# Coarse-to-Fine Minimization of Some Common Nonconvexities

Hossein Mobahi and John W. Fisher III

Computer Science and Artificial Intelligence Lab. (CSAIL)
Massachusetts Institute of Technology (MIT)
{hmobahi,fisher}@csail.mit.edu

**Abstract.** The continuation method is a popular heuristic in computer vision for nonconvex optimization. The idea is to start from a simplified problem and gradually deform it to the actual problem while tracking the solution. There are many choices for how to map the nonconvex objective to some convex task. One popular principle for such construction is Gaussian smoothing of the objective function. This involves an integration which may be expensive to compute numerically. We argue that often simple tricks at the problem formulation plus some mild approximations can make the resulted task amenable to closed form integral.

**Keywords:** Continuation Method, Diffusion Equation, Nonconvex Optimization, Graduated Nonconvexity

## 1 Introduction

Nonconvex optimization tasks are ubiquitous in computer vision [25,32,21,13,23]. However, solving such problems (to global optimality) is generally intractable. Hence, often the nonconvex problem is either relaxed to a convex task [28,15], or heuristic optimization methods are utilized [7,5,29,8,24]. Each of these two methods has its own pros and cons. The global optimum of the convex relaxed task can be found efficiently. However, some aspects of the original problem may be lost because of the relaxation, which sometimes could be crucial. On the other hand, heuristic methods are not guaranteed to find the global optimum. In return, they directly target solving the nonconvex task. Hence, they sometimes can offer good local minima if not the global one.

A long standing deterministic heuristic for handling nonconvex tasks in computer vision is Blake and Zisserman's *Graduated Non-Convexity* (GNC) [6]. The idea, introduced about three decades ago, is to start from a convex problem. The latter is then progressively deformed to the actual objective while tracking the solution along the way. Around the same time, Terzopoulos used similar ideas for surface interpolation problems [30]. Outside of computer vision field, GNC technique is known under a broader class of optimization by *homotopy continuation* method [33].

The idea of optimization by continuation has been utilized in several interesting works. For example, Brox's thesis on image segmentation relies on this technique for optimization [7]. Black and Rangarajan used continuation for analyzing

spatial discontinuities and applied it to several problems in early vision [5]. In fact, coarse-to-fine image representation which is widely used in computer vision is related to the continuation method [24]. Note that state-of-the-art solutions for optical flow [29,8] and shape estimation [1] rely on multiscale representation to avoid poor local minima.

There is also a growing interest in using similar optimization methods within machine learning community. Some example applications include semi-supervised kernel machines [27], multiple instance learning [14,18], semi-supervised structured output [11], and language modeling [2]. It has also been suggested that recent training algorithms for deep architectures [17,12], which have made a breakthrough, in fact approximate continuation methods [3].

There is an infinite number of ways to progressively deform the nonconvex objective to some convex task. One possible principle is by Gaussian smoothing of the objective function [26,24]. In fact, we have recently shown that Gaussian smoothing has optimality for homotopy construction in a certain sense [22]. The Gaussian smoothing method convolves the nonconvex objective with an isotropic Gaussian kernel. This results in a collection of functions ranging from a highly smoothed to the actual nonconvex function, depending on the bandwidth parameter of the Gaussian. The continuation method processes this collection successive, starting the smoothed function and ending at the actual nonconvex function. Since going from high to low bandwidth reveals more details of the objective function, optimization by Gaussian homotopy continuation is also called *coarse-to-fine optimization*.

From practical viewpoint, the key challenge for using Gaussian homotopy is computing the convolution integral. In fact, using this approach makes sense only if this integral can be computed analytically[1]. This may seem disappointing at first for several reasons. First, the integrands that lead to a closed form integration are often rare and must have a very simple and nice form. In addition, some applications involve objective functions defined over discrete variables, for which Gaussian convolution is not well-defined.

Despite these challenges, we argue that sometimes simple tricks at the problem formulation and some mild approximations can make the resulted task amenable to closed form integral. To be concrete, we demonstrate this within two example tasks[2]. The first one focuses on handling discrete valued variables in a combinatorial setting. The example application is establishing correspondence between a pair of point clouds. The second example shows the use indicator functions as well as robust loss functions within an image denoising setup. For both applications, we show that the objective becomes convex after enough smooth-

[1] The dimension of the integration domain is the number optimization variables. The numerical computation of this integral can be as expensive as exhaustive search of the domain for finding the global optimum.

[2] Both applications are formulated in their simplest form to allow focusing on the homotopy construction task. We do not aim at beating state of the art in such a simple setup, but rather produce comparable results against common alternatives.

ing. In addition, we demonstrate that the minimizer of the convexified (i.e. highly smoothed) problem can be expressed in closed form.

Although the paper investigates two example applications, the underlying ideas may be generalized to some other tasks. Specifically, the energy function in both applications consist of *polynomials* and *Gaussian functions*. Both of these forms are amenable to closed form Gaussian convolution. Hence, Gaussian smoothing can be analytically computed for any energy function that can be constructed from these components. Note that these two components are very rich. For example, in the alignment example we will show that *discrete* variables can be replaced by continuous through simple polynomial penalties. Furthermore, in the denoising example, we will show that *indicator* functions and some *robust loss* functions can be expressed by a Gaussian form.

