Computer Aided Geometric Design 67 (2018) 1–20

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

p-Laplace diffusion for distance function estimation, optimal transport approximation, and image enhancement

Pierre-Alain Fayolle [a,*], Alexander G. Belyaev [b]

[a] Computer Graphics Laboratory, University of Aizu, Aizu-Wakamatsu, Japan
[b] Institute of Sensors, Signals and Systems, School of Engineering & Physical Sciences, Heriot-Watt University, Edinburgh, UK

ARTICLE INFO

Article history:
Available online 19 September 2018

Keywords:
Power-law diffusion
p-Laplacian
Distance function estimation
Optimal transport approximation
Image enhancement

ABSTRACT

In this paper, we propose ADMM-based numerical schemes for power-law diffusion equations involving the p-Laplacian $\operatorname{div}\left(|\nabla u|^{p-2}\nabla u\right)$ with constant ($p = \text{const}$) and variable ($p = p(\boldsymbol{x})$) exponents. Applications to distance function and optimal transport approximations (p is large and constant) and image enhancement (p is small and variable) are considered.

1. Introduction

Power-law diffusion equations involving p-Laplacian $\operatorname{div}\left(|\nabla u|^{p-2}\nabla u\right)$ with constant ($p = \text{const}$) and variable ($p = p(\boldsymbol{x})$) exponents attract a considerable attention in connection with studies on non-Newtonian fluids (Ružička, 2000), turbulence modeling (Wukie and Orkwis, 2017), phase transitions (Alamgir and von Luxburg, 2011), data clustering (Bühler and Hein, 2011), machine learning (Slepčev and Thorpe, 2017), and image processing (Blomgren et al., 1997; Chen et al., 2006). These and other studies devoted to power-law diffusions call for development of efficient numerical methods for solving second-order elliptic partial differential equations with p-type gradient nonlinearities and some very recent results are reported in Toulopoulos and Wick (2017), Caliari and Zuccher (2017).

In this paper, we apply the alternating direction method of multipliers (ADMM) to p-Laplace-type diffusion problems arising in distance function and optimal transport approximations (p is large and constant) and image enhancement (p is small and variable).

ADMM belongs to a family of splitting methods for solving variational minimization problems and has recently become a very popular tool for problems arising in imaging, machine learning, mobile communications, and many other scientific and engineering disciplines (Han et al., 2013; Hastie et al., 2015; Glowinski et al., 2017).

The basic problem we are dealing with in this paper is as follows. Let Ω be a bounded domain in $\mathbb{R}^n$ (for the sake of simplicity, we consider only the cases when $n = 2$ or $n = 3$), $p = p(\boldsymbol{x}) > 0$, and $u(\boldsymbol{x})$ be the solution to the p-Poisson equation

$$-\Delta_p u \equiv -\operatorname{div}\left(|\nabla u|^{p-2}\nabla u\right) = f(\boldsymbol{x}) \quad \text{in} \quad \Omega \tag{1}$$

* Corresponding author.
E-mail addresses: fayolle@u-aizu.ac.jp (P.-A. Fayolle), a.belyaev@hw.ac.uk (A.G. Belyaev).

https://doi.org/10.1016/j.cagd.2018.09.003
0167-8396/

subject to proper boundary conditions. Equivalently, $u(\boldsymbol{x})$ can be defined as the minimizer of the following p-diffusion energy

$$E_p(u) \equiv \int_\Omega \frac{|\nabla u|^p}{p} d\boldsymbol{x} - \int_\Omega f u \, d\boldsymbol{x} \longrightarrow \min . \tag{2}$$

Several approaches for numerically solving (1) have been proposed recently in Toulopoulos and Wick (2017) for $p = \text{const} > 1$ and (Caliari and Zuccher, 2017) for $p = p(\boldsymbol{x}) \geq 1$. These include combining finite elements with quasi-Newton and Newton-like methods (Toulopoulos and Wick, 2017; Caliari and Zuccher, 2017) and augmented Lagrangian techniques (Toulopoulos and Wick, 2017). When $p \geq 1$, since we deal with large and constant p, our ADMM-based approach seems appropriate and computationally efficient, as each of its iterations requires

- solving a Poisson equation with a properly updated right-hand side (so the corresponding matrix and its factorization are computed only once) and
- numerically solving a 1D polynomial equation.

Setting $0 < p < 1$ leads to a non-convex energy minimization problem. p-Laplace diffusion with $p = \text{const} \in (0, 1)$ and its applications to signal and image processing were considered in Nikolova (2015), Chartrand and Yin (2016) (see also references therein). Our approach is essentially an iteratively reweighted L_1 minimization scheme (Candes et al., 2008; Chartrand and Yin, 2008) but it seems we are the first who consider the case of variable exponent $p = p(\boldsymbol{x}) \in (0, 1]$, which, in addition, is adaptively constructed.

We consider the following applications of our numerical schemes for p-Laplace diffusion problems.

Distance function approximation. As shown in Bhattacharya et al. (1989), Kawohl (1990), $u(\boldsymbol{x})$, the solution to (1) with $f(\boldsymbol{x}) \equiv 1$ and subject to homogeneous Dirichlet boundary conditions $u(\boldsymbol{x}) = 0$ on $\partial\Omega$, converges to $\text{dist}(\boldsymbol{x})$, the distance from $\partial\Omega$, as $p \to \infty$. Thus our numerical approach can be used for approximating the distance function, as detailed in Section 3.1.

Optimal transportation. According to Evans and Gangbo (1999) (see also Ambrosio, 2003; Villani, 2003) the solution to (1) with $f(\boldsymbol{x}) = f_1(\boldsymbol{x}) - f_2(\boldsymbol{x})$, $\int_\Omega f(\boldsymbol{x}) \, d\boldsymbol{x} = 0$, and subject to the Neumann boundary conditions $\frac{\partial u}{\partial \boldsymbol{n}} = 0$, converges, as $p \to \infty$, to the solution of an optimal mass transportation of $f_1(\boldsymbol{x})$ to $f_2(\boldsymbol{x})$. So our numerical approach can be also used for approximating optimal transportation plans, as discussed in Section 3.2.

Image enhancement. As demonstrated in Section 4 a proper modification of (2) with variable and adaptively chosen image-dependent exponent $p = p(\boldsymbol{x}) \in (0, 1]$ can be used for image enhancement purposes. We show that our simple modification of the ADMM method leads to impressive image dehazing and low-light image enhancing results.

The distance function estimation problem we deal with differs from traditional problems of computing distances between points on surfaces or inside volumes bounded by surfaces (Crane et al., 2013; Solomon et al., 2014) (see also references therein). The distance-from-surface problem we consider in this paper is important for a number of applications including efficient implementation of level-set methods (Burger and Osher, 2005), surface reconstruction from scattered points (Calakli and Taubin, 2011), medical image analysis and visualization (Jones et al., 2006), computer graphics (Koschier et al., 2017), computational mechanics (Biswas et al., 2004), and computational fluid dynamics (Tucker, 2011; Roget and Sitaraman, 2013). Last but not least, we would like to mention the characteristic function method of Kantorovich (Babuška et al., 2003) and its extension by Rvachev and others (Freytag et al., 2011).

As mentioned above, our approach to the distance-from-surface estimation problem consists of applying ADMM to the variational problem (2). As demonstrated in Belyaev and Fayolle (2015), p-Laplace diffusion can be used for computing accurate approximations of the distance function. So we hope that the computational approach to (2) developed in this paper turns p-Laplace diffusion into a highly competitive tool for distance function estimation. It is also worth to note that in Belyaev and Fayolle (2015) ADMM was applied to a different energy minimization problem to achieve an accurate distance function approximation; in this paper, we show that (2) coupled with ADMM yields better results.

Applications of p-Laplace diffusion for approximating optimal transportation problems were previously considered in Mazón et al. (2014) (see also references therein). In Glowinski and Marroco (1975) (2) was minimized by an iterative scheme. Augmented Lagrangian methods are frequently used for numerical solution of optimal transportation problems (Benamou and Carlier, 2015; Solomon et al., 2014). In our study, we approximate the Monge transportation problem (the so-called earth mover's distance problem) by (2) and propose a variant of ADMM which shows a fast convergence rate. Our goal here is not to compete with the elegant and computationally efficient method of Solomon et al. (2014) and/or even the faster method of Li et al. (2018). Instead we are interested in developing a sufficiently flexible approach which, after proper adjustments, can be extended to deal with modifications of the Monge transportation problem such as, for example, those considered in Benamou and Carlier (2015), Mazón et al. (2014).

Finally, we consider an application of (2) with variable exponent $p = p(\boldsymbol{x})$, $0 < p(\boldsymbol{x}) \leq 1$, to image dehazing and low-light image enhancement problems. The novelty of our approach is twofold. Firstly, p-Laplace diffusion with $0 < p(\boldsymbol{x}) \leq 1$ has not been considered before to the best of our knowledge. Secondly, the variable exponent $p(x)$ is learnt (automatically

and adaptively determined) from an input image. To validate our approach as a proof-of-concept we compare it with two state-of-the-art image dehazing and low-light image enhancing methods (He et al., 2011; Guo et al., 2017).

2. Numerical solutions to p-Laplace diffusion equations

2.1. Splitting and relaxation

To solve (2), let us introduce a dual variable $\boldsymbol{\xi}(\boldsymbol{x})$ and consider instead the constrained problem

$$\int_\Omega \frac{1}{p}|\boldsymbol{\xi}|^p d\boldsymbol{x} - \int_\Omega f u\, d\boldsymbol{x} \longrightarrow \min, \quad \text{where} \quad \boldsymbol{\xi} = \nabla u,$$

then relaxing the constraint $\boldsymbol{\xi} = \nabla u$, we arrive at

$$\int_\Omega \frac{1}{p}|\boldsymbol{\xi}|^p d\boldsymbol{x} + \frac{r}{2}\int_\Omega (\nabla u - \boldsymbol{\xi})^2 d\boldsymbol{x} - \int_\Omega f u\, d\boldsymbol{x} \longrightarrow \min, \tag{3}$$

where $r > 0$.

If $\boldsymbol{\xi}(\boldsymbol{x})$ is fixed, then minimization of (3) w.r.t. $u(\boldsymbol{x})$ leads to the Poisson equation

$$-\Delta u = -\operatorname{div}\boldsymbol{\xi} + \frac{1}{r} f \quad \text{in } \Omega.$$

Let us now fix $u(\boldsymbol{x})$ and optimize w.r.t. $\boldsymbol{\xi}(\boldsymbol{x})$. For a given $\boldsymbol{x} \in \Omega$, the optimal $\boldsymbol{\xi}(\boldsymbol{x})$ can be found in the form $\boldsymbol{\xi}(\boldsymbol{x}) = c(\boldsymbol{x})\nabla u(\boldsymbol{x})$. Indeed, if $\boldsymbol{x}$ is fixed, the optimal $\boldsymbol{\xi}$ minimizes

$$\frac{1}{p}|\boldsymbol{\xi}|^p + \frac{r}{2}(\nabla u - \boldsymbol{\xi})^2$$

and, therefore, is situated on the straight segment connecting the origin of coordinates with ∇u. In other words, $\boldsymbol{\xi} = c\nabla u$ for some scalar c. To determine for each $\boldsymbol{x}$ the scalar function $c(\boldsymbol{x})$, $\boldsymbol{\xi}(\boldsymbol{x}) = c(\boldsymbol{x})\nabla u(\boldsymbol{x})$ is substituted in (3) leading to

$$\frac{1}{p}c^p|\nabla u|^p + \frac{r}{2}|\nabla u|^2(c-1)^2 \longrightarrow \min \tag{4}$$

or, equivalently

$$g(c) \equiv c^{p-1}|\nabla u|^{p-2} + r(c-1) = 0. \tag{5}$$

For a given $\boldsymbol{x} \in \Omega$, if $|\nabla u(\boldsymbol{x})| \neq 0$, equation (5) always has a root $0 < c < 1$, because $g(0) = -r < 0$ and $g(1) = |\nabla u|^{p-2} > 0$. Although in general it is not possible to find the root of (5) analytically, we can compute it numerically.

