HAL open science

Temporally consistent gradient domain video editing

Gabriele Facciolo, Rida Sadek, Aurélie Bugeau, Vicent Caselles

To cite this version:

Gabriele Facciolo, Rida Sadek, Aurélie Bugeau, Vicent Caselles. Temporally consistent gradient domain video editing. International Conference on Energy Minimization Methods, Jul 2011, Saint Petersburg, Russia. pp.59-73. inria-00601455

HAL Id: inria-00601455

https://inria.hal.science/inria-00601455

Submitted on 20 Sep 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Temporally consistent gradient domain video editing

Gabriele Facciolo[1], Rida Sadek[1], Aurelie Bugeau[2], and Vicent Caselles[1]

[1] DTIC, Universitat Pompeu Fabra, 08023 Barcelona, Spain,
{gabriele.facciolo,rida.sadek,vicent.caselles}@upf.edu
[2] LaBRI, Université Bordeaux 1, 33405 Talence, France
aurelie.bugeau@labri.fr

Abstract. In the context of video editing, enforcing spatio-temporal consistency is an important issue. With that purpose, the current variational models for gradient domain video editing include space and time regularization terms. The spatial terms are based on the usual space derivatives, the temporal ones are based on the convective derivative, and both are balanced by a parameter β. However, the usual discretizations of the convective derivative limit the value of β to a certain range, thus limiting these models from achieving their full potential. In this paper, we propose a new numerical scheme to compute the convective derivative, the *deblurring convective derivative*, which allows us to lift this constraint. Moreover, the proposed scheme introduces less errors than other discretization schemes without adding computational complexity. We use this scheme in the implementation of two gradient domain models for temporally consistent video editing, based on Poisson and total variation type formulations, respectively. We apply these models to three video editing tasks: inpainting correction, object insertion and object removal.

Key words: Video editing, Poisson editing, temporal consistency, total variation, convective derivative, numerical methods

1 Introduction

In the context of static image editing, the insertion/removal of content in an image is generally performed using gradient domain methods. These methods allow editing an image without introducing artifacts at the boundaries of the edited regions. Consequently, gradient domain methods are widely used in image processing for: seamless cloning and compositing [1, 2], shadow removal [3], HDR compression [4], image inpainting [5–7], and matting [8] among others. Essentially, gradient domain image editing is based on the manipulation of the gradients of an image instead of its graylevels. The modified gradients are then integrated to recover the resulting image. This procedure prevents the appearance of seams at the boundaries of the edited region. For a more detailed introduction to gradient-domain methods, the reader is referred to [9]. In particular,

Poisson image editing [1] is one of such techniques; it formulates the problem variationally as

$$\min_u \int_{O \subset \Omega} \|\nabla u - g\|^2 dx; \quad \text{with} \quad u|_{\partial O} = u_0,$$

where $\Omega \subset \mathbb{R}^2$ is the image domain, $O \subset \Omega$ is the region to be edited, $g : O \to \mathbb{R}^2$ is the guidance vector field (e.g. gradient of the image to be composed), $u : O \to \mathbb{R}$ is the solution which best approximates the field g, and $u_0 : \Omega \to \mathbb{R}$ is the original image which provides the boundary conditions needed for reconstructing the solution u. The solution of this problem is computed solving the Poisson equation with Dirichlet boundary conditions $u|_{\partial O} = u_0$.

Since a video is nothing but a stack of images captured at evenly spaced times, it is natural to apply image editing techniques for video editing tasks. However, video editing poses some challenges that were not present in still image editing such as preserving the temporal consistency of the video. To cope with these challenges, video editing tasks are generally broken down into three steps: a) tracking the region where the editing is being performed [10, 11], a task that may also require detecting occlusions of the region; b) computing the new content of the region; and c) blending the new content with the original video so that the modification is not noticeable. The first two steps are usually handled by tracking and video inpainting techniques.

The focus of this work lies in computing the missing content in the editing region and enforcing the temporal consistency during the blending step. We consider three application scenarios: the correction of artifacts due to illumination changes in inpainted video sequences; the insertion of objects in a video; and the removal of objects. We restrict our study to the case where the inserted or removed object is affixed to a surface (for instance a sign on a wall). However, we do not restrict the motions of the camera nor of the surface in the scene. Our method also copes with occlusions and disocclusions of the inserted/removed object.

Temporal inconsistencies are sometimes more conspicuous than spatial ones. They tend to manifest in the edited video by introducing an annoying flickering. A simple solution to avoid the flickering is to consider the video as a three-dimensional volume and manipulate the spatio-temporal gradient to perform the editing operations. In [12], it is shown that setting the spatio-temporal gradient as the guidance field eliminates the flickering artifact. However, this approach is only valid when the edited region is not moving. That is, if the target region is moving, then the temporal consistency must be enforced by following its movement in the scene. This may be the case of inserting new content in a video, like the advertisement in Figure 1. In [13] the temporal consistency is imposed by a Kalman smoothing process applied along the trajectories of the pixels, which are computed by integrating the optical flow. Other variational approaches, such as [14, 15], enforce temporal consistency by adding a spatial regularization term to a temporal one, based on a derivative along the direction of the movement (the convective derivative). Both terms are then balanced by a parameter $\beta > 0$.

