

Article

# Fluid Simulation with an $L_0$ Based Optical Flow Deformation

Kun Li, Na Qi \* and Qing Zhu

Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China;  
lkun@emails.bjut.edu.cn (K.L.); ccgszq@bjut.edu.cn (Q.Z.)

\* Correspondence: qina@bjut.edu.cn; Tel.: +86-1067396598

Received: 20 July 2020; Accepted: 9 September 2020; Published: 12 September 2020



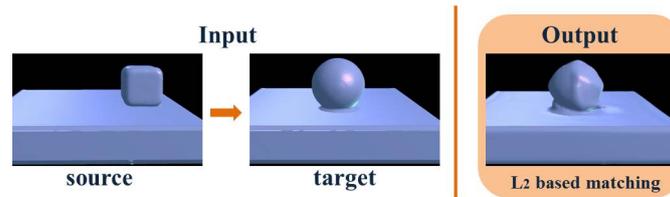
**Abstract:** Fluid simulation can be automatically interpolated by using data-driven fluid simulations based on a space-time deformation. In this paper, we propose a novel data-driven fluid simulation scheme with the  $L_0$  based optical flow deformation method by matching two fluid surfaces rather than the  $L_2$  regularization. The  $L_0$  gradient smooth regularization can result in prominent structure of the fluid in a sparsity-control manner, thus the misalignment of the deformation can be suppressed. We adopt the objective function using an alternating minimization with a half-quadratic splitting for solving the  $L_0$  based optical flow deformation model. Experiment results demonstrate that our proposed method can generate more realistic fluid surface with the optimal space-time deformation under the  $L_0$  gradient smooth constraint than the  $L_2$  one, and outperform the state-of-the-art methods in terms of both objective and subjective quality.

**Keywords:** fluid simulation; space-time deformation;  $L_0$  regularization; sparsity; data-driven

## 1. Introduction

Fluid simulations are increasingly important in computer graphics and have made an impact in real-time games. The simulation algorithms typically include physics-based methods and data-driven approaches. The physics-based fluid solvers [1–4] for working with the large amounts of simulation data are challenging, as the simulation data is typically just passed on to a rendering stage and any required change means re-starting a new simulation from scratch [5]. A category of data-driven fluid simulations [6] interpolates between two existing fluid animations based on a space-time deformation. Once the matches of two 4D shapes are computed, a user can freely choose any version in between the two extremes efficiently, without starting a new simulation.

Matching two fluid surfaces means finding the corresponding points between the source surface and the target surface, and this procedure needs to deform the source onto the target. To choose the suitable overall match, the deformation should be regularized with some constraints. A novel optical flow approach [5] is to register deforming space-time source surfaces and target surfaces for the robust registration and blend the complex volumetric phenomena by minimizing energy function with the  $L_2$  norm smoothness regularizer and the Tikhonov regularizer, which is inherently based on closest distances in four-dimensional Euclidean space. The  $L_2$  norm based smoothness regularization in [5] can be enforced on the gradient of the deformations to suppress the differences of the neighbors, however, this method can not always require an ideal deformation for the large differences of the source and target surfaces. An example of this is demonstrated in Figure 1, where the droplets of water with different shapes are released at different positions. As the errors of the deformation run into the registration procedure, the inaccurate matching caused by the sub-optimal deformation will create ambiguous corresponding points between the surface of the pool and that of the droplet, which brings out the artifacts in the interpolated results.



**Figure 1.** This image illustrates the blending result with the  $L_2$  based method [5]. While final output retrieves large scale motions, the artifacts between the droplet and surface are noticeable.

We propose an  $L_0$  based optical flow deformation optimization framework to reduce the undesirable artifacts by employing sparse smoothness constraint, and then to match the appropriate points of each shape. Because some large differences between source and target shapes have non-zero deformation gradient values while other regions keep zero, which indicates the sparse property, we present a new measure via  $L_0$  norm as smoothness regularization constraint. The sparse gradient counting measure can confine the number of deformation gradient changes, which allows large gradient magnitude by nature. In addition, to retrieve the motion optimally, this sparse property shows that different degrees of smoothness deformation priors should be introduced to different regions. As a consequence, our proposed  $L_0$  based method can retrieve an optimal deformation to yield better motion than the  $L_2$  based method [5] and avoid suffering from the error accumulation.

A preliminary version of this work appears in [7]. In this paper, we have made efforts to improve our paper in terms of the approach, theoretical analysis, and performance evaluation. We add more scenarios for evaluation and discuss the convergence, execution time, limitation of our proposed method. We demonstrate that our proposed  $L_0$  based optical flow deformation method can robustly and automatically match the surfaces of the fluid simulation data produced by semi-Lagrangian advection method than the  $L_2$  ones [5,6]. Our proposed method with the sparsity measure can outperform both the user guidance registration and the third data-point support in terms of accuracy with small details and moderate correspondences. Our proposed method can generate more realistic fluid surface with the optimal space-time deformation under the  $L_0$  gradient smooth constraint than the  $L_2$  one, and outperform the state-of-the-art methods in terms of both objective and subjective quality.

The rest of this paper is organized as follows. Section 2 provides a brief survey of related work. Section 3 introduces the background of the fluid flow and the optical flow solve. Section 4 presents the proposed fluid simulation method with the  $L_0$  based optical flow deformation. In Section 5, experiments are conducted for our proposed method in comparison with the state-of-the-art methods. The conclusion is given in Section 6.

## 2. Related Work

In this section, we briefly introduce the related work about the fluid simulation method and the motion reconstruction method with sparse constraints.

Fluid simulation has a long history in computer graphics research. One of the influential works in this area is published in Foster and Metaxas [4], and the stable advection approach introduced by [3] has made a significant progress of fluid simulation for visual effects. The FLIP method [8] which combines the particle and the grid data, and the purely particle based SPH approach [9] have been widely used in movie productions. Numerous extensions and improvements have been proposed, e.g., to introduce small-scale turbulent details [10], or to address the problem emerged in reality from fluid–air and fluid–solid interactions with the surface tension force and the adhesion force [11].

Several works have used guide behaviors to make liquid surfaces follow a desired shape or motion [12,13]. Unlike this modification approaches on existing fluid simulation, the work-flow that concentrates on align pre-computed space-time surfaces and interpolates between them for intermediate motions is mainly based on non-rigid ICP algorithms [6], and optical flow solve [5].

The non-rigid ICP algorithm has been applied to reconstruct the temporally coherent dynamic objects from data captured by the multi-view acquisition system [14] or the monocular depth

camera [15]. This registration technique has also been applied to the tracking of changing topology shapes such as liquid surfaces [16]. These method will run into the problem of local minima that is associated with non-rigid ICP. Raveendran et al. [6] roughly prevents the problem with user-guided correspondences.