Newton's method. One step of the standard Newton's method for solving numerically $g(c) = 0$ is given by

$$c_{k+1} = c_k - \frac{g(c_k)}{g'(c_k)}. \tag{6}$$

We have $g'(c) = (p-1)|\nabla u|^{p-2}c^{p-2} + r$, and thus

$$c_{k+1} = c_k - \frac{c_k^{p-1}|\nabla u|^{p-2} + r(c_k - 1)}{(p-1)|\nabla u|^{p-2}c_k^{p-2} + r}$$

Given that $0 < c < 1$, we can start from $c_1 = 1$. Then, given that $g(1) = |\nabla u|^{p-2}$, and $g'(1) = (p-1)|\nabla u|^{p-2} + r$, we arrive at these first two steps for Newton's method

$$c_1 = 1, \qquad c_2 = 1 - \frac{|\nabla u|^{p-2}}{(p-1)|\nabla u|^{p-2} + r},$$

which provides already a good approximation.

If $|\nabla u| \leq 1$, Newton's method with $c_1 = 1$ converges quickly. However, if $|\nabla u| > 1$, Newton's method for solving (5) may demonstrate a slow convergence if started from $c_1 = 1$. Fig. 1 demonstrates such a case. To find a better starting value for solving (5) by Newton's iterations when $|\nabla u| > 1$, let us consider the polynomial $g(c\alpha)$ and note that setting

$$\alpha = \frac{r^{1/(p-1)}}{|\nabla u|^{(p-2)/(p-1)}}$$

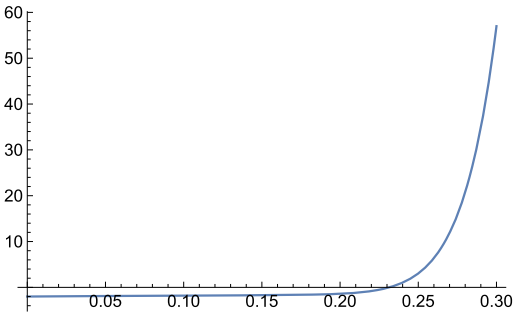

Fig. 1. The graph of $g(c)$ near its root ($p = 15$, $|\nabla u| = 5$ and $r = 2$). In this case, starting from $c_1 = 0$ or $c_1 = 1$ may require a dozen of Newton steps to achieve a reasonably accurate root localization.

turns $g(c\alpha) = 0$ into

$$c^{p-1} + \alpha c - 1 = 0. \tag{7}$$

Since (7) has a root between 0 and 1 and the influence of the term c^{p-1} in (7) is rather small, starting the Newton iterations for (7) from $c = 1$ is a good approximation. This justifies the following heuristic choice of starting point c_1 for the Newton iterations for (5)

$$c_1 = \begin{cases} 1 & \text{if } |\nabla u| \le 1 \\ \alpha & \text{if } |\nabla u| > 1. \end{cases}$$

With this selection of c_1, only few iterations are needed to obtain a good numerical approximation of the root of (5) in $[0, 1]$. In the example shown in Fig. 1, the starting point is $c_1 = \alpha = 0.236$ which is sufficiently close to the root.

Approximation. Instead of searching for $\boldsymbol{\xi}(\boldsymbol{x}) = c(\boldsymbol{x})\nabla u(\boldsymbol{x})$ and substituting in (3) to arrive at (4), we can use an approximation of the pth power of the L_p norm of $\boldsymbol{\xi}$ by

$$\int \frac{|\boldsymbol{\xi}|}{|\nabla u|^{1-p}} d\boldsymbol{x} \tag{8}$$

in (3). Here, as before, an iterative procedure is applied: ∇u_k determines $\boldsymbol{\xi}_k$ which, in its turn, determines ∇u_{k+1}. So we can assume that $\boldsymbol{\xi}$ is close to ∇u and, therefore, (8) approximates

$$\int |\boldsymbol{\xi}|^p d\boldsymbol{x}. \tag{9}$$

Now optimizing (3) with (8) instead of (9) w.r.t. $\boldsymbol{\xi}$ leads to a standard shrinkage operation used in sparse optimization (see, for example, Han et al., 2013, Section 4.4) and yields the closed form solution

$$\boldsymbol{\xi} = \max\left\{|\nabla u| - \frac{1/p}{r|\nabla u|^{1-p}}, 0\right\} \frac{\nabla u}{|\nabla u|}. \tag{10}$$

This approach works well for low values of p and thus is suitable for the image processing applications that we consider in Section 4, where $0 < p \le 1$. Unfortunately, this approximation does not work as well for the problems considered in Section 3, where p is large and where we use instead a few iterations of (6).

Iterative solver. To summarize, to compute a numerical solution to (2), we iteratively repeat two steps consisting in:

- minimizing (3) w.r.t. $u(\boldsymbol{x})$ with $\boldsymbol{\xi}(\boldsymbol{x})$ being fixed, which involves solving a simple Poisson problem,
- and then minimizing (3) w.r.t. $\boldsymbol{\xi}(\boldsymbol{x})$ with $u(\boldsymbol{x})$ being fixed, by solving numerically (5) or using the approximation (10).

2.2. Alternating direction method of multipliers (ADMM)

In order to improve the convergence of the precedent iterative scheme, the alternating direction method of multipliers (ADMM) can be used to solve (2). Let us add a Lagrange multiplier term in (3)

$$\int_\Omega \frac{1}{p}|\boldsymbol{\xi}|^p d\boldsymbol{x} + \int_\Omega \boldsymbol{\mu} \cdot (\boldsymbol{\xi} - \nabla u) d\boldsymbol{x} + \frac{r}{2}\int_\Omega (\boldsymbol{\xi} - \nabla u)^2 d\boldsymbol{x} - \int_\Omega f u \, d\boldsymbol{x} \longrightarrow \min, \tag{11}$$

where $r > 0$ and $\boldsymbol{\mu}(\boldsymbol{x})$ is the vector of the Lagrange multipliers.

Let us rewrite (11) as

$$\int_\Omega \left\{ \frac{1}{p}|\boldsymbol{\xi}|^p + \frac{r}{2}\left[\boldsymbol{\xi} - (\nabla u - \frac{\boldsymbol{\mu}}{r})\right]^2 - \frac{\boldsymbol{\mu}^2}{2r} - fu \right\} d\boldsymbol{x} \longrightarrow \min. \tag{12}$$

Then, if u and $\boldsymbol{\mu}$ are fixed the optimal $\boldsymbol{\xi}$ is proportional to $\nabla u - \frac{\boldsymbol{\mu}}{r}$ and thus has the form $c(\boldsymbol{x})\left(\nabla u(\boldsymbol{x}) - \frac{\boldsymbol{\mu}(\boldsymbol{x})}{r}\right)$. Substituting this expression for $\boldsymbol{\xi}$ in (12) leads to solving for c

$$\frac{1}{p}c^p \left|\nabla u - \frac{\boldsymbol{\mu}}{r}\right|^p + \frac{r}{2}\left|\nabla u - \frac{\boldsymbol{\mu}}{r}\right|^2 (c-1)^2 \longrightarrow \min$$

or equivalently

$$\left|\nabla u - \frac{\boldsymbol{\mu}}{r}\right|^{p-2} c^{p-1} + r(c-1) = 0, \tag{13}$$

which can be solved numerically using the same techniques that were used for solving (5). Alternatively, we can also apply an approximation to the pth power of the L_p norm of $\boldsymbol{\xi}$ leading to the closed form

$$\begin{aligned} \boldsymbol{\xi} &= \max\left\{|\boldsymbol{s}| - \frac{1/p}{r|\nabla u|^{1-p}}, 0\right\}\frac{\boldsymbol{s}}{|\boldsymbol{s}|}, \\ \boldsymbol{s} &= \nabla u - \frac{\boldsymbol{\mu}}{r}. \end{aligned} \tag{14}$$

With fixed $\boldsymbol{\xi}$ and $\boldsymbol{\mu}$, optimizing (11) w.r.t. $u(\boldsymbol{x})$ leads to solving the Poisson equation

$$-\Delta u = -\operatorname{div}\boldsymbol{\xi} - \operatorname{div}\frac{\boldsymbol{\mu}}{r} + \frac{1}{r}f \quad \text{in } \Omega. \tag{15}$$

Finally, we arrive at an iterative procedure for solving (1)

1: **repeat**
2: Optimize (12) for $\boldsymbol{\xi}^{(k+1)}$ by solving (13) or using (14),
3: Solve $-\Delta u^{(k+1)} = -\operatorname{div}\boldsymbol{\xi}^{(k+1)} - \operatorname{div}\frac{\boldsymbol{\mu}^{(k)}}{r} + \frac{1}{r}f$ for $u^{(k+1)}$,
4: Update $\boldsymbol{\mu}^{(k+1)} = \boldsymbol{\mu}^{(k)} + r\left(\boldsymbol{\xi}^{(k+1)} - \nabla u^{(k+1)}\right)$
5: **until** convergence

Here the rate of convergence is measured by the relative residual error

$$\frac{\|u^{(k+1)} - u^{(k)}\|_2}{\|u^{(k)}\|_2} \tag{16}$$

between two successive iterations and the above iterative process is stopped when (16) is less than a user-specified threshold ε. In our experiments we used a value of $\varepsilon = 10^{-3}$.

3. Distance function and optimal transport approximations with p-diffusion

3.1. Distance function

Formulation. Setting $f(\boldsymbol{x}) \equiv 1$ in (1) and considering the homogeneous Dirichlet boundary condition $u_p = 0$ on $\partial\Omega$ we arrive at

$$-\Delta_p u_p = 1 \text{ in } \Omega, \quad u_p = 0 \text{ on } \partial\Omega \tag{17}$$

with $2 \le p < \infty$.

