Computers & Geosciences 165 (2022) 105146

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

Spatiotemporal flow features in gravity currents using computer vision methods

F.D. Vianna *, B.A. Farenzena, M.S. Pinho, J.H. Silvestrini

School of Technology, Pontifical Catholic University of Rio Grande do Sul, Porto Alegre, Brazil

ARTICLE INFO

Keywords:
Gravity currents
Lobes and clefts structures
Computer vision methods
Feature point tracking

ABSTRACT

Relationships between the features visually identified at the front of the flow's current and parameters regarding its velocity and turbulence were observed in early experimental works on the characterization of gravity currents. Researches have associated front features, like lobes and clefts, with the flow's turbulence, and have used these associations ever since. In more recent works using numerical simulations, these connections were still being validated for various flow parameters at higher front velocities. The majority of works regarding measurements at the front of a gravity current rely on the front's images for making its analysis and establish relationships. Besides that, there is an interdisciplinary field related to computer science called computer vision, devoted to study how digital images can be analyzed and how these results can be automated. This paper describes the use of computer vision algorithms, particularly corner detection and optical flow, to automatically track features at the front of gravity currents, either from physical or numerical experiments. To determine the proposed approach's accuracy, we establish a ground-truth method and apply it to numerical simulation results data sets. The technique used to trace the front features along the flow showed promising results, especially with higher Reynolds numbers flows.

1. Introduction

Data analysis from numerical simulations of fluid dynamics usually involves the graphical representation of values obtained as results from these simulations, such as temperature, density, velocity, among others. Researchers studying these phenomena use scientific visualization software tools to help bring meaning and a proper interpretation of the large amount of data generated by simulations (J. Whitlock et al., 2016). In many cases, these software tools generate images representing aspects of the analyzed flow (Fabian et al., 2011), and these images can be further interpreted to generate scientific conclusions.

Among many of these physical phenomena, there are the gravity currents which are fluid flows where two fluids with different densities interact driven by a gravitational field resulting in a primarily horizontal flow (Simpson, 1999). There are numerous examples of gravity currents occurrences in nature, like an avalanche (Fig. 1), a pyroclastic flow from a volcano eruption, a sandstorm, among others. Many of these natural phenomena may have impact on man made infrastructure (Kaye et al., 2009; Al-Hemoud et al., 2019; Stefanski and Sivakumar, 2009). One particular type of gravity current is called turbidity current, which occurs when the density difference is originated from sediment particles in suspension (Meiburg and Kneller, 2010; Kneller and McCaffrey, 1999). Such flows are often associated as a consequence of submarine landslides, which affects the integrity of submarine structures like piping systems and cables (Zakeri et al., 2008; Nisbet and Piper, 1998). Besides that, turbidity currents are also linked with the formation of hydrocarbon reservoirs (Meiburg and Kneller, 2010; Syvitski et al., 1996). Gravity currents are studied using numerical simulations with data visualization (Cantero et al., 2007b; Nasr-Azadani et al., 2013; Francisco et al., 2018; Espath et al., 2014; Lucchese et al., 2019; Farenzena, 2020; Härtel et al., 2000a; Frantz et al., 2021), also with images obtained from physical experiments (Simpson, 1972; Neufeld, 2002; McElwaine et al., 2004) and with images from the actual phenomenon in nature (Horner-Devine and Chickadel, 2017).

Simpson (1999) acknowledges that gravity currents have two main parts, the frontal part which the author calls head, followed by its back portion. This back portion is usually divided in the body and the tail (Hacker et al., 1996; Meiburg and Kneller, 2010) as shown in Fig. 2. At the head of the gravity current there is an interface between the fluids with different densities. This interface is called the front of the gravity current and at this interface we observe characteristic structures called lobes and clefts (Simpson, 1982) as illustrated in Fig. 2. When observing the planform of the gravity current, a human observer, using the proper technique to distinguish the fluids with different

* Corresponding author.
E-mail address: filipi@pucrs.br (F.D. Vianna).

https://doi.org/10.1016/j.cageo.2022.105146
Received 15 June 2021; Received in revised form 13 May 2022; Accepted 15 May 2022
Available online 23 May 2022

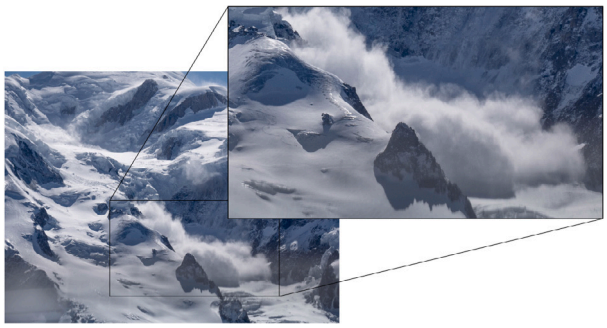

Fig. 1. Detail showing a gravity current during an avalanche at the Mont Blanc.
Source: Adapted from Kowalik (2019).

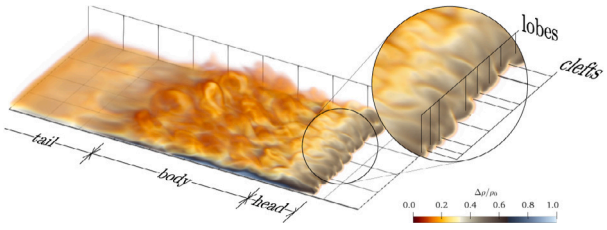

Fig. 2. The image shows the normalized density fluctuation field volume rendering of data from gravity current simulation results where each spacing interval on the grid corresponds to one length unit. The anatomy of a gravity current is also presented and the detail indicates the lobe and cleft structures in the frontal region of its head.

densities (Shin et al., 2004; Garvine, 1984; Dai, 2013), is able to spot these characteristics on the front. These structures can be identified in natural systems (Garvine, 1984; Horner-Devine and Chickadel, 2017) and were also reproduced in visualizations of numerical and physical simulations (Simpson, 1999; Härtel et al., 2000b; Neufeld, 2002).

The spatiotemporal evolution analysis of these lobes and clefts structures considers how a lobe reduces its span-wise characteristic length, making clefts merge and how lobes grow onto a particular size in which one lobe splits in two, making a new clef between them (Simpson, 1972). The number of clefts associated with physical parameters such as the front velocity (Espath et al., 2015) and the frequency of clefts and the lobe sizes (Neufeld, 2002) are fundamental for the correct physical evaluation of the simulations.

A straightforward approach to experimentally reproduce gravity current is the lock-release configuration. This approach consists of a quick release of a fixed volume of a dense fluid into another fluid with a smaller density by removing a plate that initially keeps both fluids apart.

With this approach (Simpson, 1972, 1999), using an opaque marker to distinguish the fluids with different densities to make shadowgraphs, has first manually mapped the evolution of a gravity current's front and tracked the behavior of the clefts. This procedure enabled the measurement of the span-wise characteristic length of the lobe and its behavior as the front of the gravity current moves. Due to technology limitations imposed by the ciné photograph and shadowgraph techniques, this experiment had an image acquisition rate high enough to register 20 front positions. The experimental results, obtained by the author, had 20 fronts marked and the clear front outlines enabled the manual clefts tracking. Fig. 3 shows a schematic bottom view of the gravity current's front outlined from subsequent photos taken in fixed intervals, where the clefts are manually marked, and their paths are represented with dashed lines. With this method, Simpson (1972) proposed scaling laws that enabled predicting the gravity current's front behavior at higher velocities. However, these scaling laws are still to be validated to higher front velocities and different lock-release configurations.

