# Seeing through the Blur

Hossein Mobahi[1], C. Lawrence Zitnick[2], and Yi Ma[3,4]

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

## Abstract

*This paper addresses the problem of image alignment using direct intensity-based methods for affine and homography transformations. Direct methods often employ scale-space smoothing (Gaussian blur) of the images to avoid local minima. Although, it is known that the isotropic blur used is not optimal for some motion models, the correct blur kernels have not been rigorously derived for motion models beyond translations. In this work, we derive blur kernels that result from smoothing the alignment objective function for some common motion models such as affine and homography. We show the derived kernels remove poor local minima and reach lower energy solutions in practice.*

## 1. Introduction

It is one of the most fundamental problems in computer vision to establish alignment between 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 [17]. 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) [28].

In many applications intensity-based methods are appealing due to their direct access to richer information (i.e. to every single pixel) [17]. This can be useful, for example, when working with semi-regular patterns that are difficult to match by local features [31]. However, the practical performance of direct intensity methods can be undermined by the associated optimization challenge [39]. Specifically, it is well-known that optimizing a cost function that directly compares intensities of an image pair is highly susceptible to finding local minima [14]. Thus, unless very good initialization is provided, plain direct alignment of image intensities may lead to poor results.

Lucas and Kanade addressed the problem of reducing local minima in direct methods by adopting a *coarse-to-fine* Gauss-Newton scheme [29]. The Gauss-Newton scheme uses a first order Taylor's approximation to estimate displacements, which is typically violated except when the displacements are small. Lucas and Kanade overcome this limitation by initially using a coarse resolution image to reduce the relative magnitude of the displacements. The displacements are iteratively refined using the displacements computed at coarser scales to initialize the finer scales. This results in the first order Taylor's expansion providing a better approximation and a reduction in local minima at each iteration. An alternative to reducing the resolution of the image is to utilize an isotropic Gaussian kernel to blur the image, so that the higher-order terms in the Taylor's expansion are negligible.

It was later shown that such a coarse-to-fine scheme is guaranteed to recover the optimal *displacement* under some mild conditions [24]. Although a guarantee of correctness is only established for translational motion, the notion of coarse-to-fine smoothing followed by local approximation has been adopted in computer vision to matching with almost all parametric transformation models [3, 18, 29, 37, 41, 42]. Despite its popularity, there are serious theoretical and practical issues with the Lucas-Kanade scheme when applied to non-translational motions. For example, if the transformation is scaling, it is easy to show that the Hessian of the image function may grow proportional to the distance from the origin. To compensate for this effect, stronger smoothing is required for points farther from the origin.

In this paper, we propose Gaussian smoothing the objective function of the alignment task, instead of the images, which was in fact the original goal of coarse-to-fine image smoothing techniques. In particular, we derive the theoretically correct image blur kernels that arise from (Gaussian) smoothing an alignment objective function. We show that, for common motion models, such as affine and homography, there exists a corresponding integral operator on the image space. We refer to the kernels of such integral operators as *transformation kernels*. As we show, all of these kernels are spatially varying as long as the transformation is not a pure translation, and vary from those heuristically suggested by [7] or [40].

Our goal is to improve 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.

## 2. Related Works

This paper focused on optimization by Gaussian smoothing the objective function combined with a path-following scheme. In fact, this optimization approach belongs to a large family of methods. All these methods hope to escape from brittle local minima by starting from a "simplified" problem and gradually deforming it to the desired problem. This concept has been widely used across different disciplines under different names and variations. Examples includes graduated optimization [9] in computer vision, homotopy continuation [1] in numerical methods, deterministic annealing [35] in machine learning, the diffusion equation method [33] in chemistry, reward shaping [20] in robotics, etc. Similar concept for discrete spaces has also been used [4, 5, 21, 38].

The popularity of "*isotropic Gaussian convolution*" for image blurring is, in part, a legacy of *scale-space theory*. This influential theory emerged from a series of seminal articles in the 80's [19, 46, 47]. This theory shows that "isotropic Gaussian convolution" is the "unique" linear operator obeying some least commitment axioms [25]. In particular, this operator is unbiased to location and orientation, due to its convolutional and isotropic nature. Later, Lindeberg extended scale-space theory to cover affine blur by anisotropic *spatially invariant* kernels [27, 26].

Nevertheless, it is known that the human eye has progressively less resolution from the center (fovea) toward the periphery [32]. In computer vision, spatially varying blur is believed to benefit matching and alignment tasks. In that direction, Berg and Malik [7] introduced the notion of "geometric blur" and suggested some spatially varying kernels inspired by that. However, their kernels are derived heuristically, without a rigorous connection to the underlying geometric transformations. Some limitations of traditional image smoothing are discussed in [36] and coped using stacks of binary images. That work, however, still uses isotropic Gaussian kernel for smoothing images of the stack.

**Algorithm 1 Alignment by Gaussian Smoothing.**

1: Input: $f_1 : \mathcal{X} \to \mathbb{R}$, $f_2 : \mathcal{X} \to \mathbb{R}$, $\boldsymbol{\theta}_0 \in \Theta$, ...
   The set $\{\sigma_k\}$ for $k = 1, \ldots, K$ s.t. $0 < \sigma_{k+1} < \sigma_k$
2: **for** $k = 1 \to K$ **do**
3: $\quad \boldsymbol{\theta}_k =$ local maximizer of $z(\boldsymbol{\theta}; \sigma_k)$ initialized at $\boldsymbol{\theta}_{k-1}$
4: **end for**
5: Output: $\boldsymbol{\theta}_K$

## 3. Notation and Definitions

The symbol $\triangleq$ is used for equality by definition. We use $x$ for scalars, $\boldsymbol{x}$ for vectors, $\boldsymbol{X}$ for matrices, and $\mathcal{X}$ for sets. scalar valued and vector valued functions are respectively denoted by $f(.)$ and $\boldsymbol{f}(.)$. We use $\|\boldsymbol{x}\|$ for $\|\boldsymbol{x}\|_2$ and $\nabla$ for $\nabla_{\boldsymbol{x}}$. Finally, $\star$ and $\circledast$ denote convolution operators in spaces $\Theta$ and $\mathcal{X}$ respectively.

Given a signal $f : \mathcal{X} \to \mathbb{R}$, e.g. a 2D image, we define a signal warping or *domain transformation* parameterized by $\boldsymbol{\theta}$ as $\boldsymbol{\tau} : \mathcal{X} \times \Theta \to \mathcal{X}$. Here $\boldsymbol{\theta}$ is concatenation of all the parameters of a transformation. For example, in case of affine $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$ with $\boldsymbol{x} \in \mathbb{R}^2$, $\boldsymbol{\theta}$ is a 6 dimensional vector containing the the elements of $\boldsymbol{A}$ and $\boldsymbol{b}$.

The *isotropic normalized Gaussian* with covariance $\sigma^2 \boldsymbol{I}$ is denoted by $k(\boldsymbol{x}; \sigma^2)$, where $\boldsymbol{I}$ is the identity matrix. The *anisotropic normalized Gaussian* with covariance matrix $\boldsymbol{\Sigma}$ is denoted by $K(\boldsymbol{x}; \boldsymbol{\Sigma})$. The *Fourier transform* of a real valued function $f : \mathbb{R}^n \to \mathbb{R}$ is $\hat{f}(\boldsymbol{\omega}) \triangleq \int_{\mathbb{R}^n} f(\boldsymbol{x}) e^{-i\boldsymbol{\omega}^T \boldsymbol{x}} d\boldsymbol{x}$ and the *inverse Fourier transform* is $\check{f}(\boldsymbol{x}) = (2\pi)^{-n} \int_{\mathbb{R}^n} f(\boldsymbol{\omega}) e^{i\boldsymbol{\omega}^T \boldsymbol{x}} d\boldsymbol{\omega}$.

## 4. Smoothing the Objective

We use the inner product between the transformed $f_1$ 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 in many scenarios may be different from the original signals. For example, they may be mean subtracted or normalized by their $\ell_2$ norm. The alignment objective function is denoted by $h(\boldsymbol{\theta})$ and defined as follows,

$$h(\boldsymbol{\theta}) \triangleq \int_{\mathcal{X}} f_1(\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{\theta}))\, f_2(\boldsymbol{x})\, d\boldsymbol{x}\,, \tag{1}$$

where $f_1(\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{\theta}))$ is signal $f_1$ warped by $\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{\theta})$. Our goal is to find the parameters $\boldsymbol{\theta}^*$ that optimize the objective function (1). In practice, $h$ may have multiple local optima. Thus, instead of directly optimizing $h$, we iteratively optimize a smoothed version of $h$ in a coarse-to-fine approach. We denote the objective function $h(\boldsymbol{\theta})$ obtained after smoothing as $z(\boldsymbol{\theta}, \sigma)$, where $\sigma$ determines the amount

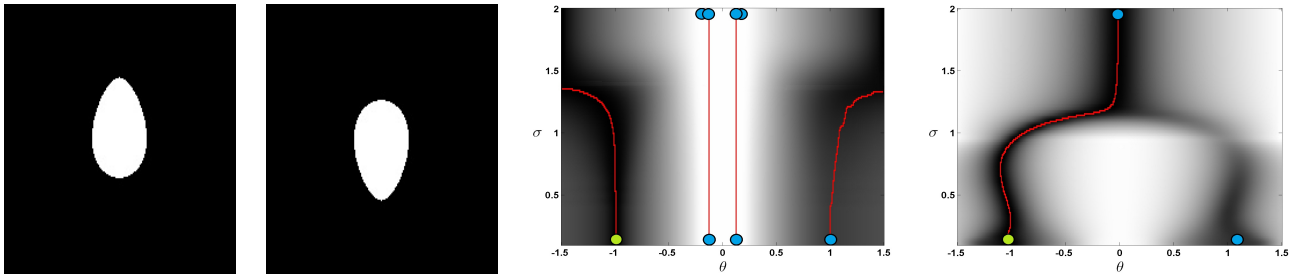



Figure 1. Basin of attraction for scale alignment. Egg shape input images are shown in (a) and (b), where black and white pixels are respectively by -1 and 1 intensity values. Obviously, the correct alignment is attained at $\theta = -1$, due to reflection symmetry. The objective function for $z_{LK}$ is shown in (c) and for $z$ in (d). Blue, green and red respectively indicate local maxima, global maximum and basin of attraction originating from local maxima of highest blur.

of smoothing. Given $z(\boldsymbol{\theta}, \sigma)$, we adopt the standard optimization approach described by Algorithm 1. That is, we use the parameters $\boldsymbol{\theta}_{k-1}$ found at a coarser scale to initialize the solution $\boldsymbol{\theta}_k$ found at each progressively finer scale.

