Video velocity measurement: a two-stage flow velocity prediction method based on deep learning

Xiaolong Wang\textsuperscript{a,}\textasteriskcentered, Qiang Ma\textsuperscript{b}, Genyi Wang\textsuperscript{a} and Guocheng An\textsuperscript{a}

\textsuperscript{a}Artificial Intelligence Research Institute of Shanghai Huaxun Network System Co., LTD., Chengdu 610074, China
\textsuperscript{b}China Institute of Water Resources and Hydropower Research, Beijing 610074, China

*Corresponding author. E-mail: 799468338@qq.com

ABSTRACT

Due to the uncertainty in output caused by environmental changes, significant discrepancies are expected between the surface flow velocities predicted using deep learning methods and the instantaneous flow velocities. In this paper, a two-stage deep learning flow velocity measurement algorithm is proposed. During the external calibration process, the upper and lower frames of the recorded water flow video are cyclically traversed to acquire predicted flow velocity values using the deep learning velocity measurement algorithm. Meanwhile, the pixel displacement is obtained using the sparse optical flow tracking method and then post-processed to derive the velocity calibration value and pixel calibration value. During the detection process, the deep learning-predicted flow velocity is internally calibrated using the velocity calibration value and pixel calibration value to adapt to changes in water flows. Compared with the pre-improved algorithm, the method achieves the minimum root mean square error in five different flow velocity videos and maintains high accuracy when the flow velocity changes rapidly. The obtained results are very promising and can help improve the reliability of video flow rate assessment algorithms.

Key words: deep learning, external calibration, flow velocity, sparse optical flow tracking, two-stage

HIGHLIGHTS

- New methods are needed to ensure the instability of the video-based surface velocity prediction due to environmental changes.
- Avoid direct measurement of high velocity, using the slow to fast process to indirectly get accurate high velocity.
- We provide a river data set for deep learning model training, which is convenient for researchers interested in this to expand their research.

INTRODUCTION

Hydrologic data, such as water levels, precipitation, and discharge, have been collected since ancient times. All of this field data have supported hydrological analyses, as well as the design of efficient infrastructure and management strategies (Sheng 2017). With the continued threat of climate change, the need for accurate data to document all hydrological processes, including the most extreme ones, is now more important than ever. The information generated from this data can facilitate rapid mobilization and decision-making processes, supporting related systems (Nohara \textit{et al.} 2018; Gourbesville 2020; Ma \& Gourbesville 2020). Among all the fundamental data that need to be collected, the flow and velocity of natural streams, especially under extreme conditions, have been a major research issue for hydrologists for centuries (Bao \textit{et al.} 2012). The extreme flood conditions obviously pose significant challenges for direct observation. They require comprehensive monitoring and measurement solutions that are not easily deployable in multiple locations across various catchments. Moreover, they involve a significant financial investment that may not be feasible in many situations.

With the development of sensor and image analysis technologies, non-contact flow velocity detection methods have attracted the attention of hydrologists due to their advantages of low cost, easy maintenance, and safety compared with traditional flow velocity detection methods such as acoustic waves and flow meters (Collins \& Emery 1988; Yang \textit{et al.} 2008). Subsequently, the large-scale particle image velocimetry (LSPIV) technology uses bubbles, ripples, and other information
floating on the surface of the fluid as large particles, which is widely used to replace the artificial tracer particles in particle image velocimetry. This method utilizes image processing techniques to obtain the displacement of large particles on the image, and the average velocity of the particles on the image can be obtained with a known exposure interval (Fujita & Komura 1994; Bechle et al. 2012). Then, the Spatiotemporal Image Velocimetry (STIV) solution judges the texture features of the water flow surface based on a marked velocity line without the need for tracers, and then obtains the one-dimensional velocity of the water flow surface. However, due to the interference of image noise, the measurement results are greatly deviated (Zhang et al. 2018). With the development of machine learning technology, some progress has been made in the research on the relationship between surface flow velocity and texture features by feature extraction and data dimensionality reduction of river surface images (Tauro et al. 2014). Researchers also use compressed sensing image analysis method to extract image features to establish river surface velocity estimation scheme (Wang et al. 2018). Meanwhile, a generative adversarial network flow measurement method based on conditional boundaries was proposed to address the problem that small differences between different flow velocity classes lead to classification difficulties (Wang et al. 2019). In order to utilize the motion information of water flow, the optical flow estimation method between frame differences was shown to be superior to the traditional block matching method (Chen & Mied 2013). In recent years, several scholars have also proposed optical flow estimation velocimetry methods based on tracing and deep learning (Pizarro et al. 2020; Yao 2022).