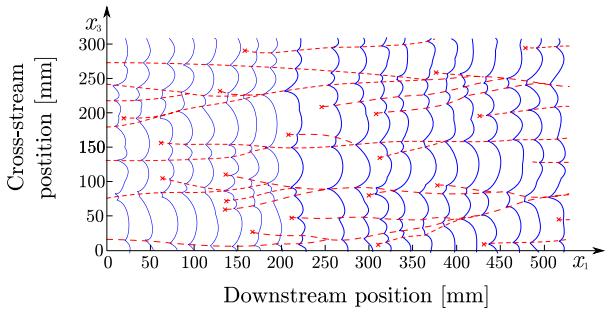

Fig. 3. The image shows a schematic bottom view of the gravity current's front outlined from subsequent shadowgraphs taken at fixed time intervals. The flow direction was from left to right. The clefts were manually marked, and their path was shown with dashed lines.
Source: Modified from Simpson (1972).

Furthermore, Espath et al. (2015) have compared cleft's path with the deposit signature on the bottom of the flow for channel lock release configuration. Their work presents evidence of a correlation between impressions at the bottom of the flow with the path of the cleft. Cantero et al. (2008) also trace the clefts evolution to show its correlation with the location of the quasi-streamwise vortices at the nose of the gravity current, and Dai and Huang (2022) did the same to show the continuity of the clefts. In the absence of a tool to automatically track clefts, they have to do it manually at each output time. If the clefts can be automatically detected using computer vision techniques and can be tracked between different images using an optical flow algorithm, and assuming these techniques can be combined into a single tracking tool. Such an automatic tool would positively impact further investigations of these correlations based on more recent simulations with result sets larger than those from early experiments. Also, a general approach for tracking clefts can also be used in other configurations different from the planar releases. This would enable the analysis of configurations like the cylindrical release (Cantero et al., 2004, 2005, 2007a), the basin or channel-basin (Francisco et al., 2018; Inghilesi et al., 2018) without the necessity of modeling the front line function.

Moreover, we can analyze the projection of the gravity current in the bottom as a function in the span-wise direction. The local minima of this function can be assumed to be the clefts. Thus, its positions can be used to verify those obtained using computer vision.

The human eye naturally processes and associates visual information (Ballard and Brown, 1982) which can be in the form of images and video representation of numeric simulation results. However, the vast amount of data generated by numerical simulations makes analyzing these data a costly process with a high level of difficulty to observe the behavior of the particular phenomena being studied (Camata et al., 2018).

Computer vision is widely used for detecting, extracting, and tracking features on images (Brown, 1988). There are several computer vision algorithms for feature detection and tracking. The edge detectors and corner detectors are used to detect these kind of feature points on images, and optical flow algorithms to track them with small movements between frames in a computer movie or animation.

In order to evaluate the accuracy of the proposed computer vision techniques, this study is restricted to planar lock-release configurations. The measurements will be compared with cleft location obtained using local minima of each density front line from the numerical simulation data. With such an accuracy evaluation, we can enable automatic measurements and cleft tracking. Finally, this paper is organized as follows: firstly, in Section 2, we present a brief review of the methods employed automating the lobe and cleft structures detection; in Section 2.1, we describe the chosen method; followed by a description of the numerical simulations performed to acquire data; the results are shown in Section 3 and, finally, in Section 4 we present a summary of all observations and the conclusions.

2. Methods and materials

Regarding the detection and tracking of the lobe and cleft structures, some works (Neufeld, 2002; McElwaine et al., 2004; Horner-Devine and Chickadel, 2017) show automatic methods using image processing techniques. Neufeld (2002) analyzed several data sets obtained using image acquisition of a lock-release experiment. In this experiment, the images were recorded at a shorter time interval compared to Simpson (1972), resulting in more clefts to track. The algorithm used to identify clefts in the acquired images, scans the front contour in the span-wise direction for each time step. With this approach the author was able to automatically detect the clefts and plot marks in their locations, enabling the visual tracking of them along with the flow. This technique was applied to several experiments that had replicated previous numerical simulations made by Härtel et al. (2000b).

Horner-Devine and Chickadel (2017), were able to extract the cleft locations from thermal infrared and acoustic footage of a river plume using image processing techniques, enabling the analysis of a real-world scale event. The authors also show similar results with their data set (Figure 2d in Horner-Devine and Chickadel, 2017). In both works, the results were qualitatively evaluated by measuring lobe sizes as the distance between clefts. Additionally, the authors have also used Simpson's scale laws to evaluate the accuracy of the lobe characteristic length.

Regarding the tracking of clefts along the evolution path of the gravity current, McElwaine et al. (2004) proposed a method to automatically trace the clefts' paths from video sequences of physical experiments using a planar lock-release configuration. For their method, it is also necessary to handle the frontal contour line as a function using rectangular coordinates in the span-wise direction therefore it would not be suitable to others configurations with different types of coordinates systems. To detect and track the clefts, they first convert a video sequence of the moving front of a gravity current into one image with color levels. Besides this method, no other work has been found applying automatic algorithms to track the clefts' positions.

2.1. Detecting clefts using corner detection methods

In order to identify and track one or many features from the front of the gravity current using computer vision, it is necessary to have a set of images of the flow. Each image should have visual information regarding the gravity current front position and shape. The images could come from simulations or experiments, but we can also extract them from figures and illustrations. For example, the lines representing the front at Fig. 3 can be extracted, resulting in one animation with 20 frames, one line per frame. In particular, we will treat clefts as corners, and there are several algorithms for corner detection, such as the Harris corner detector (Harris and Stephens, 1988) or the Shi–Tomasi corner detector (Shi and Tomasi, 1994).

The Harris corner detector (Harris and Stephens, 1988) starts computing an image intensity difference as

$$E(\Delta x_1, \Delta x_3) = \sum_{x_1,x_3} \Big\{ \underbrace{w(x_1,x_3)}_{\substack{\text{window}\\ \text{function}}} \big[\underbrace{I(x_1+\Delta x_1, x_3+\Delta x_3)}_{\substack{\text{shifted}\\ \text{intensity}}} - \underbrace{I(x_1,x_3)}_{\substack{\text{intensity}}} \big]^2 \Big\}, \tag{1}$$

where E is the intensity difference for a displacement $(\Delta x_1, \Delta x_3)$ in both stream-wise (x_1) and span-wise (x_3) directions (see Fig. 3), $I(x_1, x_3)$ is the image intensity function for a two-dimensional image, $I(x_1 + \Delta x_1, x_3 + \Delta x_3)$ is the image intensity function shifted by the $\Delta x_1, \Delta x_3$ displacement. $w(x_1, x_3)$ is a rectangular window function, and gives the weights of the pixels below the window. For corner detection, the intensity difference (Eq. (1)) must be maximized, to do that the shifted intensity has to be maximized.

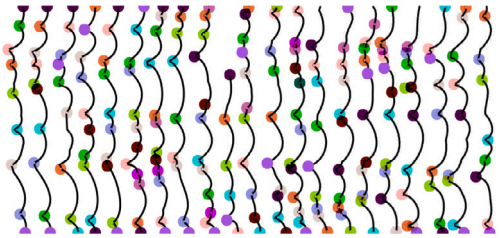

Fig. 4. Output result of our prototype detecting clefts using Shi–Tomasi corner detection (Shi and Tomasi, 1994) on data from Fig. 3. The corners were marked with randomly different colors to evidence the distinction between nearby clefts.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

After applying Taylor series expansion to the Eq. (1), we can approximate the intensity difference as

$$E(\Delta x_1, \Delta x_3) \approx \begin{bmatrix} \Delta x_1 & \Delta x_3 \end{bmatrix} \mathbf{M} \begin{bmatrix} \Delta x_1 \\ \Delta x_3 \end{bmatrix}, \tag{2}$$

and the $\mathbf{M}$ matrix as

$$\mathbf{M} = \sum_{x_1,x_3} w(x_1,x_3) \begin{bmatrix} \frac{\partial I}{\partial x_1}\frac{\partial I}{\partial x_1} & \frac{\partial I}{\partial x_1}\frac{\partial I}{\partial x_3} \\ \frac{\partial I}{\partial x_1}\frac{\partial I}{\partial x_3} & \frac{\partial I}{\partial x_3}\frac{\partial I}{\partial x_3} \end{bmatrix}. \tag{3}$$

So, according to Harris and Stephens (1988), to determine if a window contains a corner or not, it is necessary to compute the R score as

$$R = \det(\mathbf{M}) - k\,\mathrm{trace}(\mathbf{M})^2, \tag{4}$$

where k is a constant used as a tuning parameter usually between 0.04 and 0.06 (Sánchez et al., 2018). If $|R|$ is small, the region is flat; if $R < 0$, the region is an edge and if R is large, the region is a corner. Operating $\det(\mathbf{M}) = \chi_1\chi_2$ and $\mathrm{trace}(\mathbf{M}) = \chi_1 + \chi_2$, being χ_1 and χ_2 the eigenvalues of $\mathbf{M}$, we can express the R score as

$$R = \chi_1\chi_2 - k\left(\chi_1 + \chi_2\right)^2. \tag{5}$$

Shi and Tomasi (1994) proposed a simpler way to evaluate the score function as

$$R_{ST} = \min\left(\chi_1, \chi_2\right). \tag{6}$$

If R_{ST} is greater than a threshold value, the region contains a corner. Using that, we can assume that the most accentuated inflections on the front function, the clefts, will be classified as corners, while the lobes would be classified as edge or flat.

