
t_4 &\triangleq 2t_0(c_1(1 + \|\boldsymbol{x}\|^2) + x_1\boldsymbol{y}^T(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} - \boldsymbol{y})) && (48)\\
t_5 &\triangleq 2t_0(c_2(1 + \|\boldsymbol{x}\|^2) + x_2\boldsymbol{y}^T(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} - \boldsymbol{y})) && (49)\\
t_6 &\triangleq t_0 \frac{4\pi}{(4t_1t_2 - t_3^2)^{\frac{5}{2}}\sigma^2}\,. && (50)
\end{aligned}$$

8.4 Proofs

Proposition 0 *The following identity holds for the product of two Gaussians.*

$$k(\boldsymbol{\tau} - \boldsymbol{\mu}_1; \sigma_1^2)\; k(\boldsymbol{\tau} - \boldsymbol{\mu}_2; \sigma_2^2) = \frac{e^{-\frac{\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|^2}{2(\sigma_1^2 + \sigma_2^2)}}}{(\sqrt{2\pi(\sigma_1^2 + \sigma_2^2)})^m} k(\boldsymbol{\tau} - \frac{\sigma_2^2\boldsymbol{\mu}_1 + \sigma_1^2\boldsymbol{\mu}_2}{\sigma_1^2 + \sigma_2^2}; \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2})\,.$$

Proof

$$\begin{aligned}
& k(\boldsymbol{\tau}-\boldsymbol{\mu}_1;\sigma_1^2)\; k(\boldsymbol{\tau}-\boldsymbol{\mu}_2;\sigma_2^2) && (51)\\
= \; & \frac{1}{(\sigma_1\sqrt{2\pi})^m} e^{-\frac{\|\boldsymbol{\tau}-\boldsymbol{\mu}_1\|^2}{2\sigma_1^2}} \frac{1}{(\sigma_2\sqrt{2\pi})^m} e^{-\frac{\|\boldsymbol{\tau}-\boldsymbol{\mu}_2\|^2}{2\sigma_2^2}} && (52)\\
= \; & \frac{1}{(2\pi\sigma_1\sigma_2)^m} e^{-\frac{\|\boldsymbol{\tau}-\boldsymbol{\mu}_1\|^2}{2\sigma_1^2}-\frac{\|\boldsymbol{\tau}-\boldsymbol{\mu}_2\|^2}{2\sigma_2^2}} && (53)\\
= \; & \frac{1}{(2\pi\sigma_1\sigma_2)^m} e^{-\frac{\|\boldsymbol{\tau}-\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}(\frac{\boldsymbol{\mu}_1}{\sigma_1^2}+\frac{\boldsymbol{\mu}_2}{\sigma_2^2})\|^2}{2\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}}-\frac{\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|^2}{2(\sigma_1^2+\sigma_2^2)}} && (54)\\
= \; & \frac{e^{-\frac{\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|^2}{2(\sigma_1^2+\sigma_2^2)}}}{(\sqrt{2\pi(\sigma_1^2+\sigma_2^2)})^m} k(\boldsymbol{\tau}-\frac{\sigma_2^2\boldsymbol{\mu}_1+\sigma_1^2\boldsymbol{\mu}_2}{\sigma_1^2+\sigma_2^2};\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2})\,. && (55)
\end{aligned}$$

Note that (54) is derived by completing the square.

□

Proposition 5 *The following choice of u,*

$$\begin{aligned}
& u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y})\\
= \; & \frac{1}{(2\pi)^n}\int_\Omega\Big(\int_\Theta e^{i\boldsymbol{\omega}^T(\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{t})-\boldsymbol{y})}k(\boldsymbol{t}-\boldsymbol{\theta};\sigma^2)\,d\boldsymbol{t}\Big)d\boldsymbol{\omega} && (56)
\end{aligned}$$

is a solution to the definition of kernel provided in (33). Here $\mathcal{X}=\Omega=\mathbb{R}^n$, and $k(\boldsymbol{t};\sigma^2)$ is some function $k(\,.\,;\sigma):\mathcal{X}\to\mathbb{R}$ with some parameter σ, which in our case is simply an isotropic Gaussian with bandwidth σ.