However, due to numerical problems that arise in the discretization of the convective derivative, β is restricted to a certain range of values. This restriction manifests itself in practice by requiring the guidance field to be present in all frames.

In this paper, we propose a novel numerical scheme for computing the convective derivative that allows β to take any desired value and, by that, allowing the model to achieve its full potential. In particular, in some applications, we may avoid computing the guidance field in all frames. Furthermore, the proposed scheme introduces fewer errors than other discretization schemes. We use this scheme in the definition of two gradient domain video editing formulations that use L^2- and L^1-norms, which lead to Poisson [1] and total variation type models, respectively. While the L^2-norm forces smooth solutions, the L^1-norm allows sharp transitions which may be desirable to better preserve textures.

Let us describe the organization of the paper. Section 2 presents our formulation for gradient domain video editing enforcing temporal consistency. In Section 3 we discuss the discretization of the convective derivative. In Section 4 we present the proposed new discretization scheme. Section 5 shows applications of the presented models to inpainting correction, object insertion, and object removal. The experiments show the ability of our model to handle occlusions automatically. Finally, in Section 6 we include some concluding remarks and future work.

Fig. 1: Temporal consistency in gradient domain video editing. From left to right, a video frame with motion vectors superimposed, the result of video editing imposing the temporal consistency disregarding the motion of the scene, and the result of video editing considering the motion of the scene.

2 Video editing model (continuous setting)

We propose a functional to perform gradient domain editing in a video while enforcing the temporal consistency. For that we minimize an energy defined on the spatio-temporal volume $\Pi := \Omega \times \mathbb{T}$

$$E_p(u) = \int_{O \subset \Pi} |\partial_v u(x,t)|^p + \beta \|\nabla_x u(x,t) - g(x,t)\|^p \, \mathrm{d}x \, \mathrm{d}t, \tag{1}$$

where $\Omega \subset \mathbb{R}^2$ is the rectangular image domain, $\mathbb{T} = [0, T]$ is the temporal domain, $O \subset \Omega \times \mathbb{T}$ is the spatio-temporal domain where the editing is performed (Figure 2 illustrates these domains), $u : \Omega \times \mathbb{T} \to \mathbb{R}$ is a scalar function representing the video, and $v : \Omega \times \mathbb{T} \to \mathbb{R}^2$ is a known velocity field obtained from the original video u_0. We consider $p \in \{1, 2\}$ and $\beta \geq 0$. Equation (1) is solved

with Dirichlet boundary conditions $u|_{\partial O \setminus \partial \Pi} = u_0$ at $\partial O \setminus \partial \Pi$ and homogeneous Neuman boundary conditions at $O \cap \partial \Pi$.

The second term in (1) is similar to other gradient domain image editing models [1, 2] and it is responsible for image editing at each frame. The spatial gradient with respect to the two spatial dimensions is denoted by ∇_x and the field $g : O \to \mathbb{R}^2$ is the guidance field which determines the new content of the frames.

The first term in (1) imposes the temporal consistency on the video. We denote by $\partial_v u(x,t)$ the *convective derivative* of u along v (a derivative of the video u along the optical flow field v) which we describe next. Please, note that throughout the text, we will refer to *temporal* boundary as the set of points $(x,t) \in \partial O$ such that the scalar product $\langle [v_1(x,t), v_2(x,t), 1], n(x,t) \rangle \neq 0$, where $n(x,t)$ is the normal to ∂O, and $v_i(\cdot)$ denotes the i-th component of v.

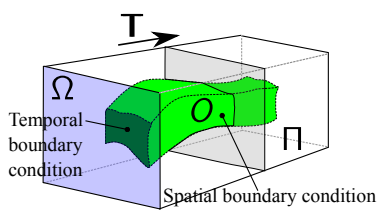


Fig. 2: Illustration of the spatio-temporal domain.

Temporal consistency and the convective derivative. A particle moving in the real world describes a trajectory $s : \mathbb{T} \to \Omega$ when seen in a video. The velocity field $v(s(t),t) = \frac{\mathrm{d}}{\mathrm{d}t} s(t)$ characterizes the motion of all the particles in the video. Then, given any particle and the associated trajectory s, the *temporal consistency* means that along s the graylevel of $u(s(t),t)$ is constant, which can be stated as

$$\frac{\mathrm{d}}{\mathrm{d}t} u(s(t),t) = 0.$$

Applying chain's rule to this equation we obtain

$$\partial_v u(x,t) := \nabla_x u(x,t) \cdot v(x,t) + \partial_t u(x,t) = 0, \quad \forall (x,t) \in \Omega \times \mathbb{T}. \tag{2}$$

Here, ∂_t denotes the derivative with respect to time ($\frac{\partial}{\partial t}$).