Fig. 4 shows the result of a prototype which we wrote using OpenCV's implementation of Shi–Tomasi corner detection on a data set of 20 frames obtained from the image of Simpson's lock exchange experiment. For this particular test, we choose to mark the corners with randomly different colors to distinguish nearby clefts. A quantitative accuracy evaluation was not made due to the small number of clefts marked in the original image. However, a qualitative visual inspection of the results shown in Fig. 4 and comparison with the clefts marked on Fig. 3 assure that all spotted corners are actual clefts, even those too close to each other that the original work has treated them as a single one.

The same experiment, running only the corner detector for each frame, was conducted using a data set comprised of 76 images extracted from the Figure 4.11 found in the work of Neufeld (2002). That figure represents the data set he called "may26b" and is available in a vector graphic format, which enabled the extraction and segmentation of each time step's contour line. Fig. 5a shows the results obtained by Neufeld (2002). Comparing his results with our initial corner detection test (Fig. 5b), we can spot some differences, although clefts were detected.

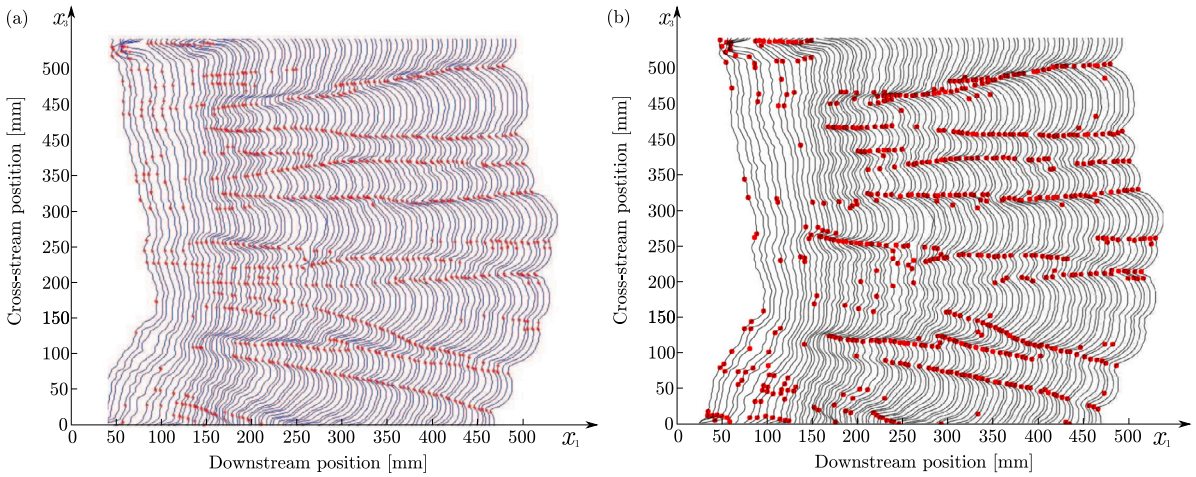

Fig. 5. The red dots mark the clefts. In (a), using Neufeld (2002) technique which we have previously described, and in (b), the clefts are marked using corner detection using our proposed approach in which the input data was extracted from one vector illustration found in Neufeld (2002). These images are shown together for a side-by-side comparison of results from two different cleft location techniques.
Source: (a): Modified from Neufeld (2002).

2.2. Tracking clefts with optical flow

The main goal of optical flow methods is to compute an approximation of a two-dimensional motion from spatiotemporal patterns of image intensity (Barron et al., 1994). In this case we have to add the time to the image intensity function. So we can define the image intensity function as $I(x_1, x_3, t)$ where (x_1, x_3) are the two-dimensional spatial location of a point and t is the frame number of a set of images composing an animation. Assuming that in a short time increment, an image feature changes its location, this can be modeled as

$$I(x_1 + v_1, x_3 + v_3, t + \Delta t) \approx I(x_1, x_3, t). \tag{7}$$

Applying the Taylor expansion to the Eq. (7) and neglecting the high order terms we obtain

$$\frac{\partial I}{\partial x_1} v_1 + \frac{\partial I}{\partial x_3} v_3 + \frac{\partial I}{\partial t} \Delta t = 0 \tag{8}$$

and $\Delta t = 1$ (one frame increment), we have the primary optical flow constraint expressed as

$$\nabla I \cdot \boldsymbol{v} + \frac{\partial I}{\partial t} = 0 \tag{9}$$

where the image intensity gradient is defined as

$$\nabla I = \left[\frac{\partial I}{\partial x_1}, \frac{\partial I}{\partial x_3} \right]^T \tag{10}$$

and $\boldsymbol{v} = [v_1, v_3]^T$ is the optical flow vector.

Considering the Eq. (9) we can estimate the displacement in the image intensity gradient direction. Unfortunately, this constraint is not enough to estimate every component of the motion field. Because of this limitation, we cannot correctly estimate the movement on edges, in which the apparent motion will always be normal to the edge. Therefore, the corners provide enough information for the direction they move to be properly estimated. This condition is called the "aperture problem" (Beauchemin and Barron, 1995).

To tackle the aperture problem, it is necessary to define constraints to estimate the optical flow not only at the image intensity gradient direction (Eq. (10)). From these necessity arose several different approaches to compute the optical flow. Barron et al. (1994) classifies these approaches as: differential methods (Horn and Schunck, 1981; Lucas and Kanade, 1981; Nagel, 1987), methods based on regions (Bergen et al., 1992), energy based methods (Heeger, 1988) and phase based methods (Waxman et al., 1988; Fleet and Jepson, 1990).

There are some surveys on optical flow methods (Fortun et al., 2015; Beauchemin and Barron, 1995; Chao et al., 2013, 2014), which analyze and describe each method and its applications. Moreover (Barron et al., 1994) did an extensive review and performance analysis of the main optical flow methods. They have concluded that the first order, local differential method of Lucas and Kanade (1981), and the local phase-based method of Fleet and Jepson (1990) were "the most reliables".

Therefore, for this work, we will concentrate on the use of the Lucas–Kanade method. This method uses a weighted window around each pixel to solve the optical flow, usually varying the size according to the number of desired neighbor elements. Finally, the solution is obtained by solving a linear system with each pixel's neighborhood size. The algorithm also differentiates an intensity matrix in order to identify the displacement of pixels between two images, and because of the assumptions made (Eq. (8)), the pixel displacement should be small.

To implement a feature tracker using a combination of corner detection and optical flow requires an iterative algorithm where new features should be discovered and certain features should be declared lost. The KLT (Kanade–Lucas–Tomasi) algorithm (Suhr, 2009; Tomasi and Kanade, 1991), combines the Shi–Tomasi corner detector and an interactive for of the Lucas–Kanade optical flow. Applying this algorithm to a video containing gravity currents moving fronts should automatically trace the clefts' path.