Throughout this paper, we use $x$ for scalars, $\boldsymbol{x}$ for vectors, $\boldsymbol{X}$ for matrices, and $\mathcal{X}$ for sets. Here $\|\boldsymbol{x}\|$ means $\|\boldsymbol{x}\|_2$ and $\nabla$ means $\nabla_{\boldsymbol{x}}$, and $\triangleq$ means equality by definition. The convolution operator is denoted by $\star$. The isotropic Gaussian kernel with standard deviation $\sigma$ is shown by $k_\sigma$,

$$k_\sigma(\boldsymbol{x}) \triangleq \frac{1}{(\sqrt{2\pi}\sigma)^n} e^{-\frac{\|\boldsymbol{x}\|^2}{2\sigma^2}} .$$

## 2 Discrete Valued Variables

In this section we show how simple tricks at the problem formulation level can handle discrete valued variables. We use 3D point cloud alignment as the example task.

### 2.1 Formulation

Given two sets of points $\mathcal{P} = \{\boldsymbol{p}_i\}_{i=1}^m$ and $\mathcal{Q} = \{\boldsymbol{q}_j\}_{j=1}^n$, where each point is in $\mathbb{R}^d$ and $d = 3$ for 3D point clouds. Consider the affine transformation $\boldsymbol{A}\boldsymbol{p} + \boldsymbol{b}$, where $\boldsymbol{A}$ is $d \times d$ and $\boldsymbol{b}$ is $d \times 1$. We define the optimal affine alignment as the following,

$$\begin{aligned} (\boldsymbol{A}^*, \boldsymbol{b}^*, \boldsymbol{c}^*) \;&=\; \arg\min_{\boldsymbol{A},\boldsymbol{b},\boldsymbol{c}} \sum_{i=1}^{m}\sum_{j=1}^{n} (c_{i,j}\|\boldsymbol{A}\boldsymbol{p}_i + \boldsymbol{b} - \boldsymbol{q}_j\|)^2 \qquad (1) \\ &\text{s.t. } \forall j \sum_{i=1}^{m} c_{i,j} = 1, \quad , \quad \forall i \forall j \quad c_{i,j} \in \{0,1\} . \end{aligned}$$

The binary variables $c_{i,j}$ determine the correspondence among the point pairs in $\mathcal{P}$ and $\mathcal{Q}$. We arranged the elements $c_{i,j}$ into vector of size $m\,n$ denoted by $\boldsymbol{c}$. Without loss of generality, we assume that the point set $\mathcal{P}$ is uncorrelated and has zero mean, with the largest variance being one as below,

$$\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{p}_i = \boldsymbol{0} \quad , \quad \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{p}_i\boldsymbol{p}_i^T = \text{diag}([1\ ,\ \lambda_2\ ,\ \lambda_3])\,, \tag{2}$$

where $1 \geq \lambda_2 \geq \lambda_3 > 0$. If $\mathcal{P}$ does not have this property, we can easily process the points to get them this way [3], as explained in the following. Suppose the original point sets are named $\mathcal{P}^\circ$ and $\mathcal{Q}^\circ$. Let the spectral decomposition of $\mathcal{P}^\circ$ be as below,

$$\frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{p}_i^\circ - \bar{\boldsymbol{p}}^\circ)(\boldsymbol{p}_i^\circ - \bar{\boldsymbol{p}}^\circ)^T = \boldsymbol{V}\,\text{diag}(\boldsymbol{d})\boldsymbol{V}^T\,, \tag{3}$$

where $\bar{\boldsymbol{p}}^\circ = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{p}_i^\circ$, $\boldsymbol{d}$ is the vector of eigenvalues and $\boldsymbol{V}$ is a matrix with columns being eigenvectors of the covariance of $\mathcal{P}^\circ$. Then we define the sets $\mathcal{P}$ and $\mathcal{Q}$ by applying the shift $\bar{\boldsymbol{p}}^\circ$, rotation $\boldsymbol{V}$, and scaling $1/\max(\boldsymbol{d})$ to the initial sets $\mathcal{P}^\circ$ and $\mathcal{Q}^\circ$ as shown below,

$$\forall \boldsymbol{p}_i^\circ \in \mathcal{P}^\circ, \boldsymbol{p}_i \triangleq \frac{1}{\max(\boldsymbol{d})}\boldsymbol{V}^T(\boldsymbol{p}_i^\circ - \bar{\boldsymbol{p}}^\circ)\boldsymbol{V} \tag{4}$$

$$\forall \boldsymbol{q}_j^\circ \in \mathcal{Q}^\circ, \boldsymbol{q}_j \triangleq \frac{1}{\max(\boldsymbol{d})}\boldsymbol{V}^T(\boldsymbol{q}_j^\circ - \bar{\boldsymbol{p}}^\circ)\boldsymbol{V}\,. \tag{5}$$

It is easy to check that the transformed set $\mathcal{P}$ now has the assumed properties. Thus, we can apply the proposed affine alignment algorithm. Suppose the algorithm returns affine parameters $\boldsymbol{A}$ and $\boldsymbol{b}$ in the sense that it best transforms the set $\mathcal{P}$ to the set $\mathcal{Q}$ via $\boldsymbol{A}\boldsymbol{p} + \boldsymbol{b}$. We can easily use this solution to relate the original sets $\mathcal{P}^\circ$ and $\mathcal{Q}^\circ$ via $\boldsymbol{A}^\circ\boldsymbol{p}^\circ + \boldsymbol{b}^\circ$, where $\boldsymbol{A}^\circ = \boldsymbol{V}\boldsymbol{A}\boldsymbol{V}^T$ and $\boldsymbol{b}^\circ = (\boldsymbol{I} - \boldsymbol{A}^\circ)\bar{\boldsymbol{p}}^\circ + \max(\boldsymbol{d})\boldsymbol{V}\boldsymbol{b}\boldsymbol{V}^T$.