The left hand side of (2) corresponds to the definition of the *convective derivative* of u, which is nothing else than the directional derivative in the direction of the flow v. Therefore the temporal consistency can be stated in terms of the convective derivative: $\partial_v u(x,t) = 0$, $\forall (x,t) \in \Omega \times \mathbb{T}$.

Analysis of the energy. In order to highlight the roles of the terms of (1), let us re-write it as

$$\int_{\mathbb{T}} \int_{O_t} |\partial_v u(x,t)|^p \, \mathrm{d}x \, \mathrm{d}t + \beta \int_{\mathbb{T}} \int_{O_t} \|\nabla_x u(x,t) - g(x,t)\|^p \, \mathrm{d}x \, \mathrm{d}t, \tag{3}$$

where O_t denotes the slice of O at time t. Observe that the rightmost term is a stack of independent gradient domain image editing problems, each spatial term attaches to its guidance field $g(\cdot, t)$ and to the corresponding spatial boundary condition. The leftmost term enforces the temporal consistency by introducing a dependence between consecutive instants. In particular, it penalizes the changes of intensity along the trajectories specified by v, while attaching to the temporal boundary conditions.

The parameter $\beta \geq 0$ determines the mixing between the temporal consistency and the spatial term. When $\beta \to \infty$, the temporal consistency is irrelevant and the solution is equivalent to solving many independent image editing problems. On the other hand, when $\beta = 0$ the functional only enforces the temporal consistency. In this case, the information at the temporal boundary is transported along the trajectories specified by v; however, we are not able to deal with the illumination changes that are incorporated by the space boundary conditions. Moreover, in the case $\beta = 0$, if we consider only the boundary condition at time $t = 0$, then solving (1) amounts to solve the advection equation (2). A common choice of β for enforcing the temporal consistency is $\beta \sim \frac{1}{10}$ [15]. However, in some cases it is interesting to consider even lower values of β. This could be the case in the absence of the guidance field, here the information needs to be transported from the temporal boundaries. As we will see in Section 3.1 standard discretizations of (1) do not always behave in a satisfactory way when $\beta \to 0$.

The choice of $p \in \{1, 2\}$ leads to two models with different characteristics of the solution. Let us discuss these two cases.

Case $p = 2$: Here, the energy (1) is quadratic and its solution is computed by solving the linear system

$$(\partial_v^* \partial_v \cdot + \beta \mathrm{div}_x \nabla_x \cdot)\, u = \beta \mathrm{div}_x\, g, \tag{4}$$

where div_x denotes the spatial divergence and ∂_v^* is given by $\partial_v^* f = -\partial_t f + \mathrm{div}_x(vf)$. Equation (4) is of Poisson-type, and we solve it by using the conjugate gradient method.

The solution of (4) smoothly adapts to the boundary conditions of O. Moreover, any error due to inconsistencies between the boundary conditions and the potential field g is smoothly spread across the whole domain O.

Case $p = 1$: Here, the energy (1) takes the form of a total variation minimization problem. To solve it, we perform an implicit gradient descent:

$$u^{j+1} = \arg\min_u E_1(u) + \frac{1}{2\lambda}\|u - u^j\|^2, \tag{5}$$

where λ is a positive number. Each iteration of the gradient descent entails the resolution of a convex problem similar to the total variation model for denoising [16, 17]. The solution of (5) is computed by solving its dual problem [17].Defining

the dual variables $\psi : O \to R$ and $\xi : O \to \mathbb{R}^2$, we perform the fixed point iteration with time step $\tau \leq 1/8$:

$$\begin{aligned}\psi^{k+1} =& \frac{\psi^k + \tau\partial_v[\partial_v^*\psi^k + \beta \mathrm{div}_x \xi^k + u^j/\lambda]}{1 + \tau|\partial_v[\partial_v^*\psi^k + \beta \mathrm{div}_x \xi^k + u^j/\lambda]|},\\ \xi^{k+1} =& \frac{\xi^k + \tau\nabla_x[\partial_v^*\psi^k + \beta \mathrm{div}_x \xi^k + u^j/\lambda] + g/\lambda}{1 + \tau\|\nabla_x[\partial_v^*\psi^k + \beta \mathrm{div}_x \xi^k + u^j/\lambda] + g/\lambda\|},\end{aligned}$$

and at convergence the solution is recovered as $u = u^j + \lambda(\partial_v^*\psi + \beta \mathrm{div}_x \xi)$.

This method allows discontinuities of u in O and at its boundary. Its solution attaches to the boundary conditions reducing the effects of the illumination changes, but as opposed to the case $p = 2$ the transitions may not be smooth. While setting $p = 2$ favors smooth transitions, the model with $p = 1$ produces sharp transitions which may be desirable in some circumstances. It also allows a better preservation of textures. Figure 7e shows the distribution of the error in both cases and it also highlights the fact that texture is better preserved (we refer to Section 5 for details).