Regular optical flow aims at recovering the motion of dynamic objects, especially for consecutive frames. Up to now, the Horn–Schunck algorithm [17] is extended to estimate high quality flow field. The connection between the optical flow and fluid flow are established for typical flow visualizations [18]. The optical flow techniques have also been used to acquire the fluid velocities based on physical priors [19]. Optical flow estimation commonly minimizes an energy function to obtain an optimal result, where the smoothness regularization is incorporated as a convex regularizer. The optical flow deformation approach [5] is essentially similar to the non-rigid iterative closest point algorithm that incorporates user guidance [6] in term of the registration process. Likewise, this optical flow deformation approach robustly recovers a match among the inputs based on local smoothness constraints and obtains a deformation field to align the surfaces. The  $L_2$  based smoothness regularization [5] may cause the error propagation and the accumulated errors. In this paper, we utilize a specific sparse regularization as a constraint to improve the results.

In fact, sparse representation [20] has successfully applied in field of image processing [21], computer vision [22], and graphics [15]. Especially, the sparsity priors have been utilized in the optical flow estimation and motion reconstruction. For example, a new sparsity prior of the flow field in certain domains is introduced to estimate dense flow fields, and a stronger additional sparsity constraint on the flow gradients is incorporated into the model to cope the measurement noises [22]. While [23] decomposes the motion field into sparse and non-sparse components to restrain the over-sparsity. In the motion tracking [15], the  $L_0$  based motion regularizer is incorporated into the non-rigid deformation to propose the  $L_0$ - $L_2$  non-rigid motion tracking method to efficiency stop the error propagation.

As we know, the points on the same surface part mostly share similar motion transformations, thus, the gradient field of the deformation exhibits the sparse property. The straightforward solution to exploit the sparsity is integrated the  $L_0$  norm regularization to the energy function. However, the optimization problem with the  $L_0$  norm minimization is non-convex problem, which can not be easily solved. In image smoothing problems [21], an  $L_0$  gradient minimization method is proposed to solve the similar issue, where an alternating optimization strategy with half-quadratic splitting is utilized. In this paper, we propose an  $L_0$  based optical flow deformation model and present an effective approach by adopting a similar alternating optimization strategy with half quadratic splitting. In addition, we conduct experiments on the fluid droplets and dam break scenarios to evaluate our proposed method.

### 3. Background

For brevity, we will give a brief overview of fluid flow, and then introduce the previous method of optical flow solve [5]. Throughout this paper we will use symbol (e.g.,  $u$ ) to denote continuous value, and bold symbol (e.g.,  $\mathbf{u}$ ) to denote the discrete counterpart. The symbol with subscript (e.g.,  $\phi_r$ ) is usually used to represent the derivative of the corresponding variable. Thus, a subscript will not denote the component of a vector except for the special statement.

**Fluid Flow.** The fluid flow is governed by incompressible Navier–Stokes equations [24], and a set of partial different equations that are supposed to hold throughout the fluid

$$\frac{\partial v}{\partial t} + v \cdot \nabla v = f - \frac{\nabla P}{\rho} + \gamma \nabla^2 v \quad \text{and} \quad \nabla \cdot v = 0 \quad (1)$$

where  $v$  is used in fluid for the velocity of the fluid,  $\frac{\partial v}{\partial t}$  is the commonly used material derivative w.r.t. time  $t$  for advection,  $f$  is the external body forces(to make the fluid behave in some desired way), the letter  $\rho$  stands for the fluid density,  $P$  is the force due to pressure that the fluid exerts on anything, and  $\gamma$  is the “kinematic viscosity”.

Following the description of [19], the observable concentration of a marker fluid is denoted with  $\phi$  and the advection of a scalar field  $\phi$  based on a velocity field  $v$  can be expressed as  $\phi_t + \nabla \cdot (v\phi) - \gamma \nabla^2 \phi = s$ , where  $s$  is a source term, including internal and external sources. Assume that the marker fluid and the ambient fluid have approximately the same mass density, the flow will perform a pure advection when the negligible diffusivity ( $\gamma = 0$ ), incompressibility ( $\nabla \cdot v = 0$ ), and  $s = 0$ , the standard form of advection is given by

$$\phi_t + v \cdot \nabla \phi = 0. \tag{2}$$

Equation (2) is equivalent to the brightness-constancy constraint of optical flow as introduced by Horn and Schunck algorithm [17].

**Optical Flow solve.** The work of [5] uses the  $L_2$  norm based optical flow solve to perform a registration step. In this setting, two animations from different simulation setups are taken as the inputs: a source and a target. Each animation is a sequence of surfaces over time, where the surfaces can be concatenated as time slices of a 4D volume. The registration mainly transforms source to further fit it into target through deformations  $u$ .

A flow simulations can be represented by 4D signed-distance values  $\phi$ , and the parameter  $r$  is an external control knob to modify the simulation from source to target. The regular optical flow usually recovers the motion of dynamic object from two images taken at different times. Thurey [5] instead uses optical flow to recover the motion among surfaces of different simulations. The algorithm employs the brightness-constancy assumption to recover the flow with motion discontinuities. Please refer to [5] for further details.

Applying the optical flow alone is underdetermined because of the non-linearity, so that various additional constraints or priors are imposed to regularize the problem. The Tikhonov regularizer term restricts the deformation to be as small as possible.  $E_{smooth}$  defines a  $L_2$  regularizer which smoothes the consistent motions along spatial as well as temporal dimensions. Given the optical flow constraint and some common regularizers, the estimation of deformation field  $u$  is achieved by minimizing the following energy

$$\min_u E_{data}(u) + \beta_T E_{Tikhonov}(u) + \beta_S E_{smooth}(u) \tag{3}$$

where  $\beta_S$  and  $\beta_T$  are the weights directly controlling the importance of each energy term. The energy to be minimized in a least-squares sense is given by

$$E_{total}(u) = \frac{1}{2} \int_{\Omega} |\phi_r + u \cdot \nabla \phi|^2 + \beta_T |u|^2 + \beta_S \sum_m |\nabla u_m|^2 d\Omega \tag{4}$$

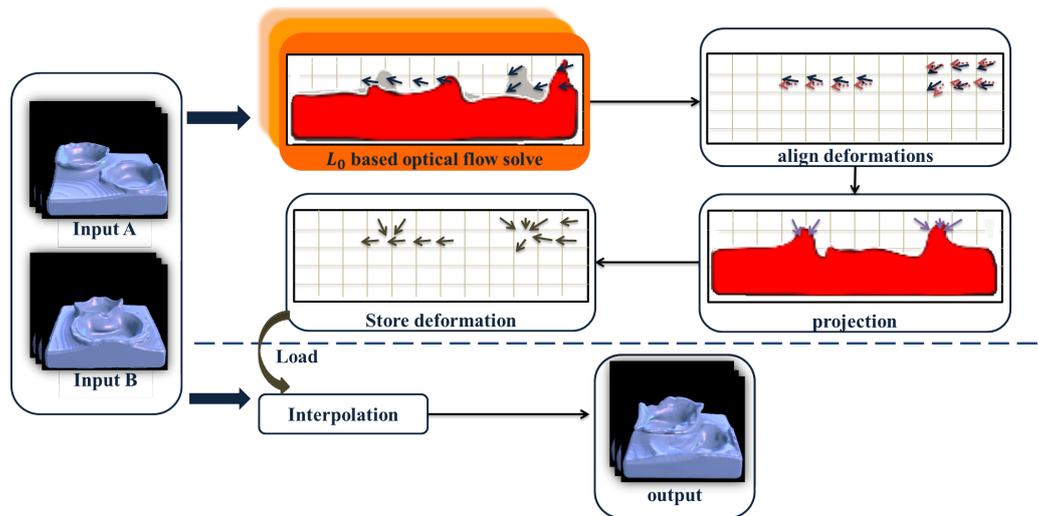
where the data derivative  $\phi_r$  corresponds to the time derivation  $\phi_t$  for optical flow applications. The deformation vector  $u$  of (4) is devoted to move a points to its corresponding position from the source to the target, and is calculated over four dimensions ( $m = 4$ ). The spatial gradient  $\nabla \phi$  is computed with the target surface.