### 2.2 Smoothing

In order to apply Gaussian smoothing, the optimization must be in continuous variables and unconstrained. To achieve the first property, we express the discrete constraint $c_{i,j} \in \{0, 1\}$ equivalently by the continuous equality constraint of form $c_{i,j}(1 - c_{i,j}) = 0$. Thus, the optimization task becomes as the following,

$$\begin{aligned}(\boldsymbol{A}^*, \boldsymbol{b}^*, \boldsymbol{c}^*) \ &= \ \arg\min_{\boldsymbol{A},\boldsymbol{b},\boldsymbol{c}} \sum_{i,j}(c_{i,j}\|\boldsymbol{A}\boldsymbol{p}_i + \boldsymbol{b} - \boldsymbol{q}_j\|)^2 \\ &\text{s.t. } -1 + \sum_{i=1}^{m} c_{i,j} = 0 \quad , \quad c_{i,j}(1 - c_{i,j}) = 0.\end{aligned} \tag{6}$$

[3] If the point set is degenerate, i.e. its covariance matrix has some eigenvalues equal to zero, then we cannot have $\mathcal{P}$ in the desired form. In that case, the null space of the data can be removed to obtain a lower dimensional representation for the points. Everything else in the paper remains the same for the new set, as there is nothing special in our analysis to force $d = 3$.

To satisfy the second property, we approximate the problem by replacing equality constraints $h(\boldsymbol{c}) = 0$ by the objective penalty $h^2(\boldsymbol{c})$. Thus, the approximate objective function becomes as the following,

$$f(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}) \triangleq \frac{\epsilon}{m\,n}\Big(\sum_{i=1}^{m}\sum_{j=1}^{n} c_{i,j}^2 \|\boldsymbol{A}\boldsymbol{p}_i + \boldsymbol{b} - \boldsymbol{q}_j\|^2\Big) + \frac{1}{m\,n}\big(\sum_{j=1}^{n}(1 - \sum_{i=1}^{m} c_{i,j})^2 + \sum_{i=1}^{m}\sum_{j=1}^{n} c_{i,j}^2(1 - c_{i,j})^2\big)\,, \tag{7}$$

where $\epsilon > 0$ is a small number. We can now convolve the objective function with the Gaussian kernel $k_\sigma\big(\mathrm{vec}(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})\big)$, where vec concatenates all variables into a long vector as below,

Due to diagonal form of the covariance, we first compute convolution w.r.t. variables $\{c_{i,j}\}$, and then convolve the result with variables $\mathrm{vec}(\boldsymbol{A}, \boldsymbol{b})$. Convolution in variables $\{c_{i,j}\}$ is easily computed as follows,

$$\begin{aligned} g_1(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}; \sigma) &\triangleq \left[f \star k(\,.\,; \sigma^2)\right](\boldsymbol{c}) \\ &= \frac{\epsilon}{m\,n}\sum_{i=1}^{m}\sum_{j=1}^{n}(c_{i,j}^2 + \sigma^2)\big(\|\boldsymbol{A}\boldsymbol{p}_i + \boldsymbol{b} - \boldsymbol{q}_j\|^2\big) \\ &\quad + \frac{1}{m\,n}\sum_{j=1}^{n}(1 - \sum_{i=1}^{m} c_{i,j})^2 + \frac{1}{m\,n}\sum_{i=1}^{m}\sum_{j=1}^{n}(c_{i,j} - 1)^2 c_{i,j}^2 + 6\sigma^2(c_{i,j} - \frac{1}{2})^2\,. \end{aligned}$$

We now apply the convolution w.r.t. $\boldsymbol{\theta} \triangleq \mathrm{vec}(\boldsymbol{A}, \boldsymbol{b})$, and denote the affine transform by as $\boldsymbol{\tau}\big(\boldsymbol{p}; \boldsymbol{\theta}\big) \triangleq \boldsymbol{A}\boldsymbol{p} + \boldsymbol{b}$.

$$\begin{aligned} g(\boldsymbol{\theta}, \boldsymbol{c}; \sigma) &\triangleq \Big[g_1(\,.\,, \boldsymbol{c}; \sigma) \star k(\,.\,; \sigma^2)\Big](\boldsymbol{\theta}) \\ &= \frac{\epsilon}{m\,n}\big(\sum_{i,j}^{m,n}(c_{i,j} + \sigma^2)\int_{\mathbb{R}^3} \|\boldsymbol{r} - \boldsymbol{q}_j\|^2 k_{\sigma\sqrt{1+\|\boldsymbol{p}_i\|^2}}(\boldsymbol{\tau}(\boldsymbol{p}_i, \boldsymbol{\theta}) - \boldsymbol{r})\,d\boldsymbol{r}\big) \quad (8) \\ &\quad + \frac{1}{m\,n}\sum_{j=1}^{n}(1 - \sum_{i=1}^{m} c_{i,j})^2 + \frac{1}{m\,n}\sum_{i,j}^{m,n}(c_{i,j} - 1)^2 c_{i,j}^2 + 6\sigma^2(c_{i,j} - \frac{1}{2})^2 \\ &= \frac{\epsilon}{m\,n}\sum_{i,j}^{m,n}(c_{i,j}^2 + \sigma^2)\big(\|\boldsymbol{A}\boldsymbol{p}_i + \boldsymbol{b} - \boldsymbol{q}_j\|^2 + 3\sigma^2(1 + \|\boldsymbol{p}_i\|^2)\big) \\ &\quad + \frac{1}{m\,n}\big(\sum_{j=1}^{n}(1 - \sum_{i=1}^{m} c_{i,j})^2 + \sum_{i,j}^{m,n}(c_{i,j} - 1)^2 c_{i,j}^2 + 6\sigma^2(c_{i,j} - \frac{1}{2})^2\big)\,. \quad (9) \end{aligned}$$

where (8) uses the *transformation kernel* for the affine map [24]. This kernel allows writing the high dimensional convolution w.r.t. $\boldsymbol{\theta}$ equivalently by a $d$-dimensional integral transform, where here $d = 3$.