Remark 1. The formulation of (1) can also easily accommodate spatial weights for controlling the blending as in [18], and the temporal consistency term can be modified as in [15] to keep the reflectance properties of the objects.

3 Discretization of the model

This section deals with the discretization of (1). The convective derivative is defined using the velocity field v, so we start by commenting on the use of the *optical flow* as an approximation for v. We argue that the optical flow is suited for imposing the temporal consistency. Then, we study some numerical schemes for computing the convective derivative and their limitations. In next section, we introduce the *deblurring convective derivative*, a new numerical scheme for computing it.

In what follows we work in a discrete setting. The spatial domain is now a square lattice in $\mathbb{Z}^2$, $\Omega = \{0, 1, \ldots, N\}^2$, and the temporal domain is $\mathbb{T} := \{0, 1, \ldots, T\}$; therefore a video is represented as a stack of $T + 1$ digital images (frames). The spatial gradient is computed using forward differences and is denoted by ∇_x^+.

About the optical flow: As a discrete approximation of the velocity field, we use the optical flow computed on the original video $u_0 : \Pi \to \mathbb{R}$. We define the *forward optical flow* $v^+ : \Omega \times \mathbb{T} \to \mathbb{R}^2$ between two frames $u(\cdot, t)$ and $u(\cdot, t+1)$ as a vector field such that $u(x, t)$ and $u(x + v^+(x, t), t + 1)$ correspond to the same point in the scene. Similarly, the *backward optical flow* v^-, relates the frame at t with the one at time $t-1$. We discretize the temporal consistency constraint (2) using the forward optical flow as

$$\partial_v^+ u(x, t) := \hat{u}(x + v^+(x, t), t + 1) - u(x, t) = 0, \tag{6}$$

Fig. 3: Results with and without pre-processing of the optical flow. From left to right: a diagram describing the pre-processing, the removed flows, the result obtained using the pre-processed flow and without using it.

where $\hat{u}(x+v^+(x,t),t+1)$ is the bilinear interpolation of $u(\cdot,t+1)$ at $x+v^+(x,t)$. In a similar way, we can also discretize the temporal consistency constraint using the backward optical flow. The optical flow used in our experiments is obtained with the algorithm described in [19], but any optical flow algorithm with sub-pixel precision and regularization (or better, with edge preserving regularization) could be used.

As a last remark, the optical flow is not defined at the occluded/disoccluded areas of a frame. In these cases there may be no correspondence for a pixel in the next frame. While some optical flow algorithms produce occlusion maps, many others do not, so we pre-process the flow to identify the occlusion/disocclusion areas and remove them from the energy. A simple and direct way to achieve that is to compute $|\hat{u}_0(x+v^+(x,t),t+1)-u_0(x,t)|>tol$, where u_0 is the original video, and tol is a tolerance for the change in the gray levels. Flow vectors that do not satisfy the tolerance criteria are removed from the energy. The discrete energy becomes:

$$E_p(u)=\sum_{t\in\mathbb{T}}\sum_{x\in\Omega}\|Occ(x,t)\bar{\partial}_v u(x,t)\|^p+\beta\sum_{t\in\mathbb{T}}\sum_{x\in\Omega}\|\nabla_x^+ u(x,t)-g(x,t)\|^p, \quad (7)$$

where $\bar{\partial}_v$ denotes a discretization of the convective derivative, for instance $\partial_v^+ u(x,t)$, and $Occ:\Omega\times\mathbb{T}\to\{0,1\}$ indicates if a vector $v^+(x,t)$ satisfies the tolerance criterion or not. The inclusion of $Occ(\cdot,\cdot)$ implies that occluded/disoccluded pixels are only influenced by the spatial regularization term. Figure 3 illustrates with an experiment the benefits of pre-processing the optical flow.

3.1 Discretization of the convective derivative

The discretization of the convective derivative given in (6) corresponds to an *implicit upwind scheme with an adaptive stencil* [20]. That is, for computing the bilinear interpolation $\hat{u}(x+v(x,t),t+1)$ the convective derivative considers stencil points surrounding $(x+v^I(x,t),t+1)$ (where $v^I(x,t)$ is the integer part of the vector $v^+(x,t)$), this results in a potentially different stencil for each x. When used for simulating the advection equation, this adaptive scheme allows to achieve stability with time steps beyond the one prescribed by the Courant–Friedrichs–Lewy condition[20].