The minimization of  $E_{total}$  is performed by an hierarchical scheme. The hierarchical procedure is used to recursively re-sample and deform the simulation data on coarser resolutions. The first deformation  $u^1$  calculated by optical flow deformation model transforms source inputs closer to target inputs. Then another solve is performed for the remaining difference, yielding the second deformation  $u^2$ . The deformation is continually refined with residual iterations until the refinement does not yield gains in quality anymore. We exactly follow [5] to align multiple consecutive Eulerian deformations so that they can be merged into a single, smooth linear deformation from the source to the target.

#### 4. Method

The 4D registrations and interpolation pipeline were processed automatically as illustrated in Figure 2. The optical flow solve computed a dense field of four-component deformation vectors  $\mathbf{u}$ ,

which mapped points of input  $A$  onto input  $B$ . The residual optical flow was performed until the refinement did not yield gains in quality. Then the deformation sequences from each iteration were aligned into a single deformation field. It was observed that the source surface should ideally deform to end up on the target surface. Based on this goal, the projection step [16] was adopted to retrieve the fine details at the surface that prevented by regularization. It is notable that the projection only gave good results when a suitable overall match was obtained by optical flow step. After deformations were calculated, the interpolation step [5] was further utilized to produce intermediate surface animations. Given two input sequences  $\psi_1, \psi_2$  with deformations  $u_{1 \rightarrow 2}$  and  $u_{2 \rightarrow 1}$ , the linear interpolation could significantly generate the final quality.



**Figure 2.** The pipeline of the registration and interpolation scheme. The inputs animation show two crown splashes. Our proposed  $L_0$  based optimization is applied into the optical flow solve.

#### 4.1. $L_0$ Based Optical Flow Deformation

The goal of optical flow in this work is to compute deformations and to recover the discontinuous fluid motions. In the following, a sparsity constraint on the deformation gradients is incorporated into the optical flow estimation model to reduce the undesirable errors.

The  $L_2$  based optical flow solve [5] for the certain inputs produced by semi-Lagrangian advection algorithm cannot robustly and accurately match surfaces in the nearest-neighbor fashion. Especially in the inputs with little similarities because of the surface flickering, the deformations will try to match pieces in proximity but not necessarily the whole target shape. We propose an  $L_0$  based deformation regularizer over the existing optical flow solve to implicitly define the appropriate correspondences and to prevent motion changes on other regions.

Specifically, the deformation is denoted by  $\mathbf{u}$  to transform the source surface closer into the target surface. We formulate the  $L_0$  based deformation by minimizing the following energy function

$$\min_{\mathbf{u}} E_{data}(\mathbf{u}) + \beta_T E_{Tikhonov}(\mathbf{u}) + \beta_S E'_{smooth}(\mathbf{u}) \tag{5}$$

where a new smoothness regularizer  $E'_{smooth}$  is introduced to the general optical flow solve to replace the  $L_2$  norm smoothness term.  $E'_{smooth}$  measures the deformation differences between all pairs of neighboring points, and is given by

$$E'_{smooth} = S(\mathbf{u}) \tag{6}$$

$$S(\mathbf{u}) = \# \{ p \mid |\partial_x \mathbf{u}^p| + |\partial_y \mathbf{u}^p| + |\partial_z \mathbf{u}^p| + |\partial_t \mathbf{u}^p| \neq 0 \} \tag{7}$$

where  $S(\mathbf{u})$  is the  $L_0$  norm for the deformation gradient  $\nabla \mathbf{u}$ . The gradient  $\nabla \mathbf{u}^p = \{ \partial_x \mathbf{u}^p, \partial_y \mathbf{u}^p, \partial_z \mathbf{u}^p, \partial_t \mathbf{u}^p \}$  for each point  $p$  of the deformation field is calculated based on the two

4D SDFs between neighboring points in space and time. The gradient magnitude  $|\partial \mathbf{u}^p|$  is defined as the sum of gradient magnitudes over four dimensions. Our smoothness constraint with  $L_0$  norm regularization is used to control the number of non-zero gradients globally and present the sparsity of deformation gradients.

We estimated the optimized deformation by minimizing the energy function

$$\min_{\mathbf{u}} \left\{ \sum_p |\boldsymbol{\psi}_r^p + \nabla \boldsymbol{\psi}^p \cdot \mathbf{u}^p|^2 + \beta_T |\mathbf{u}^p|^2 + \beta_S S(\mathbf{u}) \right\} \tag{8}$$

The  $E'_{smooth}$  term of  $L_0$  norm regularizer in (5) is a non-convex problem. Due to the computationally intractability of the  $L_0$  norm, the half-quadratic splitting strategy utilized in image smoothing achieves a high qualitative effect [21]. Inspired by the strategy described in [21], we split the objective function (8) into two sub-problems by introducing auxiliary variables into the energy function. We argue that this method found appropriate matches and retrieved the surface motions robustly.

#### 4.2. Solver

We introduced an auxiliary variable  $\mathbf{v}^p$ , corresponding to  $\nabla \mathbf{u}^p$ , and reformulated the energy function as

$$\min_{\mathbf{u}, \mathbf{v}} \left\{ \sum_p |\boldsymbol{\psi}_r^p + \nabla \boldsymbol{\psi}^p \cdot \mathbf{u}^p|^2 + \beta_T |\mathbf{u}^p|^2 + \beta_S S(\mathbf{v}) + \beta_A |\nabla \mathbf{u}^p - \mathbf{v}^p|^2 \right\} \tag{9}$$

where  $S(\mathbf{v}) = \# \{ p \mid |\partial_x \mathbf{v}^p| + |\partial_y \mathbf{v}^p| + |\partial_z \mathbf{v}^p| + |\partial_t \mathbf{v}^p| \neq 0 \}$ , and  $\beta_A$  is an automatically adapting parameter to control the similarity between the gradients  $\nabla \mathbf{u}^p$  and their corresponding variables  $\mathbf{v}^p$ . To solve this problem, we alternatively fix variables  $\mathbf{u}^p$  to solve  $\mathbf{v}^p$  and fix variables  $\mathbf{v}^p$  to solve  $\mathbf{u}^p$  in each iteration.