The implementation of all computer vision prototype software developed for this work was done using the open-source computer vision library OpenCV (Bradski and Kaehler, 2000), using the Python language. We also implemented a KLT according to Suhr (2009), following the steps presented in the flow chart in Fig. 6. For that, we used the OpenCV's implementations of Shi–Tomasi, function `goodFeaturesToTrack` in Python `cv2` library and, from the same library, Lucas–Kanade in implemented in the function `calcOpticalFlowPyrLK`.

2.3. Numerical simulations

The data acquisition for features detection and tracking is performed using numerical simulations of planar lock-release gravity currents. The canonical set-up, previously described and illustrated by Fig. 7 has x_1 as the stream-wise direction, x_2 the vertical direction and the x_3 the span-wise direction. The released volume of the dense fluid is initially enclosed in $L_{1,b} \times H \times L_3$. Removing the lock plate, which separates the two fluids, starts a gravity current that propagates horizontally. Such flow configuration, is assumed to be an incompressible flow under the Boussinesq approximation. The corresponding governing

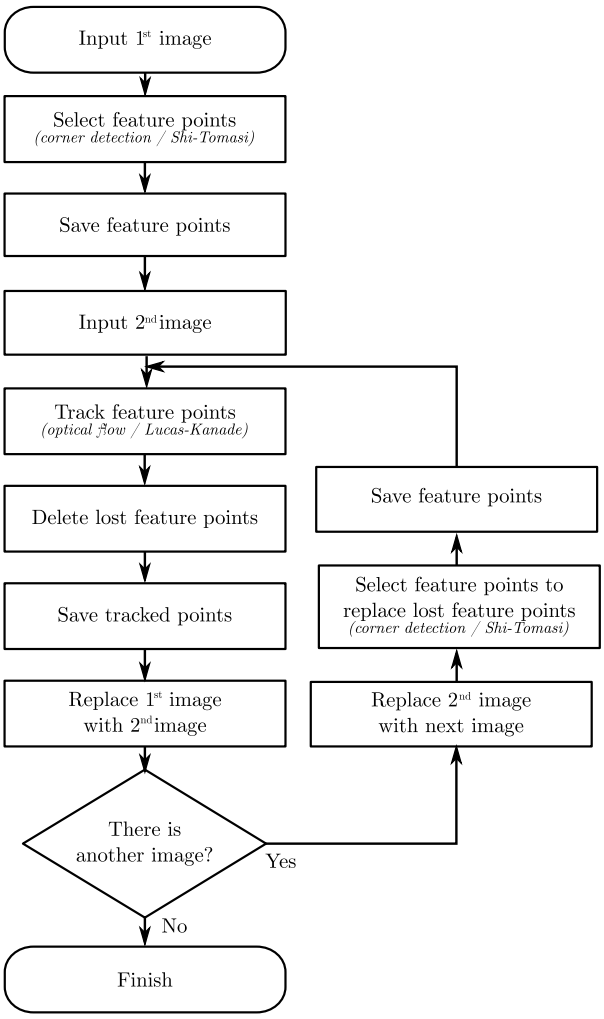

Fig. 6. Flow chart of the implemented feature tracker.
Source: Modified from Suhr (2009).

equations, under these hypothesis, in the cartesian coordinate system $\boldsymbol{x} = \left[x_1, x_2, x_3\right]^T$, are given in the dimensionless form as

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0, \tag{11}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\,\boldsymbol{u} = -\boldsymbol{\nabla} p + \frac{1}{Re}\nabla^2 \boldsymbol{u} - \varphi \boldsymbol{e}_2, \tag{12}$$

$$\frac{\partial \varphi}{\partial t} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\,\varphi = \frac{1}{Sc\,Re}\nabla^2 \varphi, \tag{13}$$

where t is the time; $\boldsymbol{u} = \left[u_1, u_2, u_3\right]^T$ is the velocity field; p is the pressure; $\boldsymbol{e}_2$ is the unitary vector oriented in the vertical direction; Re is the Reynolds number; Sc the Schmidt number and φ the density fluctuation. The dimensionless density fluctuation is related with the fluid density by the equation:

$$\rho = \Delta\rho\,\varphi + \rho_0, \tag{14}$$

being $\Delta\rho$ the density difference between both fluids and ρ_0 the light fluid density.

The velocity field is made dimensionless with the buoyancy velocity scale $u_b = \sqrt{g' H}$, being $g' = \Delta\rho g/\rho_0$ the reduced gravity acceleration (Simpson, 1999) and H is the length scale, which in this case is the channel height (see Fig. 7). Consequently, the pressure and time scales are $\rho_0 u_b^2$ and H/u_b respectively.

The Reynolds number in such configuration is defined as

$$Re = \frac{u_b H}{\nu}, \tag{15}$$

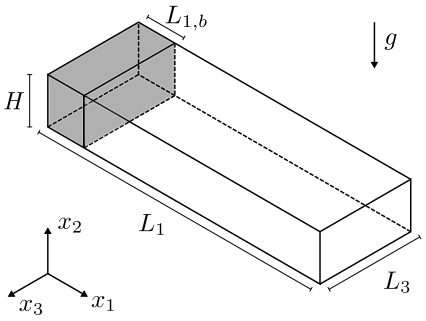

Fig. 7. The image shows a schematic view of the initial configuration of a lock-release gravity current.

where ν is the kinematic viscosity. It is noteworthy that the Richardson number is not explicit as a free parameter due to the convenient choice of characteristic scales, which reduces this dimensionless number to the unity. The Schmidt number is defined as the ratio between the fluid kinematic viscosity and the mass diffusivity. In order to keep the computational cost minimum, $Sc = 1$ is adopted (Cantero et al., 2007b; Espath et al., 2014, 2015; Farenzena and Silvestrini, 2022).

The boundary conditions considered for the velocity are: no-slip in $x_2 = 0$, free-slip in $x_1 = 0$, $x_1 = L_1$ and $x_2 = 1$; and for the scalar field no-flux in $x_1 = 0$, $x_1 = L_1$, $x_2 = 0$ and $x_2 = 1$; and periodicity is imposed in the x_3 direction. The initial condition of the scalar field is prescribed with the use of a smooth function in order to avoid local discontinuities

$$\varphi(\boldsymbol{x}, 0) = \frac{1}{2} + \frac{1}{2}\tanh\left[\sqrt{ReSc}\,(x_1 - L_{1,b})\right]. \tag{16}$$

A weak random number perturbation, with amplitude $O(10^{-1})$, is superimposed in all components of the velocity field localized at the neighborhood of the lock position ($x_1 = L_{1,b}$).

Simulations were carried out with the open-source parallel code Incompact3D (Bartholomew et al., 2020), designed for Direct Numerical Simulations and Large Eddy Simulations of incompressible flows. The code is based on sixth-order compact finite difference schemes for spatial-discretization (Lele, 1992). The momentum equation's nonlinear term is computed in the skew-symmetric form to reduce aliasing and increase stability (Kravchenko and Moin, 1997). A third-order explicit Adams–Bashforth method is employed for time integration. The pressure is obtained via a fractional step (prediction/correction method) by solving a Poisson equation in the spectral space with a distributed Fast Fourier Transform (FFT) package.

Based on the work of Cantero et al. (2007b), three numerical simulations in the lock-release configuration are performed varying the Reynolds number (Re) and keeping the computational domain lengths constant, considered as $(L_1 \times L_2 \times L_3) = (25 \times 1 \times 4)$ with $L_{1,b} = 1$. This computational domain corresponds to the "planar small release" setting series from Cantero et al. (2007b), but we opt to use a larger domain in the x_1 and x_3 directions to have a greater sampling of the lobe and cleft structures when compared with the reference study. The numerical parameters, including the number of grid points in each direction (n_1, n_2 and n_3), time step increment (Δt) and simulation time (t_f), and the corresponding approach for each simulation are summarized in the Table 1. The simulation LR895 is solved with the Direct Numerical Simulation (DNS) approach, while the cases LR3450 and LR8950 with Implicit Large Eddy Simulation (ILES) based on the method proposed by Dairay et al. (2017). As reported by Frantz et al. (2021), the ILES approach can be employed to reduce the simulations' computational cost as larger Reynolds number values are considered, which can be considered a good test to the proposed method of features detection due to the reduced grid resolution compared with the DNS approach.