### 2.3 Asymptotic Minimizer

It is easy to check that, as $\sigma \to \infty$, the Hessian of the objective converges to a matrix with zero off-diagonals and positive diagonals (hence asymptotically convex). In addition, zero crossing the gradient when $\sigma \to \infty$ leads to the following asymptotic minimizer,

$$\begin{aligned} \boldsymbol{A}^* &= \Big(\frac{1}{m\,n}\sum_{i=1}^{m}\sum_{j=1}^{n}\boldsymbol{q}_j\boldsymbol{p}_i^T\Big)\,\mathrm{diag}([1\,,\,\frac{1}{\lambda_2}\,,\,\frac{1}{\lambda_3}]) \\ \boldsymbol{b}^* &= \frac{1}{n}\sum_{j=1}^{n}\boldsymbol{q}_j \quad , \quad c^*_{i,j} = \frac{1}{2+\epsilon\,(1+\|\boldsymbol{p}_i\|^2)}\,. \end{aligned} \tag{10}$$

### 2.4 Continuation Updates

The continuation process moves from the stationary point attained at the previous smoothing level $\sigma$ to the (possibly local) minimum of the current objective function formed by a smaller $\sigma$, i.e. reduced smoothing. For each fixed $\sigma$, the stationary point of the problem is obtained by looping over gradient descent with the line search until convergence. The sequence for $\sigma$ is generated by starting from $\sigma_0 = 2$ and updating it by $\sigma_{k+1} = 0.9\sigma_k$, until the value of $\sigma$ falls below 0.01.

A great advantage of the low-order polynomial formulation of the alignment task is that, we can compute the *optimal line search* in closed form as explained below. The gradient of $g(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}; \sigma)$ in (9) can be expressed as follows,

$$\frac{\partial g}{\partial \boldsymbol{A}} = 2\epsilon\sum_{i=1}^{m}\sum_{j=1}^{n} c_{i,j}^2\Big(\,(\,\boldsymbol{A}\boldsymbol{p}_i + \boldsymbol{b} - \boldsymbol{q}_j\,)\boldsymbol{p}_i^T\,\Big), \tag{11}$$

$$\frac{\partial g}{\partial \boldsymbol{b}} = 2\epsilon\sum_{i=1}^{m}\sum_{j=1}^{n} c_{i,j}^2\Big(\boldsymbol{A}\boldsymbol{p}_i + \boldsymbol{b} - \boldsymbol{q}_j\Big)\,. \tag{12}$$

$$\begin{aligned} \frac{\partial g}{\partial c_{i,j}} &= \epsilon\big(\|\boldsymbol{A}\boldsymbol{p}_i + \boldsymbol{b} - \boldsymbol{q}_j\|^2 + 3\sigma^2(1+\|\boldsymbol{p}_i\|^2)\big) - 2 + 2\sum_{k=1}^{m} c_{k,j} \\ &\quad + 2\Big(c_{i,j}(c_{i,j}-1)(2c_{i,j}-1) + 6\sigma^2(c_{i,j}-\frac{1}{2})\Big)\,. \end{aligned} \tag{13}$$

Define the updated solution as $\boldsymbol{A}^+ \triangleq \boldsymbol{A} + \alpha\frac{\partial g}{\partial \boldsymbol{A}}$, $\boldsymbol{b}^+ \triangleq \boldsymbol{b} + \alpha\frac{\partial g}{\partial \boldsymbol{b}}$, and $\boldsymbol{c}^+ \triangleq \boldsymbol{c} + \alpha\frac{\partial g}{\partial \boldsymbol{c}}$. The optimal step size $\alpha$ can be obtained by zero crossing $\frac{d}{d\alpha}g(\boldsymbol{A}^+, \boldsymbol{b}^+, \boldsymbol{c}^+; \sigma)$. By collecting different exponents of $\alpha$, the latter can be written as below,

$$\frac{d}{d\alpha}g(\boldsymbol{A}^+, \boldsymbol{b}^+, \boldsymbol{c}^+; \sigma) = t_3\alpha^3 + t_2\alpha^2 + t_1\alpha + t_0\,, \tag{14}$$

where $t_0$, $t_1$, $t_2$, and $t_3$ are constants introduced for brevity[4].

It is obvious that (14) is a *cubic equation* with the following closed-form roots for choices of $w \in \{-2, 1+i\sqrt{3}, 1-i\sqrt{3}\}$.

$$\alpha = -\frac{1}{6t_3}\Big(2t_2 + w(\frac{T_1 + \sqrt{T_1^2 - T_2}}{2})^{\frac{1}{3}} + w^*(\frac{T_1 - \sqrt{T_1^2 - T_2}}{2})^{\frac{1}{3}}\Big)\,,$$

where $w^*$ is the complex conjugate of $w$, and the auxiliary variables $T_1$ and $T_2$ are defined as follows,

$$T_1 \triangleq 2t_2^3 - 9t_1t_2t_3 + 27t_0t_3^2 \quad , \quad T_2 \triangleq 4(t_2^2 - 3t_1t_3)^3\,.$$

Obviously, we only consider the *real roots* of the above equation. We can evaluate $g(\boldsymbol{A}^+, \boldsymbol{b}^+, \boldsymbol{c}^+; \sigma)$ at all the real roots (at most three) and choose the one that attains the smallest value of $g(\boldsymbol{A}^+, \boldsymbol{b}^+, \boldsymbol{c}^+; \sigma)$.