##### 4.2.1. Minimizing $\mathbf{v}$

The terms not involving  $\mathbf{v}^p$  in (9) are removed to estimate  $\mathbf{v}^p$

$$\min_{\mathbf{v}} \left\{ \sum_p |\nabla \mathbf{u}^p - \mathbf{v}^p|^2 + \frac{\beta_S}{\beta_A} S(\mathbf{v}) \right\} \tag{10}$$

As  $\nabla \mathbf{u}^p$  is pre-fixed, (10) has a close form solution

$$\mathbf{v}^p = \begin{cases} (0, 0, 0, 0), & (\partial_x \mathbf{u}^p)^2 + (\partial_y \mathbf{u}^p)^2 + (\partial_z \mathbf{u}^p)^2 + (\partial_t \mathbf{u}^p)^2 \leq \frac{\beta_S}{\beta_A} \\ (\partial_x \mathbf{u}^p, \partial_y \mathbf{u}^p, \partial_z \mathbf{u}^p, \partial_t \mathbf{u}^p), & otherwise \end{cases} \tag{11}$$

##### 4.2.2. Minimizing $\mathbf{u}$

As  $\mathbf{v}^p$  is pre-fixed, the  $\mathbf{u}^p$  estimation can be represented in a quadratic function

$$\min_{\mathbf{u}} E(\mathbf{u}) = \min_{\mathbf{u}} \left\{ \sum_p |\boldsymbol{\psi}_r^p + \nabla \boldsymbol{\psi}^p \cdot \mathbf{u}^p|^2 + \beta_T |\mathbf{u}^p|^2 + \beta_A |\nabla \mathbf{u}^p - \mathbf{v}^p|^2 \right\} \tag{12}$$

(12) formulates a  $L_2$  based minimization. The solution is similar to the method of  $L_2$  based optical flow [5]. Setting the gradient of  $E(\mathbf{u})$  to be zero gives a closed solution for (12), which can be expressed as a linear equation

$$([\nabla \boldsymbol{\psi}]^T [\nabla \boldsymbol{\psi}] + \beta_T \mathbf{E} + \beta_A \mathbf{L}) \cdot \mathbf{u} = \beta_A \mathbf{v} - [\nabla \boldsymbol{\psi}]^T \boldsymbol{\psi}_r \tag{13}$$

The  $\psi_r$  was computed with  $\psi_1 - \psi_2$ , where the source and target inputs are denoted by  $\psi_1$  and  $\psi_2$  respectively. The spatial gradient  $\nabla\psi$  is denoted with the target surface.  $L$  denotes the discrete Laplacian.

The optical flow based on  $L_0$  regularizer is summarized in Algorithm 1. Initially, the algorithm was performed with two SDF inputs  $\psi$  and a zero deformation field  $\mathbf{u} = 0$ . The function  $deform(\psi, \mathbf{u}, \alpha)$  applied the deformation  $\mathbf{u}$  to input  $\psi$  with weight  $\alpha \in [0..1]$ . A single, smooth linear deformation from source to target was required rather than deformation sequences produced by the iteration runtime. We followed [5] to align multiple deformations with the  $alignVelocity(\mathbf{u})$  function.

The reliable error metric [5] was adopted to quantitatively evaluate whether the current optical flow solve led to an improvement. More emphasis was put on the surface region rather than the volume itself. Given two inputs  $\psi_1$  and  $\psi_2$  (which we define them in a discrete setting), the error metric becomes

$$error(\psi_1, \psi_2) = \sum_x h(\psi_1(x), \psi_2(x))V \tag{14}$$

where  $V$  is the volume of a cell. The indicator function  $h$  detecting misalignment surfaces is defined as

$$h(s_1, s_2) = \begin{cases} 0, & \text{if } sgn(s_1) = sgn(s_2) \\ \min(1, \frac{|s_1 - s_2|}{\Delta x}), & \text{otherwise} \end{cases} \tag{15}$$

where  $s_1$  and  $s_2$  represent two input SDFs  $\psi_1$  and  $\psi_2$ .

---

**Algorithm 1** Optical flow based on  $L_0$  regularizer

---

**Input:** inputs  $\psi_1$  and  $\psi_2$ ,  $\mathbf{u}$ , smoothing weight  $\beta_S$ , parameters  
1  $\beta_T, \beta_A, \alpha, \kappa$   
**Output:** Optimized  $\mathbf{u}$   
2 **Initialization :**  $\beta_A \leftarrow 2\beta_S, \mathbf{u}^{(0)} \leftarrow 0, \mathbf{v}^{(0)} \leftarrow \nabla\mathbf{u}^{(0)}$   
3 **for**  $i = 1 : max$  **do**  
4      $\psi'_1 = deform(\psi_1, \mathbf{u}^{(i-1)}, \alpha);$   
5      $\mathbf{v}^{(i)} \leftarrow \min_v(\beta_S, \beta_A, \mathbf{u}^{(i-1)})$   
6      $\mathbf{u}^{tmp(i)} \leftarrow \min_u(\psi'_1, \psi_2, \mathbf{v}^{(i)}, \beta_T, \beta_A)$   
7     **begin**  
8          $\mathbf{u}^{tmp} = \mathbf{u}^{(i-1)} + alignVelocity(\mathbf{u}^{tmp(i)});$   
9         **if**  $error(deform(\psi_1, \mathbf{u}^{(i)}, \alpha), \psi_2) \geq error(deform(\psi_1, \mathbf{u}^{tmp}, \alpha), \psi_2)$  **then**  
10              $\mathbf{u}^{(i)} = \mathbf{u}^{tmp};$   
11              $\beta_A = \kappa \cdot \beta_A$   
12         **else**  
13             **break;**  
14         **end**  
15     **end**  
16 **end**

---

In our proposed method, we also utilized the commonly used hierarchical procedure to recursively re-sample and deform the data on coarser resolutions. The deformation was continually refined with residual iterations until the refinement did not yield gains in quality anymore, e.g., the ratio of the error of two consecutive iterations was greater than a given threshold for each resolution.