Proof The key to the proof is writing $f(\boldsymbol{x})$ by its Fourier form $f(\boldsymbol{x})=(2\pi)^{-n}\int_\Omega\hat{f}(\boldsymbol{\omega})e^{i\boldsymbol{\omega}^T\boldsymbol{x}}\,d\boldsymbol{\omega}$, where $\Omega=\mathbb{R}^n$ (similar to $\mathcal{X}=\mathbb{R}^n$).

$$\begin{aligned}
& [f(\boldsymbol{\tau}(\boldsymbol{x},.))\overset{\Theta}{\circledast}k(\,.\,\sigma^2)](\boldsymbol{\theta}) && (57)\\
= \; & [\Big(\frac{1}{(2\pi)^n}\int_\Omega\hat{f}(\boldsymbol{\omega})e^{i\boldsymbol{\omega}^T\boldsymbol{\tau}(\boldsymbol{x},.)}\,d\boldsymbol{\omega}\Big)\overset{\Theta}{\circledast}k(\,.\,\sigma^2)](\boldsymbol{\theta}) && (58)\\
= \; & \frac{1}{(2\pi)^n}\int_\Theta\Big(\int_\Omega\hat{f}(\boldsymbol{\omega})e^{i\boldsymbol{\omega}^T\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{t})}\,d\boldsymbol{\omega}\Big)k(\boldsymbol{t}-\boldsymbol{\theta};\sigma^2)\,d\boldsymbol{t} && (59)\\
= \; & \frac{1}{(2\pi)^n}\int_\Omega\hat{f}(\boldsymbol{\omega})\Big(\int_\Theta e^{i\boldsymbol{\omega}^T\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{t})}k(\boldsymbol{t}-\boldsymbol{\theta};\sigma^2)\,d\boldsymbol{t}\Big)\,d\boldsymbol{\omega} && (60)\\
= \; & \frac{1}{(2\pi)^n}\int_\mathcal{X}f(\boldsymbol{y})\Big(\int_\Omega e^{-i\boldsymbol{\omega}^T\boldsymbol{y}}\Big(\int_\Theta e^{i\boldsymbol{\omega}^T\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{t})}k(\boldsymbol{t}-\boldsymbol{\theta};\sigma^2)\,d\boldsymbol{t}\Big)\,d\boldsymbol{\omega}\Big)\,d\boldsymbol{y} && (61)\\
= \; & \frac{1}{(2\pi)^n}\int_\mathcal{X}f(\boldsymbol{y})\Big(\int_\Omega\int_\Theta e^{i\boldsymbol{\omega}^T(\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{t})-\boldsymbol{y})}k(\boldsymbol{t}-\boldsymbol{\theta};\sigma^2)\,d\boldsymbol{t}\,d\boldsymbol{\omega}\Big)\,d\boldsymbol{y} && (62)\\
= \; & \int_\mathcal{X}f(\boldsymbol{y})u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y})\,d\boldsymbol{y}\,, && (63)
\end{aligned}$$

where (61) uses the *Parseval* theorem, and (63) uses proposition's assumption (56).

□

Proposition 6 *Suppose* $f_1 = \sum_{k=1}^{p} a_k \phi(\boldsymbol{y}; \boldsymbol{x}_k, \delta_k)$*, where* $\phi(\boldsymbol{x}; \boldsymbol{x}_k, \delta_k) = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x}_k\|^2}{2\delta_k^2}}$. *Assume that* $u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$ *is Gaussian in variable* $\boldsymbol{y}$*. Then the following identity holds.*

$$\int_{\mathcal{X}} f_1(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \tag{64}$$

$$= \sum_{i=1}^{p} a_i \Big(\frac{\delta_i}{\sqrt{\delta_i^2 + s^2}}\Big)^n e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{\tau}\|^2}{2(\delta_i^2 + s^2)}}. \tag{65}$$