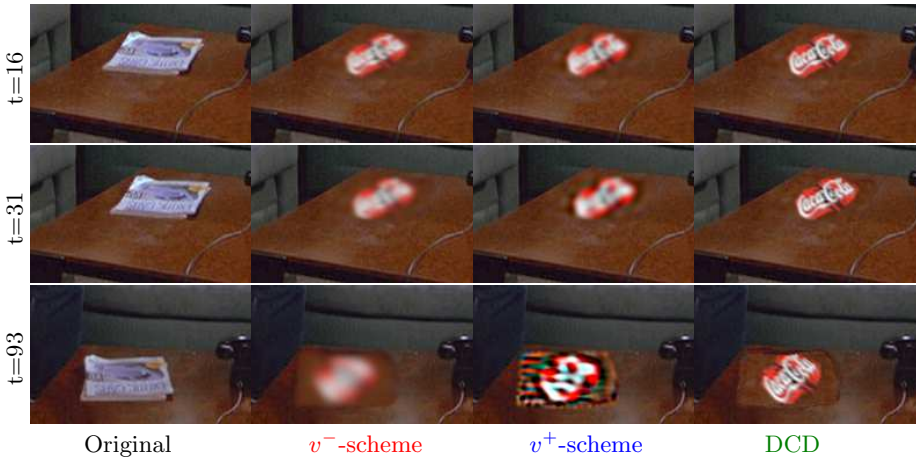


Fig. 4: Temporal propagation from frame $t = 0$, by solving (7) using $\beta = 0$ and $p = 2$. Columns from left to right show the: original frame, results obtained with v^--scheme (explicit), v^+-scheme (implicit), and DCD scheme for discretizing the convective derivative (see Section 4).

Using the forward optical flow v^+ we define the v^+-*scheme* for computing the convective derivative as

$$\partial_v^+ u(x,t) := \begin{cases} \hat{u}(x + v^+(x,t), t+1) - u(x,t) & \text{if } t < T, \\ 0 & \text{if } t = T \text{ or } (x + v^+(x,t), t+1) \notin O, \end{cases}$$

where $\hat{u} : \mathbb{Z}^2 \times \mathbb{T} \to \mathbb{R}$ is a bilinear interpolation of $u(\cdot, t+1)$ at $x + v^+(x,t)$, modified in order to account for the Neumann boundary conditions at $\partial(\Omega \times \mathbb{T})$.

Similarly, with the backward optical flow v^- we define the discrete v^--*scheme* for computing the convective derivative as

$$\partial_v^- u(x,t) := \begin{cases} u(x,t) - \hat{u}(x + v^-(x,t), t-1) & \text{if } t > 0, \\ 0 & \text{if } t = T \text{ or } (x + v^-(x,t), t-1) \notin O. \end{cases}$$

We implement these operators as sparse matrices, which allow us to easily compute their adjoints.

Evaluation: Let us consider the following experiment. We solve (7) with $\beta = 0$ (spatial term disabled) and with a Dirichlet boundary condition only at $t = 0$. Note that the guidance field g is undefined in this experiment. In the solution, we expect the content of the first frame to be *transported* according to the known optical flow v from time $t = 0$ to all subsequent frames.

This experiment is related to simulating the advection equation $\partial_v u = 0$ with initial condition at $t = 0$. When applied in this setting, the v^+- and v^--schemes for computing the convective derivative correspond to applying explicit or implicit upwind schemes for solving the advection equation, respectively. Using the v^--scheme can be seen as applying an explicit scheme, where a filter is applied to $u(\cdot, t-1)$ to obtain $u(\cdot, t)$, while the v^+-scheme behaves as an implicit one

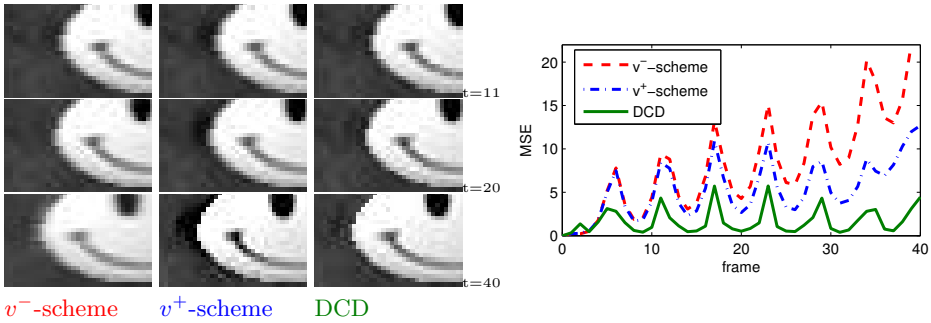


Fig. 5: Temporal propagation with $\beta = 0.01$, horizontal motion 0.175px/frame, g known and $p = 2$. After 40 frames the results of v^--scheme (explicit scheme) or v^+-scheme (implicit scheme) are more distorted than with DCD. The plots of the evolution of the error w.r.t. the ground truth (right) confirm it.

where an inverse filter relates $u(\cdot, t)$ with $u(\cdot, t+1)$. Figure 4 shows the results of this transport experiment. It is not surprising to see that for both, the explicit and implicit discretizations, the solutions are completely distorted only a few frames away from $t = 0$. The reason being that these schemes introduce a numerical diffusion (or oscillations) that is accumulated over time. These artifacts are due to approximation errors in the discretization of the differential operators. In particular, in the case of the above upwind schemes the approximation is only first-order accurate. Artifacts as those seen in Figure 4 are in general never observed, since models like (7) are always applied with $\beta > 0$ and with a known guidance field g for all the frames [14]. However, in some circumstances, like in the case of partial occlusions, the guidance field may not be easily computed. Moreover, as we see in Figure 5, even with $\beta > 0$ and with a known g, these discretizations induce noticeable errors in the result. We have seen that due to discretization issues we are unable to cope with a small value of β and, therefore, unable to use (7) to its full potential.