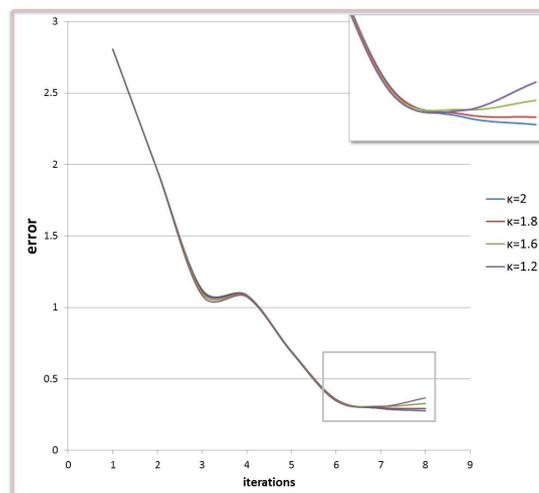
### 5. Experimental Results

We conducted the experiments on the fluid simulation of both the flying droplets and dam break scenarios. The standard fluid simulation, surface matching process and the interpolation step were

run on the extensible fluid simulation framework, called mantaflow [25]. The fluid surface sequences of both the source and target inputs were generated by running the standard fluid simulation. When the deformations were calculated, new fluid animations could be produced in the interpolation weight range from 0 to 1. We utilized our proposed  $L_0$  based fluid simulation method and the state-of-the-art method [5] to calculate the deformation and further generate the interpolation results as the outputs, respectively. The interpolation step yielded a smooth transition between source inputs and target inputs, and generated new outputs along a range of input positions. In the following, we compare our  $L_0$  based refinement results with those of the state-of-the-art [5] to demonstrate the accuracy of our proposed method.

### 5.1. Convergence Analysis

We took the flying droplets as an example to study the convergence property of our proposed  $L_0$  based optical flow solve method in Figure 3. The error metric *error* in [5] was computed between the target SDFs and the deformed one at different iterations. We observe that the errors descended with the increase of iterations so that our proposed algorithm was convergent. In order to speed up the convergence, we monotonically increased  $\beta_A$  by multiplying the parameter  $\kappa = 2$ . In addition, Figure 3 shows the influence of the error curves with different parameters  $\kappa$ . We let the parameter  $\kappa$  varying from 1.2 to 2. From the convergence curve, we can observe that the best performance was generally achieved at  $\kappa = 2$ . Thus, in our experiments with a resolution of  $64^4$ , we set  $\beta_A$  to be  $1E - 3$  in the first iteration and increase it by multiplying parameter  $\kappa = 2$  after each iteration when error was reduced.



**Figure 3.** The convergence analysis of flying droplets with variable parameter  $\kappa$ . X label means iteration number, Y label means the error metric *error*.

### 5.2. Deformation Results

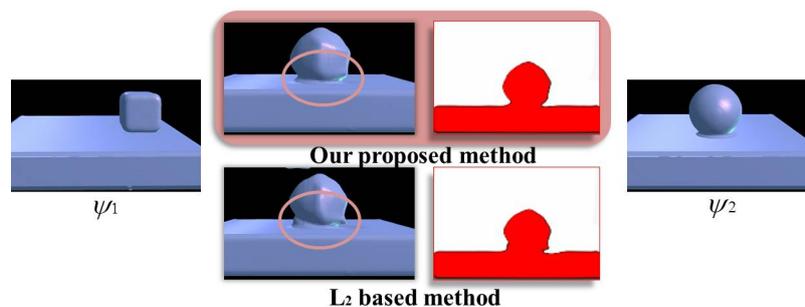
We evaluate the quality of the deformation through the visual quality of the interpolated results, where the uni-directional nearest-neighbor interpolation is used, such as the deformation direction from  $\psi_1$  to  $\psi_2$ .

We consider two flying droplets with different initial shapes and different positions. They drop in the pool at different times to create different water ripples. We try to deform the source one to the target one by our proposed fluid simulation method and state-of-the-art method [5] respectively, with the interpolation weight as 1. Figure 4 shows the interpolated results of the  $L_2$  based and the  $L_0$  based deformation methods respectively. In order to easily observe, the sectional view images at a fixed frame are shown along with the interpolated results. We can observe that our results are closer to the target input  $\psi_2$  and with less artifacts than those of the  $L_2$  based method. In addition,

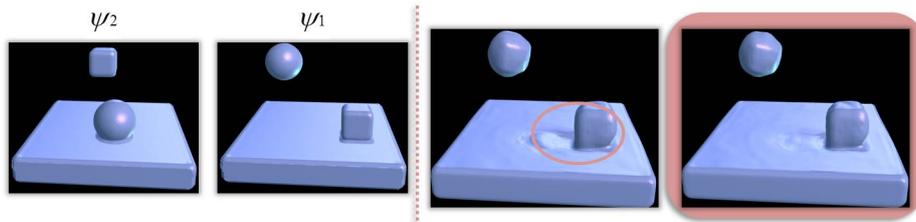
the interpolated results of these two different methods with the deformation from the input  $\psi_2$  to the input  $\psi_1$  are shown in Figure 5. The  $L_2$  based deformation method may create some ambiguous correspondences between the input  $\psi_1$  to the input  $\psi_2$ , and the undesired ripples appear at the surface due to the suboptimal matching. Our proposed method can alleviate the artificial motions brought by the  $L_2$  based method effectively.

Three examples of interpolated results at different frames for flying droplets are demonstrated in Figure 6, where the droplets were falling into the pool and causing the varying splashes. Figure 7 demonstrates the interpolation results of the droplets falling into the pool and the varying splashes. We observe that our interpolated results could preserve the large-scale structures and recover the small-scale details. Our  $L_0$  based method could reduce the artificial motions brought by the  $L_2$  based method, which demonstrates that our  $L_0$  based method could calculate moderate correspondences and a good deformation field and could further reduce the undesired errors of complex surfaces of the liquid.

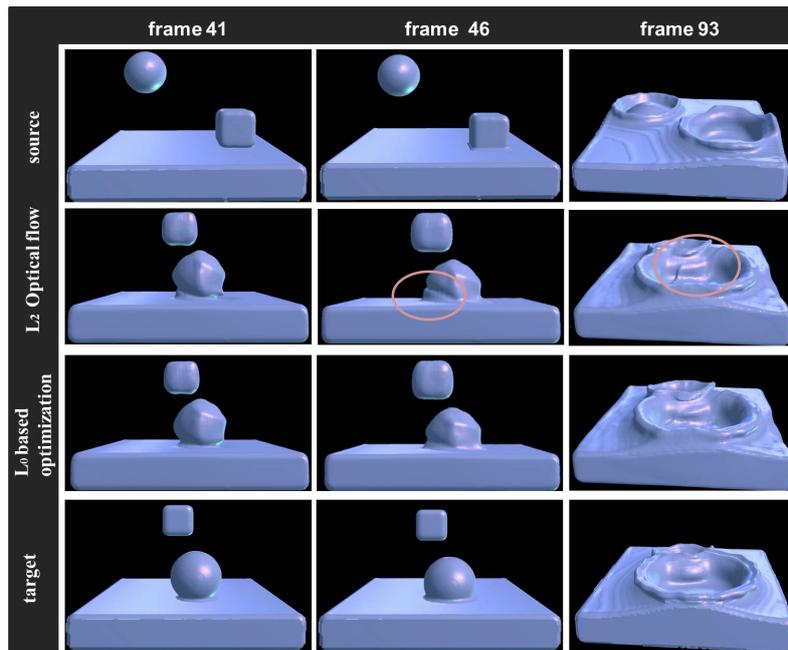
We take the dam break as an example to demonstrate the accuracy of our optimization method. In this set of animations, a block of water was released at one side of a pool, and this formed a wave that swept the pool. As shown in Figure 8, we produced fluid animations for two different widths for this block of water. The interpolated results were generated with the interpolation weight as 1 to compare the generated surfaces with the target surfaces visually. We compared the results of our proposed  $L_0$  based method with those of [5]. As shown in the right columns of Figure 8, the  $L_2$  based method caused the undesired ripples at the surface, and the  $L_0$  optimization method captured the accurate motions of the waves. Therefore, the surface of our results based on the  $L_0$  regularization was closer to the target one than that based on the  $L_2$  optimization. Figure 9 shows the whole range of the interpolation results of dam break deformed from  $\psi_1$  to  $\psi_2$  with different interpolation weights. The inputs had inconsistent behaviors in this static frame. Compared with the  $L_2$  based regularizer which restricted the deformation locally across the surface region, the  $L_0$  based regularizer effectively recovered the small-scale details. The animation simulation of all cases by using the  $L_2$  based method [5] and our  $L_0$  methods can be found in the supplemental video. The accompanying video shows a direct comparison to illustrate the difference between the results of [5] and our method.