All the simulations were carried out on a cluster made of Intel® Xeon® CPU E5-2620, with 24 compute cores per node, running at

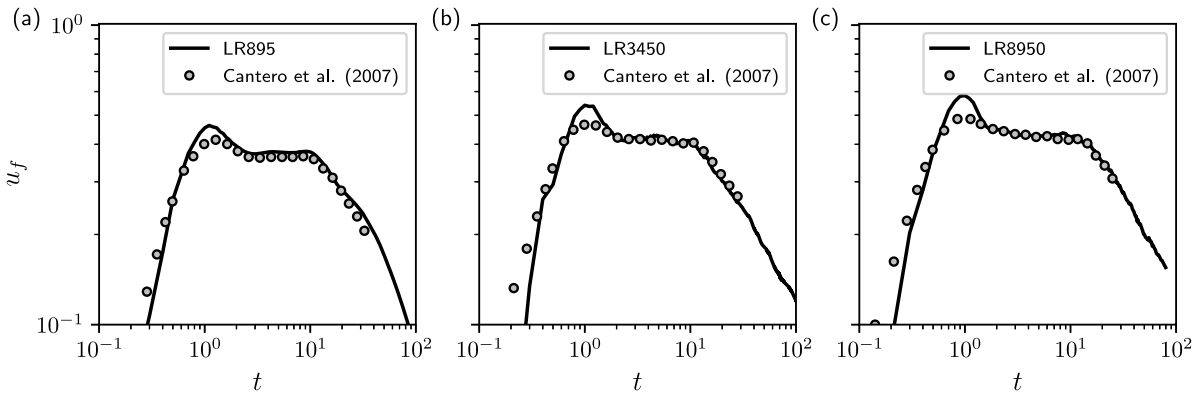

Fig. 8. Front velocity as function of the time of our simulations and the data available in Cantero et al. (2007b), for $Re = 895$ (a), $Re = 3450$ (b) and $Re = 8950$ (c).

Table 1
Numerical simulations parameters. All simulations have $L_1 = 25$, $L_{1,b} = 1$, $L_2 = 1$, $L_3 = 4$ and $Sc = 1$.

Case	Re	$n_1 \times n_2 \times n_3$	Δt	t_f	Numerical approach
LR895	895	$1351 \times 55 \times 216$	10^{-3}	160	DNS
LR3450	3450	$801 \times 93 \times 128$	10^{-3}	120	ILES
LR8950	8950	$1601 \times 145 \times 256$	5×10^{-4}	90	ILES

2.00 GHz. The case LR3450 used 4 cluster nodes and took around 26 h to complete. The case LR895 used 4 cluster nodes and took around 25 h to complete. Finally, the case LR8950 used 8 cluster nodes and took around 144 h to complete.

In order to have a better comparison of the data acquired from numerical simulations with the scaling laws proposed by Simpson (1972) and render figures similar to Fig. 3, we use the projection of the density current in the (x_1, x_3) plane for each time. Such projection of the three-dimensional φ field would be the numerical equivalent of taking a picture from the top of the channel and have the form:

$$\varphi_{\mathrm{p}}(x_1, x_3, t) = \int_0^1 \varphi(\boldsymbol{x}, t) dx_2. \tag{17}$$

Combining the numerical simulations and the proposed data treatment, we validate our analysis by comparing the results with the scaling laws proposed by Simpson (1972). The scaling law of interest in this work describes the mean lobe size ($\bar{\lambda}$) normalized by the head height (h), as a function of the front Reynolds number (Re_f):

$$\frac{\bar{\lambda}}{h} = 7.4 Re_f^{-0.39 \pm 0.02}, \quad \text{with} \quad Re_f \geq 200, \tag{18}$$

where the front Reynolds number is obtained by re-scaling the global Reynolds number (Re) with the dimensionless head height and front velocity, in the form:

$$Re_f = u_f h Re. \tag{19}$$

The front velocity is derived from the front position in the form of:

$$u_f(t) = \frac{dx_f}{dt}, \tag{20}$$

and the front position (x_f) is defined, for each time, as the maximum x_1 value where the span-wise average of φ_{p} is greater than 10^{-2}.

3. Results

3.1. Numerical simulations

Fig. 8 presents the front velocity as a function of time, determined using the previously described method, for each simulation case and a comparison with the results of Cantero et al. (2007b). The authors describes that the higher the Reynolds number, the faster the resulting gravity current propagates in the stream-wise direction. Firstly the flow has a rapid acceleration, then it propagates with nearly constant velocity, then the front velocity decays proportional to $t^{-1/3}$, and finally, the front velocity decays proportional to $t^{-5/8}$. Such behaviors after the rapid acceleration phase are referred as the slumping phase, inertial phase and viscous phase respectively.

Evaluating the Eq. (17) for each time step and adopting the reference isovalue, figures similar to Figs. 3 and 5a are generated, as are shown in Fig. 9. To graph each isoline, we adopt a nearly constant front position increment due to its temporal behavior. Initially ($x_1 - L_{1,b} < 5$), all simulation cases develop lobe and cleft structures with the wavelength as a function of the Reynolds number. As the gravity current propagates, these small structures start to merge and form larger structures, and after some time, when a lobe reaches a critical length, it splits into two new structures as described in Simpson (1972). For the lower Reynolds number, after the lobes reach a critical length, they saturate and do not split to form new structures.

The mean lobe length during the slumping phase is determined using the isolines presented in Fig. 9. The lobe length is defined as the distance between two local minima of a φ_p isoline using signal analysis tools to provide base results for the computer vision method's accuracy evaluation. The local minima were extracted using the routine `scipy.signal.argrelextrema` from the scipy (Virtanen et al., 2020) python library signal analysis tools. The results of such analysis is presented in Fig. 10, where is shown the time mean lobe length as function of the front Reynolds number for the LR3450 and LR8950 cases and Eq. (18). We observe that these simulation cases are in accordance with the results initially predicted by Simpson (1972), presenting similar slope and being inside the uncertainty range of Eq. (18).

3.2. Evaluating computer vision methods

In Section 2.1 we have presented the results of two early tests conducted exclusively using corner detection to find the cleft positions. Figs. 5b and 4 show the result of a prototype code written using OpenCV implementation of Shi–Tomasi corner detector on image data sets manually extracted from figures showing gravity currents front evolution in time. We were able to qualitatively evaluate the corner detection algorithm with these early results and confirm that it is suitable to detect clefts in gravity currents.

The approach using only corner detection has shown a promising result when applied to a lean data set from the first test (see Fig. 4). However, when tested against a data set with a high number of lobe and clefts structures, the results were qualitatively unsatisfying. From Fig. 5 wen can observe clefts detected in one time step but not in the previous or the next. Nevertheless, our KLT implementation has enabled us to address this issue because it discards the feature point if