In the Lucas-Kanade algorithm [29], instead of smoothing the objective function, they directly blur the images. This results in the following form for the objective function,

$$z_{LK}(\boldsymbol{\theta}, \sigma) \triangleq \int_{\mathcal{X}} [f_1(\boldsymbol{\tau}(\,.\,, \boldsymbol{\theta})) \circledast k(\,.\,; \sigma^2)]\, [f_2 \circledast k(\,.\,; \sigma^2)](\boldsymbol{x})\; d\boldsymbol{x}\,.$$

Image smoothing is done in hope of eliminating the brittle local optima in the objective function. However, if the latter is our goal, we propose the correct approach is to blur the objective function directly[1],

$$z(\boldsymbol{\theta}, \sigma) \triangleq [h \star k(\,.\,, \sigma^2)](\boldsymbol{\theta})\,.$$

The optimization landscape of these two cases may differ significantly. To illustrate, consider the egg shape images in Figures 1(a) and 1(b). If we assume the only parameter subject to optimization is the scale factor, i.e. $\boldsymbol{\tau}(\boldsymbol{x}, \theta) = \theta \boldsymbol{x}$, the associated optimization landscape is visualized in figure 1(c) for $z_{LK}$ and 1(d) for $z$. Clearly, $z$ has a single basin of attraction that leads to the global optimum, unlike $z_{LK}$ whose basins do not necessarily land at the global optimum.

## 5. Transformation Kernels

Our is to perform optimization on the *smoothed* objective function. Smoothing the objective function refers to a convolution in the space of transformation parameters with a Gaussian kernel. Unfortunately, performing this convolution may be computationally expensive when the dimensionality of the *transformation space* is large, e.g. eight for homography of 2D images. This section introduces the notion of transformation kernels, which enables us to equivalently write the smoothed objective function using some *integral transform* of the signal. This integration is performed in the *image space* (e.g. 2D for images), reducing the computational complexity.

**Definition** 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 the *transformation kernel* associated with $\boldsymbol{\tau}$ as $u_{\boldsymbol{\tau},\sigma} : \Theta \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ to be the function satisfying the following *integral equation* for *all* Schwartz[2] functions $f$,

$$[f(\boldsymbol{\tau}(\boldsymbol{x}, \cdot)) \star k(\cdot; \sigma^2)]\,(\boldsymbol{\theta}) = \int_{\mathcal{X}} f(\boldsymbol{y})\, u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{y} \quad (2)$$

Using this definition, the smoothed alignment objective $z$ can be equivalently written as the following,

$$\begin{aligned} & z(\boldsymbol{\theta}, \sigma) && (3)\\ \triangleq\; & [h \star k(\,.\,, \sigma^2)](\boldsymbol{\theta}) && (4)\\ =\; & \int_{\mathcal{X}} \left(f_2(\boldsymbol{x}) [f_1(\boldsymbol{\tau}(\boldsymbol{x}, .)) \star k(\,.\,, \sigma^2)](\boldsymbol{\theta})\right) d\boldsymbol{x} && (5)\\ =\; & \int_{\mathcal{X}} \left( f_2(\boldsymbol{x}) \left( \int_{\mathcal{X}} f_1(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \right) \right) d\boldsymbol{x}\,, && (6) \end{aligned}$$

where the integral transform in (6) uses the definition of kernel provided in (2). A procedure for computing the integral transform (6) will be provided in section 6.

### 5.1. Derivation of Kernels

**Proposition 1** *The following choice of $u$ is a solution to the definition of a kernel provided in (2). Here $\mathcal{X} = \Omega = \mathbb{R}^n$.*

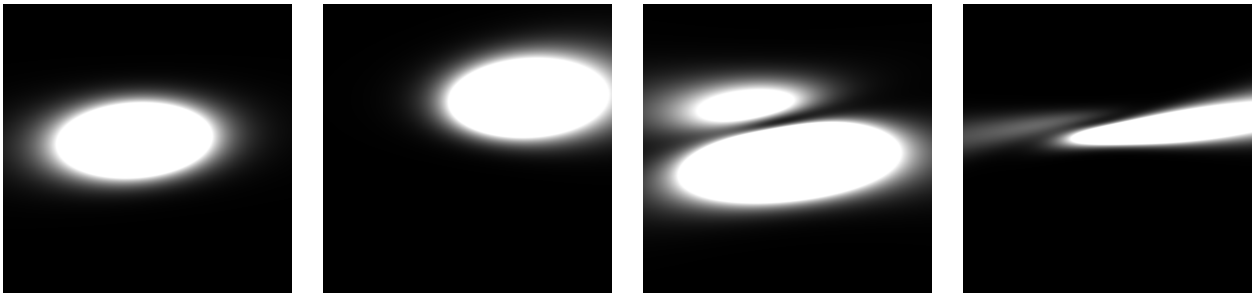
$$\begin{aligned} & u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) \\ & = \frac{1}{(2\pi)^n} \int_{\Omega} \left( \int_{\Theta} e^{i\boldsymbol{\omega}^T (\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{t}) - \boldsymbol{y})} k(\boldsymbol{t} - \boldsymbol{\theta}; \sigma^2)\, d\boldsymbol{t} \right) d\boldsymbol{\omega} \quad (7) \end{aligned}$$

[1] $[h \star k(\,.\,, \sigma^2)](\boldsymbol{\theta})$ is bounded when either signals decay rapidly enough or have bounded support. In image scenario, the latter always holds.

[2] A Schwartz function is one whose derivatives are rapidly decreasing.

| 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 | $[\,vec(\boldsymbol{A}_{n\times n})\,,\,\boldsymbol{b}_{n\times 1}\,]$ | $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$ | $k(\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{\theta}) - \boldsymbol{y}; \sigma^2(1 + \|\boldsymbol{x}\|^2))$ |
| Homography | $[\,vec(\boldsymbol{A}_{n\times n})\,,\,\boldsymbol{b}_{n\times 1}\,,\,\boldsymbol{c}_{n\times 1}\,]$ | $\frac{1}{1+\boldsymbol{c}^T\boldsymbol{x}}(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$ | $q(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}, \sigma)\; e^{-p(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}, \sigma)}$ |

Table 1. Kernels for some of the common transformations arising in vision (for all kernels $n \geq 1$ except homography where $n = 2$).



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

Figure 2. Visualization of affine and homography kernels specified by $\boldsymbol{A}_0 = [2 \quad 0.2\,;\, -0.3 \quad 4]$, $\boldsymbol{b}_0 = [0.15 \quad -0.25]$ (also $\boldsymbol{c}_0 = [1 \quad -5]$ for homography). Here $\boldsymbol{x} \in [-1, 1] \times [-1, 1]$, $\sigma = 0.5$ and $\boldsymbol{y} = (1, 1)$ or $\boldsymbol{y} = (0, 0)$. More precisely, affine kernels in (a) $u(\boldsymbol{\theta} = \boldsymbol{\theta}_0, \boldsymbol{x}, \boldsymbol{y} = (0, 0))$ (b) $u(\boldsymbol{\theta} = \boldsymbol{\theta}_0, \boldsymbol{x}, \boldsymbol{y} = (1, 1))$ and homography kernels in (c) $u(\boldsymbol{\theta} = \boldsymbol{\theta}_0, \boldsymbol{x}, \boldsymbol{y} = (0, 0))$ (d) $u(\boldsymbol{\theta} = \boldsymbol{\theta}_0, \boldsymbol{x}, \boldsymbol{y} = (1, 1))$ .

The proof uses the Fourier representation $f(\boldsymbol{x}) = (2\pi)^{-n} \int_\Omega \hat{f}(\boldsymbol{\omega}) e^{i\boldsymbol{\omega}^T\boldsymbol{x}}\, d\boldsymbol{\omega}$, and then application of Parseval's theorem. See the appendix for details.

Now by applying the result of proposition 1 to the desired transformation $\boldsymbol{\tau}$, we can compute the integrals[3] and derive the corresponding kernel function as shown in Table 1 (see also figure 2 for some visualization). The functions $q$ and $p$, associated with the homography kernel, are each a ratio of polynomials[4].

The complete derivation of these kernels is provided in the appendix. Nevertheless, below we present a relatively easy way to check the correctness of the kernels. Specifically, we check two necessary conditions of the heat equation and the limit behavior, which must hold for the kernels.

[3] Although the integral in (7) 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.

[4] Complete expression for the homography kernel $qe^{-p}$ is as below:

$$
\begin{aligned}
\gamma_0 &\triangleq \frac{1}{1 + \|\boldsymbol{x}\|^2} \\
\gamma_1 &\triangleq 1 + \boldsymbol{c}^T\boldsymbol{x} \\
\boldsymbol{v} &\triangleq \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \\
q &\triangleq \gamma_0 \frac{(\gamma_0\|\boldsymbol{x}\|^2\boldsymbol{y}^T\boldsymbol{v} + \gamma_1)^2 + \sigma^2\|\boldsymbol{x}\|^2(1 + \gamma_0\|\boldsymbol{x}\|^2\|\boldsymbol{y}\|^2)}{2\pi\sigma^2(1 + \gamma_0\|\boldsymbol{x}\|^2\|\boldsymbol{y}\|^2)^{\frac{5}{2}}} \\
p &\triangleq \frac{\|\gamma_1\boldsymbol{y} - \boldsymbol{v}\|^2 + \gamma_0\|\boldsymbol{x}\|^2(v_2y_1 - v_1y_2)^2}{2\sigma^2(1 + \|\boldsymbol{x}\|^2(1 + \|\boldsymbol{y}\|^2))}\,.
\end{aligned}
$$

#### 5.1.1 Heat Equation

Consider the convolution $[f(\boldsymbol{\tau}(\boldsymbol{x}, \cdot)) \star k(\cdot\,; \sigma)](\boldsymbol{\theta})$. Such Gaussian convolution obeys the *heat equation* [45]:

$$
\begin{aligned}
& \sigma\Delta_{\boldsymbol{\theta}}[f(\boldsymbol{\tau}(\boldsymbol{x}, \cdot)) \star k(\cdot\,; \sigma)](\boldsymbol{\theta}) \\
= \;& (\partial/\partial\sigma)[f(\boldsymbol{\tau}(\boldsymbol{x}, \cdot)) \star k(\cdot\,; \sigma)](\boldsymbol{\theta})\,. \qquad (8)
\end{aligned}
$$

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:

$$
\begin{aligned}
& \sigma\Delta_{\boldsymbol{\theta}} \int_{\mathcal{X}} f(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \\
= \;& \frac{\partial}{\partial\sigma} \int_{\mathcal{X}} f(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \qquad (9) \\
\equiv \;& \int_{\mathcal{X}} f(\boldsymbol{y}) \sigma\Delta_{\boldsymbol{\theta}} u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \\
= \;& \int_{\mathcal{X}} f(\boldsymbol{y}) \frac{\partial}{\partial\sigma} u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \qquad (10) \\
\Leftarrow \;& \sigma\Delta_{\boldsymbol{\theta}} u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) = \frac{\partial}{\partial\sigma} u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\,, \qquad (11)
\end{aligned}
$$

where $\Leftarrow$ in (11) means sufficient condition. Now it is much easier to check the identity (11) for the provided kernels. For example, in the case of an 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 $(\frac{\|\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{\theta}) - \boldsymbol{y}\|^2}{\sigma^3(1+\|\boldsymbol{x}\|^2)} - \frac{n}{\sigma})\; k(\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{\theta}) - \boldsymbol{y}; \sigma^2(1 + \|\boldsymbol{x}\|^2))$.