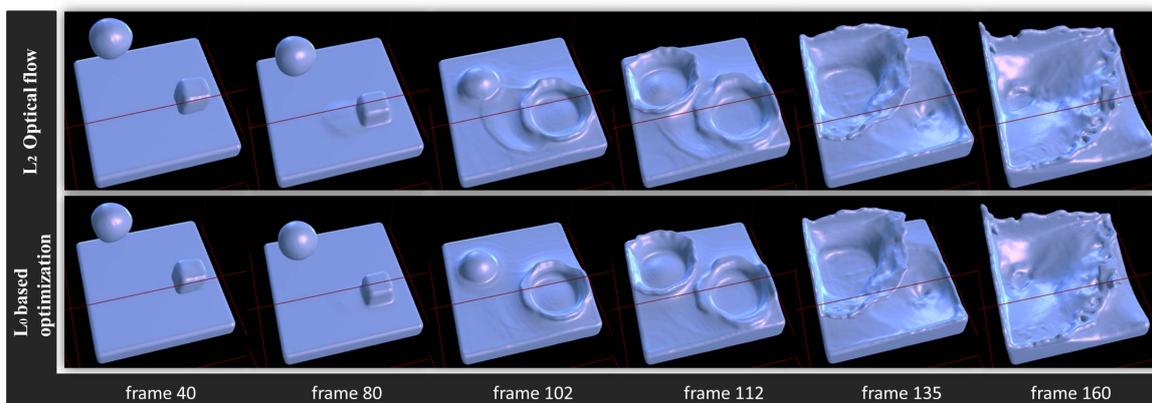
**Figure 4.** The visual quality comparison of flying droplets via different deformation methods based on  $L_2$  and  $L_0$  smooth regularization. Left: the source input  $\psi_1$ . Middle: the results of  $L_2$  based method [5] and  $L_0$  based method, and the sectional view images correspond to the results. Right: the target input  $\psi_2$ .



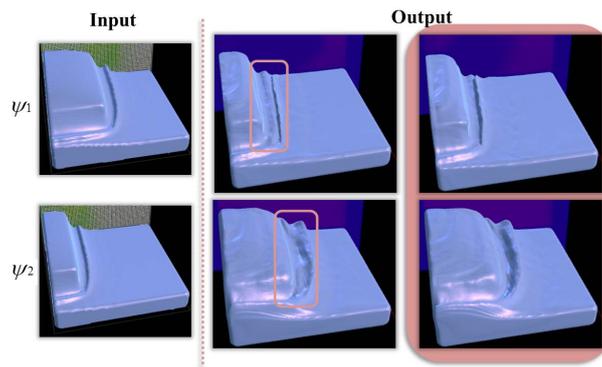
**Figure 5.** The results of flying droplets with the deformation from source to the target. Left: the source input  $\psi_2$  and the target inputs  $\psi_1$  are as the input. Right: the interpolated results of [5] and our proposed  $L_0$  based deformation method, respectively.



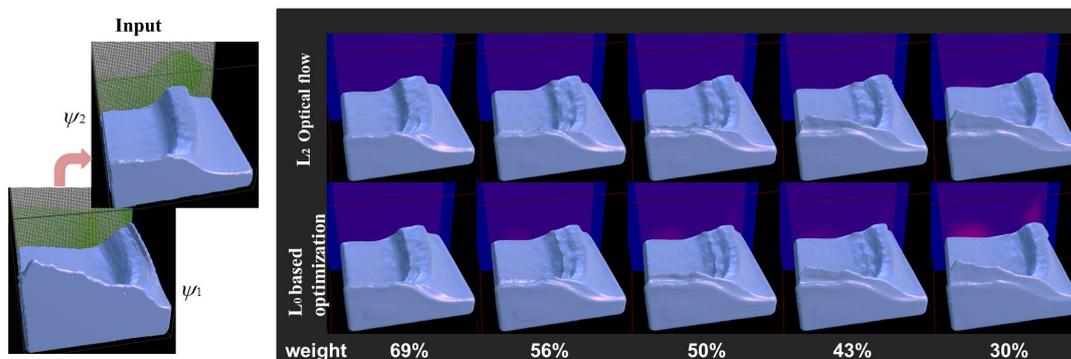
**Figure 6.** A visual comparison between the  $L_2$  based optical flow deformation, and our proposed  $L_0$  based optimization method. From top to bottom: the source fluid, the interpolated fluid results with deformation solved by the  $L_2$  based method [5] as well as our  $L_0$  based optimization method, and the target fluid respectively. The figure shows two drops falling into a pool, which causes the crown splash.



**Figure 7.** Some static frames of droplets falling into the pool. The target motions of the fluid animation are shown as the input  $\psi_1$ . The top and bottom rows show the results of [5] and our  $L_0$  based method, respectively.



**Figure 8.** A comparison of the  $L_2$  based optical flow against our  $L_0$  based optimization output for the dam break scenario. Left: the input  $\psi_1$  and  $\psi_2$ . Middle: the interpolated results generated by using the  $L_2$  based method with two different directional deformations including  $\psi_1 \rightarrow \psi_2$  and  $\psi_2 \rightarrow \psi_1$  with the interpolation weight 1, respectively. Right: the corresponding interpolated results by using the  $L_0$  based method.



**Figure 9.** The range of interpolation for the dambreak setup of inputs at a fixed point in time. The numbers on bottom of each image indicate the interpolation weights. Direction of the deformation is employed from  $\psi_1$  to  $\psi_2$ .

### 5.3. Evaluation

The error metric mentioned in Section 4.2 could quantitatively and significantly detect the differences between the deformed surface and the target surface. We utilized the error metric (14) to calculate the errors between the target surfaces and the generated surfaces which were deformed with the  $L_2$  based and the  $L_0$  based deformation methods respectively. The error values can be found in Table 1, which includes two aforementioned scenarios (droplet, dam break). We show the results with the deformation direction from  $\psi_1$  to  $\psi_2$  and the inverse one. We can observe that our proposed  $L_0$  based fluid simulation method could obtain less errors than the  $L_2$  based method [5], and generate the desired correspondences.