Let $\operatorname{dist}(\boldsymbol{x})$ denote the shortest distance from $\boldsymbol{x}$ to $\partial\Omega$. As shown in Bhattacharya et al., Kawohl (1989, 1990),

$$u_p(\boldsymbol{x}) \to \operatorname{dist}(\boldsymbol{x}) \quad \text{as} \quad p \to \infty, \tag{18}$$

and the convergence is strong in Sobolev space $W^{1,k}(\Omega)$ for arbitrary $k > 1$. One can improve the above distance function approximation by considering

$$v_p(\boldsymbol{x}) = -\left|\nabla u_p\right|^{p-1} + \left[\frac{p}{p-1}u_p + \left|\nabla u_p\right|^p\right]^{\frac{p-1}{p}} \tag{19}$$

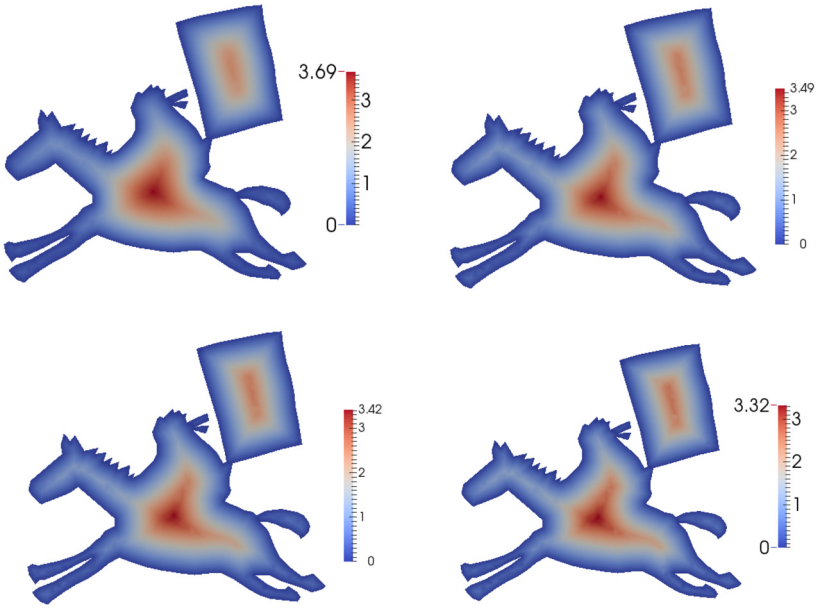

Fig. 2. Approximation to the distance function by the solution of the p-Poisson problem (17). Top row: $p = 5$ (left) and $p = 15$ (right). Bottom row: $p = 25$ (left) and the exact distance (right).

for which

$$v_p = 0 \quad \text{and} \quad \frac{\partial v_p}{\partial \boldsymbol{n}} = 1 \quad \text{on} \quad \partial\Omega. \tag{20}$$

Thus $v_p(\boldsymbol{x})$ delivers a more accurate approximation of $\text{dist}(\boldsymbol{x})$ near $\partial\Omega$. See Belyaev and Fayolle (2015) for a justification of this normalization and additional details. Interestingly normalization (19) has been recently employed in turbulence modeling for estimating the so-called wall distance (Wukie and Orkwis, 2017).

Initialization and Poisson distance. To initialize the numerical schemes from Section 2 to solve (17), we start from the solution to the homogeneous Dirichlet problem for the following Poisson equation

$$-\Delta u = 1 \quad \text{in} \quad \Omega, \qquad u = 0 \quad \text{on} \quad \partial\Omega. \tag{21}$$

This so-called Poisson distance function approximation (21) is used in several geometric modeling and image processing applications including action recognition (Gorelick et al., 2007; Gorelick et al., 2006), shape skeletonization (Aubert and Aujol, 2012), estimating the wall distance in turbulence modeling (Tucker, 1998), and geometric de-featuring (Xia et al., 2012).

Numerical experiments. Distance function approximations obtained by the p-Poisson distance are illustrated in Fig. 2 for a 2D domain. The solution to (17) is computed numerically by using ADMM (Section 2.2). The geometrical domain is triangulated, and the Poisson problems (15) are solved by finite elements using linear basis functions and homogeneous Dirichlet boundary conditions.

Fig. 2 illustrates the behavior of the solution to (17) for different values of $p = 5, 15$ and 25. Notice that the approximation to the exact distance function improves as p increases (compare to the image corresponding to the exact distance to the boundary).

The point-wise absolute and relative errors w.r.t. the exact distance to the boundary $\text{dist}(\boldsymbol{x})$ for the 2D domain of Fig. 2 are shown in Fig. 3 for $p = 25$. One can observe that the largest errors are located at and near distance function singularities (the medial axis of Ω) and concave angles of $\partial\Omega$. In particular, concave angles of $\partial\Omega$ worsen smoothness properties of the solutions to Poisson equation (15) near $\partial\Omega$ (Grisvard, 2011). In addition, a concave angle of $\partial\Omega$ weakens smoothness of $\text{dist}(\boldsymbol{x})$ near the vertex of the angle.

This phenomenon occurs for other methods based on solving elliptic boundary value problems, e.g. Poisson problems. It also occurs, for example, with the geodesics-in-heat method (Crane et al., 2013) adapted for estimating the distance from a 2D curve or a surface in 3D. This is illustrated in the bottom row of Fig. 3 (the time-step recommended in Crane et al. (2013) was used). The images also demonstrate that, in term of approximation quality, our approach delivers better results than the geodesics-in-heat method.

The relative residual error (16) between two successive iterations of the ADMM, is shown in Fig. 4 for different values of p during a few iterations of the ADMM. Only a few iterations are necessary to reach convergence. In contrast to Fig. 2,

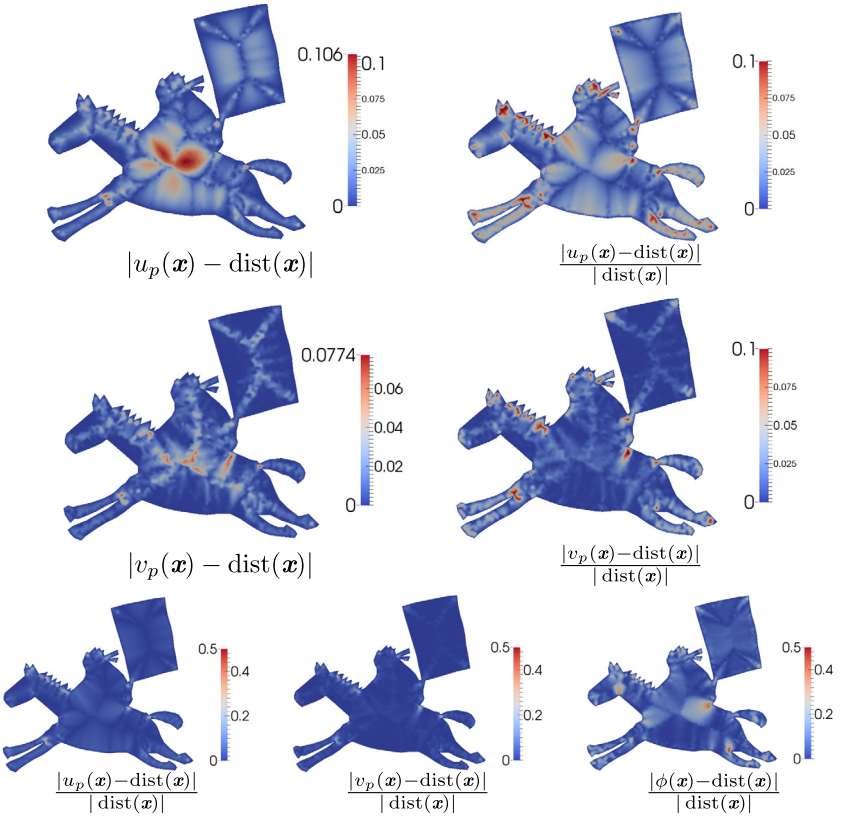

Fig. 3. Point-wise absolute (left) and relative (right) errors for the p-Poisson distance (top) and its normalization (19) (middle) w.r.t. the exact distance to the boundary of a 2D domain. Here $p = 25$. Bottom row: relative point-wise error for the p-Poisson distance (v_p, $p = 25$) on the left, its normalization (19) in the middle and the geodesics-in-heat method (ϕ) (Crane et al., 2013) on the right; the same colormap is used for comparison.

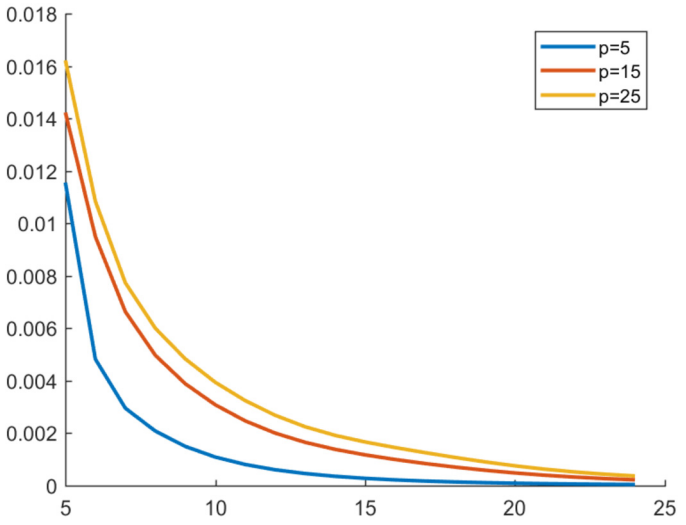

Fig. 4. Relative residual error for the solution of (17) on the rider domain for different values of p.

Fig. 4 shows advantages of using smaller values of p: the p-Laplacian is less degenerate and ADMM demonstrates a better convergence.

In Fig. 5, we compare the result obtained for $p = 25$ with the approach described in Belyaev and Fayolle (2015, Section 5), which is also based on ADMM iterations but is used to solve a different variational problem

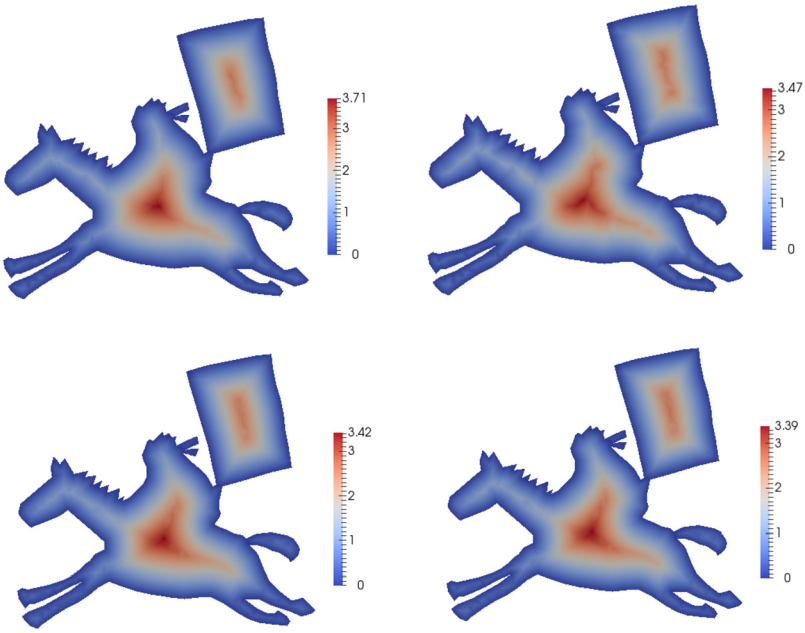

Fig. 5. Approximation to the distance function by: top row – the ADMM approach of (Belyaev and Fayolle, 2015) for 20 iterations, and until the residual error is below some threshold; bottom row – the solution of the p-Poisson problem (17) for $p = 25$, and its normalization (19).