#### 5.1.2 Limit Behavior

When the amount of smoothing approaches zero, the integral transform must recover the original function. Formally, we want the following identity to hold,

$$\lim_{\sigma \to 0^+} \int_{\mathcal{X}} f(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} = f(\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{\theta}))\,. \quad (12)$$

The 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 $\delta$ is Dirac's delta function. This is trivial for the kernels of affine and its special cases; 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.

### 5.2. Remarks

Two interesting observations can be made about Table 1. First, from a *purely objective* standpoint, the derived kernels exhibit "foveation", similar to that in the eye. Except for 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 spatially varying Gaussian kernels whose variance depends and increases in $\|\boldsymbol{x}\|$.

Second, the derived kernels are not necessarily rotation invariant. Therefore, the blur kernels proposed by Berg and Malik [7] are unable to represent the geometric transformations listed in Table 1. In fact, to the best of our knowledge, this work is the first that rigorously derives kernels for such transformations.

### 5.3. Image Blurring vs. Objective Blurring

It is now easy to check that for the "translation transformation", Gaussian convolution of the alignment objective with respect to the optimization variables is equivalent to applying a "Gaussian convolution" to the image $f_1$. This is easy to check by plugging the translation kernel from Table 1 into the smoothed objective function (6) as below:

$$\begin{aligned} & z(\boldsymbol{\theta}, \sigma) && (13)\\ &= \int_{\mathcal{X}} \left( f_2(\boldsymbol{x}) \int_{\mathcal{X}} f_1(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \right) d\boldsymbol{x} && (14)\\ &= \int_{\mathcal{X}} \left( f_2(\boldsymbol{x}) \int_{\mathcal{X}} f_1(\boldsymbol{y}) k(\boldsymbol{\theta} + \boldsymbol{x} - \boldsymbol{y}; \sigma^2)\, d\boldsymbol{y} \right) d\boldsymbol{x} && (15)\\ &= \int_{\mathcal{X}} \left( f_2(\boldsymbol{x}) \left[ f_1(\,\cdot\,) \circledast k(\,\cdot\,; \sigma^2) \right](\boldsymbol{\theta} + \boldsymbol{x}) \right) d\boldsymbol{x}\,. && (16) \end{aligned}$$

However, such equivalence does not hold for other transformations, e.g. affine. There, Gaussian convolution of the alignment objective with respect to the optimization variables is equivalent to an "integral transform" of $f_1$, which cannot be expressed by the convolution of $f_1$ with some spatially invariant convolution kernel in image space as shown below for affine case:

$$\begin{aligned} & z(\boldsymbol{\theta}, \sigma) && (17)\\ &= \int_{\mathcal{X}} \left( f_2(\boldsymbol{x}) \int_{\mathcal{X}} f_1(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \right) d\boldsymbol{x} && (18)\\ &= \int_{\mathcal{X}} \int_{\mathcal{X}} \frac{f_2(\boldsymbol{x}) f_1(\boldsymbol{y}) e^{-\frac{\|\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}-\boldsymbol{y}\|^2}{2\sigma^2(1+\|\boldsymbol{x}\|^2)}}}{(\sigma^2 2\pi(1+\|\boldsymbol{x}\|^2))^{\frac{n}{2}}} d\boldsymbol{y} d\boldsymbol{x}. && (19) \end{aligned}$$

## 6. Computation of the Integral Transform

Kernels can offer computational efficiency when computing the smoothed objective (3).

If the kernel $u$ is affine or one of its special cases, then it is a Gaussian form[5] in variable $\boldsymbol{y}$ according to Table 1. In such cases, expressing $f_1$ by Gaussian Basis Functions[6], piecewise constant or piecewise polynomial forms leads to a closed form of the integral transform. Details are provided in sections 6.1 and 6.2.

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 (6) compared to the original form (3). For example, 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 $\dim(\boldsymbol{y}) = 2$.

### 6.1. Gaussian RBF Representation of $f_1$

The following result addresses the representation of $f_1$ 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.*

$$\int_{\mathcal{X}} f_1(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} = \sum_{i=1}^{p} a_i \left(\frac{\delta_i}{\sqrt{\delta_i^2 + s^2}}\right)^n e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{\tau}\|^2}{2(\delta_i^2 + s^2)}}\,.$$

See the appendix for a proof.

[5] We say a kernel is Gaussian in $\boldsymbol{y}$ when it can be written as $u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) = k(\boldsymbol{\tau}(\boldsymbol{\theta}, \boldsymbol{x}) - \boldsymbol{y};\, s^2(\boldsymbol{\theta}, \boldsymbol{x}))$, where $s : \Theta \times \mathcal{X} \to \mathbb{R}_+$ is an arbitrary map and the maps $\boldsymbol{\tau}$ and $s$ are *independent* of $\boldsymbol{y}$.

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

### 6.2. Piecewise Constant Representation of $f_1$

The following result addresses the representation of $f_1$ as piecewise constant; the extension to piecewise polynomial is straightforward.

**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 $u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$ has the form $K(\boldsymbol{q}_{\boldsymbol{\tau}}(\boldsymbol{\theta}) - \boldsymbol{y}; \mathrm{diag}(s_1^2, \cdots, s_n^2))$, where $\boldsymbol{q}_{\boldsymbol{\tau}} : \Theta \to \mathbb{R}^n$ is some map. Then the following identity holds:*

$$\begin{aligned} &\int_{\mathcal{X}^\dagger} f_1(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\, d\boldsymbol{y} \\ = &\prod_{k=1}^n \frac{1}{2}\Big( \mathrm{erf}\,(\frac{q_{\boldsymbol{\tau}\,k} - \overline{x_k}}{\sqrt{2} s_k}) - \mathrm{erf}\,(\frac{q_{\boldsymbol{\tau}\,k} - \underline{x_k}}{\sqrt{2} s_k})\Big). \end{aligned}$$

The proof uses separability of integrals for diagonal $K$.

## 7. Regularization

Regularization may compensate for the numerical instability caused by excessive smoothing of the objective function and *improve the well-posedness* of the task. The latter means if there are multiple transformations that lead to equally good alignments (e.g. when image content has *symmetries*), the regularization prefers the closest transformation to some given $\boldsymbol{\theta}_0$. This makes existence of a *unique global optimum* more presumable. We achieve these goals by replacing $f_1$ with the following regularized version:

$$\tilde{f}_1(\boldsymbol{\tau}(\,\cdot\,,\cdot\,), \boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\theta}_0, r) \triangleq k(\boldsymbol{\theta} - \boldsymbol{\theta}_0; r^2) f_1(\boldsymbol{\tau}(\boldsymbol{x}; \boldsymbol{\theta})). \quad (20)$$

Regularization shrinks the signal $f_1$ for peculiar transformations with very large $\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|$. Typically $\boldsymbol{\theta}_0$ is set to the identity transformation $\boldsymbol{\tau}(\boldsymbol{x}; \boldsymbol{\theta}_0) = \boldsymbol{x}$. Using (20), the regularized objective function can be written as below:

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

Consequently, the smoothed *regularized* objective is as follows:

$$\tilde{z}(\boldsymbol{\theta}, \boldsymbol{\theta}_0, r, \sigma) \triangleq [\tilde{h}(\,\cdot\,, \boldsymbol{\theta}_0, r) \star k(\,\cdot\,; \sigma^2)](\boldsymbol{\theta}). \quad (21)$$

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.*

$$\begin{aligned} &\tilde{z}(\boldsymbol{\theta}, \boldsymbol{\theta}_0, r, \sigma) && (22)\\ = &[\tilde{h}(\,\cdot\,, \boldsymbol{\theta}_0, r) \star k(\,\cdot\,; \sigma^2)](\boldsymbol{\theta}) && (23)\\ = &\int_{\mathcal{X}} \Big( k(\boldsymbol{\theta} - \boldsymbol{\theta}_0; r^2 + \sigma^2) f_2(\boldsymbol{x}) \ldots && (24)\\ &\cdot \int_{\mathcal{X}} \Big( f_1(\boldsymbol{y}) u_{\boldsymbol{\tau}, \frac{r\sigma}{\sqrt{r^2+\sigma^2}}}(\frac{r^2\boldsymbol{\theta} + \sigma^2\boldsymbol{\theta}_0}{r^2 + \sigma^2}, \boldsymbol{x}, \boldsymbol{y}) \Big)\, d\boldsymbol{y} \Big) d\boldsymbol{x}\,. \end{aligned}$$

See the appendix for the proof.

## 8. Experiments

We evaluate the performance of the proposed scheme for smoothing the alignment objective function against traditional Gaussian blurring and no blurring at all. We use the images provided by [13] (see figure 3-top). This dataset consists of five planar scenes, each having six different views of increasingly dramatic transformations.

For the proposed method, we use the homography kernel. The goal is to maximize the correlation between a pair of views by transforming one to the other. The local maximization in Algorithm 1 and the other methods used for comparison is achieved by a coordinate ascent method[7] with a naive line search. The images are represented by a piecewise constant model. The integral transform in Algorithm 1 is approximated by the Laplace method.