**Table 1.** Errors between the deformed surface and the target surface.

Fluid Example	Directions	Initial Error	The Error after Deformation	
			$L_2$	$L_0$
Flying	$\psi_1 \rightarrow \psi_2$	4.1207	1.1113	0.9440
Droplets	$\psi_2 \rightarrow \psi_1$	3.9823	1.9747	1.8369
Dam Break	$\psi_1 \rightarrow \psi_2$	4.0991	0.6768	0.6082
	$\psi_2 \rightarrow \psi_1$	5.0944	1.5366	1.3622

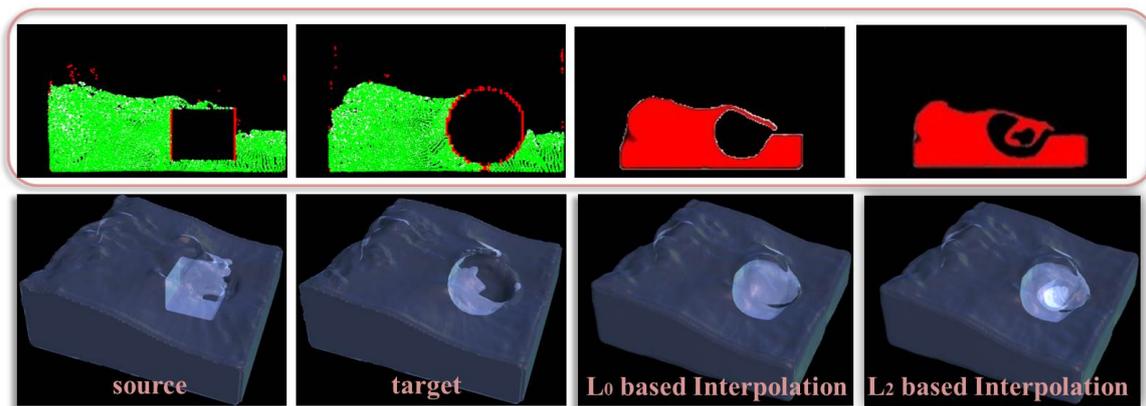
Our experiments used data-sets with a resolution of  $64 \times 64 \times 64 \times 64$ . We utilized the commonly used hierarchical procedure to recursively re-sample and deform the data on coarser resolutions, including the resolutions of  $16 \times 16 \times 16 \times 16$ ,  $32 \times 32 \times 32 \times 32$ , and  $64 \times 64 \times 64 \times 64$ , denoted by  $16^4$ ,  $32^4$ , and  $64^4$ , respectively. The changes of resolution were very useful to reduce the inaccuracies of the optical flow solve. Table 2 lists the error between the deformed surface and the target surface, the factor (the ratio of the error of this iteration to that of last iteration), and the time to calculate the deformation during each iteration. In our experiment, we stopped the iterations when the factors for the resolution of  $16^4$ ,  $32^4$ , and  $64^4$  were greater than 70%, 95%, and 95%, respectively. Particularly for the resolution of  $16^4$ , the iterations stopping criterion should exceed three times. Note that the deformation was not refined if the gains in quality could not be yielded anymore. Thus, we did not use the deformation obtained by the last iteration. From the Table 2, we can see the execution time of the deformation by our  $L_0$  based optical flow method with the half quadratic splitting approach was less than that of the  $L_2$  based optical flow method.

**Table 2.** Errors and timings for the registration step performed with two constraint methods.

Resolution	Iteration	$L_0$ Based			$L_2$ Based		
		Error	Factor	Time	Error	Factor	Time
$16 \times 16 \times 16 \times 16$	0	4.1207	69.37%	0.073 s	4.1207	69.37%	0.076 s
	1	3.0787	74.71%	0.084 s	3.1301	75.96%	0.077 s
	2	1.9302	62.69%	0.067 s	2.0283	64.8%	0.073 s
	3	1.3648	70.71%	0.068 s	1.5698	77.39%	0.097 s
$32 \times 32 \times 32 \times 32$	0	1.9837	62.98%	3.793 s	1.8924	68.36%	3.78 s
	1	1.3720	69.16%	3.436 s	1.4362	75.89%	4.434 s
	2	0.8306	60.54%	2.829 s	0.8947	62.29%	4.486 s
	3	0.6313	76.01%	2.979 s	0.7171	70.15%	4.484 s
	4	0.5498	87.09%	3.199 s	0.6488	90.47%	4.257 s
$64 \times 64 \times 64 \times 64$	0	0.9440	69.54%	81.41 s	1.1113	77.72%	87.496 s
	1	1.0097	106.9%	75.616 s	1.1269	105.7%	91.247 s
total time		-	-	176.79 s	-	-	204.838 s

#### 5.4. Limitation

We take an example to illustrate in which case the interpolation method yielded undesirable results. Figure 10 shows the interpolation results of the water hitting different obstacles. The obstacle in the pool led to different liquid motions for the splash near the obstacle. The surface of both the  $L_0$  and the  $L_2$  based interpolation results demonstrate that part of the water was pushed into the obstacle. Even though the  $L_0$  based result outperformed the  $L_2$  based result, both them were undesirable results. Because the large differences of the source and the target surface had difficulty finding the corresponding points between the source and target surface, the inaccuracy deformation calculation led to incorrect matching and further resulted in undesirable results and artifacts. Through our experiments, we observed that either the  $L_0$  or the  $L_2$  based interpolation method failed when the fluid had relatively complex structures and large differences between the source and the target. Therefore, there is still room to improve the fluid simulation approach.



**Figure 10.** A failure case to yield undesirable interpolation results. Two inputs of water wave are released to hit obstacles with different shapes.

## 6. Conclusions

In this paper, we proposed a  $L_0$ -norm based deformation for fluid simulation to obtain a good alignment of two inputs. The main contribution of our work is a  $L_0$ -norm optimization approach to align fluid simulation with changing topologies. The  $L_0$  based optical flow takes advantage of the sparsity property of the fluid motion of neighboring frames to constrain the deformation. Moreover, we propose a novel effective approach to solve the  $L_0$ -based optimization problem by making use of  $L_0$  gradient minimization, which can globally control the non-zero gradients to approximate the deformation in a sparsity-control manner. Experimental results demonstrate that our method outperforms the state-of-the-art  $L_2$  based optical flow [5] and accurately recovers motions between the space-time surfaces of different simulations.

There are two different applications of the interpolation method. The time step of the input data influences the continuity of the animations visually. We can make use of our proposed fluid simulation scheme to produce a highly continuous behavior for arbitrary neighboring sequences with a similar topology. Alternatively, we can utilize different scenarios to interpolate new fluid animations frame-by-frame. Our proposed approach is a data-driven optical flow solve, however, the underlying physics is ignored. In the future, we will incorporate with other physical optimization of the surface motions, such as the constraints for mass conservation.