Proof

$$\int_{\mathcal{X}} f_1(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \tag{66}$$

$$= \int_{\mathcal{X}} f_1(\boldsymbol{y}) k(\boldsymbol{\tau} - \boldsymbol{y}; s^2)\, d\boldsymbol{y} \tag{67}$$

$$= \int_{\mathbb{R}^n} \left(\sum_{k=1}^{p} a_k \phi(\boldsymbol{y}; \boldsymbol{x}_k, \delta_k) \right) k(\boldsymbol{\tau} - \boldsymbol{y}; s^2)\, d\boldsymbol{y} \tag{68}$$

$$= \sum_{k=1}^{p} a_k \left(\int_{\mathbb{R}^n} \phi(\boldsymbol{y}; \boldsymbol{x}_k, \delta_k) k(\boldsymbol{\tau} - \boldsymbol{y}; s^2)\, d\boldsymbol{y} \right) \tag{69}$$

$$= \sum_{k=1}^{p} a_k (\delta_k \sqrt{2\pi})^n \left(\int_{\mathbb{R}^n} k(\boldsymbol{y} - \boldsymbol{x}_k; \delta_k^2) k(\boldsymbol{\tau} - \boldsymbol{y}; s^2)\, d\boldsymbol{y} \right) \tag{70}$$

$$= \sum_{k=1}^{p} a_k (\delta_k \sqrt{2\pi})^n \left(\int_{\mathbb{R}^n} \frac{e^{-\frac{\|\boldsymbol{x}_k - \boldsymbol{\tau}\|^2}{2(\delta_k^2 + s^2)}}}{(\sqrt{2\pi(\delta_k^2 + s^2)})^n} k(\boldsymbol{y} - \frac{s^2 \boldsymbol{x}_k + \delta_k^2 \boldsymbol{\tau}}{\delta_k^2 + s^2}; \frac{\delta_k^2 s^2}{\delta_k^2 + s^2})\, d\boldsymbol{y} \right) \tag{71}$$

$$= \sum_{k=1}^{p} a_k \Big(\frac{\delta_k}{\sqrt{\delta_k^2 + s^2}}\Big)^n e^{-\frac{\|\boldsymbol{x}_k - \boldsymbol{\tau}\|^2}{2(\delta_k^2 + s^2)}} \left(\int_{\mathbb{R}^n} k(\boldsymbol{y} - \frac{s^2 \boldsymbol{x}_k + \delta_k^2 \boldsymbol{\tau}}{\delta_k^2 + s^2}; \frac{\delta_k^2 s^2}{\delta_k^2 + s^2})\, d\boldsymbol{y} \right) \tag{72}$$

$$= \sum_{k=1}^{p} a_k \Big(\frac{\delta_k}{\sqrt{\delta_k^2 + s^2}}\Big)^n e^{-\frac{\|\boldsymbol{x}_k - \boldsymbol{\tau}\|^2}{2(\delta_k^2 + s^2)}}, \tag{73}$$

where in (71) we use the Gaussian product result from proposition 0. □

Proposition 7 *The regularized objective function* $\tilde{z}(\boldsymbol{\theta}, \boldsymbol{\theta}_0, r, \sigma)$ *can be written using transformation kernels as follows.*

$$\tilde{z}(\boldsymbol{\theta}, \boldsymbol{\theta}_0, r, \sigma) \tag{74}$$

$$= [\tilde{h}(\,.\,, \boldsymbol{\theta}_0, r) \overset{\ominus}{\circledast} k(\,.\,; \sigma^2)](\boldsymbol{\theta}) \tag{75}$$

$$= \int_{\mathcal{X}} \left(k(\boldsymbol{\theta} - \boldsymbol{\theta}_0; r^2 + \sigma^2) f_2(\boldsymbol{x}) \int_{\mathcal{X}} \Big(f_1(\boldsymbol{y}) u_{\boldsymbol{\tau}, \frac{r\sigma}{\sqrt{r^2+\sigma^2}}} (\frac{r^2 \boldsymbol{\theta} + \sigma^2 \boldsymbol{\theta}_0}{r^2 + \sigma^2}, \boldsymbol{x}, \boldsymbol{y}) \Big)\, d\boldsymbol{y} \right) \tag{76}$$

Proof For computing $\tilde{z}$, we proceed as below.