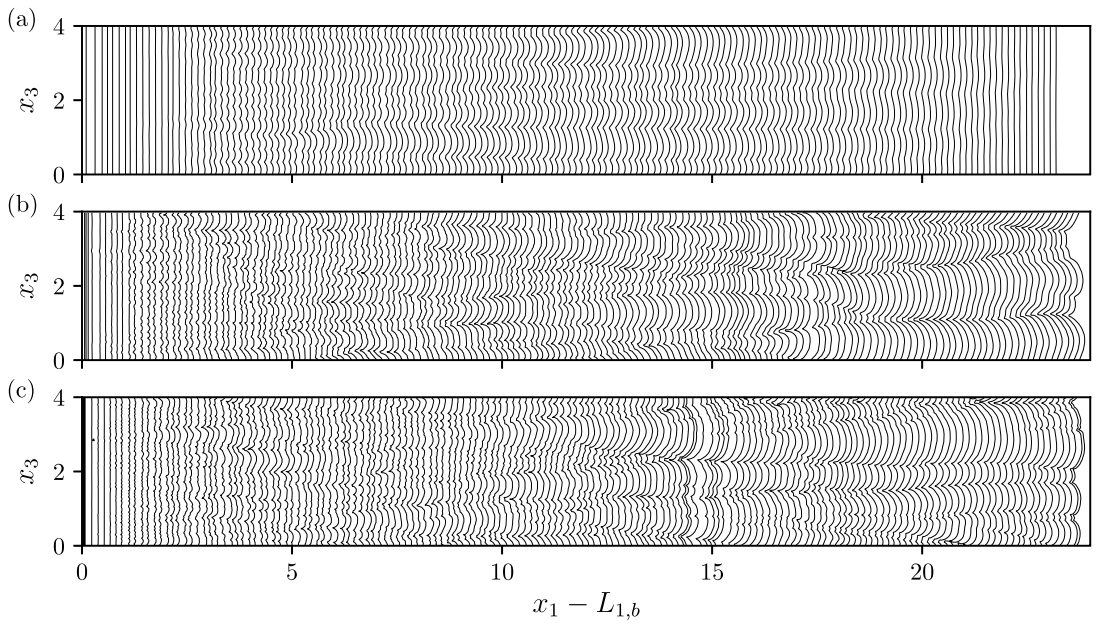

Fig. 9. Temporal evolution of the concentration projection isolines for the cases LR895 (a), LR3450 (b) and LR8950 (c). An isovalue of $\varphi_p = 10^{-2}$ was adopted to draw each isoline with a variable time step that corresponds a constant front position variation.

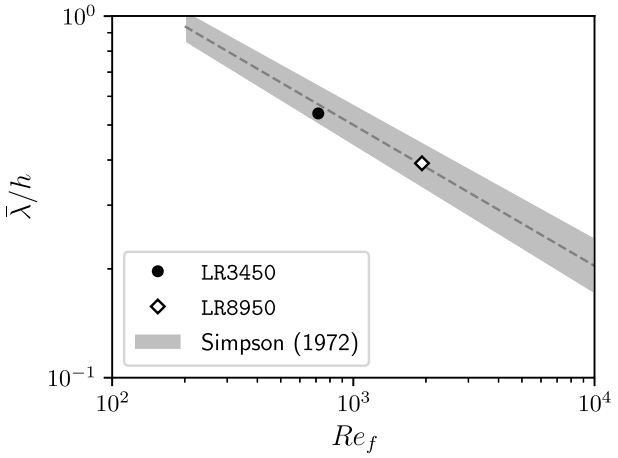

Fig. 10. Comparison of the mean lobe size as function of the front Reynolds number of the simulation cases LR3450 and LR8950 (during the slumping phase) with Eq. (18). Results obtained with signal analysis tools.

it is not detected in previous and next frames. In this way, the optical flow algorithm helped to refine the feature detection.

The Fig. 11 shows the results of the KLT algorithm applied to the three simulation cases. In order to evaluate the proposed method, two approaches can be taken, one quantitative and another qualitative. The cleft's tracks of Fig. 11 can be used to qualitatively evaluate the results obtained. As new corners are detected, the tracked movement of these features is consistent with the pattern observed in Fig. 9.

From Fig. 11, we can check that when using the KLT algorithm, it was possible to track the features once they were detected. The new found features have also been tracked correctly, not interfering with the path of previously detected features. The technique also shows better results for flows with higher Reynolds numbers (Figs. 11b and 11c).

The computer vision technique used to find clefts can be quantitatively evaluated by comparing its results with the local minima obtained using signal analysis tools. The locations of all local minima in this function should be considered a cleft. The results obtained with signal analysis tools, computer vision methods and Eq. (18) were compared using the mean lobe size as function of the front Reynolds number of the simulation cases LR3450 and LR8950 (see Fig. 12). We observe that for both cases, the computer vision technique performed with less accuracy when compared with the ground-truth, but a similar trend slope can be perceived. We can also observe the results for higher Reynolds numbers are closer to the (Simpson, 1972) scaling law than the lower Reynolds results.

The reduced accuracy for flows with lower Reynolds numbers can be attributed to the corner detection process. As the clefts at the gravity current's front are smaller, they are classified as flat. This results in larger lobe sizes, outside of the uncertainty range.

4. Summary and conclusions

This work has evaluated the use of computer vision techniques to analyze spatiotemporal flow features in gravity currents. The technique used in this work's implementation, called KLT, for Kanade–Lucas–Tomasi, consists of using a corner detection algorithm (Shi and Tomasi, 1994) to detect clefts and an optical flow algorithm (Lucas and Kanade, 1981) to track the detected clefts along with the flow. Early results with data sets from physical experiments had shown potential for the use of corner detection. The implementation of the KLT helped refine the feature point detection. For a better quantitative evaluation of the method, numerical simulations were performed. The simulations provided data sets larger than the first tests made using data extracted from figures of other works. That allowed measuring the mean lobe size with a ground truth method. This enabled comparing the ground truth measurement and the computer vision method against a scaling law. The flow configuration of the simulations was the lock-release, and the flow parameters were chosen to replicate simulations performed by Cantero et al. (2007b), with three scenarios, each with different Reynolds numbers.

Compared with a ground truth technique to find clefts position, the corner detection performed poorly for the lower Reynolds number (Re) flows, but with higher Reynolds numbers, and increased accuracy is perceived. The results were out of the uncertainty range determined by early Simpson's experiments. However, the KLT showed promising results in tracking the clefts' movement and tracing its path. The results were qualitatively better with flows with higher Reynolds numbers. Due to the geometric nature of turbulent features to behave more

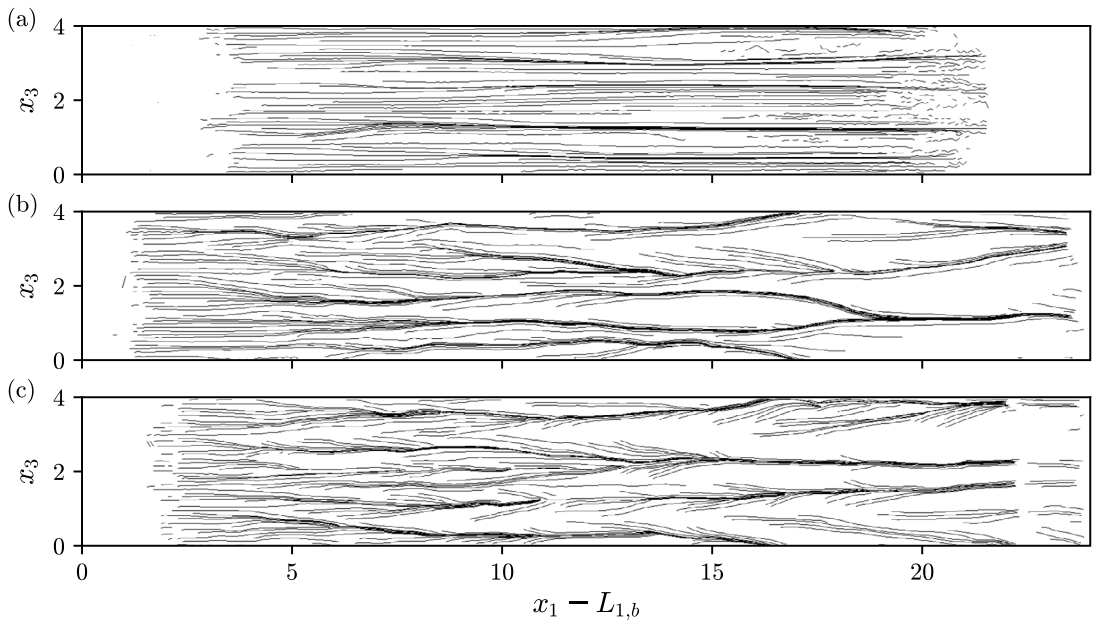

Fig. 11. Clefts' tracks for the cases LR895 (a), LR3450 (b) and LR8950 (c) obtained after applying the KLT algorithm to the simulation results.