Pixel coordinates were normalized to range in $[-1, 1]$. Images $f_1$ and $f_2$ were converted to grayscale and were subtracted by their joint mean (i.e. $(\bar{f}_1 + \bar{f}_2)/2$, where $\bar{f}_1$ is the average intensity of $f_1$) as a preprocessing step. The sequence of $\sigma$ (for both the proposed kernel and Gaussian kernel) starts from $\sigma = 0.1$, and is multiplied by $2/3$ in each iteration of algorithm until it falls below $0.0001$. The initial transformation $\boldsymbol{\theta}_0$ was set to the identity, i.e. $\boldsymbol{A}_0 = \boldsymbol{I}$ and $\boldsymbol{b}_0 = \boldsymbol{c}_0 = \boldsymbol{0}$. Since the initial $\sigma$ is large and the images lack significant areas of symmetry, no regularization was used.

The performance of these methods is summarized in figure 3-bottom. Each plot corresponds to one of the scenes in the dataset. For each scene, there is one rectified view that is used as $f_1$. The rest of five 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 a much higher NCC value[9].

[7] A block coordinate ascent is performed by partitioning the 8 parameters of homography to three classes, those that comprise $\boldsymbol{A}$, $\boldsymbol{b}$ and $\boldsymbol{c}$. This improves numerical stability because the sensitivity of parameters within each partition are similar.

[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.

[9] The code for reproducing our results is available at http://perception.csl.illinois.edu/smoothing.

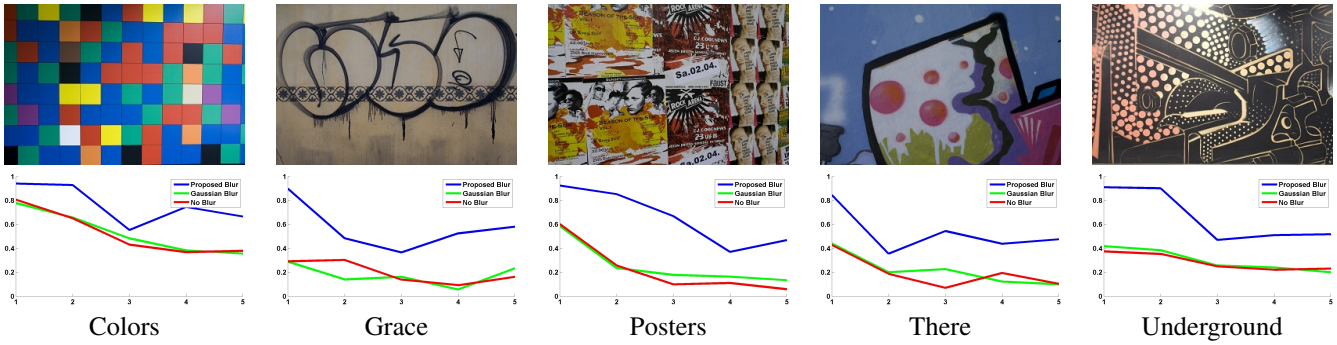



Figure 3. **Top**: Representative rectified views from the dataset provided in [13]. **Bottom**: NCC value after alignment. Horizontal axis is the view index (increasing in complexity) of the scene. Four views are used for each scene, each one being as $f_2$ and compared against $f_1$, which is a rectified view in the dataset.

## 9. Conclusion & Future Directions

This paper studied the problem of image smoothing for the purpose of alignment by direct intensity-based methods. We argued that the use of traditional Gaussian image blurring, mainly inspired by the work of Lucas and Kanade [29], may not be suitable for non-displacement motions. Instead, we suggested directly smoothing the alignment objective function. This led to a rigorous derivation of spatially varying kernels required for smoothing the objective function of common model-based alignment tasks including affine and homography models.

The derivation process of the kernels in this paper may provide some insights for blur kernels in other tasks such as image deblurring, motion from blur, matching, optical flow, etc. For example, in image deblurring, the blur caused by the motion of the camera or by scene motion typically leads to spatially varying blur. The estimation of such kernels is very challenging [10, 44, 16]. Yet if the motion is close to the models discussed in this paper, our results may provide new insights for estimation of the blur kernel. Similarly, our kernels could be relevant to tasks involving motion blur [12], due to the physical relationship between motion estimation and blur estimation [11]. The coarse-to-fine scheme is a classic and very effective way to escape from poor local minima in optical flow estimation [2, 43]. Using the proposed kernels may boost the quality of the computed solution.

Another possible application which may benefit from our proposed kernels is visual detection and recognition. Heuristic spatially-varying kernels [7, 40] have been successfully utilized in face detection [8] and object recognition [6, 15]. Thus, our results may provide new perspective on using blur kernels for such tasks in a more principled way. Another related machinery for visual recognition tasks is convolutional deep architectures [22, 23, 30, 34, 48]. These methods apply learnable convolution filters to the scale-space representation of the images, hence gain translation and scale invariance. Utilizing the proposed kernels instead of traditional convolutional filters and scale-space representation between layers might extend the invariance of these methods to a broader range of transformations.

Finally there is a lot of room to improve the computational efficiency of using the proposed kernels. In this work, the integral transforms are evaluated on a dense grid. However, since the kernels are smooth and localized in space, one might be able to get a good approximate of the integral transform by merely evaluating it at a small subset of image points.

## Acknowledgement

We thank John Wright (Columbia University) and Vadim Zharnitsky (UIUC) for fruitful discussions and anonymous reviewers for their thoughtful comments. Hossein Mobahi was supported by the CSE PhD fellowship of UIUC. This research was partially supported by ONR N00014-09-1-0230, NSF CCF 09-64215, and NSF IIS 11-16012.

## References

[1] E. L. Allgower and K. Georg. *Numerical Continuation Methods: An Introduction*. Springer-Verlag, 1990. 2

[2] L. Alvarez, J. Weickert, and J. Sánchez. A scale-space approach to nonlocal optical flow calculations. Scale-Space'99, pages 235–246, 1999. 7

[3] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56:221–255, 2004. 1

[4] Y. Bengio. *Learning Deep Architectures for AI*. Now Publishers Inc., 2009. 2

[5] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *International Conference on Machine Learning, ICML*, 2009. 2

[6] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. volume 1 of *CVPR'05*, pages 26–33, 2005. 7

[7] A. C. Berg and J. Malik. Geometric blur for template matching. CVPR'01, pages 607–614, 2001. 2, 5, 7

[8] 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. CVPR, 2004. 7

[9] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987. 2

[10] A. Chakrabarti, T. Zickler, and W. T. Freeman. Analyzing spatially-varying blur. CVPR'10, 2010. 7

[11] W.-G. Chen, N. Nandhakumar, and W. N. Martin. Image motion estimation from motion smear-a new computational model. *IEEE PAMI*, 18(4):412–425, 1996. 7

[12] T. S. Cho, A. Levin, F. Durand, and W. T. Freeman. Motion blur removal with orthogonal parabolic exposures. In *Int. Conf. in Comp. Photography (ICCP)*, 2010. 7

[13] K. Cordes, B. Rosenhahn, and J. Ostermann. Increasing the accuracy of feature evaluation benchmarks using differential evolution. In *IEEE Symp. on Differential Evolution (SDE)*, 2011. 6, 7

[14] M. Cox, S. Lucey, and S. Sridharan. Unsupervised alignment of image ensembles. Technical Report CI2CV-MC-20100609, Auto. Sys. Lab, CSIRO ICT Centre, 2010. 1

[15] A. Frome, F. Sha, Y. Singer, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. ICCV'07, 2007. 7

[16] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *ECCV*, 2010. 7

[17] M. Irani and P. Anandan. All about direct methods. In *Workshop on Vision Algorithms*, ICCV'99, pages 267–277. 1

[18] J. Jiang, S. Zheng, A. W. Toga, and Z. Tu. Learning based coarse-to-fine image registration. pages 1–7, 2008. 1

[19] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50(5):363–370, 1984. 2

[20] G. Konidaris and A. Barto. Autonomous shaping: knowledge transfer in reinforcement learning. ICML '06, pages 489–496, 2006. 2

[21] M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *NIPS*, pages 1189–1197. Curran Associates, Inc., 2010. 2

[22] Y. Lecun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In *Int. Symp. on Circuits and Systems (ISCAS'10)*, 2010. 7

[23] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML'09*, pages 609–616, 2009. 7

[24] M. Lefebure and L. D. Cohen. Image registration, optical flow and local rigidity. *J. Math. Imaging Vis.*, 14:131–147, 2001. 1

[25] 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*, 1994. 2

[26] T. Lindeberg. Generalized gaussian scale-space axiomatics comprising linear scale-space, affine scale-space and spatio-temporal scale-space. *J. Math. Imaging Vis.*, 40:36–81, 2011. 2

[27] T. Lindeberg and J. Garding. Shape-adapted smoothing in estimation of 3-d depth cues from affine distortions of local 2-d brightness structure. In *Image and Vision Computing*, pages 389–400, 1994. 2

[28] D. G. Lowe. Object recognition from local scale-invariant features. ICCV'99, pages 1150–1157, 1999. 1

[29] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. IJCAI'81, pages 674–679, 1981. 1, 3, 7

[30] H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. ICML'09, pages 737–744, 2009. 7

[31] H. Mobahi, Z. Zhou, A. Y. Yang, and Y. Ma. Holistic 3d reconstruction of urban structures from low-rank textures. In *3DRR Workshop*, ICCV'11, pages 593–600, 2011. 1

[32] G. Osterberg. *Topography of the layer of rods and cones in the human retina*. Acta ophthalmologica: Supplementum. Levin & Munksgaard, 1935. 2

[33] L. Piela, J. Kostrowicki, and H. A. Scheraga. On the multiple-minima problem in the conformational analysis of molecules. *Journal of Physical Chemistry*, 93(8):3339–3346, 1989. 2

[34] M. Ranzato, J. Susskind, V. Mnih, and G. Hinton. On deep generative models with applications to recognition. In *CVPR'11*, 2011. 7

[35] K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. pages 2210–2239, 1998. 2

[36] L. Sevilla-Lara and E. Learned-Miller. Distribution fields for tracking. CVPR'12, 2012. 2

[37] P. Simard, Y. LeCun, and J. S. Denker. Efficient pattern recognition using a new transformation distance. NIPS'92, pages 50–58, 1993. 1