$$\begin{aligned}
& \tilde{z}(\boldsymbol{\theta}, \boldsymbol{\theta}_0, r, \sigma) && (77) \\
= \; & [\tilde{h}(\,.\,, \boldsymbol{\theta}_0, r) \star k(\,.\,; \sigma^2)](\boldsymbol{\theta}) && (78) \\
= \; & [\Big(\int_{\mathcal{X}} \big(k(\boldsymbol{\theta} - \boldsymbol{\theta}_0; r^2) f_1(\boldsymbol{\tau}(\boldsymbol{x}; \boldsymbol{\theta})) f_2(\boldsymbol{x}) \big) \, d\boldsymbol{x} \Big) \star k(\,.\,; \sigma^2)](\boldsymbol{\theta}) && (79) \\
= \; & \int_{\mathcal{X}} \left(f_2(\boldsymbol{x}) [\Big(k(\boldsymbol{\theta} - \boldsymbol{\theta}_0; r^2) f_1(\boldsymbol{\tau}(\boldsymbol{x}; \boldsymbol{\theta})) \Big) \star k(\,.\,; \sigma^2)](\boldsymbol{\theta}) \right) d\boldsymbol{x} && (80) \\
= \; & \int_{\mathcal{X}} \left(f_2(\boldsymbol{x}) \int_{\Theta} \Big(k(\boldsymbol{\theta}_0 - \boldsymbol{t}; r^2) f_1(\boldsymbol{\tau}(\boldsymbol{x}; \boldsymbol{t})) k(\boldsymbol{\theta} - \boldsymbol{t}; \sigma^2) \Big) \, d\boldsymbol{t} \right) d\boldsymbol{x} && (81) \\
= \; & \int_{\mathcal{X}} \left(f_2(\boldsymbol{x}) \int_{\Theta} \Big(f_1(\boldsymbol{\tau}(\boldsymbol{x}; \boldsymbol{t})) \frac{e^{-\frac{\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|^2}{2(r^2 + \sigma^2)}}}{\left(\sqrt{2\pi(r^2 + \sigma^2)} \, \right)^m} k(\boldsymbol{t} - \frac{\sigma^2 \boldsymbol{\theta}_0 + r^2 \boldsymbol{\theta}}{r^2 + \sigma^2}; \frac{r^2 \sigma^2}{r^2 + \sigma^2}) \Big) \, d\boldsymbol{t} \right) d\boldsymbol{x} && (82) \\
= \; & \int_{\mathcal{X}} \left(\frac{e^{-\frac{\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|^2}{2(r^2 + \sigma^2)}}}{\left(\sqrt{2\pi(r^2 + \sigma^2)} \, \right)^m} f_2(\boldsymbol{x}) \int_{\mathcal{X}} \Big(f_1(\boldsymbol{y}) u_{\boldsymbol{\tau}, \frac{r\sigma}{\sqrt{r^2 + \sigma^2}}} (\frac{r^2 \boldsymbol{\theta} + \sigma^2 \boldsymbol{\theta}_0}{r^2 + \sigma^2}, \boldsymbol{x}, \boldsymbol{y}) \Big) \, d\boldsymbol{y} \right) d\boldsymbol{x} && (83)
\end{aligned}$$

Thus, regularized objective function from (27) leads to the following result.

$$\begin{aligned}
& \tilde{z}(\boldsymbol{\theta}, \boldsymbol{\theta}_0, r, \sigma) && (84) \\
= \; & [\tilde{h}(\,.\,, \boldsymbol{\theta}_0, r) \overset{\Theta}{\circledast} k(\,.\,; \sigma^2)](\boldsymbol{\theta}) && (85) \\
= \; & \int_{\mathcal{X}} \left(k(\boldsymbol{\theta} - \boldsymbol{\theta}_0; r^2 + \sigma^2) f_2(\boldsymbol{x}) \int_{\mathcal{X}} \Big(f_1(\boldsymbol{y}) u_{\boldsymbol{\tau}, \frac{r\sigma}{\sqrt{r^2 + \sigma^2}}} (\frac{r^2 \boldsymbol{\theta} + \sigma^2 \boldsymbol{\theta}_0}{r^2 + \sigma^2}, \boldsymbol{x}, \boldsymbol{y}) \Big) \, d\boldsymbol{y} \right) d\boldsymbol{x} && (86)
\end{aligned}$$

□