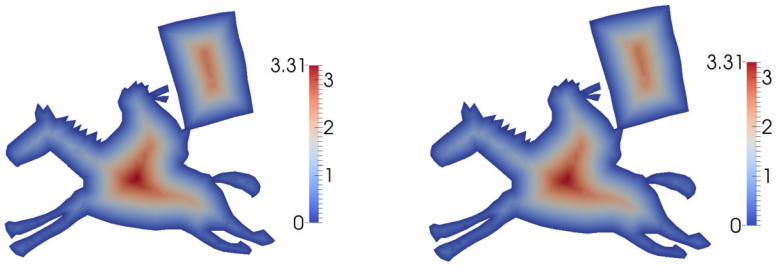

Fig. 6. Solution to (17) for large values of p (left: $p = 100$ and right: $p = 200$).

$$\int_{\Omega} (|\nabla u| - 1)^2 d\boldsymbol{x} \to \min, \quad u = 0 \text{ on } \partial\Omega. \tag{22}$$

One advantage of the p-Laplacian approach (17) over (22) for the distance function estimation consists of the normalization (19), which makes the approximation very accurate near the boundary. Another advantage is that for sufficiently high value of p, one can obtain a better approximation of the distance, with only a few iterations of ADMM.

While an approximation of the distance function based on the solution of the p-Poisson problem (17) was also presented in Belyaev and Fayolle (2015, Section 7), the numerical solution was computed with a traditional non-linear solver and limited to small values of p (the numerical examples shown in Belyaev and Fayolle (2015) are up to $p = 10$). The numerical approach proposed in this paper allows instead to solve efficiently (17) for much larger values of p, as already illustrated for $p = 25$ in Fig. 2 and for $p = 100$ and 200 in Fig. 6.

3D numerical experiments. Our approach consisting of applying ADMM iterations to (17) and using normalization (19) naturally works in 3D for approximating the distance to an input surface represented by a triangle mesh. The Poisson problems (15) are now solved on a tetrahedralization of the 3D domain by finite element with linear basis functions. Fig. 7 shows the numerical solution to (17) for $p = 15$ after a couple of iterations of ADMM. One notices that the normalization (19) allows to improve significantly the distance approximation (compare the left and middle images in Fig. 7 against the exact distance on the right).

The point-wise absolute error $|u_p(\boldsymbol{x}) - \text{dist}(\boldsymbol{x})|$ and relative error

$$\frac{|u_p(\boldsymbol{x}) - \text{dist}(\boldsymbol{x})|}{|\,\text{dist}(\boldsymbol{x})|}$$

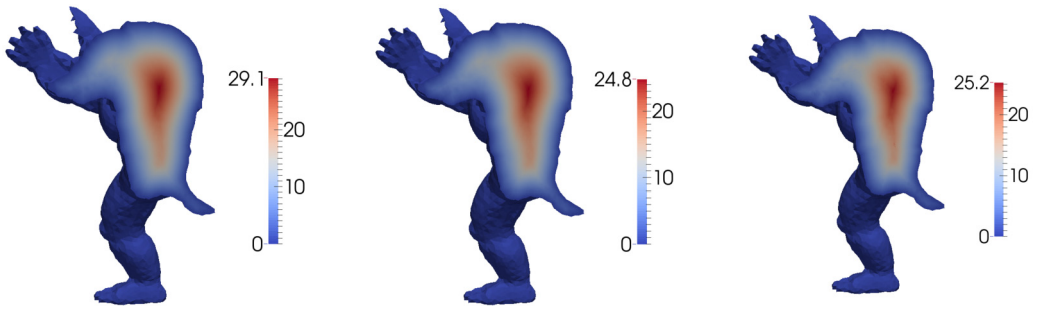

Fig. 7. p-Poisson distance (left), normalized p-Poisson distance (middle) obtained by solving (17) with ADMM (Section 2.2) and distance (right) to the surface of the Armadillo triangle mesh. In this example $p = 15$ is used.

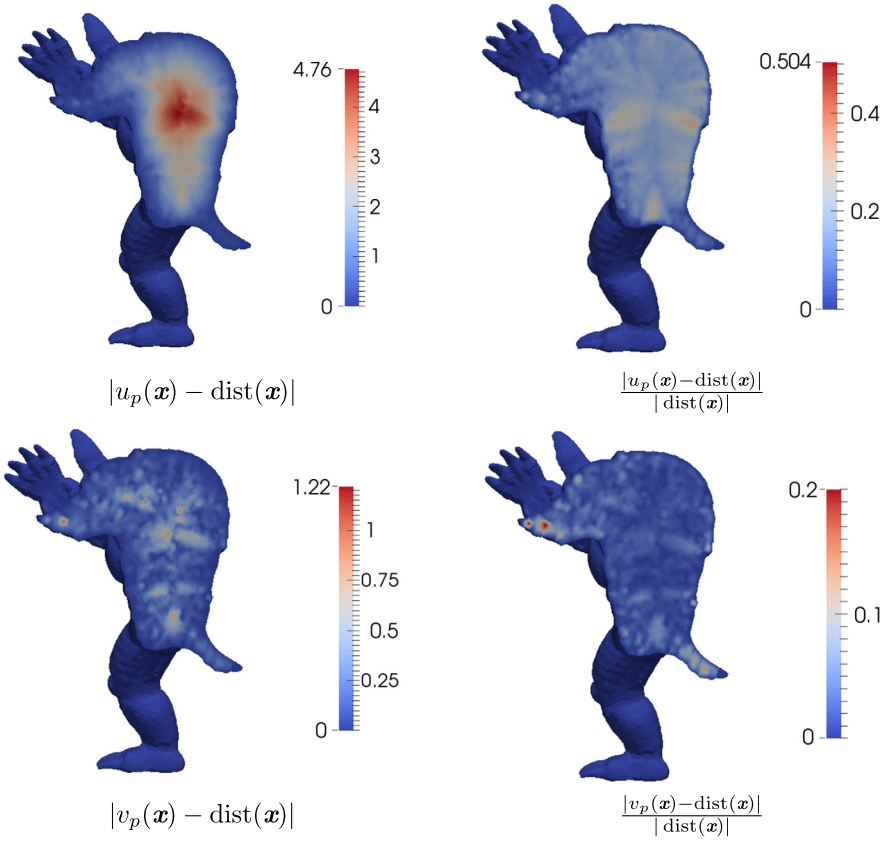

Fig. 8. Absolute (left) and relative (right) error for the p-Poisson distance (top) and its normalization (19) (bottom) w.r.t. the exact distance to the boundary (see also Fig. 7).

w.r.t. the exact distance $\mathrm{dist}(\boldsymbol{x})$ are shown in Fig. 8 for further comparison. Similar to the 2D case in Fig. 3, higher errors are located near distance function singularities of the domain and concave angles of the boundary. Additionally, higher errors can occur at poorly shaped tetrahedra, near small features of the domain boundary. One could use moving mesh approaches (Cao et al., 2003) and/or techniques for improving the shapes of the generated tetrahedra such as Klingner and Shewchuk (2008), Tournois et al. (2009) to improve the result.

Behavior near the boundary. The normalization (19) allows us to improve the approximation of the distance near the boundary. This is illustrated in Fig. 9 on a selected region of the Armadillo model (left). Both $p = 15$ and $p = 100$ are shown. For $p = 15$ the normalized solution (19) delivers already a close approximation of the distance (compare for the center-left column the top and bottom row). For increased values of p (for example $p = 100$ is shown in Fig. 9) the difference between the solution of (17) and its normalization becomes less evident, as illustrated in the center-right column.

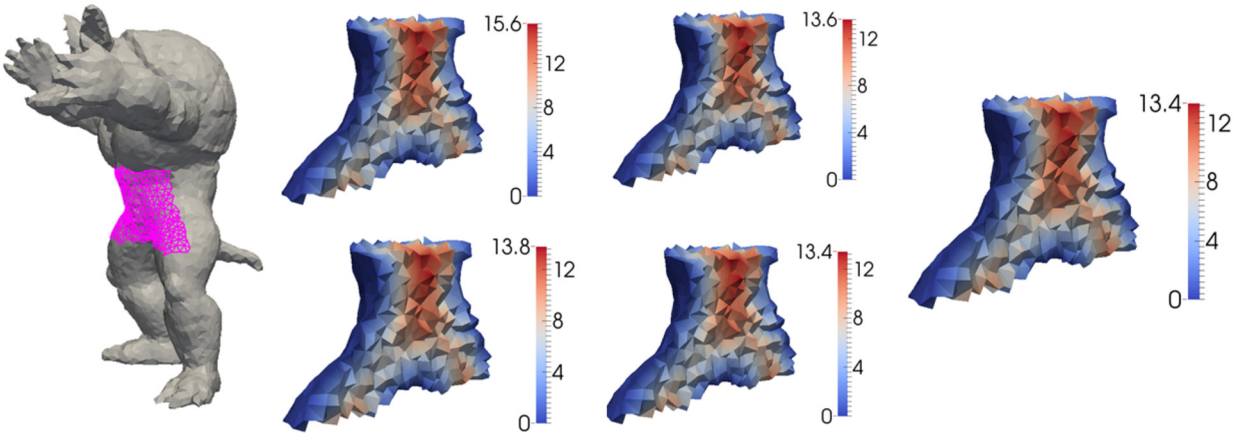

Fig. 9. Behavior near the boundary. Left-most: selected region on the Armadillo model. Four central images correspond to $p = 15$ (top-left), $p = 100$ (top-right), $p = 15$ normalized by (19) (bottom-left), and $p = 100$ normalized by (19) (bottom-right). Right-most: the exact distance from the boundary. The maximum values achieved by the approximate and exact distances provide us with a reliable indicator of approximation quality.

Table 1
Timings for solving (17) to approximate the distance function for triangulated 2D and tetrahedral 3D domains.

2D model	#nodes	#tris	time (s)
Rider, $p = 5$	1258	2228	0.08
Rider, $p = 15$	1258	2228	0.11
Rider, $p = 25$	1258	2228	0.15
3D model	**#nodes**	**#tets**	**time (s)**
Armadillo, $p = 5$	16962	88927	15.45
Armadillo, $p = 15$	16962	88927	16.1
Armadillo, $p = 25$	16962	88927	17.6
Fertility, $p = 5$	21811	114769	20.27
Fertility, $p = 15$	21811	114769	21.198
Fertility, $p = 25$	21811	114769	23.31

Computational timings. Table 1 shows computational timings for approximating the distance to a curve in 2D, and surface in 3D by solving (17) for different values of p and different domains. The times correspond to evaluating the distance approximation at every node of the domain. ADMM (Section 2.2) is used. The Poisson problems are solved on a triangulation or tetrahedralization of the domain using linear elements. The timings are more expensive in the 3D case as the corresponding matrices are less sparse. The implementation is written entirely in MATLAB and run on an Intel Core i3 at 3.3 GHz with 4GB of RAM. The code is not particularly optimized outside of using vectorized operations. Linear systems are solved with the backslash ('\') operator.