Algorithm 1 shows the procedure for affine alignment by Gaussian smoothing and path following.

### 2.5 Results

In this section, we present that result obtained by Algorithm 1. We use Iterative Closest Point (ICP) algorithm [4] as a baseline result. The idea of ICP is to alternate between creating a correspondence between pair of points (of the two

[4] The constants $t_0$, $t_1$, $t_2$, and $t_3$ have the following form,

$$\begin{aligned}
t_0 \triangleq & \sum_{i,j} 2\frac{\partial g}{\partial c_{i,j}}(2c_{i,j}-1)\big((c_{i,j}-1)c_{i,j} + 3\sigma^2\big) \\
& +3\epsilon\sigma^2\frac{\partial g}{\partial c_{i,j}}(1 + \|\boldsymbol{p}_i\|^2) + \epsilon\frac{\partial g}{\partial c_{i,j}}\|\boldsymbol{A}\boldsymbol{p}_i + \boldsymbol{b} - \boldsymbol{q}_j\|^2 \\
& +2c_{i,j}\epsilon(\boldsymbol{A}\boldsymbol{p}_i + \boldsymbol{b} - \boldsymbol{q}_j)^T(\frac{\partial g}{\partial \boldsymbol{A}}\boldsymbol{p}_i + \frac{\partial g}{\partial \boldsymbol{b}}) + 2\sum_{j=1}^{n}((\sum_{i=1}^{m}\frac{\partial g}{\partial c_{i,j}})(-1 + \sum_{i=1}^{m} c_{i,j})) \\
t_1 \triangleq & \sum_{i,j} 2(\frac{\partial g}{\partial c_{i,j}})^2\big(1 + 6c_{i,j}(c_{i,j}-1) + 6\sigma^2\big) \\
& +2\epsilon c_{i,j}\|\frac{\partial g}{\partial \boldsymbol{A}}\boldsymbol{p}_i + \frac{\partial g}{\partial \boldsymbol{b}}\|^2 + 2\sum_{j=1}^{n}(\sum_{i=1}^{m}\frac{\partial g}{\partial c_{i,j}})^2 \\
& +4\epsilon\frac{\partial g}{\partial c_{i,j}}(\boldsymbol{A}\boldsymbol{p}_i + \boldsymbol{b} - \boldsymbol{q}_j)^T(\frac{\partial g}{\partial \boldsymbol{A}}\boldsymbol{p}_i + \frac{\partial g}{\partial \boldsymbol{b}}) \\
t_2 \triangleq & \sum_{i,j} 3\frac{\partial g}{\partial c_{i,j}}\big(2(\frac{\partial g}{\partial c_{i,j}})^2(2c_{i,j}-1) + \epsilon\|\frac{\partial g}{\partial \boldsymbol{A}}\boldsymbol{p}_i + \frac{\partial g}{\partial \boldsymbol{b}}\|^2\big) \\
t_3 \triangleq & \sum_{i,j} 4(\frac{\partial g}{\partial c_{i,j}})^4\,.
\end{aligned}$$

**Algorithm 1** Point Cloud Alignment by Gaussian Homotopy Continuation

Input: Point clouds $\mathcal{P}$ (as eq. (2)) and $\mathcal{Q}$, a small $\epsilon > 0$, a sequence $\sigma_1 > \sigma_2 > \cdots > \sigma_N > 0$.
$\boldsymbol{A} = (\frac{1}{m\,n} \sum_{i=1}^{m} \sum_{j=1}^{n} \boldsymbol{q}_j \boldsymbol{p}_i^T)\, \mathrm{diag}([1\,,\, \frac{1}{\lambda_2}\,,\, \frac{1}{\lambda_3}])$
$\boldsymbol{b} = \frac{1}{n} \sum_{j=1}^{n} \boldsymbol{q}_j$
$c_{i,j} = \frac{1}{2+\epsilon\,(1+\|\boldsymbol{p}_i\|^2)}$
**for** $k = 1 \rightarrow N$ **do**
  **repeat**
    $\boldsymbol{A} = \boldsymbol{A} + \alpha \frac{\partial g}{\partial \boldsymbol{A}}$.
    $\boldsymbol{b} = \boldsymbol{b} + \alpha \frac{\partial g}{\partial \boldsymbol{b}}$.
    $\boldsymbol{c} = \boldsymbol{c} + \alpha \frac{\partial g}{\partial \boldsymbol{c}}$
  **until** Convergence
**end for**
Output: $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$

clouds) and refining the geometric transformation between the corresponding points. Both algorithms share the same initialization, which is the asymptotic minimizer of the alignment objective presented in (10). For the continuation algorithm, we set $\epsilon = 0.01$.

We use some of the 3D objects provided by Stanford's dataset [9,19,31], each of which comprises a set $\mathcal{P}$. We then create $\mathcal{Q}$ by rotating points in $\mathcal{P}$ by $n$ degrees along all three $x$, $y$ and $z$ axes, where $n$ varies between 30 degrees to 90 degrees, in steps of 15 degrees. This way, for each $\mathcal{P}$, we derive a set of problems $\{\mathcal{Q}_n\}$ that are increasingly more challenging as $n$ grows.

Figures 1 and 2 indicate that ICP gets stuck is poor local minima more often than the continuation method, before reaching a reasonable alignment.