4 The deblurring convective derivative (DCD)

It is interesting to note that, in the experiment shown in Figure 4, the effects of the explicit and implicit schemes for discretizing the convective derivative are somehow opposite. The explicit scheme introduces blurring in the solution, while the implicit scheme sharpens the solution but also introduces oscillations. This motivates the search for a new scheme which allows the convective derivative to preserve the transported information for longer periods of time. The idea of the *deblurring convective derivative* (DCD for short) is to attain this objective as the balance of two opposing processes: implicit and explicit schemes. That is, applying both the v^+- and v^--schemes to moderate each other's effects.

Let us analyze in detail the effect of using the explicit upwind scheme (v^--scheme) while solving (7) by considering an illustrative example. We consider a similar problem as before, where only the initial frame ($t = 0$) is known, and we have a constant optical flow $v^- = [0.3, 0]^T$, $v^+ = [-0.3, 0]^T$ (with this flow the

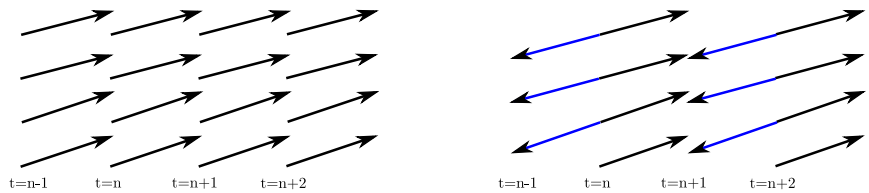


Fig. 6: On the left, we show a diagram depicting the v^+-scheme for discretizing the convective derivative. On the right, we show a diagram corresponding to the DCD: it alternates between v^- and v^+-schemes.

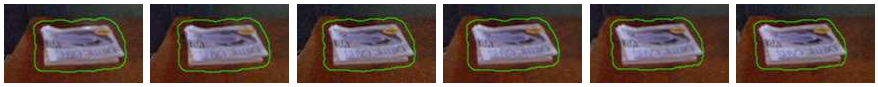
problem reduces to 1D). Taking $\beta = 0$ and $p = 2$ in (7) and analyzing the first two frames of the sequence we get: $\min_u \sum_{x\in\Omega} \|\partial_v^- u(x,1)\|^2$. It is easy to see that the values of $u(\cdot,1)$ which minimize this energy are explicitly determined by applying the filter $[0.7, 0.3]$ (coefficients specific for this flow) to the rows of frame $u(\cdot,0)$. Similarly, frame $u(\cdot,2)$ is obtained by filtering $u(\cdot,1)$, and so on. Denoting the filtering operator as M_{v^-}, then we can describe this relation as: $u(\cdot,t+1) = M_{v^-}u(\cdot,t)$. In this case, the minimum of the complete energy (for all the frames) is 0, and the solution is increasingly blurry with t since $u(\cdot,t) = (M_{v^-})^t u(\cdot,0)$, which confirms what we observed in Figure 4.

A similar analysis for the implicit upwind scheme (v^+-scheme) reveals that the solution of the problem restricted to the first frame, $\min_u \sum_{x\in\Omega} \|\partial_v^+ u(x,0)\|^2$, satisfies the relation $M_{v^+}u(\cdot,1) = u(\cdot,0)$, where M_{v^+} applies the filter $[0.3, 0.7]$ to the rows of $u(\cdot,1)$. Therefore, $u(\cdot,1)$ is given by the pseudo-inverse of M_{v^+} applied to $u(\cdot,0)$. The repeated application of the pseudo-inverse acts as an inverse smoothing. This sharpening enhances the high frequencies in the solution but also introduces numerical artifacts, as seen in Figure 4. Let us mention that, by solving the linear system using the conjugate gradient method, this problem is mitigated by its regularization effect.

The DCD takes advantage of the fact that these two schemes have opposite effects on the data (one blurs while the other sharpens) and alternates between them. Shortly, if between $t = 0$ and $t = 1$ we apply the v^+-scheme (implicit), then from frame $t = 1$ to $t = 2$ we apply v^--scheme (explicit) and so on. The diagrams in Figure 6 shows how the temporal derivatives are taken.

Note that there is no computational overhead in the use of DCD with respect to implicit or explicit schemes. It can also be applied sequentially on pairs of frames [14] without the need of solving the complete system associated to (7).