3.2. Optimal transport

Formulation. The connection between p-Poisson equations and optimal transportation problems was established by Evans and Gangbo in Evans and Gangbo (1999) (see also Ambrosio, 2003 and the description in Villani (2003, Section 2.4.6), where a connection to the diffusion of particles is made). Given two densities f_1 and f_2, with equal mass $\int_\Omega f_1 \, d\boldsymbol{x} = \int_\Omega f_2 \, d\boldsymbol{x}$, let us consider the p-Poisson problem

$$-\Delta_p u_p \equiv -\operatorname{div}\left(|\nabla u_p|^{p-2}\nabla u_p\right) = f_1 - f_2. \tag{23}$$

Evans and Gangbo showed that the limit of u_p, the solution to (23), as $p \to \infty$ is a function u_∞, which satisfies

$$-\operatorname{div}(a\nabla u_\infty) = f_1 - f_2, \quad a(\boldsymbol{x}) \geq 0, \quad |\nabla u_\infty| \leq 1.$$

This solution u_∞ is the transportation potential, and its gradient $-\nabla u_\infty$ gives the direction of optimal transport. Furthermore, the transport density $a(\boldsymbol{x})$ allows to compute the distance in the direction $-\nabla u_\infty$.

Another possible interpretation is as follows. The problem

$$-\Delta u = f_1 - f_2 \quad \text{in} \quad \Omega, \quad \frac{\partial u}{\partial \boldsymbol{n}} = 0 \quad \text{on} \quad \partial\Omega$$

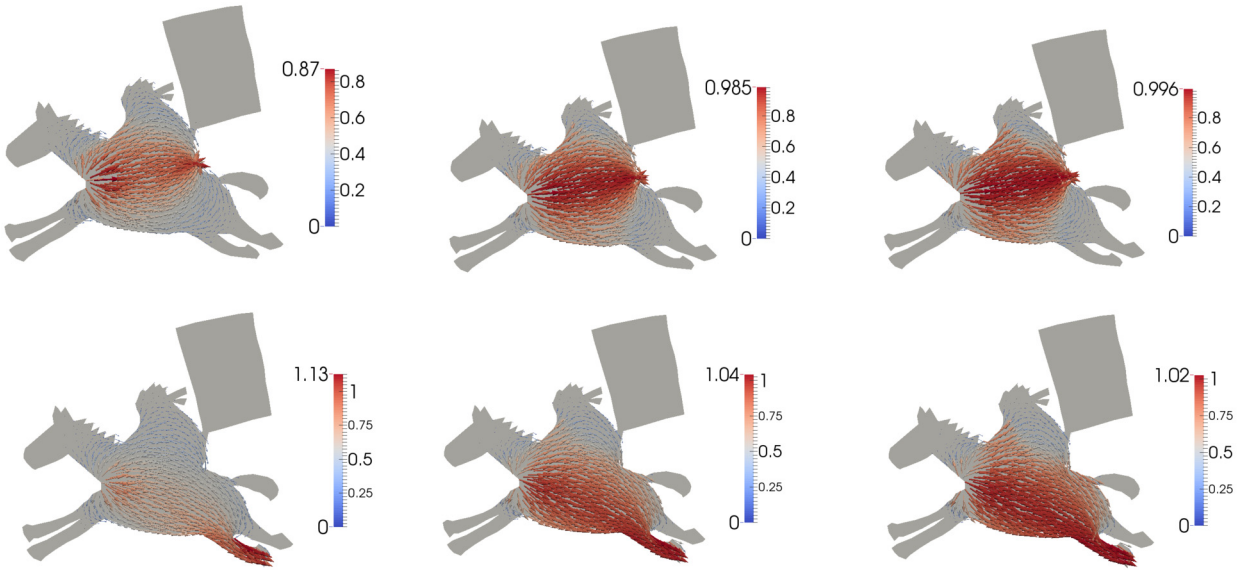

Fig. 10. Transport direction given by $-\nabla u_p$, with u_p solution to (23), for different sources and sinks (top and bottom row). From left to right: $p = 5, 15$ and 25.

defines a diffusion of particles born with rate $f_1(\boldsymbol{x})$ and dying with rate $f_2(\boldsymbol{x})$. The optimal trajectories for the particles happen where $|\nabla u|$ is large. Thus, in order to accelerate the diffusion, we can use instead

$$-\operatorname{div}(a\nabla u) = f_1 - f_2,$$

where $a(\boldsymbol{x})$ is large when $|\nabla u(\boldsymbol{x})|$ is large. For example, we can set $a(\boldsymbol{x}) = |\nabla u(\boldsymbol{x})|^{p-2}$ with $p > 2$, leading back to (23).

We arrive at (23) from (1) by setting $f(\boldsymbol{x}) \equiv f_1(\boldsymbol{x}) - f_2(\boldsymbol{x})$. Thus, the problem (23) can be solved with the numerical techniques developed in Section 2.

The method described here presents similarities with Solomon et al. (2014). Indeed, both approaches correspond to different formulations and approximations of the same problem, while both relying on ADMM for the numerical solution. In Solomon et al. (2014), the approximation comes from the spectral approximation of the curl free part, while in this work the approximation comes from solving a p-Poisson problem (p controls the approximation accuracy).

Boundary conditions. The reflecting boundary condition

$$\frac{\partial u_p}{\partial \boldsymbol{n}} = 0 \quad \text{on} \quad \partial\Omega$$

is used with the problem (23). Both methods in Section 2 contain a step that requires solving a simple Poisson sub-problem. In the case of the ADMM, it is

$$-\Delta u = -\operatorname{div}\boldsymbol{\xi} - \operatorname{div}\frac{\boldsymbol{\mu}}{r} + \frac{1}{r} f \text{ in } \Omega.$$

The corresponding boundary conditions for this problem are given by

$$\frac{\partial u}{\partial \boldsymbol{n}} = \left(\boldsymbol{\xi} + \frac{\boldsymbol{\mu}}{r}\right) \cdot \boldsymbol{n} \text{ on } \partial\Omega. \tag{24}$$

Numerical experiments. Fig. 10 illustrates the results obtained from the numerical solution of (23) for different values of p, when f_1 and f_2 are Dirac distributions. ADMM (Section 2.2) is used with a few Newton iterations (6). The linear Poisson problems (15) are solved by the finite element method, using linear basis functions on the triangulated domain with Neumann boundary conditions (24). As Dirac distributions are used for f_1 and f_2, the direction of optimal transport given by $-\nabla u_\infty$ correspond to the geodesics, as illustrated in Fig. 10, where $-\nabla u_p$ is shown for increasing values of p. The vector field $-\nabla u_p$ is colored according to its magnitude, and the size of the arrows is set proportional to the magnitude as well. The corresponding relative residual error at each iteration is shown in Fig. 11 for the same domain (for the choice of source and sink shown in the first row of Fig. 10).

Similar computations can be performed as well on a surface in $\mathbb{R}^3$, with the surface represented by a triangle mesh. The numerical approaches described in Section 2 requires discrete versions of the gradient, divergence and Laplace–Beltrami operators on a triangle mesh. Discrete exterior calculus (see for example Desbrun et al., 2008) can be used to obtain

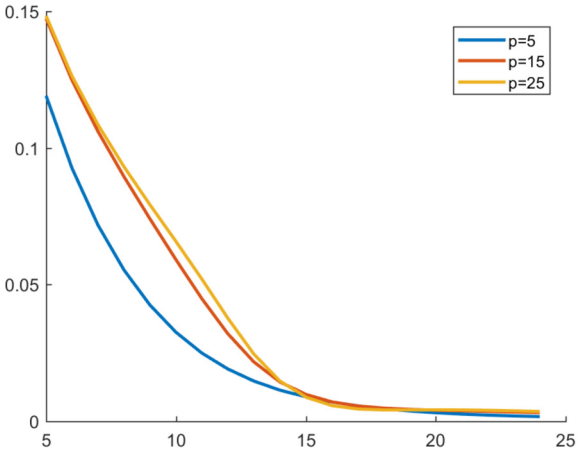

Fig. 11. Relative residual error for solving the optimal transport problem (23) on the rider domain for $p = 5, 15, 25$.

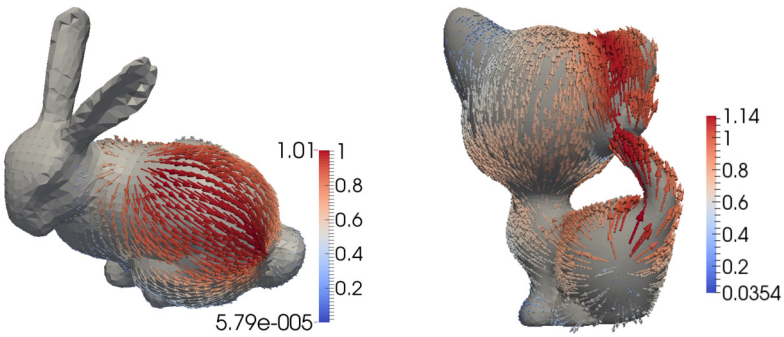

Fig. 12. Transport direction given by $-\nabla u_p$ for u_p solution to (23) on different triangle meshes. $p = 25$.

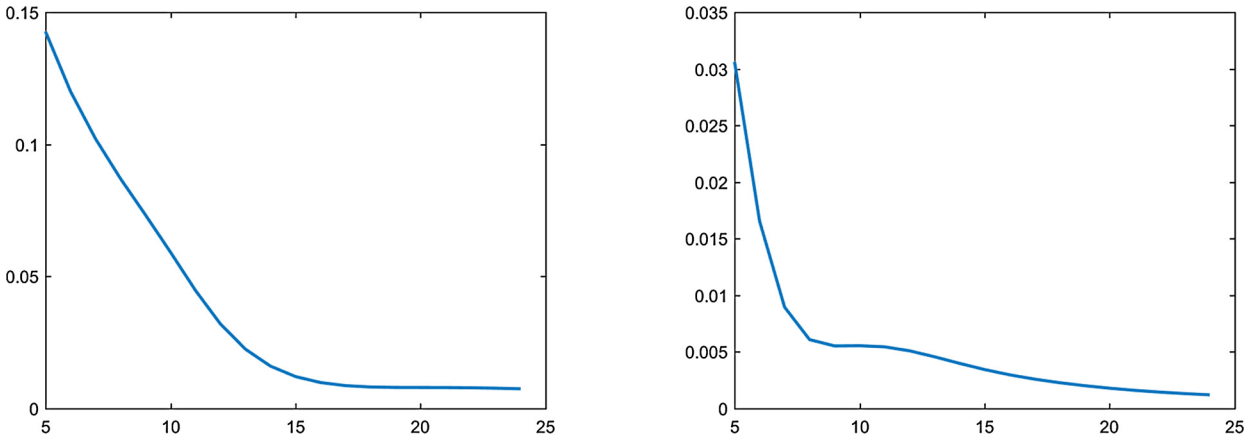

Fig. 13. Relative residual error for the examples of Fig. 12. Left: Bunny; Right: Kitten.

expressions for these discrete operators. Fig. 12 illustrates the results of numerical experiments on two different input triangle meshes corresponding to the solution of (23) for $p = 25$, with Dirac distributions for $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$. In these examples, (23) is solved by ADMM iterations (Section 2.2). The vector field $-\nabla u_p$ is shown in Fig. 12, with the vectors color and length proportional to the magnitude. As Dirac distributions are used for f_1 and f_2, the gradient flow of the optimal transportation can be interpreted as the geodesic from the source to the sink on the surface. The corresponding relative residual error for each domain is shown in Fig. 13.

Computational timings. Table 2 shows timings for computing an approximation to the optimal transport problem by solving (23) for different values of p and different triangulated domains in 2D and surfaces in 3D. In this case as well, the times

Table 2
Timings for solving (23) by ADMM for two Dirac masses on different triangulated 2D domains and triangle mesh surfaces in 3D.