[38] V. I. Spitkovsky, H. Alshawi, and D. Jurafsky. From Baby Steps to Leapfrog: How "Less is More" in unsupervised dependency parsing. In *Proc. of NAACL-HLT*, 2010. 2

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

[40] E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. CVPR'08, 2008. 2, 7

[41] F. d. l. Torre and M. J. Black. Robust parameterized component analysis. ECCV'02, pages 653–669, 2002. 1

[42] N. Vasconcelos and A. Lippman. Multiresolution tangent distance for affine-invariant classification. pages 843–849, 1998. 1

[43] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers. Statistical and geometrical approaches to visual motion analysis. chapter An Improved Algorithm for TV-L1 Optical Flow, pages 23–45. 2009. 7

[44] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 7

[45] D. V. Widder. *The Heat Equation*. Academic Press, 1975. 4

[46] A. P. Witkin. Scale-Space Filtering. volume 2 of *IJCAI'83*, pages 1019–1022, Karlsruhe, 1983. 2

[47] A. L. Yuille and T. A. Poggio. Scaling theorems for zero crossings. *IEEE PAMI*, 8(1):15–25, 1986. 2

[48] M. D. Zeiler, D. Kirshnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *CVPR'10*, 2010. 7

# Seeing through the Blur

## The Supplementary Appendix

Hossein Mobahi
hmobahi2@illinois.edu
Department of Computer Science
University of Illinois at Urbana Champaign (UIUC)
Urbana, IL

Charles Lawrence Zitnick
larryz@microsoft.com
Interactive Visual Media Group
Microsoft Research
Redmond, WA

Yi Ma
yima@illinois.edu , mayi@microsoft.com
Department of Electrical and Computer Engineering
University of Illinois at Urbana Champaign (UIUC)
Urbana, IL
&
Visual Computing Group
Microsoft Research Asia
Beijing, China

April 2012

# 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 $\nabla$ means $\nabla_{\boldsymbol{x}}$. Finally, $\star$ denotes convolution operator in space $\Theta$.

# 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, in 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]**

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

**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 $\boldsymbol{\tau} : \Theta \times \mathcal{X} \times \Theta \to \mathcal{X}$, 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*,

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

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.

$$\tilde{z}(\boldsymbol{\theta}, \boldsymbol{\theta}_0, r, \sigma) \triangleq [\tilde{h}(\,\cdot\,, \boldsymbol{\theta}_0, r) \star k(\,\cdot\,; \sigma^2)](\boldsymbol{\theta}). \tag{3}$$

# 3 Proof of Results in the Paper

**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**

$$\begin{aligned}
& k(\boldsymbol{\tau}-\boldsymbol{\mu}_1;\sigma_1^2)\ k(\boldsymbol{\tau}-\boldsymbol{\mu}_2;\sigma_2^2) \\
= \ & \frac{1}{(\sigma_1\sqrt{2\pi})^m} e^{-\frac{\|\boldsymbol{\tau}-\boldsymbol{\mu}_1\|^2}{2\sigma_1^2}}\ \frac{1}{(\sigma_2\sqrt{2\pi})^m} e^{-\frac{\|\boldsymbol{\tau}-\boldsymbol{\mu}_2\|^2}{2\sigma_2^2}} \\
= \ & \frac{1}{(2\pi\sigma_1\sigma_2)^m} e^{-\frac{\|\boldsymbol{\tau}-\boldsymbol{\mu}_1\|^2}{2\sigma_1^2}-\frac{\|\boldsymbol{\tau}-\boldsymbol{\mu}_2\|^2}{2\sigma_2^2}} \\
= \ & \frac{1}{(2\pi\sigma_1\sigma_2)^m} e^{-\frac{\|\boldsymbol{\tau}-\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}(\frac{\boldsymbol{\mu}_1}{\sigma_1^2}+\frac{\boldsymbol{\mu}_2}{\sigma_2^2})\|^2}{2\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}}-\frac{\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|^2}{2(\sigma_1^2+\sigma_2^2)}} \qquad (4) \\
= \ & \frac{e^{-\frac{\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|^2}{2(\sigma_1^2+\sigma_2^2)}}}{(\sqrt{2\pi(\sigma_1^2+\sigma_2^2)})^m} k(\boldsymbol{\tau}-\frac{\sigma_2^2\boldsymbol{\mu}_1+\sigma_1^2\boldsymbol{\mu}_2}{\sigma_1^2+\sigma_2^2};\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2})\,.
\end{aligned}$$

Note that (4) is derived by completing the square.

□

**Proposition 1** *The following choice of $u$,*

$$\begin{aligned}
& u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y}) \\
= \ & \frac{1}{(2\pi)^n}\int_\Omega \Big(\int_\Theta e^{i\boldsymbol{\omega}^T(\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{t})-\boldsymbol{y})} k(\boldsymbol{t}-\boldsymbol{\theta};\sigma^2)\,d\boldsymbol{t}\Big)\,d\boldsymbol{\omega} \qquad (5)
\end{aligned}$$

*is a solution to the definition of kernel provided in (2). 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 $\sigma$, which in our case is simply an isotropic Gaussian with bandwidth $\sigma$.*

**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 $\mathcal{X}=\mathbb{R}^n$).

$$
\begin{aligned}
& [f(\boldsymbol{\tau}(\boldsymbol{x},.)) \star k(\,.\,;\sigma^2)](\boldsymbol{\theta}) \\
= \; & [\Big(\frac{1}{(2\pi)^n}\int_{\Omega} \hat{f}(\boldsymbol{\omega}) e^{i\boldsymbol{\omega}^T\boldsymbol{\tau}(\boldsymbol{x},.)}\, d\boldsymbol{\omega}\Big) \star k(\,.\,;\sigma^2)](\boldsymbol{\theta}) \\
= \; & \frac{1}{(2\pi)^n}\int_{\Theta}\left(\int_{\Omega} \hat{f}(\boldsymbol{\omega}) e^{i\boldsymbol{\omega}^T\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{t})}\, d\boldsymbol{\omega}\right) k(\boldsymbol{t}-\boldsymbol{\theta};\sigma^2)\, d\boldsymbol{t} \\
= \; & \frac{1}{(2\pi)^n}\int_{\Omega} \hat{f}(\boldsymbol{\omega})\left(\int_{\Theta} e^{i\boldsymbol{\omega}^T\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{t})} k(\boldsymbol{t}-\boldsymbol{\theta};\sigma^2)\, d\boldsymbol{t}\right) d\boldsymbol{\omega} \\
= \; & \frac{1}{(2\pi)^n}\int_{\mathcal{X}} f(\boldsymbol{y})\left(\int_{\Omega} e^{-i\boldsymbol{\omega}^T\boldsymbol{y}}\left(\int_{\Theta} e^{i\boldsymbol{\omega}^T\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{t})} k(\boldsymbol{t}-\boldsymbol{\theta};\sigma^2)\, d\boldsymbol{t}\right) d\boldsymbol{\omega}\right) d\boldsymbol{y} \quad (6) \\
= \; & \frac{1}{(2\pi)^n}\int_{\mathcal{X}} f(\boldsymbol{y})\left(\int_{\Omega}\int_{\Theta} e^{i\boldsymbol{\omega}^T(\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{t})-\boldsymbol{y})} k(\boldsymbol{t}-\boldsymbol{\theta};\sigma^2)\, d\boldsymbol{t}\, d\boldsymbol{\omega}\right) d\boldsymbol{y} \\
= \; & \int_{\mathcal{X}} f(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y})\, d\boldsymbol{y}\,, \quad (7)
\end{aligned}
$$

where (6) uses the *Parseval* theorem, and (7) uses proposition's assumption (5). □

**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.*

$$
\begin{aligned}
& \int_{\mathcal{X}} f_1(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y})\, d\boldsymbol{y} \\
= \; & \sum_{i=1}^{p} a_i \Big(\frac{\delta_i}{\sqrt{\delta_i^2+s^2}}\Big)^n e^{-\frac{\|\boldsymbol{x}_i-\boldsymbol{\tau}\|^2}{2(\delta_i^2+s^2)}}\,.
\end{aligned}
$$

**Proof**

$$
\begin{aligned}
& \int_{\mathcal{X}} f_1(\boldsymbol{y}) u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{y})\, d\boldsymbol{y} \\
= & \int_{\mathcal{X}} f_1(\boldsymbol{y}) k(\boldsymbol{\tau}-\boldsymbol{y}; s^2)\, d\boldsymbol{y} \\
= & \int_{\mathbb{R}^n} \left( \sum_{k=1}^{p} a_k \phi(\boldsymbol{y}; \boldsymbol{x}_k, \delta_k) \right) k(\boldsymbol{\tau}-\boldsymbol{y}; s^2)\, d\boldsymbol{y} \\
= & \sum_{k=1}^{p} a_k \left( \int_{\mathbb{R}^n} \phi(\boldsymbol{y}; \boldsymbol{x}_k, \delta_k) k(\boldsymbol{\tau}-\boldsymbol{y}; s^2)\, d\boldsymbol{y} \right) \\
= & \sum_{k=1}^{p} a_k (\delta_k \sqrt{2\pi})^n \left( \int_{\mathbb{R}^n} k(\boldsymbol{y}-\boldsymbol{x}_k; \delta_k^2) k(\boldsymbol{\tau}-\boldsymbol{y}; s^2)\, d\boldsymbol{y} \right) \\
= & \sum_{k=1}^{p} a_k (\delta_k \sqrt{2\pi})^n \left( \int_{\mathbb{R}^n} \frac{e^{-\frac{\|\boldsymbol{x}_k-\boldsymbol{\tau}\|^2}{2(\delta_k^2+s^2)}}}{(\sqrt{2\pi(\delta_k^2+s^2)})^n} k(\boldsymbol{y} - \frac{s^2\boldsymbol{x}_k + \delta_k^2\boldsymbol{\tau}}{\delta_k^2+s^2}; \frac{\delta_k^2 s^2}{\delta_k^2+s^2})\, d\boldsymbol{y} \right) \quad (8) \\
= & \sum_{k=1}^{p} a_k (\frac{\delta_k}{\sqrt{\delta_k^2+s^2}})^n e^{-\frac{\|\boldsymbol{x}_k-\boldsymbol{\tau}\|^2}{2(\delta_k^2+s^2)}} \left( \int_{\mathbb{R}^n} k(\boldsymbol{y} - \frac{s^2\boldsymbol{x}_k + \delta_k^2\boldsymbol{\tau}}{\delta_k^2+s^2}; \frac{\delta_k^2 s^2}{\delta_k^2+s^2})\, d\boldsymbol{y} \right) \\
= & \sum_{k=1}^{p} a_k (\frac{\delta_k}{\sqrt{\delta_k^2+s^2}})^n e^{-\frac{\|\boldsymbol{x}_k-\boldsymbol{\tau}\|^2}{2(\delta_k^2+s^2)}}\,,
\end{aligned}
$$

where in (8) we use the Gaussian product result from proposition 0. □

**Proposition 3** *The regularized objective function $\tilde{z}(\boldsymbol{\theta}, \boldsymbol{\theta}_0, r, \sigma)$ can be written using transformation kernels as follows.*

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

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

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

Thus, regularized objective function from (3) leads to the following result.