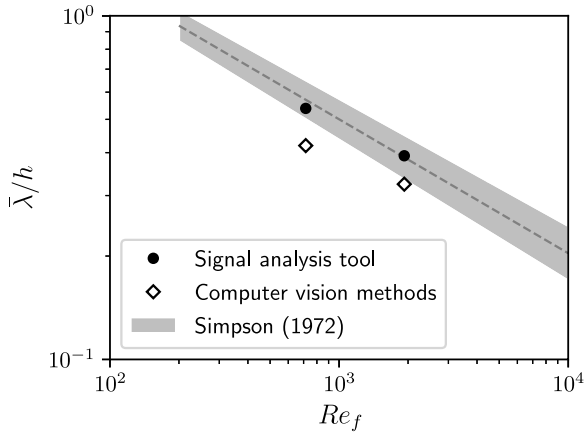

Fig. 12. Mean lobe size as function of the front Reynolds number of the simulation cases LR3450 and LR8950 (during the slumping phase), with comparison between the results obtained with signal analysis tools, computer vision methods and Eq. (18).

like corners than edges the authors are confident that higher Reynolds numbers would lead to a better corner detection as well as optical flow. One concern however, is that too sharp lobes, resulting from a high Reynolds number turbulent flow can be misclassified as clefts. This can be solved using a minimum value test check. These conclusions enable the use of KLT to further investigate flows with higher Reynolds in more complex flow configurations.

Additionally, these computer vision approaches can also be used in a configuration different from the planar release; one example is the cylindrical release (Cantero et al., 2004, 2005, 2007a). The cylindrical release gravity current can be represented with an axisymmetric configuration, which is parameterized by a polar function (Cantero et al., 2007a). The cleft positions can be accurately located in both cases simply by finding the local minima of each density front line. Complementary, in different lock release configurations like the basin or channel-basin (Francisco et al., 2018), the parametrization of the contour line of the gravity current's front may even be harder. For example, Inghilesi et al. (2018) assume an elliptic function to model the front function from a particular basin configuration. Computer vision methods, such as corner detection and optical flow, do not require the parametrization of the gravity current front to any function. In these methods, the entire image is treated as a function, a brightness intensity function. Furthermore, the features are obtained by applying mathematical operations to the image function (Harris and Stephens, 1988; Shi and Tomasi, 1994; Lucas and Kanade, 1981).

5. Code availability

The numerical simulations were performed with the open-source code Incompact3d, an OS and platform independent, high-order flow solver, written in Fortran-90 available since 2006. Incompact3d's current version is available at https://github.com/xcompact3d/Incompact3d repository licensed under the GNU General Public License v3.0. Incompact3D heavily uses the 2DECOMP&FFT libraries, but dependency it not an issue as all dependencies are already at Incompact3D main development tree. It runs in parallel using the MPI (Message Passing Interface) protocol, thus a MPI environment has to be properly installed and setup. The hardware requirements to run this code will depend on the size of the problem, and adding more compute nodes decreases the computing time. The KLT prototype scripts developed for this work, namely cleftTracker and all the data sets used, are publicly available at the https://github.com/filipi/cleftTracker repository since 2020. The cleftTracker prototype, also independent of operating system and computing platform, was written in python 3 and is licensed under the MIT License. These prototype scripts depend on OpenCV and Numpy python libraries which, of course, have to be properly installed for them to run. Hardware requirements for both, Incompact3D and cleftTracker, will depend on the size of the simulation domain. For the problems presented in this work, a seventh generation intel core processor with 32 GB of RAM and 500 GB of disk space where enough for the computer vision post-processing scripts, the Incompact3D simulations were run on a computer cluster as mentioned in Section 2.3.

CRediT authorship contribution statement

F.D. Vianna: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. **B.A. Farenzena:** Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – review & editing. **M.S. Pinho:** Conceptualization, Supervision, Resources, Writing – review & editing. **J.H. Silvestrini:** Project administration, Conceptualization, Supervision, Resources, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors are grateful to Petrobras S.A., which partially funded this work (grant number 5900.0110995.19.9.), for supporting this study. Computing resources were provided by the high-performance facility LAD-PUCRS at Pontifícia Universidade Católica do Rio Grande do Sul in Porto Alegre.

References

Al-Hemoud, A., Al-Dousari, A., Misak, R., Al-Sudairawi, M., Naseeb, A., Al-Dashti, H., Al-Dousari, N., 2019. Economic impact and risk assessment of sand and dust storms (SDS) on the oil and gas industry in Kuwait. Sustainability 11 (1), 200.

Ballard, D.M., Brown, C.M., 1982. Computer Vision. Prentice-Hall, Englewood Cliffs, N.J..

Barron, J.L., Fleet, D.J., Beauchemin, S.S., 1994. Performance of optical flow techniques. Int. J. Comput. Vis. 12 (1), 43–77.

Bartholomew, P., Deskos, G., Frantz, R.A., Schuch, F.N., Lamballais, E., Laizet, S., 2020. Xcompact3D: An open-source framework for solving turbulence problems on a Cartesian mesh. SoftwareX 12, 100550.

Beauchemin, S.S., Barron, J.L., 1995. The computation of optical flow. ACM Comput. Surv. 27 (3), 433–466.

Bergen, J.R., Anandan, P., Hanna, K.J., Hingorani, R., 1992. Hierarchical model-based motion estimation. In: European Conference on Computer Vision. Springer, pp. 237–252.

Bradski, G., Kaehler, A., 2000. Opencv. Dr. Dobb's J. Softw. Tools 3.

Brown, C.M., 1988. Advances in Computer Vision, Vol. 1. Lawrence Erlbaum Associates, Hillside, N.J..

Camata, J.J., Silva, V., Valduriez, P., Mattoso, M., Coutinho, A.L., 2018. In situ visualization and data analysis for turbidity currents simulation. Comput. Geosci. 110, 23–31.

Cantero, M.I., Balachandar, S., García, M.H., 2007a. High-resolution simulations of cylindrical density currents. J. Fluid Mech. 590 (437–469), 131, 1956.

Cantero, M.I., Balachandar, S., García, M.H., Bock, D., 2008. Turbulent structures in planar gravity currents and their influence on the flow dynamics. J. Geophys. Res. Oceans 113 (C8).

Cantero, M.I., Balachandar, S., García, M.H., Ferry, J.P., 2005. Direct numerical simulations of planar and cylindrical density currents. J. Appl. Mech. 73 (6), 923–930.

Cantero, M.I., Ferry, J.P., Balachandar, S., García, M.H., 2004. Direct numerical simulations of axisymmetric density currents. Mec. Comput. 2031–2046.

Cantero, M.I., Lee, J.R., Balanchandar, S., García, M.H., 2007b. On the front velocity of gravity currents. J. Fluid Mech. 586, 1–39.

Chao, H., Gu, Y., Napolitano, M., 2013. A survey of optical flow techniques for uav navigation applications. In: 2013 International Conference on Unmanned Aircraft Systems. ICUAS, IEEE, pp. 710–716.

Chao, H., Gu, Y., Napolitano, M., 2014. A survey of optical flow techniques for robotics navigation applications. J. Intell. Robot. Syst. 73 (1), 361–372.

Dai, A., 2013. Experiments on gravity currents propagating on different bottom slopes. J. Fluid Mech. 731, 117–141.

Dai, A., Huang, Y.-L., 2022. On the merging and splitting processes in the lobe-and-cleft structure at a gravity current head. J. Fluid Mech. 930.

Dairay, T., Lamballais, E., Laizet, S., Vassilicos, J.C., 2017. Numerical dissipation vs. subgrid-scale modelling for large eddy simulation. J. Comput. Phys. 337, 252–274.

Espath, L., Pinto, L., Laizet, S., Silvestrini, J., 2014. Two- and three-dimensional direct numerical simulation of particle-laden gravity currents. Comput. Geosci. 63, 9–16.