2D model	#nodes	#tris	time (s)
Rider, $p = 5$	1258	2228	1.21
Rider, $p = 15$	1258	2228	1.98
Rider, $p = 25$	1258	2228	2.18
3D model	**#nodes**	**#tris**	**time (s)**
Bunny, $p = 5$	2762	5534	7.65
Bunny, $p = 15$	2762	5534	8.38
Bunny, $p = 25$	2762	5534	8.01
Kitten, $p = 5$	9447	18894	16.7
Kitten, $p = 15$	9447	18894	20.66
Kitten, $p = 25$	9447	18894	22.92

correspond to evaluating u_p, the solution to (23), at every node of the computational domain. ADMM is used and the Poisson problems are solved using linear elements. The same experimental conditions as for the distance approximation (Section 3.1) are used.

4. Image processing applications

We now turn our attention to using the numerical solutions of the p-Poisson equation (1) developed in Section 2 to image processing applications, namely to the task of image enhancement.

Curiously, image enhancement often starts from a proper image smoothing step. Let us start from the following p-diffusion minimization problem

$$\int_\Omega \frac{1}{p}|\nabla u|^p + \frac{\lambda}{2}\int_\Omega (H[u-g])^2 \longrightarrow \min, \tag{25}$$

where $g(\boldsymbol{x})$ is a given image and H is a smoothing operator (for example, H is the convolution with a Gaussian kernel). For $p = 1$, variational problem (25) becomes a member of the famous total variation (TV) family of methods introduced in Rudin et al. (1992). High popularity of TV-based methods is based on two pillars

- they deal with convex energies and
- they promote sparsity in the gradient domain.

The former property allows for using highly efficient computational methods developed for dealing with convex minimization problems. The latter one is crucial since typically natural images allow for accurate approximations by images, which are sparse in the gradient domain.

In particular, the variational model (25) with $p = 1$ and various choices for smoothing operators $H[\cdot]$ is widely used for texture+structure image decomposition (Meyer, 2001, Chapter 1), (Aujol et al., 2006), (Aubert and Kornprobst, 2006, Section 5.2), (Vese and Le Guyader, 2016, Chapter 5). Indeed, let us set $p = 1$ in (25) and assume that an input image $g(\boldsymbol{x})$ can be decomposed into the sum of an high-frequency texture component and a structure component sparse in the gradient domain. Then one can hope that the smoothing operator H in (25) suppresses the image texture and the TV term allows for recovering the structure component of $g(\boldsymbol{x})$.

Recently, however, non-convex regularization methods have started to attract interest, because they allow to recover sparser solutions from fewer measurements (Chartrand and Yin, 2016). In particular, the L_p-based regularization (corresponding to the p-diffusion energy) with $0 < p < 1$ has become a popular tool for sparsity enforcement (Selesnick and Bayram, 2014; Nikolova, 2015; Chartrand and Yin, 2016).

In our study, we deal with a variational model which is more general than (25). Namely we consider

$$\int_\Omega \left(a(\boldsymbol{x})|\nabla u|^{p(\boldsymbol{x})}\right) + \frac{\lambda}{2}\int_\Omega (H[u-g])^2 \longrightarrow \min, \tag{26}$$

with variable exponent $0 < p \equiv p(\boldsymbol{x}) \leq 1$ and weight function $a(\boldsymbol{x})$. Further we suggest to select $p(\boldsymbol{x})$ and $a(\boldsymbol{x})$ adaptively in order to achieve a better suppression of high-frequency image details.

While the use of p-diffusion energy term with variable exponent $p(\boldsymbol{x}) \geq 1$ has been previously considered in a number of works, such as Diening et al. (2011), Radulescu and Repovs (2015), Tang et al. (2017) for image processing applications, it seems that the case $0 < p(\boldsymbol{x}) \leq 1$ has not been considered before.

4.1. Adaptive image smoothing

We assume that small-scale image details and texture contribute to the high-frequency part of the level-set curvature

$$k(\boldsymbol{x}) = \operatorname{div}\left(\frac{\nabla g(\boldsymbol{x})}{|\nabla g(\boldsymbol{x})|}\right)$$

of the input image $g(\boldsymbol{x})$. Let $\tilde{k}(\boldsymbol{x})$ be obtained from $k(\boldsymbol{x})$ by suppressing high frequencies of $k(\boldsymbol{x})$. We would like to find an image which is close to the structure component of $g(\boldsymbol{x})$ and whose level-set curvature is close to $\tilde{k}(\boldsymbol{x})$. We achieve this by using a two-step procedure.

In the first step, we determine (learn) the variable exponent $p(\boldsymbol{x})$ and weight $a(\boldsymbol{x})$ from the input image $g(\boldsymbol{x})$. We solve (26) with $p = 1$ and $a = 1$ and measure a disparity between $k_1(\boldsymbol{x})$, the level-set curvature of the resulting image, and $\tilde{k}(\boldsymbol{x})$. The disparity indicates where additional smoothing should be applied to $g(\boldsymbol{x})$ and is therefore used to define $p(\boldsymbol{x})$ and $a(\boldsymbol{x})$.

In the second step, once $p(\boldsymbol{x})$ and $a(\boldsymbol{x})$ are given, we use a modification of the numerical methods described in Section 2 to solve (26).

Computation of $p(\boldsymbol{x})$ and $a(\boldsymbol{x})$. We obtain $\tilde{k}(\boldsymbol{x})$ by applying a simple low-pass filter, such as a Butterworth low-pass filter $\left[1 + \left(\frac{|\omega|}{b}\right)^m\right]^{-1}$ with cutoff frequency b, to $k(\boldsymbol{x})$ the level set curvature of the input image $g(\boldsymbol{x})$.

The next stage of our two-step approach consists of minimizing (26) with fixed values $p = 1$ and $a = 1$. We use a variant of the method described in Section 2.2. The variational problem to be minimized becomes

$$L(u, \boldsymbol{\xi}, \boldsymbol{\mu}) = \int_\Omega \Big\{ |\boldsymbol{\xi}| + \frac{\lambda}{2} H[u - g]^2 + \boldsymbol{\mu} \cdot (\boldsymbol{\xi} - \nabla u) + \frac{r}{2} |\boldsymbol{\xi} - \nabla u|^2 \Big\}, \tag{27}$$

where $\boldsymbol{\mu}(\boldsymbol{x})$ is a vector of Lagrange multipliers and $r > 0$.

For fixed u and $\boldsymbol{\mu}$, minimizing the subproblem w.r.t. $\boldsymbol{\xi}$ allows for a closed-form solution

$$\boldsymbol{\xi} = \max\left\{ |\boldsymbol{s}| - \frac{1}{r}, 0 \right\} \frac{\boldsymbol{s}}{|\boldsymbol{s}|}, \quad \boldsymbol{s} = \nabla u - \frac{\boldsymbol{\mu}}{r}. \tag{28}$$

For fixed $\boldsymbol{\xi}$ and $\boldsymbol{\mu}$, computing the optimal u leads to solving a linear PDE

$$-r(\Delta u - \operatorname{div}(\boldsymbol{\xi})) + \operatorname{div}(\boldsymbol{\mu}) + \lambda H^* H[u - g] = 0$$

which can be done in this case with the Fast Fourier Transform (FFT).

This leads to the following iterative procedure

repeat
  $\boldsymbol{\xi}^{(k+1)} = \max\left\{ |\boldsymbol{s}| - \frac{1}{r}, 0 \right\} \frac{\boldsymbol{s}}{|\boldsymbol{s}|}$, where $\boldsymbol{s} = \nabla u^{(k)} - \frac{\boldsymbol{\mu}^{(k)}}{r}$
  $u^{(k+1)} = \arg\min_u L(u, \boldsymbol{\xi}^{(k+1)}, \boldsymbol{\mu}^{(k)})$,
  $\boldsymbol{\mu}^{(k+1)} = \boldsymbol{\mu}^{(k)} + r\left(\boldsymbol{\xi}^{(k+1)} - \nabla u^{(k+1)}\right)$.
until convergence

Let $k_1(\boldsymbol{x})$ be the level set curvature of the solution obtained from minimizing (26) with $p = 1$ and $a = 1$. We estimate the difference between $\tilde{k}$ and k_1 by $|k_1 - \tilde{k}|$, smooth it by convolving it with the Gaussian kernel G_σ, and normalize

$$d(\boldsymbol{x}) = G_\sigma * \frac{|k_1 - \tilde{k}|}{\max\left(G_\sigma * |k_1 - \tilde{k}|\right)}. \tag{29}$$

Now the disparity $d(\boldsymbol{x})$ is used to define the weight $a(\boldsymbol{x})$ and variable exponent $p(\boldsymbol{x})$ by

$$a(\boldsymbol{x}) = \alpha\, d(\boldsymbol{x}) \quad \text{and} \quad p(\boldsymbol{x}) = 1 - \frac{d(\boldsymbol{x})}{2},$$

where α is a positive constant (we use $\alpha = 8$ in our experiments). Thus the bigger disparity $d(\boldsymbol{x})$ is, the bigger weight $a(\boldsymbol{x})$ and smaller exponent $p(\boldsymbol{x})$ are used.

Minimizing the weighted $p(\cdot)$-diffusion energy. Following ideas of Oh et al. (2013), Aström and Schnorr (2015) we employ a slight modification of ADMM to solve (26) numerically with adaptive weight $a(\boldsymbol{x})$ and exponent $p(\boldsymbol{x})$. Instead of (27), we now consider

$$L_{a,p}(u,\boldsymbol{\xi},\boldsymbol{\mu}) = \int_{\Omega} \Big\{ a(\boldsymbol{x})|\boldsymbol{\xi}|^{p(\boldsymbol{x})} + \frac{\lambda}{2} H[u-g]^2 + \boldsymbol{\mu}\cdot(\boldsymbol{\xi}-\nabla u) + \frac{r}{2}|\boldsymbol{\xi}-\nabla u|^2 \Big\}, \tag{30}$$

where $\boldsymbol{\mu}(\boldsymbol{x})$ is a vector of Lagrange multipliers and $r>0$ is a constant.

We follow the same iterative procedure as used previously (for $p=1$ and $a=1$). However, we don't have a closed-form solution for optimizing w.r.t. $\boldsymbol{\xi}$ and instead we resort to the approximation introduced in Section 2.2 for the pth power of the L_p norm of $\boldsymbol{\xi}$. This leads us to the following modification of (14)

$$\begin{aligned} \boldsymbol{\xi} &= \max\left\{ |\boldsymbol{s}| - \frac{a(\boldsymbol{x})}{r|\nabla u|^{1-p(\boldsymbol{x})}}, 0 \right\} \frac{\boldsymbol{s}}{|\boldsymbol{s}|}, \\ \boldsymbol{s} &= \nabla u - \frac{\boldsymbol{\mu}}{r}. \end{aligned} \tag{31}$$

The iterative procedure becomes

1: **repeat**
2: $\quad \boldsymbol{\xi}^{(k+1)} = \max\left\{ |\boldsymbol{s}| - \frac{a(\boldsymbol{x})}{r|\nabla u^{(k)}|^{1-p(\boldsymbol{x})}} \right\} \frac{\boldsymbol{s}}{|\boldsymbol{s}|}$, where $\boldsymbol{s} = \nabla u^{(k)} - \frac{\boldsymbol{\mu}^{(k)}}{r}$
3: $\quad u^{(k+1)} = \arg\min_u L_{a,p}(u, \boldsymbol{\xi}^{(k+1)}, \boldsymbol{\mu}^{(k)})$,
4: $\quad \boldsymbol{\mu}^{(k+1)} = \boldsymbol{\mu}^{(k)} + r\left(\boldsymbol{\xi}^{(k+1)} - \nabla u^{(k+1)}\right)$.
5: **until** convergence

Here, as before, the relative residual error (16) is used as a measure of convergence and the iterations are stopped once it goes below some threshold (10^{-3} was used in our experiments).