In the DCD, a step with the v^+-scheme permits to recover the frequencies smoothed by the previous step (v^--scheme). However, it may also introduce other high frequencies, which on the long term will build up as high frequency artifacts. These artifacts can then be removed using a low-pass filter as a post-processing step. In general the DCD preserves the transported information for much longer periods of time without noticeable decay. The performance of the proposed method is shown in Figures 4 and 5, where it is compared with the implicit and explicit upwind schemes. Observe that the texture is preserved for more than 70 frames (which is considered as a very long time).



(a) Original sequence.

(b) Inpainted sequence.

(c) Corrected inpainting with $p = 1$

(d) Corrected inpainting with $p = 2$

(e) See caption.

Fig. 7: Correcting an inpainted domain to impose spatio-temporal consistency using Equation (7) with $p = \{1, 2\}$ and $\beta = 1$. Figure (e), left (resp. right), shows the modulus of the difference of the magnitude of the gradients corresponding to the first column of images 7b and 7c (resp. 7b and 7d). These images have been jointly scaled to take values in $[0, 255]$ (max value $= 16$). Note that in the case $p = 2$ the differences are spread across the domain.

5 Applications

What we have presented so far can be used for a variety of applications. Here we present some experiments that illustrate the usefulness of these techniques in three applications:

1. Correcting an inpainted domain by making it temporally consistent.
2. Inserting an object on a surface while handling situations where the object is partially occluded/disoccluded.
3. Removing an object from a surface while handling situations where the object is partially occluded/disoccluded.

In what follows we discuss the experimental results for each application. The full set of results can be viewed at `http:\\gpi.upf.edu\static\emmcvpr11`.

Inpainting correction: In this scenario, we are given an inpainted video sequence. Thus, we can compute the guidance field in all frames. We would like to correct this inpainted sequence so that it becomes spatio-temporally consistent. In order to achieve that, we use the inpainted video as our input and solve (7) for $p = \{1, 2\}$ and $\beta = 1$. Figure 7a shows the original sequence. Figure 7b shows the resulting inpainted sequence. The sequence has been inpainted with an algorithm that generates a temporally consistent inpainting, yet not consistent with

(a) Original sequence. Frames $t = 1..6$ and $t = 24..30$.

(b) Inserted object using Equation (7) with $p = 1$

(c) Inserted object using Equation (7) with $p = 2$

Fig. 8: Object insertion experiment with handling of occlusions and disocclusions. Figures (a) show some frames of the original sequence, with the editing region superimposed. Figures (b) and (c) show the results obtained after solving (7) with $p = 1$ and $p = 2$, respectively. In this experiment the new object is only provided in the first and last frames of the sequence, and the editing region in-between is unknown. A closer inspection of the results corresponding to $p = 1$ and $p = 2$ reveals that the model with $p = 1$ preserves better the texture.

respect to the illumination changes at each frame. Figures 7c and 7d show the result obtained by solving (7) with $p = 1$ and $p = 2$, respectively. Notice that the inconsistency which is present in the inpainted sequence has been corrected and the result is temporally and spatially consistent. We also observe that in this case, the results of $p = 1$ and $p = 2$ are very similar.

Object insertion: As we have pointed out in Section 3, we pre-process the optical flow for detecting occlusions/disocclusions and the tolerance criterion used in the experiments is $tol = 50$. In this scenario, we wish to insert a two-dimensional object into a video sequence and affix it to a surface. Thanks to the DCD, we are able to coherently transport information present in one frame into subsequent (or previous) frames in the video sequence. Basically, we start inserting the object into a chosen frame that indicates the first appearance of the object in the video. Let us call this the first frame. Then, by solving Equation (7) with a small value of β (usually 0.01 for $p = 1$, and 0.0001 for $p = 2$) and

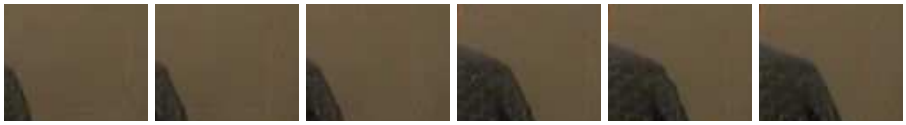


Fig. 9: Example of object removal. We remove the object only in the first frame (by inpainting) and then transport the information to subsequent frames by solving Equation (7) with $p = 1$. The result for $p = 2$ is similar. The complete sequence is available at the webpage mentioned at the beginning of Section 5.

setting the first frame as a Dirichlet temporal boundary condition, we are able to transport the first frame to the others. Since we are affixing an object on a surface, the inserted object inherits the optical flow from this surface. The occlusions are handled automatically by the temporal consistency using the preprocessed optical flow and it suffices to insert the object into the first frame. If we want to handle disocclusions, then we need also to insert the object into a later frame in the sequence and set it as another Dirichlet boundary condition. Let us call this the last frame. We note that in this setting we only need the information in the first and the last frames. For the intermediate frames we just have a hole that we fill-in with the new information.