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

□

# 4 Derivation of Affine and Homography Kernels

**Proposition 4** *Suppose $n \geq 1$ is some integer and let $t : \mathbb{R}^n \to (\mathbb{R} - \{0\})$. Then for any real $n \times n$ matrix $\boldsymbol{A}^\dagger$ and any real $n \times 1$ vectors $\boldsymbol{b}^\dagger$, $\boldsymbol{x}$, and $\boldsymbol{y}$, the following identity holds:*

$$
\begin{aligned}
& (2\pi)^{-n} \int_{\Omega} \int_{\mathcal{A}} \int_{\mathcal{B}} e^{\frac{i\boldsymbol{\omega}^T \boldsymbol{A}\boldsymbol{x} + i\boldsymbol{\omega}^T \boldsymbol{b}}{t(\boldsymbol{x})} - i\boldsymbol{\omega}^T \boldsymbol{y}} k_\sigma(\boldsymbol{A} - \boldsymbol{A}^\dagger) k_\sigma(\boldsymbol{b} - \boldsymbol{b}^\dagger) \, d\boldsymbol{A} \, d\boldsymbol{b} \, d\boldsymbol{\omega} \\
= \; & k(\frac{\boldsymbol{A}^\dagger \boldsymbol{x} + \boldsymbol{b}^\dagger}{t(\boldsymbol{x})} - \boldsymbol{y}; \frac{\sigma^2 (1 + \|\boldsymbol{x}\|^2)}{t^2(\boldsymbol{x})}) \, ,
\end{aligned}
$$

*where $\Omega = \mathcal{B} = \mathbb{R}^n$ and $\mathcal{A} = \mathbb{R}^n \times \mathbb{R}^n$.*

**Proof** We proceed as below,

$$
\begin{aligned}
& \int_{\mathcal{A}}\int_{\mathcal{B}} e^{\frac{i\boldsymbol{\omega}^T\boldsymbol{A}\boldsymbol{x}+i\boldsymbol{\omega}^T\boldsymbol{b}}{t(\boldsymbol{x})}-i\boldsymbol{\omega}^T\boldsymbol{y}} k_\sigma(\boldsymbol{A}-\boldsymbol{A}^\dagger)k_\sigma(\boldsymbol{b}-\boldsymbol{b}^\dagger)\,d\boldsymbol{A}\,d\boldsymbol{b} \\
= & \int_{\mathcal{A}}\int_{\mathcal{B}} e^{\frac{i\sum_{j=1}^n\sum k=1^n w_j a_{jk}x_k+i\sum_{k=1}^n\omega_k b_k}{t(\boldsymbol{x})}-i\sum_{k=1}^n\omega_k y_k} k_\sigma(\boldsymbol{A}-\boldsymbol{A}^\dagger)k_\sigma(\boldsymbol{b}-\boldsymbol{b}^\dagger)\,d\boldsymbol{A}\,d\boldsymbol{b} \\
= & e^{-i\sum_{j=1}^n\omega_j y_j}\Pi_{j=1}^n\Big(\int_{\mathcal{B}_j} e^{\frac{i\omega_j}{t(\boldsymbol{x})}b_j}k_\sigma(b_j-b_j^\dagger)\,db_j\Big) \\
& \Pi_{j=1}^n\Pi_{k=1}^n\Big(\int_{\mathcal{A}_{jk}} e^{\frac{iw_jx_k}{t(\boldsymbol{x})}a_{jk}}k_\sigma(a_{jk}-a_{jk}^\dagger)\,da_{jk}\Big) \\
= & e^{-i\sum_{j=1}^n\omega_j y_j}\Pi_{j=1}^n\Big(e^{\frac{i\omega_j}{t(\boldsymbol{x})}b_j^\dagger+\frac{1}{2}\sigma^2(\frac{i\omega_j}{t(\boldsymbol{x})})^2}\Big) \\
& \Pi_{j=1}^n\Pi_{k=1}^n\Big(e^{\frac{iw_jx_k}{t(\boldsymbol{x})}a_{jk}^\dagger+\frac{1}{2}\sigma^2(\frac{iw_jx_k}{t(\boldsymbol{x})})^2}\Big),
\end{aligned}
\tag{9}
$$

where (9) uses the identity $\int_{\mathbb{R}} e^{ax}k_\sigma(x^\dagger-x)\,dx = e^{ax^\dagger+\frac{1}{2}\sigma^2a^2}$. We proceed by factorizing $\omega_j$ and $\omega_j^2$ in the exponent as the following.

$$
\begin{aligned}
= & \int_{\mathcal{A}}\int_{\mathcal{B}} e^{\frac{i\boldsymbol{\omega}^T\boldsymbol{A}\boldsymbol{x}+i\boldsymbol{\omega}^T\boldsymbol{b}}{t(\boldsymbol{x})}-i\boldsymbol{\omega}^T\boldsymbol{y}} k_\sigma(\boldsymbol{A}-\boldsymbol{A}^\dagger)k_\sigma(\boldsymbol{b}-\boldsymbol{b}^\dagger)\,d\boldsymbol{A}\,d\boldsymbol{b} \\
= & e^{-i\sum_{j=1}^n\omega_j y_j}\Pi_{j=1}^n\Big(e^{\frac{i\omega_j}{t(\boldsymbol{x})}b_j^\dagger+\frac{1}{2}\sigma^2(\frac{i\omega_j}{t(\boldsymbol{x})})^2}\Big) \\
& \Pi_{j=1}^n\Pi_{k=1}^n\Big(e^{\frac{iw_jx_k}{t(\boldsymbol{x})}a_{jk}^\dagger+\frac{1}{2}\sigma^2(\frac{iw_jx_k}{t(\boldsymbol{x})})^2}\Big) \\
= & \Pi_{j=1}^n e^{-i\omega_jy_j+\frac{i\omega_j}{t(\boldsymbol{x})}b_j^\dagger+\frac{1}{2}\sigma^2(\frac{i\omega_j}{t(\boldsymbol{x})})^2+\frac{i(\sum_{k=1}^n a_{jk}^\dagger x_k)}{t(\boldsymbol{x})}w_j+\frac{1}{2}\sigma^2\frac{\sum_{k=1}^n x_k^2}{t^2(\boldsymbol{x})}(iw_j)^2} \\
= & \Pi_{j=1}^n e^{i\omega_j\frac{b_j^\dagger-y_j+\sum_{k=1}^n a_{jk}^\dagger x_k}{t(\boldsymbol{x})}-\frac{1}{2}w_j^2\frac{\sigma^2(1+\sum_{k=1}^n x_k^2)}{t^2(\boldsymbol{x})}}
\end{aligned}
\tag{10}
$$

Now dividing both sides by $(2\pi)^{-n}$ and integrating w.r.t. $\boldsymbol{\omega}$, we obtain the following.

$$
\begin{aligned}
= & (2\pi)^{-n}\int_\Omega\int_{\mathcal{A}}\int_{\mathcal{B}} e^{\frac{i\boldsymbol{\omega}^T\boldsymbol{A}\boldsymbol{x}+i\boldsymbol{\omega}^T\boldsymbol{b}}{t(\boldsymbol{x})}-i\boldsymbol{\omega}^T\boldsymbol{y}} k_\sigma(\boldsymbol{A}-\boldsymbol{A}^\dagger)k_\sigma(\boldsymbol{b}-\boldsymbol{b}^\dagger)\,d\boldsymbol{A}\,d\boldsymbol{b}\,d\boldsymbol{\omega} \qquad (11) \\
= & (2\pi)^{-n}\int_\Omega \Pi_{j=1}^n e^{i\omega_j\frac{b_j^\dagger-y_jt(\boldsymbol{x})+\sum_{k=1}^n a_{jk}^\dagger x_k}{t(\boldsymbol{x})}-\frac{1}{2}w_j^2\frac{\sigma^2(1+\sum_{k=1}^n x_k^2)}{t^2(\boldsymbol{x})}}\,d\boldsymbol{\omega} \\
= & \Pi_{j=1}^n\Big(\int_{\Omega_j}(2\pi)^{-1}e^{i\omega_j\frac{b_j^\dagger-y_jt(\boldsymbol{x})+\sum_{k=1}^n a_{jk}^\dagger x_k}{t(\boldsymbol{x})}-\frac{1}{2}w_j^2\frac{\sigma^2(1+\sum_{k=1}^n x_k^2)}{t^2(\boldsymbol{x})}}\,d\omega_j\Big) \\
= & \Pi_{j=1}^n\Big(k(\frac{b_j^\dagger-y_jt(\boldsymbol{x})+\sum_{k=1}^n a_{jk}^\dagger x_k}{t(\boldsymbol{x})};\frac{\sigma^2(1+\sum_{k=1}^n x_k^2)}{t^2(\boldsymbol{x})})\Big) \qquad (12) \\
= & k(\frac{\boldsymbol{A}^\dagger\boldsymbol{x}+\boldsymbol{b}^\dagger}{t(\boldsymbol{x})}-\boldsymbol{y};\frac{\sigma^2(1+\|\boldsymbol{x}\|^2)}{t^2(\boldsymbol{x})}),
\end{aligned}
$$

where (12) uses the identity $(2\pi)^{-1}\int_{\mathbb{R}} e^{i\omega x-\frac{\omega^2}{2y}}\,d\omega = k(x;y)$ for $y>0$.