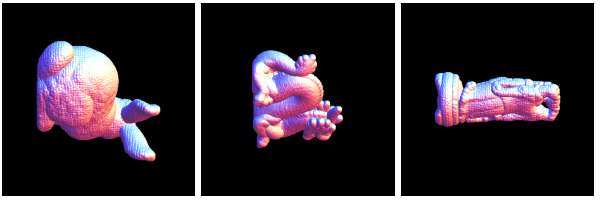

**Fig. 1.** The point set $\mathcal{Q}$ for each 3D object.

## 3 Indicator Function and Robust Loss

In this section we show how indicator function and a robust loss function (truncated quadratic) can be approximated in a way that become amenable to closed form integration. This is shown through an example task for image denoising.

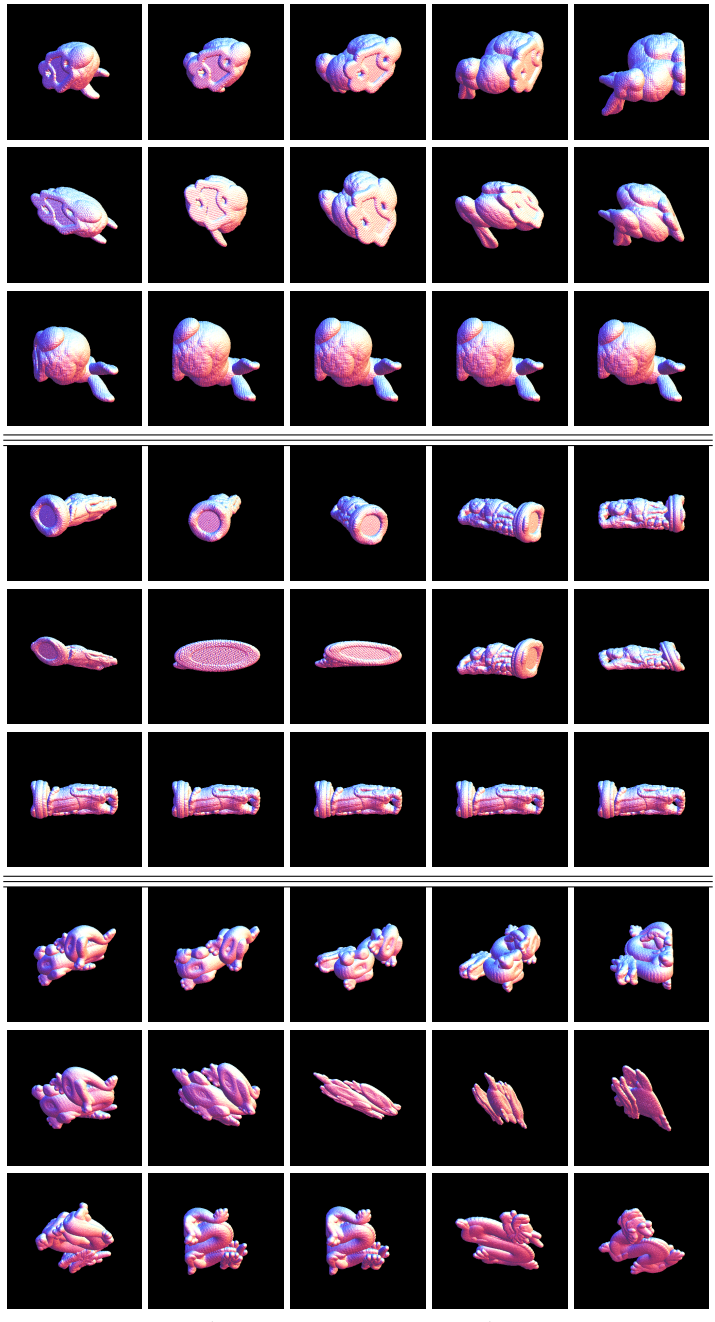


**Fig. 2.** Each object occupies there successive rows, where each row has the following role. (Top) Input $\mathcal{P}$, which is a rotated version of $\mathcal{Q}$. (Middle) Transformed $\mathcal{P}$ to match $\mathcal{Q}$ using ICP. (Bottom) Transformed $\mathcal{P}$ to match $\mathcal{Q}$ using proposed method.

### 3.1 Formulation

Given an image matrix $\boldsymbol{V}$ whose entries are affected additively by independent Gaussian noise. The resulted noisy image is denoted by $\tilde{\boldsymbol{V}}$. The noise has zero mean and unknown variance. Suppose the original image has a piecewise constant structure[5]. The denoising problem can be formulated as the following,

$$\boldsymbol{U}^* = \arg\min_{\boldsymbol{U}} \sum_{i,j} \lambda \left(u_{i,j} - \tilde{v}_{i,j}\right)^2 + I_{\|\nabla u_{i,j}\| \neq 0}, \tag{15}$$

where $I$ is an indicator function that is one if its argument is true and zero otherwise. This regularization resembles the $\ell_0$ norm of gradient's magnitude map. Here $\nabla u_{i,j}$ is the finite difference approximation of the gradient at the entry $u_{i,j}$, i.e. $\|\nabla u_{i,j}\|^2 \triangleq (u_{i+1,j} - u_{i,j})^2 + (u_{i,j+1} - u_{i,j})^2$. The parameter $\lambda$ balances between fidelity and regularization.