The traditional optical flow estimation algorithm (Farneback 2003) generally uses mathematical formulas to model the image, and then solves the formula coefficients based on the similarity between continuous images to calculate the approximate representation of each pixel neighborhood, to obtain the position information of each pixel in the previous frame moving to the next frame. However, it is usually affected by natural light variations that are common under natural conditions (Liu et al. 2014). In contrast, the optical flow estimation algorithm based on deep learning uses a large number of images during the training phase. This solution allows to limit the interference of illumination. The classification algorithm then combines the optical flow estimation images to predict the flow rate without the need for distance calibration of the camera (Wang et al. 2019). However, the existing deep learning flow measurement algorithm has not given had a good solution to the problem of output uncertainty. To overcome this difficulty, this paper proposes a new calibration method of predicted velocity based on the general velocity measurement architecture of optical flow model combined with a detection model. The main contributions of this paper are as follows:

(1) A flow velocity external calibration (EC) algorithm and a pixel EC algorithm are designed to minimize the error between the predicted flow velocity output from the model and the reference flow velocity (Le Coz et al. 2014), and to provide an a priori calibration reference value for the internal flow rate calibration.

(2) Internal calibration of deep learning-predicted flow rates based on the reference values provided by the EC to obtain more accurate prediction outputs.

(3) Pixel calibration values from the EC are used to calibrate the flow rate calibration values to adapt to the rapid changes in the current flow velocity.

**DEEP LEARNING FLOW VELOCITY DETECTION ALGORITHM**

**Recurrent All-pairs Field Transform (RAFT) algorithm use**

RAFT (Teed & Deng 2020) is an optical flow estimation algorithm based on deep learning. Unlike previous work (Ilg et al. 2017), RAFT maintains a high-resolution optical flow field, eliminating the structural design of estimating optical flow at low resolution. It is mainly composed of three parts, as shown in Figure 1.

**Feature extractor**

The feature extraction module is divided into two parts: feature encoder and context encoder. It consists of convolutional neural networks (CNNs) that in turn consists of six residual layers (He et al. 2016). The output feature map of each layer is halved in width and height compared to the input image, and the number of channels increases. The two consecutive images in the water flow video are used as the input of the feature extraction module. The Feature Encoder is composed of two CNNs with shared weights. The features are extracted from the two images for inner product and output to the Visual Similarity Calculator. The Context Encoder is composed of a CNN that extracts features to the Updator module for iterative update of optical flow.
Visual similarity calculator

The input is the inner product vector of two image features output by Feature Encoder, and then a four-layer pyramid feature \( \{C^0, C^1, C^2, C^3\} \) is constructed, where \( C^k \rightarrow H \times W \times H / 2^k \times W / 2^k \). When the initial state of the optical flow of the given image with respect to the \( xy \) coordinate is \( (f^1, f^2) \), each pixel \( x = (u, v) \) in Frame1 can be mapped to the corresponding point \( x' \) on Frame2, as shown in Equation (1).

\[
x' = (u + f^1(u), v + f^2(v))
\]

In order to obtain the image-rich semantic information, the neighborhood grid of \( x' \), as shown in Equation (2), where \( \eta(x', r) \) represents the set of coordinates around \( x' \) with radius less than \( r \). Combined with the initial optical flow value in Updater, the search is performed on all levels of pyramids by \( \eta(x'/2^k) \), and the lower the pyramid corresponds to the more global semantic features.

\[
\eta(x', r) = \{x' + dx | dx \in \mathbb{Z}^2, ||dx||_1 \leq r\}
\]