Espath, L.F.R., Pinto, L.C., Laizet, S., Silvestrini, J.H., 2015. High-fidelity simulations of the lobe-and-cleft structures and the deposition map in particle-driven gravity currents. Phys. Fluids 27 (5), 056604.

Fabian, N., Moreland, K., Thompson, D., Bauer, A.C., Marion, P., Gevecik, B., Rasquin, M., Jansen, K.E., 2011. The paraview coprocessing library: A scalable, general purpose in situ visualization library. In: 2011 IEEE Symposium on Large Data Analysis and Visualization. IEEE, pp. 89–96.

Farenzena, B.A., 2020. Simulação numérica de correntes de densidade hiperpicnais sob referencial móvel. (Ph.D. thesis). Pontifícia Universidade Católica do Rio Grande do Sul.

Farenzena, B.A., Silvestrini, J.H., 2022. Density currents front velocity uncertainty. Comput. & Fluids 232, 105209. http://dx.doi.org/10.1016/j.compfluid.2021.105209.

Fleet, D.J., Jepson, A.D., 1990. Computation of component image velocity from local phase information. Int. J. Comput. Vis. 5 (1), 77–104.

Fortun, D., Bouthemy, P., Kervrann, C., 2015. Optical flow modeling and computation: A survey. Comput. Vis. Image Underst. 134, 1–21.

Francisco, E., Espath, L., Laizet, S., Silvestrini, J., 2018. Reynolds number and settling velocity influence for finite-release particle-laden gravity currents in a basin. Comput. Geosci. 110, 1–9.

Frantz, R.A., Deskos, G., Laizet, S., Silvestrini, J.H., 2021. High-fidelity simulations of gravity currents using a high-order finite-difference spectral vanishing viscosity approach. Comput. & Fluids 221, 104902.

Garvine, R.W., 1984. Radial spreading of buoyant, surface plumes in coastal waters. J. Geophys. Res. Oceans 89 (C2), 1989–1996.

Hacker, J., Linden, P., Dalziel, S., 1996. Mixing in lock-release gravity currents. Dyn. Atmos. Oceans 24 (1), 183–195.

Harris, C., Stephens, M., 1988. A combined corner and edge detector. In: Proceedings of the 4th Alvey Vision Conference. pp. 147–151.

Härtel, C., Carlsson, F., Thunblom, M., 2000a. Analysis and direct numerical simulation of the flow at a gravity-current head. Part 2. The lobe-and-cleft instability. J. Fluid Mech. 418, 213–229.

Härtel, C., Meiburg, E., Necker, F., 2000b. Analysis and direct numerical simulation of the flow at a gravity-current head. part 1. flow topology and front speed for slip and no-slip boundaries. J. Fluid Mech. (418), 189–212.

Heeger, D.J., 1988. Optical flow using spatiotemporal filters. Int. J. Comput. Vis. 1 (4), 279–302.

Horn, B.K., Schunck, B.G., 1981. Determining optical flow. Artificial Intelligence 17 (1–3), 185–203.

Horner-Devine, A.R., Chickadel, C.C., 2017. Lobe-cleft instability in the buoyant gravity current generated by estuarine outflow. Geophys. Res. Lett. 44 (10), 5001–5007.

Inghilesi, R., Adduce, C., Lombardi, V., Roman, F., Armenio, V., 2018. Axisymmetric three-dimensional gravity currents generated by lock exchange. J. Fluid Mech. 851, 507–544.

J. Whitlock, B., M. Legensky, S., Forsythe, J., 2016. In situ infrastructure enhancements for data extract generation. In: 54th AIAA Aerospace Sciences Meeting, SciTech, Vol. 2016. pp. 1–12.

Kaye, G., Cole, J., King, A., Johnston, D., 2009. Comparison of risk from pyroclastic density current hazards to critical infrastructure in Mammoth lakes, California, USA, from a new Inyo craters rhyolite dike eruption versus a dacitic dome eruption on Mammoth mountain. Nat. Hazards 49 (3), 541–563.

Kneller, B., McCaffrey, W., 1999. Depositional effects of flow nonuniformity and stratification within turbidity currents approaching a bounding slope; deflection, reflection, and facies variation. J. Sediment. Res. 69 (5), 980–991.

Kowalik, K., 2019. Unsplash - Avalanche at Mont Blanc. https://unsplash.com/photos/PkXThkFaGzo (Accessed: 18 Jan 2021).

Kravchenko, A., Moin, P., 1997. On the effect of numerical errors in large eddy simulations of turbulent flows. J. Comput. Phys. 131 (2), 310–322.

Lele, S.K., 1992. Compact finite difference schemes with spectral-like resolution. J. Comput. Phys. 103 (1), 16–42.

Lucas, B.D., Kanade, T., 1981. An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2. In: IJCAI'81, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 674–679.

Lucchese, L.V., Monteiro, L.R., Schettini, E.B.C., Silvestrini, J.H., 2019. Direct numerical simulations of turbidity currents with evolutive deposit method, considering topography updates during the simulation. Comput. Geosci. 133, 104306.

McElwaine, J., Patterson, M., Dalziel, S., 2004. Lobe and cleft formation at the head of a gravity current. In: Proceedings of the XXI International Congress of Theoretical and Applied Mechanics. Warsaw, pp. 15–21.

Meiburg, E., Kneller, B., 2010. Turbidity currents and their deposits. Annu. Rev. Fluid Mech. 42 (1), 135–156.

Nagel, H.-H., 1987. On the estimation of optical flow: Relations between different approaches and some new results. Artificial Intelligence 33 (3), 299–324.

Nasr-Azadani, M., Hall, B., Meiburg, E., 2013. Polydisperse turbidity currents propagating over complex topography: comparison of experimental and depth-resolved simulation results. Comput. Geosci. 53, 141–153.

Neufeld, J., 2002. Lobe-Cleft Patterns in the Leading Edge of a Gravity Current. University of Toronto, Toronto.

Nisbet, E.G., Piper, D.J.W., 1998. Giant submarine landslides. Nature 392, 329–330.

Sánchez, J., Monzón, N., Salgado De La Nuez, A., 2018. An analysis and implementation of the harris corner detector. Image Process. Line.

Shi, J., Tomasi, C., 1994. Good features to track. In: IEEE CVPR. pp. 593–600.

Shin, J., Dalziel, S., Linden, P., 2004. Gravity currents produced by lock exchange. J. Fluid Mech. 521, 1–34.

Simpson, J.E., 1972. Effects of the lower boundary on the head of a gravity current. J. Fluid Mech. 53 (4), 759–768.

Simpson, J.E., 1982. Gravity currents in the laboratory, atmosphere, and ocean. Annu. Rev. Fluid Mech. 14 (1), 213–234.

Simpson, J., 1999. Gravity currents. In: the Environment and the Laboratory. In: the Environment and the Lab, Cambridge University Press.

Stefanski, R., Sivakumar, M., 2009. Impacts of sand and dust storms on agriculture and potential agricultural applications of a SDSWS. In: IOP Conference Series: Earth and Environmental Science, Vol. 7. (1), IOP Publishing, 012016.

Suhr, J.K., 2009. Kanade-lucas-tomasi (klt) feature tracker. Comput. Vis. (EEE6503) 9–18.

Syvitski, J.P., Alexander, C.R., Field, M.E., Gardner, J.V., Orange, D.L., Yun, J.W., 1996. Continental-slope sedimentation: The view from northern California. Oceanography 9 (3), 163–167.

Tomasi, C., Kanade, T., 1991. Detection and tracking of point. Technical Report, features. Technical Report CMU-CS-91-132, Carnegie, Mellon University.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, I., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors, 2020. SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nature Methods 17, 261–272. http://dx.doi.org/10.1038/s41592-019-0686-2.

Waxman, A.M., Wu, J., Bergholm, F., 1988. Convected activation profiles and the measurement of visual motion. In: Proceedings CVPR'88: The Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, pp. 717–718.

Zakeri, A., Høeg, K., Nadim, F., 2008. Submarine debris flow impact on pipelines – part I: Experimental investigation. Coast. Eng. 55 (12), 1209–1218.