□

**Lemma 5 (Derivation of Affine Kernel)** *Suppose $\boldsymbol{x}\in\mathbb{R}^n$, where $n\geq 1$ is some integer. The kernel $u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}^\dagger,\boldsymbol{x},\boldsymbol{y})$ for the affine transformation $\boldsymbol{\tau}(\boldsymbol{x})=\boldsymbol{A}^\dagger\boldsymbol{x}+\boldsymbol{b}^\dagger$ is equal to the following expression:*

$$k\Big(\boldsymbol{A}^\dagger\boldsymbol{x}+\boldsymbol{b}^\dagger-\boldsymbol{y};\sigma^2(1+\|\boldsymbol{x}\|^2)\Big)\,,$$

*where $\boldsymbol{A}^\dagger$ is any $n\times n$ real matrix and $\boldsymbol{b}^\dagger$ and $\boldsymbol{y}$ are any $n\times 1$ real vectors.*

**Proof** By (5) from Proposition 1, any $u$ that satisfies the following equation is a kernel for $\boldsymbol{\tau}$.

$$u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}^\dagger,\boldsymbol{x},\boldsymbol{y}) \triangleq \frac{1}{(2\pi)^n}\int_\Omega\left(\int_\Theta e^{i\boldsymbol{\omega}^T(\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{\theta})-\boldsymbol{y})}k_\sigma(\boldsymbol{\theta}-\boldsymbol{\theta}^\dagger)\,d\boldsymbol{\theta}\right)d\boldsymbol{\omega}\,.$$

We proceed with computing $u$ as follows:

$$\begin{aligned}
& u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}^\dagger,\boldsymbol{x},\boldsymbol{y}) \\
\triangleq\;& \frac{1}{(2\pi)^n}\int_\Omega\left(\int_\Theta e^{i\boldsymbol{\omega}^T(\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{\theta})-\boldsymbol{y})}k_\sigma(\boldsymbol{\theta}-\boldsymbol{\theta}^\dagger)\,d\boldsymbol{\theta}\right)d\boldsymbol{\omega} \\
=\;& \frac{1}{(2\pi)^n}\int_\Omega\int_{\mathcal{A}}\int_{\mathcal{B}} e^{i\boldsymbol{\omega}^T(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}-\boldsymbol{y})}k_\sigma(\boldsymbol{A}-\boldsymbol{A}^\dagger)k_\sigma(\boldsymbol{b}-\boldsymbol{b}^\dagger)\,d\boldsymbol{A}\,d\boldsymbol{b}\,d\boldsymbol{\omega} \quad (13)\\
=\;& k\Big(\boldsymbol{A}^\dagger\boldsymbol{x}+\boldsymbol{b}^\dagger-\boldsymbol{y};\sigma^2(1+\|\boldsymbol{x}\|^2)\Big)\,,
\end{aligned}$$

where (13) applies Lemma 4 with the particular choice of $t(\boldsymbol{x})=1$.

□

**Proposition 6** *The following indefinite integral identities hold.*

$$\begin{aligned}
&\forall t\in\mathbb{R}\,,\; c\in\mathbb{R}\,,\; p_1\in\mathbb{R}\;\; p_2\in\mathbb{R}_{++}\;\;: \\
&\int e^{-p_2t^2+p_1t}\,dt = \frac{1}{2}\sqrt{\frac{\pi}{p_2}}e^{\frac{p_1^2}{4p_2}}\operatorname{erf}\Big(\frac{2p_2t-p_1}{2\sqrt{p_2}}\Big)+c \\
&\int te^{-p_2t^2+p_1t}\,dt = \frac{p_1}{4p_2\sqrt{p_2}}e^{\frac{p_1^2}{4p_2}}\sqrt{\pi}\operatorname{erf}\Big(\frac{2p_2t-p_1}{2\sqrt{p_2}}\Big)-\frac{1}{2p_2}e^{-p_2t^2+tp_1}+c \\
&\int t^2e^{-p_2t^2+p_1t}\,dt = \frac{\sqrt{\pi}}{8p_2^2\sqrt{p_2}}(2p_2+p_1^2)e^{\frac{p_1^2}{4p_2}}\operatorname{erf}\Big(\frac{2p_2t-p_1}{2\sqrt{p_2}}\Big)-\frac{p_1+2p_2t}{4p_2^2}e^{-p_2t^2+tp_1}+c\,.
\end{aligned}$$

**Proof** The correctness of these identities can be easily checked by differentiating RHS w.r.t. $t$ and observing that it becomes equal to the integrand of LHS. Remember $\frac{d}{dt}\operatorname{erf}(t)=\frac{2}{\sqrt{\pi}}e^{-t^2}$.

□

**Corollary 7** *The following definite integral identities hold.*

$$\begin{aligned}
&\forall \quad t \in \mathbb{R}\ ,\ p_1 \in \mathbb{R} \quad p_2 \in \mathbb{R}_{++} \quad : \\
&\int_{\mathbb{R}} e^{-p_2 t^2 + p_1 t}\, dt = \sqrt{\frac{\pi}{p_2}} e^{\frac{p_1^2}{4p_2}} \\
&\int_{\mathbb{R}} t e^{-p_2 t^2 + p_1 t}\, dt = \frac{p_1}{2p_2\sqrt{p_2}} e^{\frac{p_1^2}{4p_2}} \sqrt{\pi} \\
&\int_{\mathbb{R}} t^2 e^{-p_2 t^2 + p_1 t}\, dt = \frac{\sqrt{\pi}}{4p_2^2\sqrt{p_2}} (2p_2 + p_1^2) e^{\frac{p_1^2}{4p_2}}\ .
\end{aligned}$$

**Proof** Using the identities for their indefinite counterparts provided in Proposition 6, these definite integrals are easily computed by subtracting their value at the limit $t \to \pm\infty$. Note that $\lim_{t\to\pm\infty} \operatorname{erf}(t) = \pm 1$ and that $\lim_{t\to\pm\infty} f(t)\exp(-p_2 t^2 + p_1 t) = 0$, where $p_2 > 0$ and $f : \mathbb{R} \to \mathbb{R}$ is such that $f(t)$ is a polynomial in $t$. □

**Lemma 8 (Derivation of Homography Kernel)** *Suppose $\boldsymbol{x} \in \mathbb{R}^2$. The kernel $u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}^\dagger, \boldsymbol{x}, \boldsymbol{y})$ for the homography transformation $\boldsymbol{\tau}(\boldsymbol{x}) = (\boldsymbol{A}^\dagger \boldsymbol{x} + \boldsymbol{b}^\dagger)(1 + \boldsymbol{c}^{\dagger T} \boldsymbol{x})^{-1}$ is equal to the following expression:*

$$u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}^\dagger, \boldsymbol{x}, \boldsymbol{y}) = q e^{-p}\ ,$$

*where the auxiliary variables are as below:*

$$\begin{aligned}
z_0 &\triangleq \frac{1}{1 + \|\boldsymbol{x}\|^2} \\
z_1 &\triangleq 1 + \boldsymbol{x}^T \boldsymbol{c}^\dagger \\
\boldsymbol{v} &\triangleq \boldsymbol{A}^\dagger \boldsymbol{x} + \boldsymbol{b}^\dagger \\
q &\triangleq z_0 \frac{(z_0 \|\boldsymbol{x}\|^2 \boldsymbol{y}^T \boldsymbol{v} + z_1)^2 + \sigma^2 \|\boldsymbol{x}\|^2 (1 + z_0 \|\boldsymbol{x}\|^2 \|\boldsymbol{y}\|^2)}{2\pi\sigma^2 (1 + z_0 \|\boldsymbol{x}\|^2 \|\boldsymbol{y}\|^2)^{\frac{5}{2}}} \\
p &\triangleq \frac{\|z_1 \boldsymbol{y} - \boldsymbol{v}\|^2 + z_0 \|\boldsymbol{x}\|^2 (v_2 y_1 - v_1 y_2)^2}{2\sigma^2 (1 + \|\boldsymbol{x}\|^2 (1 + \|\boldsymbol{y}\|^2))}\ .
\end{aligned}$$

*Here $\boldsymbol{A}^\dagger$ is any $2 \times 2$ real matrix and $\boldsymbol{b}^\dagger$, $\boldsymbol{c}^\dagger$, and $\boldsymbol{y}$ are any $2 \times 1$ real vectors.*

**Proof** By (5) from Proposition 1, any $u$ that satisfies the following equation is a kernel for $\boldsymbol{\tau}$.

$$u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}^\dagger, \boldsymbol{x}, \boldsymbol{y}) \triangleq \frac{1}{(2\pi)^n} \int_\Omega \left( \int_\Theta e^{i\boldsymbol{\omega}^T (\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{\theta}) - \boldsymbol{y})} k_\sigma(\boldsymbol{\theta} - \boldsymbol{\theta}^\dagger)\, d\boldsymbol{\theta} \right) d\boldsymbol{\omega}\ .$$

We proceed with computing $u$ as follows:

$$
\begin{aligned}
& u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}^\dagger, \boldsymbol{x}, \boldsymbol{y}) \\
\triangleq\ & \frac{1}{(2\pi)^n} \int_\Omega \left( \int_\Theta e^{i\boldsymbol{\omega}^T(\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{\theta})-\boldsymbol{y})} k_\sigma(\boldsymbol{\theta}-\boldsymbol{\theta}^\dagger)\, d\boldsymbol{\theta} \right) d\boldsymbol{\omega} \\
=\ & \int_{\mathcal{C}} \Big( \frac{1}{(2\pi)^n} \int_\Omega \int_{\mathcal{A}} \int_{\mathcal{B}} e^{i\boldsymbol{\omega}^T(\frac{\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}}{1+\boldsymbol{c}^T\boldsymbol{x}}-\boldsymbol{y})} k_\sigma(\boldsymbol{A}-\boldsymbol{A}^\dagger) k_\sigma(\boldsymbol{b}-\boldsymbol{b}^\dagger)\, d\boldsymbol{A}\, d\boldsymbol{b}\, d\boldsymbol{\omega} \Big) \\
& k_\sigma(\boldsymbol{c}-\boldsymbol{c}^\dagger)\, d\boldsymbol{c} \\
=\ & \int_{\mathcal{C}} \left( k\Big( \frac{\boldsymbol{A}^\dagger\boldsymbol{x}+\boldsymbol{b}^\dagger}{1+\boldsymbol{c}^T\boldsymbol{x}} - \boldsymbol{y}; \frac{\sigma^2(1+\|\boldsymbol{x}\|^2)}{(1+\boldsymbol{c}^T\boldsymbol{x})^2} \Big) k_\sigma(\boldsymbol{c}-\boldsymbol{c}^\dagger) \right) d\boldsymbol{c} \qquad (14) \\
=\ & \int_{\mathcal{C}_2} \int_{\mathcal{C}_1} \Big( k\big( \frac{\boldsymbol{A}^\dagger\boldsymbol{x}+\boldsymbol{b}^\dagger}{1+\boldsymbol{c}^T\boldsymbol{x}} - \boldsymbol{y}; \frac{\sigma^2(1+\|\boldsymbol{x}\|^2)}{(1+\boldsymbol{c}^T\boldsymbol{x})^2} \big) k_\sigma(c_1 - c_1^\dagger) dc_1 \Big) k_\sigma(c_2 - c_2^\dagger) dc_2,
\end{aligned}
$$

where (14) applies Lemma 4 with the particular choice of $t(\boldsymbol{x}) = 1 + \boldsymbol{c}^T\boldsymbol{x}$, and $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2$ with $\mathcal{C}_1 = \mathcal{C}_2 = \mathbb{R}$.