Figure 8 shows an experiment where we insert a poster on a door replacing an existing one. The area where we insert the poster is being occluded and then disoccluded by a moving man. The total number of frames of the sequence is 31. The results shown in Figure 8 are obtained by solving Equation (7) with $p = \{1, 2\}$. As we can see, our method handles both occlusions and disocclusions. Notice that sometimes, on the boundary of the occluding object, we can see some small inconsistencies due to the inaccuracy of the optical flow. We will address this issue in a future work.

Object removal: This object insertion case can be easily adapted to object removal. We can remove an object from the first and last frames and then fill-in the hole in these frames either by inpainting or by copying a zero gradient field using Poisson editing, for instance. Then we can just transport the information from the temporal boundaries to all the domain in-between. Figure 9 shows an example.

6 Conclusions and future work

In this work we have presented the deblurring convective derivative, a novel scheme for computing the convective derivative that allows us to maintain information for a long period of time without decay. We have also integrated this derivative in a functional for video editing that imposes both spatial and temporal consistency. We applied this formulation to three video editing applications: object insertion, object removal and inpainting correction.

Our solution relies extensively on the quality of the optical flow and we have seen that inconsistencies in the optical flow affect considerably the quality of the results. Our future work will mainly focus on how to detect and handle inconsistent flows in order to improve the results.

Acknoledgements: We acknowledge support by MICINN project, reference MTM2009-08171, and by GRC reference 2009 SGR 773. VC also acknowledges partial support by IP project "2020 3D Media: Spatial Sound and Vision", financed by EC, and by "ICREA Acadèmia" prize for excellence in research funded both by the Generalitat de Catalunya. We acknowledge P. Arias and E. Meinhardt for their assistance and useful comments.

References

1. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. ACM Trans. Graph. **22** (2003) 313–318
2. Georgiev, T.: Image reconstruction invariant to relighting. In: Eurographics'05. (2005) 61–4
3. Finlayson, G.D., Hordley, S.D., Lu, C., Drew, M.S.: On the removal of shadows from images. IEEE Trans. on PAMI **28** (2006) 59–68
4. Fattal, R., Lischinski, D., Werman, M.: Gradient domain high dynamic range compression. ACM Trans. Graph. **21** (2002) 249–256
5. Arias, P., Facciolo, G., Caselles, V., Sapiro, G.: A variational framework for exemplar-based image inpainting. Int. J. Comput. Vision **93** (2011) 319–347
6. Komodakis, N., Tziritas, G.: Image completion using efficient belief propagation via priority scheduling and dynamic pruning. IEEE Trans. on IP **16** (2007) 2649–61
7. Kwatra, V., Essa, I., Bobick, A., Kwatra, N.: Texture optimization for example-based synthesis. ACM Trans. Graph. **24** (2005) 795–802
8. Sun, J., Jia, J., Tang, C.K., Shum, H.Y.: Poisson matting. ACM Trans. Graph. **23** (2004) 315–321
9. Agrawal, A., Raskar, R.: Gradient domain manipulation techniques in vision and graphics (2007) ICCV'07 short course.
10. Bai, X., Sapiro, G.: A geodesic framework for fast interactive image and video segmentation and matting. In: IEEE ICCV'07. (2007) 1–8
11. Grundmann, M., Kwatra, V., Han, M., Essa, I.: Efficient hierarchical graph-based video segmentation. In: IEEE CVPR'10. (2010) 2141–2148
12. Wang, H., Xu, X., Raskar, R., Ahuja, N.: Videoshop: A new framework for spatio-temporal video editing in gradient domain. In: IEEE CVPR'05. (2005) 1201
13. Bugeau, A., Gargallo, P., D'Hondt, O., Hervieu, A., Papadakis, N., Caselles, V.: Coherent Background Video Inpainting through Kalman Smoothing along Trajectories. In: Modeling, and Visualization Workshop. (2010) 8
14. Bhat, P., Zitnick, C.L., Cohen, M., Curless, B.: Gradientshop: A gradient-domain optimization framework for image and video filtering. ACM Trans. Graph. **29** (2010) 10:1–14
15. Bhat, P., Zitnick, C.L., Snavely, N., Agarwala, A., Agrawala, M., Curless, B., Cohen, M., Kang, S.B.: Using photographs to enhance videos of a static scene. In: Eurographics Symposium on Rendering. (2007) 327–338
16. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica **D** (1992) 259–268
17. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imaging Vis. **20** (2004) 89–97
18. Tao, M.W., Johnson, M.K., Paris, S.: Error-tolerant image compositing. In: European Conf. on Comput. Vision. (2010) 31–44
19. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: European Conf. on Comput. Vision. (2004)
20. Zhou, M.H., Mascagni, M., Qiao, A.Y.: Explicit Finite Difference Schemes for the Advection Equation. Relation **10** (1998) 7098