For the case $0 < p(\boldsymbol{x}) \le 1$ and the applications that we are considering here, this approximation provides the best result experimentally. Additionally, the standard ADMM with shrinkage (28) and the modified ADMM with (31) for the variable exponent case show similar performance in terms of numerical convergence in practice.

The only user-specified parameters are the weight λ in (26) and the parameter b controlling the amount of smoothing applied to the level-set curvature $k(\boldsymbol{x})$.

Given this approach for computing a numerical solution to (26), we now turn to two possible applications in image processing: removing haze in images and enhancement of low-light images.

4.2. Application to image dehazing

Let $g(\boldsymbol{x})$ be an original image with haze (see for examples the left images in Fig. 14), the solution $u(\boldsymbol{x})$ to (26) provides us with the image structure. The image texture is then obtained from $g(\boldsymbol{x}) - u(\boldsymbol{x})$. A simple enhancement for haze removal of the image is obtained from the following uniform enhancement

$$u(\boldsymbol{x}) + \beta(g(\boldsymbol{x}) - u(\boldsymbol{x})), \tag{32}$$

with $\beta > 1$ a constant.

While this approach works well when haze is uniformly distributed in the image, it seems better to use an approach with selective enhancement when the haze is located only in some region of the image. This leads to

$$u(\boldsymbol{x}) + h(\boldsymbol{x},\beta)\,(g(\boldsymbol{x}) - u(\boldsymbol{x})), \tag{33}$$

where $h(\cdot,\cdot)$ should be selected such that it is close to 1 in the region without haze, and equal to some larger value in the region with haze. We use a simple implementation based on the dark channel prior (He et al., 2011)

$$h(\boldsymbol{x},\beta) = \beta\left[g^{\text{dark}}(\boldsymbol{x})\right]^2 + 1, \tag{34}$$

where $g^{\text{dark}}(\boldsymbol{x})$ is the dark channel prior of the input image $g(\boldsymbol{x})$.

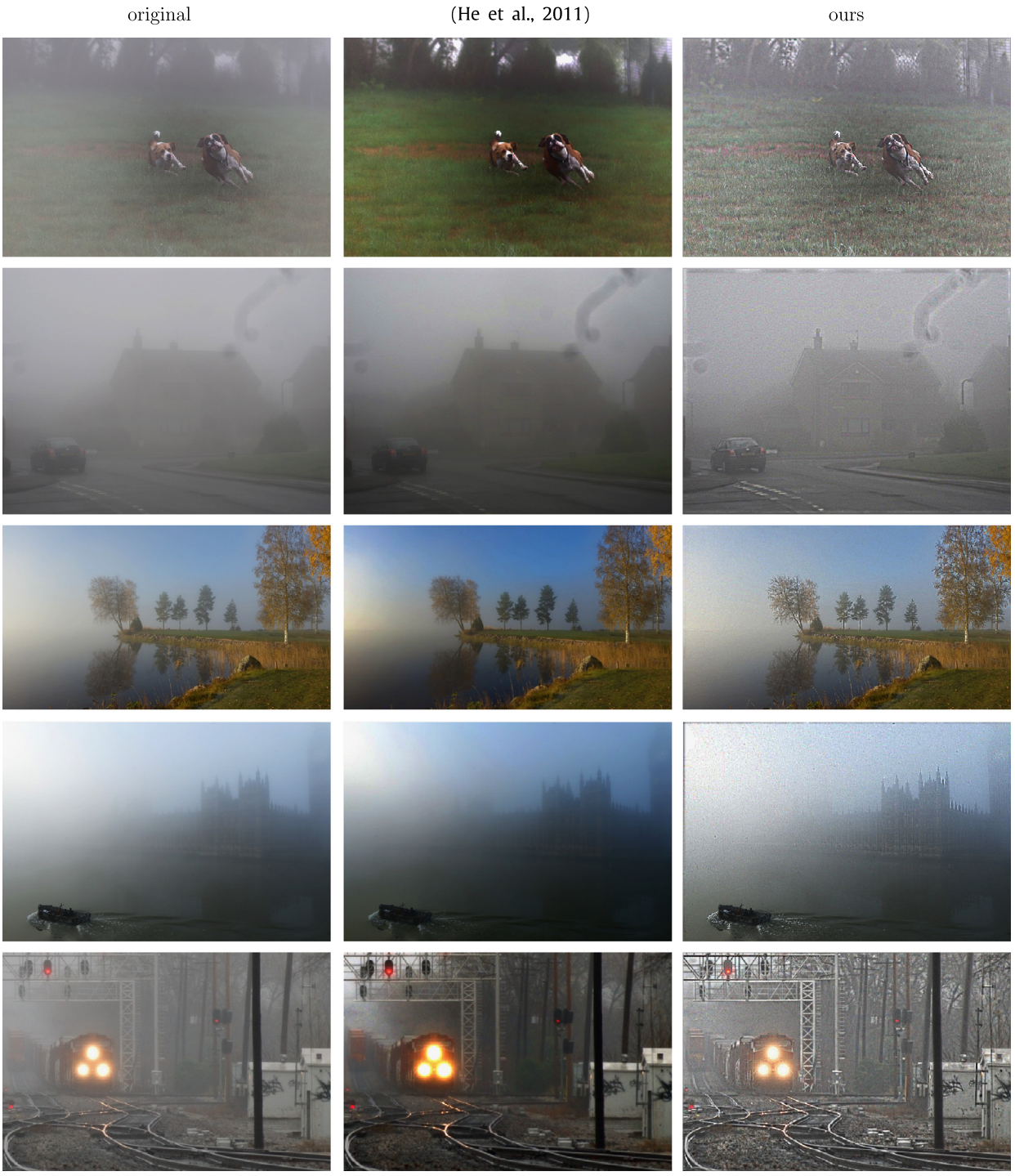

Fig. 14. Application to haze removal. From left to right: original image, results obtained with (He et al., 2011), and results obtained with our approach. These images are best seen on a computer screen.

In Fig. 14, we present a visual comparison of our approach with the popular dark channel prior method (He et al., 2011). While the dark channel prior method often generates more natural-looking results (as seen, for example, in the top row of Fig. 14), our two-pass approach allows to better extract fine image details. This better reconstruction of small-scale image features is visible, for example, on the second image from the bottom in Fig. 14. For this image, finer details of the clock and the building are better recovered by the two-pass approach based on solving (26). See also the left group of images in Fig. 16 with a magnified view on some of the features.

Fig. 15. Application to low-light image enhancement. From left to right: original image, results obtained from (Guo et al., 2017), and results obtained with our approach. These images are best seen on a computer screen.

4.3. Application to low-light image enhancement

The second application for the solution of (26) that we consider, is the enhancement of low-light images. The method described below is based on a recent work on low-light image enhancement (Guo et al., 2017). In this work, the authors consider a retinex-based model and assume that a captured image $L = R \circ T$ is the composition of the observed scene R and an illumination map T ($\circ$ stands for the element-wise multiplication). If L is a low-light image, then R can be considered as a recovered (enhanced) image. Given the illumination map T, the enhanced image is obtained by $R = \frac{L}{(T+\epsilon)}$, where the division is done element-wise and ϵ is a small regularization parameter for preventing division by zero. Thus the main ingredient of this approach consists of recovering the illumination map T which is achieved by applying a TV-based smoothing scheme to L, see Guo et al. (2017) for details.

Fig. 16. Left group of images demonstrates image dehazing: results obtained with the approach presented in this paper (left) and results obtained with (He et al., 2011) (right). Right group of images demonstrates low light image enhancement: results obtained with the approach presented in this paper (top) and results obtained with (Guo et al., 2017) (bottom). These images are best seen on a computer screen.

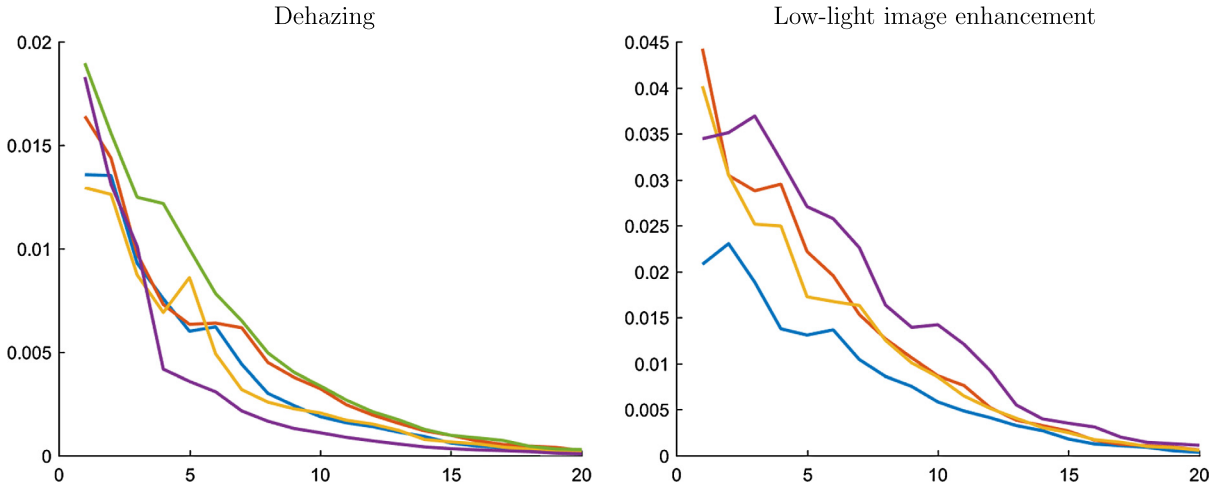

Fig. 17. Relative residual error plots (16) over a few iterations for dehazing and low-light image enhancing applications. Different colors are used for the residual plots corresponding to the different images used in Fig. 14 and 15.

It turns out that using (26) with adaptively selected weights (our two-pass scheme) instead of TV-based smoothing often allows for a better restoration of fine image details, as seen in Fig. 15, and in the magnified view in the right group of images in Fig. 16.

4.4. Visual convergence analysis

Fig. 17 shows plots of the relative residual error (16) for the ADMM algorithm used for image processing applications. Plots for both image dehazing (left) and low-light image enhancement (right) are provided for all the test images shown in Fig. 14 and 15. One can observe that the relative error quickly decreases and is below 10^{-3} within 20 iterations for both these image processing tasks.

5. Discussion and conclusion

In this work, we introduce efficient ADMM-based numerical schemes for solving p-Laplacian diffusion problems. We show that solutions to these models can be used to solve different problems in geometric modeling and image processing. With constant and large exponent $p > 1$, solutions to these models allow us to compute smooth approximants of the distance to surface function and to approximate solutions to optimal transportation plans.