Updater

The updated optical flow value, the feature vector queried in the feature pyramid of Visual Similarity Calculator and the feature vector output by Context Encoder are input to predict the update of optical flow \( \nabla f \) after Gate Recurrent Unit (Cho et al. 2014) and two convolution layers to predict the update of the optical flow, as shown in Equation (3). The update operator estimates a series of optical flow \( \{f_1, \cdots, f_{k+1}\} \) starting from the initial point \( f_0 = 0 \), and outputs the last updated optical flow result after reaching the set number of iterations, and after processing, the RGB optical flow map of the corresponding water flow image is obtained.

\[
f_{k+1} = \nabla f + f_{k+1}
\]

EfficientNet algorithm

EfficientNet is a very efficient convolutional neural network for extracting image features (Tan & Le 2019). The structure is shown in Figure 2 is composed of MBConv structure, and MBConv is composed of Inverted residuals structure and forward
residual structure. In Figure 2, the number of convolution cores of each structural block is shown in the red font. The input to predict the corresponding flow value is the RGB optical flow image output by the optical flow network, and after a series of MBConv structures and serial convolution blocks, the feature map channel dimension is reduced, as shown in the green area. Finally, the image features are fused into one pixel value through the fully connected layer, which is used as the numerical output of the prediction flow to complete the deep learning flow rate prediction algorithm.

**TWO-STAGE FLOW VELOCITY PREDICTION METHOD**

The proposed approach divides the velocity detection algorithm into two parts: EC process and detection process, as shown in Figure 3. In particular, the initial values of the two parts of the flow velocity prediction are obtained by the deep learning algorithm in Section 2. The EC algorithm is added before the detection process. The purpose of the EC algorithm is to provide...
a prior for the detection process part that is convenient to guide the calibration of the output flow velocity value. We then designed an internal calibration algorithm within the detection algorithm to allow the flow velocity calibration value from the EC and the pixel calibration value to adjust the predicted value to make the result more accurate.

**EC process**

Inside the EC process, the treatment is divided into flow velocity calibration part and pixel calibration part. The flow velocity calibration part is shown in the EC process of Figure 3, including the cyclic traversal of the upper and lower frames of the input image for optical flow estimation and then the flow velocity detection to obtain the predicted value list \( V = \{ V_1, V_2, \ldots, V_i \} \). The pixel calibration part first performs corner detection on the first frame to obtain the corner set \( C_i \), and then uses the LK sparse optical flow algorithm (Barron et al. 1992) to track the detected corners, as shown in Equation (4). \( F_i \) and \( F_{i+1} \) represent the previous frame and the current frame, and the tracked corner set \( C_{i+1} \) is obtained.

The state quantity \( ST \) selects the corner set \( C_i \) of the tracking \( F_i \) corresponding to \( F_{i+1} \) and obtains the corner set \( C_{i+1}^{new} \) of the previous frame and the corner set \( C_{i+1}^{new} \) of the current frame, as shown in formula (5).

\[
C_{i+1}^{new} = \begin{cases} C_{i+1} & \text{if } ST = 1 \\ C_i & \text{if } ST = 1 \\ C_{i+1} & \text{otherwise} \end{cases}
\]  

(4)

(5)

where \( F_i \) and \( F_{i+1} \) denote the previous frame and the current frame to get the set of corner points tracked \( C_{i+1} \). The state quantity \( ST \) filters out the set of corner points \( C_i \) of \( F_{i+1} \) corresponding to \( F_i \) to get the set of corner points \( C_{i+1}^{new} \) of the previous frame and the set of corner points \( C_{i+1}^{new} \) of the current frame after filtering, as shown in Equation (5).