We continue by first computing the inner integral, aka w.r.t. $c_1$. To reduce clutter, we introduce the following auxiliary variables which are *independent* of $c_1$.

$$
\begin{aligned}
\boldsymbol{v} &\triangleq \boldsymbol{A}^\dagger\boldsymbol{x} + \boldsymbol{b}^\dagger \\
s &\triangleq 1 + c_2 x_2 \\
z_0 &\triangleq \frac{1}{\sigma^2(1+\|\boldsymbol{x}\|^2)} \\
z_1 &\triangleq \frac{1}{2\pi\sigma\sqrt{2\pi}}.
\end{aligned}
$$

Now we proceed with integration w.r.t. $c_1$ as below.

$$
\begin{aligned}
& \int_{\mathcal{C}_1} k_\sigma(c_1 - c_1^\dagger)\, k\big( \frac{\boldsymbol{A}^\dagger\boldsymbol{x}+\boldsymbol{b}^\dagger}{1+\boldsymbol{c}^T\boldsymbol{x}} - \boldsymbol{y}; \frac{\sigma^2(1+\|\boldsymbol{x}\|^2)}{(1+\boldsymbol{c}^T\boldsymbol{x})^2} \big)\, dc_1 \\
=\ & \int_{\mathcal{C}_1} \frac{1}{\sqrt{2\pi}\sigma} \frac{(1+\boldsymbol{c}^T\boldsymbol{x})^2}{2\pi\sigma^2(1+\|\boldsymbol{x}\|^2)} e^{\frac{-(c_1-c_1^\dagger)^2}{2\sigma^2} - \frac{(1+\boldsymbol{c}^T\boldsymbol{x})^2}{2\sigma^2(1+\|\boldsymbol{x}\|^2)} \|\frac{\boldsymbol{A}^\dagger\boldsymbol{x}+\boldsymbol{b}^\dagger}{1+\boldsymbol{c}^T\boldsymbol{x}} - \boldsymbol{y}\|^2}\, dc_1 \\
=\ & z_0 z_1 \int_{\mathcal{C}_1} (q_0 + c_1 q_1 + c_1^2 q_2) e^{-p_2 c_1^2 + p_1 c_1 + p_0}\, dc_1 \\
=\ & z_0 z_1 \sqrt{\frac{\pi}{p_2}} e^{p_0 + \frac{p_1^2}{4p_2}} \Big( q_0 + q_1 \frac{p_1}{2p_2} + q_2 \frac{1}{4p_2^2}(2p_2 + p_1^2) \Big), \qquad (15)
\end{aligned}
$$

where (15) uses Corollary 7 with the particular choice of $p_i$ and $q_i$ for $i = 0, 1, 2$ as the following. Obviously the following $p_2$ satisfies $p_2 > 0$. Also not that $p_i$ and $q_i$ are independent of integration variable $c_1$.

$$
\begin{aligned}
q_0 &\triangleq s^2 \\
q_1 &\triangleq 2x_1 s \\
q_2 &\triangleq x_1^2 \\
p_0 &\triangleq -\frac{{c_1^\dagger}^2}{2\sigma^2} - \frac{z_0}{2}\|\boldsymbol{v} - s\boldsymbol{y}\|^2 \\
p_1 &\triangleq \frac{c_1^\dagger}{\sigma^2} + z_0\Big(x_1 \boldsymbol{y}^T(\boldsymbol{v} - s\boldsymbol{y})\Big) \\
p_2 &\triangleq \frac{1}{2\sigma^2} + z_0\frac{x_1^2\|\boldsymbol{y}\|^2}{2}\,.
\end{aligned}
$$

Combining (**??**) and (15) gives the following.

$$
\begin{aligned}
& u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}^\dagger, \boldsymbol{x}, \boldsymbol{y}) \\
= & \int_{\mathcal{C}_2} z_0 z_1 \sqrt{\frac{\pi}{p_2}} e^{p_0 + \frac{p_1^2}{4p_2}} \Big(q_0 + q_1\frac{p_1}{2p_2} + q_2\frac{1}{4p_2^2}(2p_2 + p_1^2)\Big)\, dc_2\,.
\end{aligned}
$$

We can compute the above integral in a similar fashion as shown below.

$$
\begin{aligned}
& u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}^\dagger, \boldsymbol{x}, \boldsymbol{y}) \\
= & z_0 z_1 \sqrt{\frac{\pi}{p_2}} \int_{\mathcal{C}_2} k_\sigma(c_2 - c_2^\dagger) e^{p_0 + \frac{p_1^2}{4p_2}} \Big(q_0 + q_1\frac{p_1}{2p_2} + q_2\frac{1}{4p_2^2}(2p_2 + p_1^2)\Big)\, dc_2 \\
= & z_0 z_1 \sqrt{\frac{\pi}{p_2}} \frac{1}{\sqrt{2\pi}\sigma} \int_{\mathcal{C}_2} e^{p_0 + \frac{p_1^2}{4p_2} - \frac{(c_2 - c_2^\dagger)^2}{2}} \Big(q_0 + q_1\frac{p_1}{2p_2} + q_2\frac{1}{4p_2^2}(2p_2 + p_1^2)\Big)\, dc_2 \\
= & \frac{z_0 z_1 z_2^2}{|x_2|} \sqrt{\frac{\pi}{p_2}} \int_{\mathcal{C}_2} \frac{1}{\sqrt{2\pi}\sigma}(Q_0 + sQ_1 + s^2 Q_2) e^{-P_2 s^2 + P_1 s + P_0}\, ds \qquad (16) \\
= & \frac{z_0 z_1 z_2^2}{\sigma|x_2|} \sqrt{\frac{\pi}{2p_2 P_2}} e^{P_0 + \frac{P_1^2}{4P_2}} \Big(Q_0 + Q_1\frac{P_1}{2P_2} + Q_2\frac{1}{4P_2^2}(2P_2 + P_1^2)\Big)\,, \qquad (17)
\end{aligned}
$$

where (16) applies change of variable $s = 1 + x_2 c_2$ to the integral. Note that, $\int_{\mathbb{R}} f(c_2)dc_2 = \text{sign}(x_2)\int_{\mathbb{R}} f((s-1)/x_2)ds/x_2 = 1/|x_2|\int_{\mathbb{R}} f((s-1)/x_2)ds$. Also, (17) uses Corollary 7 with the particular choice of $z_2$, $P_i$ and $Q_i$ for $i = 0, 1, 2$ as the following. Obviously the following $P_2$ satisfies $P_2 > 0$. Also not that $P_i$ and $Q_i$ are independent of integration variable $s$.

$$
\begin{aligned}
z_2 &\triangleq \frac{1}{1+\sigma^2 x_1^2 z_0 \|\boldsymbol{y}\|^2} \\
Q_0 &\triangleq \frac{1}{z_2}\sigma^2 x_1^2 + x_1^2 (c_1^\dagger + \sigma^2 z_0 x_1 \boldsymbol{y}^T \boldsymbol{v})^2 \\
Q_1 &\triangleq 2x_1 (c_1^\dagger + \sigma^2 z_0 x_1 \boldsymbol{y}^T \boldsymbol{v}) \\
Q_2 &\triangleq 1 \\
P_0 &\triangleq -\frac{1}{2\sigma^2 x_2^2}(1 + c_2^\dagger x_2)^2 - \frac{z_0 z_2}{2}(\|\boldsymbol{v} - c_1^\dagger x_1 \boldsymbol{y}\|^2 + \sigma^2 z_0 x_1^2 (v_2 y_1 - v_1 y_2)^2) \\
P_1 &\triangleq z_0 z_2 \boldsymbol{y}^T (\boldsymbol{v} - c_1^\dagger x_1 \boldsymbol{y}) + \frac{1}{\sigma^2 x_2^2}(1 + c_2^\dagger x_2) \\
P_2 &\triangleq \frac{1}{2\sigma^2 x_2^2} + \frac{z_0 z_2}{2}\|\boldsymbol{y}\|^2 .
\end{aligned}
$$

In fact, by plugging in the definitions for $z_i$, $P_i$, and $Q_i$ and performing elementary algebraic manipulations, one can write (17) more compactly as the following,

$$u_{\boldsymbol{\tau},\sigma}(\boldsymbol{\theta}^\dagger, \boldsymbol{x}, \boldsymbol{y}) = q e^{-p} ,$$

where the auxiliary variables are as below:

$$
\begin{aligned}
z_0 &\triangleq \frac{1}{1+\|\boldsymbol{x}\|^2} \\
z_1 &\triangleq 1 + \boldsymbol{x}^T \boldsymbol{c}^\dagger \\
\boldsymbol{v} &\triangleq \boldsymbol{A}^\dagger \boldsymbol{x} + \boldsymbol{b}^\dagger \\
q &\triangleq z_0 \frac{(z_0 \|\boldsymbol{x}\|^2 \boldsymbol{y}^T \boldsymbol{v} + z_1)^2 + \sigma^2 \|\boldsymbol{x}\|^2 (1 + z_0 \|\boldsymbol{x}\|^2 \|\boldsymbol{y}\|^2)}{2\pi\sigma^2 (1 + z_0 \|\boldsymbol{x}\|^2 \|\boldsymbol{y}\|^2)^{\frac{5}{2}}} \\
p &\triangleq \frac{\|z_1 \boldsymbol{y} - \boldsymbol{v}\|^2 + z_0 \|\boldsymbol{x}\|^2 (v_2 y_1 - v_1 y_2)^2}{2\sigma^2 (1 + \|\boldsymbol{x}\|^2 (1 + \|\boldsymbol{y}\|^2))} .
\end{aligned}
$$

□