Observe that an indicator function $I_{x \neq 0}$ can be also expressed as $1 - \lim_{\epsilon \to 0} e^{-\frac{x^2}{2\epsilon^2}}$. For practical applications, $1 - e^{-\frac{x^2}{2\epsilon^2}}$ with a small enough $\epsilon$ provides a reasonable approximation to the limit case (Figure 3-Left). The advantage of this particular approximation for the indicator function is this it allows for a closed-form Gaussian convolution. Using that, the objective can be written as below,

$$\begin{aligned} f(\boldsymbol{U}) &= \sum_{i,j} \lambda \left(u_{i,j} - \tilde{v}_{i,j}\right)^2 + 1 - e^{-\frac{\|\nabla u_{i,j}\|^2}{2\epsilon^2}} \\ &= \sum_{i,j} \lambda \left(u_{i,j} - \tilde{v}_{i,j}\right)^2 + 1 - e^{-\frac{(u_{i+1,j} - u_{i,j})^2 + (u_{i,j+1} - u_{i,j})^2}{2\epsilon^2}} . \end{aligned}$$

Note that the Gaussian function, which a larger choice of $\epsilon$, can also provide a good approximation for the truncated quadratic form (Figure 3-Right). This approximation maintains the key property of robust loss functions, which is having flat tails. In the following, however, we continue with the simple (non-robust) quadratic loss.

### 3.2 Smoothing

Convolving this objective with Gaussian kernels in variables $u_{i,j}$ and dropping constant terms leads to the following,

$$\begin{aligned} g(\boldsymbol{U}, \sigma) = &\sum_{i,j} \lambda \left(u_{i,j} - \tilde{v}_{i,j}\right)^2 \\ &- c\epsilon^{-\frac{\epsilon^2 \|\nabla u_{i,j}\|^2 + 2\sigma^2 (u_{i,j}^2 + u_{i,j+1}^2 + u_{i+1,j}^2 - u_{i,j+1} u_{i+1,j} - u_{i,j}(u_{i,j+1} + u_{i+1,j}))}{2(\sigma^2 + \epsilon^2)(3\sigma^2 + \epsilon^2)}} , \end{aligned}$$

where $c$ is a constant factor.

[5] That is the case for most shape images.

**Fig. 3.** Using the function $e^{-\frac{x^2}{2\epsilon^2}}$ to approximate indicator function (Left) by $\epsilon = 0.005$ and (robust) truncated quadratic loss (Right) by $\epsilon = 1/2$. Both cases are plotted in the range $x \in [-4, 4]$

### 3.3 Asymptotic Minimizer

As $\sigma \to \infty$, the regularization term vanishes and only the convex quadratic term remains. Hence, this problem is asymptotically convex, and its asymptotic minimizer is simply the solution of the convex quadratic part, which is $u_{i,j} = \tilde{v}_{i,j}$.

### 3.4 Continuation

The sequence for $\sigma$ is generated by starting from $\sigma_0 = 2$ and updating it by $\sigma_{k+1} = 0.9\sigma_k$, until the value of $\sigma$ falls below 0.01. Sensitivity parameter $\epsilon$ is set to $\frac{2}{255}$, which means 2 intensity levels out of 255 possible levels in an 8-bit representation. For each value of $\sigma$, gradient descent loop is performed until convergence. The loop starts by initializing the solution obtained from the previous value of $\sigma$. The exponential form appearing in this application prevents finding the optimal line search in closed form. Thus, here we use the plain gradient descent.

### 3.5 Results

The method is applied to an example shape image degraded by Gaussian noise. The intensity values of the image range between zero and one, and the standard deviation of the noise is 0.5, which is quite severe. We apply four other methods to these data for comparison. Specifically, we use isotropic and anisotropic total variation [16] using publicly available code[6], BM3D denoising package[7] [10], and KMeans clustering [20] shipped with Matlab. Total variation essentially penalizes the $\ell_1$ norm of the gradient's magnitude. Note that $\ell_1$ norm is the convex envelope for the $\ell_0$ norm, hence the best possible convex approximation of the actual problem.

[6] We used Matlab code published by Benjamin Tremoulheac.

[7] Authors of this package have made their code publicly available. We used version 2.0 of this package.

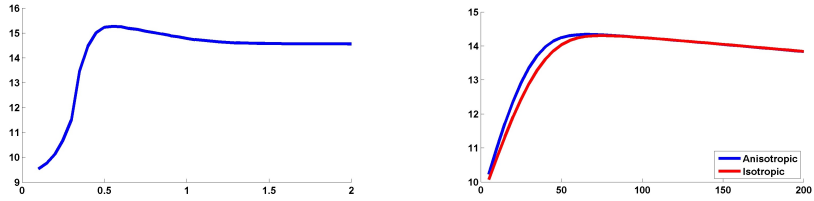


**Fig. 4.** Plots of PSNR value versus algorithm's parameter. Left: BM3D (horizontal axis shows multipliers of noise standard deviation, e.g. 2 means twice the value of the standard deviation). Right: Total Variation (the parameter balances between regularization and fidelity).

For BM3D, we provide the algorithm with the true value of noise variance, which is to the advantage of this method. The total variation method, like ours, depends on a $\lambda$ parameter that balances fidelity versus regularization. These parameters were carefully searched for each method to obtain maximally possible PSNR value (Figure 4). To ensure KMeans's solution has not been unlucky with initialization, it is run 100 times, and only the one with the lowest cost function is reported here.

The output of each method and their associated PSNR values are shown in Figure 5.

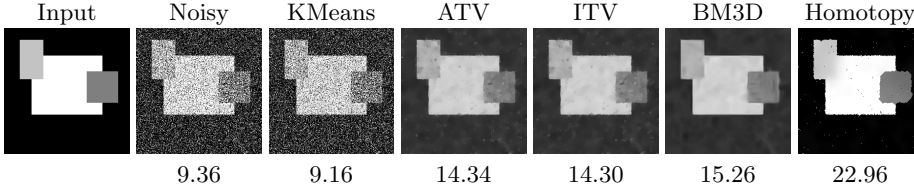


**Fig. 5.** Denoising a shape image using different methods. The best PSNR attained by each method is show below its image.