The pixel distances between \( C_{i+1}^{new} \) and \( C_{i+1}^{new} \) are calculated, the pixel distances are sorted from largest to smallest, and the distance threshold is set and then filtered once more to finally obtain \( topk \) pixel distances. The average pixel shift of the current water flow image is obtained by averaging the pixel distances. After looping through the input image, the pixel displacement list \( \text{Pix} = \{ \text{Pix}_1, \text{Pix}_2, \ldots, \text{Pix}_t \} \) is obtained, as shown in Equation (6).

\[
\text{Pix}_i = M \left( \sum_{n=1}^{\lambda} \text{dis}_{topk}(C_{i+1}^{new}, C_{i+1}^{new}) \right)
\]  

(6)

where \( \text{dis} \) represents the Euclidean distance, and \( \lambda \) is generally between 5 and 10, representing the number of corners. \( M \) is the average operation.

After obtaining the aperiodic sequence \( V \) and \( \text{Pix} \), the post-processing part of the EC process is initiated. The number of points \( m \) of the sub-multipoint average and the moving interval \( \Delta \) are set to perform the \( m \)-point moving average that is generally set to about one-fifth of the number of frames \( n \). The interval \( \Delta \) is set to about one-half of the multipoint average \( m \), as shown in Equations (7) and (8), to obtain \( F_i \) the corresponding video stream rate candidate calibration value \( V' \) and pixel candidate calibration value \( \text{Pix}' \).

\[
\bar{V}, \text{Pix} = \frac{1}{m} \left( \sum_{t=\Delta t}^{\Delta t+m-1} (V_t, \text{Pix}_t) \right)
\]  

(7)

\[
\bar{V}', \text{Pix}' = \frac{1}{n-m} \left( \sum_{t=\Delta}^{n-m} (V_t, \text{Pix}_t) \right)
\]  

(8)

where \( t = 0, 1, \ldots, n-m \), represents the number of moving averages. In Equation (7), \( \Delta t \) represents the number of moving intervals of the \( t \)th time, and \( \bar{V}_t \) represents the average of the \( t \)th time. In Equation (8), \( V' \) and \( \text{Pix}' \) represent the velocity candidate calibration value and the pixel calibration value of \( F_i \) corresponding to the flow video, respectively.
The predicted velocity value list \( V \) and the pixel displacement list \( \text{Pix} \) from smallest to largest is produced and the median is taken out as shown in Equation (9).

\[
V^*, \text{Pix}^* = \text{sort}(V, \text{Pix})[i//2 − 1]
\]  

(9)

where \( \text{sort}(V, \text{Pix}) \) means the velocity sequence \( [V, \text{Pix}] \) sorted from smallest to largest, \( [i//2 − 1] \) means taking out the median, \( V^* \) and \( \text{Pix}^* \) are the flow velocity second candidate calibration values and pixel second candidate calibration values.

The minimum values in \( [V^*, \text{Pix}^*] \) and \( [V^{''}, \text{Pix}^{''}] \) are used as the final flow velocity calibration value \( V_c \) and pixel calibration value \( \text{Pix}_c \) of the water flow video corresponding to \( F_i \), as shown in Equation (10) that is used for the internal calibration part of the detection process.

\[
V_c, \text{Pix}_c = \min([V^*, \text{Pix}^*], [V^{''}, \text{Pix}^{''}])
\]  

(10)

**Detection process**

Inside the detection process, the protocol is divided into two parts: detection and internal calibration. The detection part reads the water flow video according to the upper and lower frames. After the optical flow estimation network mentioned in Figure 1 and the multi-classification network illustrated in Figure 2, the initial predicted flow velocity value is obtained and used for the internal calibration part. The internal calibration part is then divided into flow velocity internal calibration and pixel internal calibration.

**Internal calibration of flow velocity**

The number of anomalies \( \alpha \) and the list of anomalous elements \( \beta \) are initialized to null values. The internal calibration is then performed as shown in Figure 4. When the first detection or the number of exceptions is equal to 0, the predicted value \( V_{\text{pre}} \) of the video frame is assessed to know if it is near to its calibration value \( V_c \), as shown in Equation (11).

\[
\begin{align*}
\text{Abnormal, } (V_{\text{pre}} - V_c)/V_c & \geq \delta \\
\text{Normal, } (V_{\text{pre}} - V_c)/V_c & < \delta
\end{align*}
\]  