While solving p-Poisson problems to compute smooth approximation of the distance to surface problems has already been considered before in Belyaev and Fayolle (2015), the ADMM-based numerical approach proposed in this paper exhibits a better performance.

By considering a variable exponent $0 < p(\boldsymbol{x}) \leq 1$, we show that solutions to our modified power-law diffusion model can be used for different tasks of image enhancement, such as dehazing and improvement of low-light images. Furthermore, we present a strategy for automatically and adaptively selecting the exponent $p(\boldsymbol{x})$. We demonstrate that our approach is competitive with state-of-the-art image dehazing and low-light image enhancing methods, and can even outperform them

in some cases, while being very generic. However, additional studies are needed in order to turn our approach into a powerful image processing tool.

Acknowledgements

We would like to thank the anonymous reviewers of this paper for their constructive and stimulating comments. The Bunny and Armadillo meshes are provided courtesy of the Stanford Computer Graphics Laboratory. The Kitten mesh is provided courtesy of the AIM@Shape Repository.

References

Alamgir, M., von Luxburg, U., 2011. Phase transition in the family of p-resistances. In: Neural Information Processing Systems. NIPS.
Ambrosio, L., 2003. Lecture notes on optimal transport problems. In: Mathematical Aspects of Evolving Interfaces. In: Lecture Notes in Mathematics, vol. 1812. Springer, pp. 1–52.
Aström, F., Schnorr, C., 2015. On coupled regularization for non-convex variational image enhancement. In: 2015 3rd IAPR Asian Conference on Pattern Recognition. ACPR, pp. 786–790.
Aubert, G., Aujol, J.-F., 2012. Poisson skeleton revisited: a new mathematical perspective. J. Math. Imaging Vis., 1–11.
Aubert, G., Kornprobst, P., 2006. Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations, 2nd edition. Applied Mathematical Sciences, vol. 147. Springer.
Aujol, J.-F., Gilboa, G., Chan, T., Osher, S., 2006. Structure-texture image decomposition – modeling, algorithms, and parameter selection. Int. J. Comput. Vis. 67 (1), 111–136.
Babuška, I., Banerjee, U., Osborn, J.E., 2003. Survey of meshless and generalized finite element methods: a unified approach. Acta Numer. 12, 1–125.
Belyaev, A., Fayolle, P.-A., 2015. On variational and PDE-based distance function approximations. Comput. Graph. Forum 34 (8), 104–118.
Benamou, J.D., Carlier, G., 2015. Augmented Lagrangian methods for transport optimization, mean field games and degenerate elliptic equations. J. Optim. Theory Appl. 167, 1–26.
Bhattacharya, T., DiBenedetto, E., Manfredi, J., 1989. Limits as $p \to \infty$ of $\Delta_p u_p = f$ and related extremal problems. Rend. Semin. Mat. Univ. Pol. (Torino) (Fascicolo Speciale Nonlinear PDEs), 15–68.
Biswas, A., Shapiro, V., Tsukanov, I., 2004. Heterogeneous material modeling with distance fields. Comput. Aided Geom. Des. 21, 215–242.
Blomgren, P., Chan, T.F., Mulet, P., Wong, C.K., 1997. Total variation image restoration: numerical methods and extensions. In: Proceedings of the IEEE International Conference on Image Processing, vol. 3, pp. 384–387.
Bühler, T., Hein, M., 2011. Spectral clustering based on the graph p-Laplacian. In: Proc. 26th Annual International Conference on Machine Learning. ICML '09, pp. 81–88.
Burger, M., Osher, S.J., 2005. A survey on level set methods for inverse problems and optimal design. Eur. J. Appl. Math. 16 (2), 263–301.
Calakli, F., Taubin, G., 2011. SSD: smooth signed distance surface reconstruction. Comput. Graph. Forum 30 (7), 1993–2002.
Caliari, M., Zuccher, S., 2017. Quasi-Newton minimization for the $p(x)$-Laplacian problem. Am. J. Comput. Appl. Math. 309, 122–131.
Candes, E.J., Wakin, M.B., Boyd, S.P., 2008. Enhancing sparsity by reweighted l_1 minimization. J. Fourier Anal. Appl. 14 (5–6), 877–905.
Cao, W., Huang, W., Russell, R.D., 2003. Approaches for generating moving adaptive meshes: location versus velocity. Appl. Numer. Math. 47, 121–138.
Chartrand, R., Yin, W., 2008. Iteratively reweighted algorithms for compressive sensing. In: IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP 2008, pp. 3869–3872.
Chartrand, R., Yin, W., 2016. Nonconvex sparse regularization and splitting algorithms. In: Splitting Methods in Communication, Imaging, Science, and Engineering. Springer, pp. 237–249.
Chen, Y., Levine, S., Rao, M., 2006. Variable exponent, linear growth functionals in image restoration. SIAM J. Appl. Math. 66 (4), 1383–1406.
Crane, K., Weischedel, C., Wardetzky, M., 2013. Geodesics in heat: a new approach to computing distance based on heat flow. ACM Trans. Graph. 32, 152.
Desbrun, M., Kanso, E., Tong, Y., 2008. Discrete differential forms for computational modeling. In: Discrete Differential Geometry. Springer, pp. 287–324.
Diening, L., Harjulehto, P., Hästö, P., Ruzicka, M., 2011. Lebesgue and Sobolev Spaces with Variable Exponents. Springer.
Evans, L.C., Gangbo, W., 1999. Differential equations methods for the Monge–Kantorovich mass transfer problem. Mem. Am. Math. Soc. 137, 66.
Freytag, M., Shapiro, V., Tsukanov, I., 2011. Finite element analysis in situ. Finite Elem. Anal. Des. 47 (9), 957–972.
Glowinski, R., Marroco, A., 1975. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problémes de dirichlet non linéaires. Revue française d'automatique, informatique, recherche opérationnelle, Anal. Numér. 9 (R2), 41–76.
Glowinski, R., Osher, S.J., Yin, W., 2017. Splitting Methods in Communication, Imaging, Science, and Engineering. Springer.
Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R., 2007. Actions as space–time shapes. IEEE Trans. Pattern Anal. Mach. Intell. 29 (12), 2247–2253.
Gorelick, L., Galun, M., Sharon, E., Basri, R., Brandt, A., 2006. Shape representation and classification using the Poisson equation. IEEE Trans. Pattern Anal. Mach. Intell. 28 (12), 1991–2005.
Grisvard, P., 2011. Elliptic Problems in Nonsmooth Domains. SIAM.
Guo, X., Li, Y., Ling LIME, H., 2017. Low-light image enhancement via illumination map estimation. IEEE Trans. Image Process. 26 (2), 982–993.
Han, Z., Li, H., Yin, W., 2013. Compressive Sensing for Wireless Networks. Cambridge University Press.
Hastie, T., Tibshirani, R., Wainwright, M., 2015. Statistical Learning with Sparsity: the Lasso and Generalizations. CRC Press.
He, K., Sun, J., Tang, X., 2011. Single image haze removal using dark channel prior. IEEE Trans. Pattern Anal. Mach. Intell. 33 (12), 2341–2353.
Jones, M.W., Baerentzen, J.A., Sramek, M., 2006. 3D distance fields: a survey of techniques and applications. IEEE Trans. Vis. Comput. Graph. 12 (4), 581–599.
Kawohl, B., 1990. On a family of torsional creep problems. J. Reine Angew. Math. 410 (1), 1–22.
Klingner, B.M., Shewchuk, J.R., 2008. Aggressive tetrahedral mesh improvement. In: Proceedings of the 16th International Meshing Roundtable. Springer, pp. 3–23.
Koschier, D., Deul, C., Brand, M., Bender, J., 2017. An *hp*-adaptive discretization algorithm for signed distance field generation. IEEE Trans. Vis. Comput. Graph. 23 (10), 2208–2221.
Li, W., Ryu, E.K., Osher, S., Yin, W., Gangbo, W., 2018. A parallel method for earth mover's distance. J. Sci. Comput. 75 (1), 182–197.
Mazón, J.M., Rossi, J.D., Toledo, J., 2014. Mass transport problems obtained as limits of p-Laplacian type problems with spatial dependence. Adv. Nonlinear Anal. 3 (3), 133–140.
Meyer, Y., 2001. Oscillating Patterns in Image Processing and Nonlinear Evolution Equations. University Lecture Series, vol. 22. AMS.
Nikolova, M., 2015. Energy minimization methods. In: Scherzer, O. (Ed.), Handbook of Mathematical Methods in Imaging, 2nd edition. Springer, pp. 157–204.
Oh, S., Woo, H., Yun, S., Kang, M., 2013. Non-convex hybrid total variation for image denoising. J. Vis. Commun. Image Represent. 24 (3), 332–344.
Radulescu, V.D., Repovs, D.D., 2015. Partial Differential Equations with Variable Exponents: Variational Methods and Qualitative Analysis, vol. 9. CRC Press.
Roget, B., Sitaraman, J., 2013. Wall distance search algorithm using voxelized marching spheres. J. Comput. Phys. 241, 76–94.

Rudin, L.I., Osher, S., Fatemi, E., 1992. Nonlinear total variation based noise removal algorithms. Physica D 60 (1), 259–268.
Ružička, M., 2000. Electrorheological Fluids: Modeling and Mathematical Theory. Springer.
Selesnick, I.W., Bayram, I., 2014. Sparse signal estimation by maximally sparse convex optimization. IEEE Trans. Signal Process. 62 (5), 1078–1092.
Slepčev, D., Thorpe, M., 2017. Analysis of p-Laplacian regularization in semi-supervised learning. Tech. rep., arXiv preprint, arXiv:1707.06213.
Solomon, J., Rustamov, R., Guibas, L., Butscher, A., 2014. Earth mover's distances on discrete surfaces. In: Proc. SIGGRAPH 2014. ACM Trans. Graph. 33 (4), 67.
Tang, C., Hou, C., Hou, Y., Wang, P., Li, W., 2017. An effective edge-preserving smoothing method for image manipulation. Digit. Signal Process. 63, 10–24.
Toulopoulos, I., Wick, T., 2017. Numerical methods for power-law diffusion problems. SIAM J. Sci. Comput. 39 (3), 681–710.
Tournois, J., Wormser, C., Alliez, P., Desbrun, M., 2009. Interleaving Delaunay refinement and optimization for practical isotropic tetrahedron mesh generation. ACM Trans. Graph. 28 (3), 75.
Tucker, P.G., 1998. Assessment of geometric multilevel convergence and a wall distance method for flows with multiple internal boundaries. Appl. Math. Model. 22, 293–311.
Tucker, P.G., 2011. Hybrid Hamilton–Jacobi–Poisson wall distance function model. Comput. Fluids 44 (1), 130–142.
Vese, L.A., Le Guyader, C., 2016. Variational Methods in Image Processing. CRC Press.
Villani, C., 2003. Topics in Optimal Transportation, vol. 58. American Mathematical Society.
Wukie, N.A., Orkwis, P.D., 2017. A p-Poisson wall distance approach for turbulence modeling. In: 23rd AIAA Computational Fluid Dynamics Conference.
Xia, H., Tucker, P.G., Coughlin, G., 2012. Novel applications of BEM based Poisson level set approach. Eng. Anal. Bound. Elem. 36, 907–912.