**Author Contributions:** conceptualization, N.Q.; methodology, K.L. and N.Q.; software, K.L.; validation, N.Q.; formal analysis, K.L.; investigation, K.L. and N.Q.; resources, Q.Z.; data curation, N.Q.; writing—original draft preparation, K.L. and N.Q.; writing—review and editing, K.L. and N.Q.; visualization, N.Q.; supervision, N.Q. and Q.Z.; project administration, Q.Z.; funding acquisition, N.Q. and Q.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partially supported by the National Natural Science Foundation of China under Grants (61906009 and 61976011), and the scientific Research Common Program of Beijing Municipal Commission of Education (KM202010005018).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Müller, M.; Charypar, D.; Gross, M. Particle-based Fluid Simulation for Interactive Applications. In Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation SCA '03, San Diego, CA, USA, 26–27 July 2003; Eurographics Association: Aire-la-Ville, Switzerland, 2003; pp. 154–159.
2. Solenthaler, B.; Pajarola, R. Predictive-corrective Incompressible SPH. In *ACM SIGGRAPH 2009 Papers*; SIGGRAPH '09; ACM: New York, NY, USA, 2009; pp. 40:1–40:6. [[CrossRef](#)]

3. Stam, J. Stable Fluids. In Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques SIGGRAPH '99, Los Angeles, CA, USA, 26–30 July 1999; ACM Press/Addison-Wesley Publishing Co.: New York, NY, USA, 1999; pp. 121–128. [[CrossRef](#)]
4. Foster, N.; Metaxas, D. Realistic Animation of Liquids. In Proceedings of the Conference on Graphics Interface GI '96, Toronto, ON, Canada, 22–24 May 1996; Canadian Information Processing Society: Toronto, ON, Canada, 1996; pp. 204–212.
5. Thuerey, N. Interpolations of Smoke and Liquid Simulations. *ACM Trans. Graph.* **2016**, *36*, 3:1–3:16. [[CrossRef](#)]
6. Raveendran, K.; Wojtan, C.; Thuerey, N.; Turk, G. Blending Liquids. *ACM Trans. Graph.* **2014**, *33*, 137:1–137:10. [[CrossRef](#)]
7. Li, K.; Qi, N.; Zhu, Q. Fluid Simulation with a Dense Space-time Deformation via L0 Gradient Minimization. In Proceedings of the 10th International Conference on Frontier Computing, FC2020, Singapore, 10–13 July 2020 .
8. Zhu, Y.; Bridson, R. Animating Sand As a Fluid. In *ACM SIGGRAPH 2005 Papers; SIGGRAPH '05*; ACM: New York, NY, USA, 2005; pp. 965–972. [[CrossRef](#)]
9. Solenthaler, B.; Pajarola, R. Predictive-corrective Incompressible SPH. *ACM Trans. Graph.* **2009**, *28*, 40:1–40:6. [[CrossRef](#)]
10. Kim, T.; Thürey, N.; James, D.; Gross, M. Wavelet Turbulence for Fluid Simulation. In *ACM SIGGRAPH 2008 Papers; SIGGRAPH '08*; ACM: New York, NY, USA, 2008; pp. 50:1–50:6. [[CrossRef](#)]
11. Akinci, N.; Akinci, G.; Teschner, M. Versatile Surface Tension and Adhesion for SPH Fluids. *ACM Trans. Graph.* **2013**, *32*, 182:1–182:8. [[CrossRef](#)]
12. Raveendran, K.; Thuerey, N.; Wojtan, C.; Turk, G. Controlling Liquids Using Meshes. In Proceedings of the 11th ACM SIGGRAPH/Eurographics Conference on Computer Animation EUROSCA'12, Los Angeles, CA, USA, 29–31 July 2012; Eurographics Association: Aire-la-Ville, Switzerland, 2012; pp. 255–264. [[CrossRef](#)]
13. Pan, Z.; Huang, J.; Tong, Y.; Zheng, C.; Bao, H. Interactive Localized Liquid Motion Editing. *ACM Trans. Graph.* **2013**, *32*, 184:1–184:10. [[CrossRef](#)]
14. Li, H.; Luo, L.; Vlastic, D.; Peers, P.; Popović, J.; Pauly, M.; Rusinkiewicz, S. Temporally Coherent Completion of Dynamic Shapes. *ACM Trans. Graph.* **2012**, *31*, 2:1–2:11. [[CrossRef](#)]
15. Guo, K.; Xu, F.; Wang, Y.; Liu, Y.; Dai, Q. Robust Non-rigid Motion Tracking and Surface Reconstruction Using L0 Regularization. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) ICCV '15, Santiago, Chile, 7–13 December 2015; IEEE Computer Society: Washington, DC, USA, 2015; pp. 3083–3091. [[CrossRef](#)]
16. Bojsen-Hansen, M.; Li, H.; Wojtan, C. Tracking Surfaces with Evolving Topology. *ACM Trans. Graph.* **2012**, *31*, 53:1–53:10. [[CrossRef](#)]
17. Horn, B.K.P.; Schunck, B.G. Determining Optical Flow. *Artif. Intell.* **1981**, *17*, 185–203. [[CrossRef](#)]
18. Liu, T.; Shen, L. Fluid flow and optical flow. *J. Fluid Mech.* **2008**, *614*, 253–291. [[CrossRef](#)]
19. Gregson, J.; Ihrke, I.; Thuerey, N.; Heidrich, W. From Capture to Simulation: Connecting Forward and Inverse Problems in Fluids. *ACM Trans. Graph.* **2014**, *33*, 139:1–139:11. [[CrossRef](#)]
20. Zhang, Z.; Xu, Y.; Yang, J.; Li, X.; Zhang, D. A survey of sparse representation: Algorithms and applications. *IEEE Access* **2015**, *3*, 490–530. [[CrossRef](#)]
21. Xu, L.; Lu, C.; Xu, Y.; Jia, J. Image Smoothing via L0 Gradient Minimization. In Proceedings of the 2011 SIGGRAPH Asia Conference SA '11, Hong Kong, China, 12–15 December 2011; ACM: New York, NY, USA, 2011; pp. 174:1–174:12. [[CrossRef](#)]
22. Shen, X.; Wu, Y. Sparsity model for robust optical flow estimation at motion discontinuities. In Proceedings of the Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13–18 June 2010; pp. 2456–2463. [[CrossRef](#)]
23. Chen, Z. Decomposing and Regularizing Sparse/Non-sparse Components for Motion Field Estimation. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) CVPR '12, Providence, RI, USA, 16–21 June 2012; IEEE Computer Society: Washington, DC, USA, 2012; pp. 1776–1783.

24. Session Details: Course 31: Fluid Simulation. In *ACM SIGGRAPH 2007 Courses*; SIGGRAPH '07; Bridson, R., Ed.; ACM: New York, NY, USA, 2007. [[CrossRef](#)]
25. Thuerey, N.; Pfaff, T. MantaFlow. 2018. Available online: <http://mantaflow.com> (accessed on 31 December 2018).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).