(11)

where \( \delta \) represents the abnormal probability value that is used to determine whether \( V_{\text{pre}} \) deviates from \( V_c \) too much. If \( (V_{\text{pre}} - V_c)/V_c \) is lower than \( \delta \), it means that the fluctuation of \( V_{\text{pre}} \) is within the normal range, and no abnormal processing
is performed in the next step. If the probability is greater than δ, it means that the predicted value \( V_{\text{pre}} \) is abnormal and needs to be calibrated. The calibration formula is shown in Equation (12).

\[
V_{\text{pre}} = V_{\text{pre}} \times j - s(\beta) + j - \alpha
\]

where \( s(\beta) \) represents the sum of the abnormal list \( \beta \), \( j \) is the number of image frame traversal. After the calibration of the predicted flow velocity value by the flow velocity calibration, the value is calibrated by the pixel calibration value.

**Intra-pixel calibration**

Repeat the corner detection and optical flow tracking process in Section 3.1, add the pixel displacement \( \text{Pix}'_\beta \) between the current frame and the previous frame to the pixel list \( \text{Pix}' \), and take the length of the abnormal element list \( \beta \) in Section 3.2.2 as the length of \( \text{Pix}' \). The average value of pixel displacement \( \text{Pix}'_\beta \) obtained by combining Equations (7) and (8) is compared with the pixel calibration value \( V_c \), and the updated flow velocity calibration value is obtained by combining the flow velocity EC value, as shown in Equation (13).

\[
V_c \rightarrow \frac{\text{Pix}'_\beta \times V_c}{\text{Pix}_c}
\]

**FIELD DATA AND RESULTS ANALYSIS**

**Detection process**

Several video data sets were collected from different rivers for the model training. They represent various flow conditions such as turbulence, reflection, calmness, and dark light. The five video clips have been recorded with a resolution of 1,920 × 1,080 pixels, and two velocity measurement regions are selected for each clip, a total of 10 speed measurement areas, as shown in Figure 5. The average velocity values between two adjacent frames on the river surface of each measurement region were obtained from the open-source software Fudaa-LSPIV as a reference flow velocity value for the video frames.

In order to simulate the river scene in complex environment, the speed measurement area is enhanced by affine transformation and color transformation data. Affine transformation is rotation, translation, and scaling operations, as follows:

\[
\text{Tran} = A \cdot \begin{bmatrix} x \\ y \end{bmatrix} + B
\]

\[
A = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix}
\]

\[
B = \begin{bmatrix} b_{00} \\ b_{10} \end{bmatrix}
\]

**Figure 5** | Display maps of 10 velocity measurement area for the RAFT model training.
where \((x, y)\) is a pixel coordinate of the image of the speed measurement area, \(A\) includes the rotation and scaling operation of the image, \(B\) includes the translation operation of the image, and the image of the speed measurement area after affine transformation needs to undergo color transformation to simulate the real shooting environment. The color transformation used in the approach includes the Gaussian blur and contrast change analysis:

1. **Gaussian blur**: Gaussian kernel function is used to convolve the image, and the blur degree of the image can be controlled by changing the size of the Gaussian kernel. For the water flow image taken by the camera, different Gaussian kernels can be used to simulate the camera at different focal lengths, as follows:

\[
I(x, y) = A \exp \left(-\left(\frac{(x - x_0)^2}{2\sigma_x^2} + \frac{(y - y_0)^2}{2\sigma_y^2}\right)\right)
\]

where \(A\) is the magnitude. \(x_0, y_0\) is the center point coordinate, \(\sigma_x\) and \(\sigma_y\) is the variance.

2. **Contrast change**: Under the condition of keeping the average brightness unchanged, by changing the difference of different brightness levels between the brightest white and the darkest black in the light and shade area of the image, the light and shade changes of the light in the real world are simulated, as follows:

\[
I'(x, y) = (I(x, y) - \text{Avg}) \cdot k + P(x, y)
\]

where \(k\) belongs to \([-1, 1]\), \(\text{Avg}\) is the average brightness of the image, and \(I\) is the original value of the pixel.