## 4 Conclusion

In this work we argue that the convolution integral associated with the Gaussian homotopy continuation can be computed in closed form for some interesting scenarios. Such closed form expression is of great importance and it makes the Gaussian homotopy method useful in practice. We explored this idea within two simple scenarios that involve difficult combinatorial nonconvexities.

## 5 Acknowledgment

This work is partially funded by the Shell Research. First author is grateful to William T. Freeman for the supporting this work.

## References

1. Barron, J.T., Malik, J.: Color constancy, intrinsic images, and shape estimation. In: Fitzgibbon, A.W., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV (4). Lecture Notes in Computer Science, vol. 7575, pp. 57–70. Springer (2012)
2. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: International Conference on Machine Learning, ICML (2009)
3. Bengio, Y.: Learning Deep Architectures for AI. Now Publishers Inc. (2009)
4. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Trans. Pattern Anal. Mach. Intell. 14(2) (1992)
5. Black, M.J., Rangarajan, A.: On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. International Journal of Computer Vision 19(1), 57–91 (1996)
6. Blake, A., Zisserman, A.: Visual Reconstruction. MIT Press (1987)
7. Brox, T.: From pixels to regions: partial differential equations in image analysis. Ph.D. thesis, Faculty of Mathematics and Computer Science, Saarland University, Germany (April 2005)
8. Brox, T., Malik, J.: Large displacement optical flow: Descriptor matching in variational motion estimation. IEEE Trans. Pattern Anal. Mach. Intell. 33(3), 500–513 (2011)
9. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. pp. 303–312. SIGGRAPH '96, ACM, New York, NY, USA (1996)
10. Dabov, Foi, Katkovnik, Egiazarian: Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. Image Processing, IEEE Transactions on 16(8), 2080–2095 (Aug 2007)
11. Dhillon, P.S., Keerthi, S.S., Bellare, K., Chapelle, O., Sundararajan, S.: Deterministic annealing for semi-supervised structured output learning. In: AISTATS 2012. vol. 15 (2012)
12. Erhan, D., Manzagol, P.A., Bengio, Y., Bengio, S., Vincent, P.: The difficulty of training deep architectures and the effect of unsupervised pre-training. In: AISTATS. pp. 153–160 (2009)
13. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. 32(9), 1627–1645 (2010)
14. Gehler, P., Chapelle, O.: Deterministic annealing for multiple-instance learning. In: AISTATS 2007. pp. 123–130. Microtome, Brookline, MA, USA (3 2007)
15. Goldluecke, B., Strekalovskiy, E., Cremers, D.: Tight convex relaxations for vector-valued labeling. SIAM J. Imaging Sciences 6(3), 1626–1664 (2013)
16. Goldstein, T., Osher, S.: The split bregman method for l1-regularized problems. SIAM J. Imaging Sciences 2(2), 323–343 (2009)
17. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief networks. Neural Computation 18(7), 1527–1554 (2006)

18. Kim, M., Torre, F.D.: Gaussian processes multiple instance learning pp. 535–542 (2010)
19. Krishnamurthy, V., Levoy, M.: Fitting smooth surfaces to dense polygon meshes. In: SIGGRAPH. pp. 313–324 (1996)
20. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Cam, L.M.L., Neyman, J. (eds.) Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability. vol. 1, pp. 281–297. University of California Press (1967)
21. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: Danyluk, A.P., Bottou, L., Littman, M.L. (eds.) ICML. ACM International Conference Proceeding Series, vol. 382, p. 87. ACM (2009)
22. Mobahi, H., Fisher III, J.W.: On the link between gaussian homotopy continuation and convex envelopes. In: Lecture Notes in Computer Science (EMMCVPR 2015). Springer
23. Mobahi, H., Rao, S., Ma, Y.: Data-driven image completion by image patch subspaces. In: Picture Coding Symposium (2009)
24. Mobahi, H., Ma, Y., Zitnick, L.: Seeing through the Blur. In: Proceedings of CVPR 2012 (2012)
25. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Comm. Pure Appl. Math. 42(5), 577–685 (Jul 1989)
26. Nielsen, M.: Graduated non-convexity by smoothness focusing. In: Proceedings of the British Machine Vision Conference. pp. 60.1–60.10. BMVA Press (1993)
27. Sindhwani, V., Keerthi, S.S., Chapelle, O.: Deterministic annealing for semi-supervised kernel machines. In: ICML '06. pp. 841–848. ACM, New York, NY, USA (2006)
28. Strekalovskiy, E., Chambolle, A., Cremers, D.: Convex relaxation of vectorial problems with coupled regularization. SIAM J. Imaging Sciences 7(1), 294–336 (2014)
29. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 2432–2439. IEEE (Jun 2010)
30. Terzopoulos, D.: The computation of visible-surface representations. IEEE Trans. Pattern Anal. Mach. Intell. 10(4), 417–438 (1988)
31. Turk, G., Levoy, M.: Zippered polygon meshes from range images. In: Proceedings of the 21st annual conference on Computer graphics and interactive techniques. pp. 311–318. SIGGRAPH '94, ACM, New York, NY, USA (1994)
32. Vural, E., Frossard, P.: Analysis of descent-based image registration. SIAM J. Imaging Sciences 6(4), 2310–2349 (2013)
33. Watson, L.T.: Theory of globally convergent probability-one homotopies for nonlinear programming. SIAM Journal on Optimization pp. 761–780 (2001)