After data enhancement, 10 test areas with the same resolution are used to perform two consecutive frames of optical flow calculation with RAFT optical flow network. The two-dimensional optical flow map represents the motion information of each pixel, and it is converted into RGB map for convolutional neural network training test. For the test, 10,024 water flow images with motion information generated by dense optical flow are combined with corresponding velocity labels to form an experimental data set. The RGB optical flow image corresponding to Figure 5 is shown in Figure 6. Due to the difference in texture features, the generated optical flow image is also different from its corresponding velocity label.

**Hardware configuration**

The experimental hardware configuration is shown in Table 1.

**Experiment setting**

The regression model (Tan et al. 2020) is selected as the experimental benchmark model. The training epoch is set to 300, the training batch is 32, the first three epochs are used for warming up the model, the initial learning rate is 0.01, the input size of the training and test images is set to \(640 \times 640\) pixels, and the optimizer is AdamW (Loshchilov & Hutter 2017). The experimental hyperparameter settings for adding some are shown in Table 2.

**Figure 6** | Display maps of 10 velocity measurement area optical flow.
Evaluation metrics

In the approach, the root mean square error (RMSE) is used to measure the error between the predicted flow velocity values and the reference flow velocity labels, as follows.

\[
\text{RMSE}(V_{\text{pre}}, V_{\text{label}}) = \sqrt{\frac{1}{m-1} \sum_{i=1}^{m-1} (V^i_{\text{pre}} - V^i_{\text{label}})^2}
\]  

(19)

where \( m - 1 \) represents the number of test RGB optical flow images, \( m \) represents the number of test images, \( V^i_{\text{pre}} \) represents the predicted flow velocity value of the current \( i \)th RGB optical flow image, and \( V^i_{\text{label}} \) represents the actual flow velocity labels of the current \( i \)th RGB optical flow image.

Experiment

Experimental comparison of external alignment algorithms

The water flow video recorded at Beichuan mountain torrent (Sichuan, China) is used as the test set to assess the performance of the algorithm. The maximum width of the river is 3.2 m, the average water depth is about 0.46 m, and the slope is 3%. The pixel outer calibration performs the corner point detection on the first frame to obtain the corners of each velocity measurement region. In order to ensure that the number of corners is not too small, the approach compares a variety of corner detection algorithms (Shi & Tomasi 2002; Rosten et al. 2009, 2011) in the calmer water corner detection effect, as shown in the Figure 8.

In the case of setting the same corner threshold, the Oriented Fast and Rotated Brief (ORB) algorithm detects more corners than other corner detection algorithms, and the corners can reflect more texture features. The corner point of pixel (80, 236) in Figure 7(c) reflects the texture features of the image, but (a), (b) are not detected. After comparison, the method proposed in this paper uses ORB algorithm to detect corner points of pixel displacement.

When traversing the upper and lower frame images each time of the water flow video, the predicted flow velocity output by the EC part is stored in the flow velocity list. The out-of-pixel calibration part will perform the sparse optical flow tracking based on the corner points detected in the first frame and then store the corner pixel displacement obtained each time into the displacement list. As shown in the Figure 8(a), two velocity measurement areas in the flood time flow video are selected for out-of-pixel calibration. The number of detected corner points is 4 and 5, respectively. The trajectory obtained by sparse optical flow tracking of the corner points is shown in the Figure 8(b). The sparse optical flow tracking algorithm can judge the direction of water flow and assess the pixel displacement of water flow.

In the actual scene, the displacement of each position on the surface of the river water body is different. The above detection and tracking algorithm obtains the displacement of the key feature points on the river surface, which is used to reflect the

Table 1 | Hardware configuration

<table>
<thead>
<tr>
<th>GPU</th>
<th>CPU</th>
<th>Operating system</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA</td>
<td>Intel(R) Xeon(R) Gold 6,132 CPU</td>
<td>Linux</td>
</tr>
<tr>
<td>Tesla M60/8GB</td>
<td>2.60 GHz/64GB</td>
<td>3.10.0–1160.49.1</td>
</tr>
</tbody>
</table>

Table 2 | Hyperparameter settings

<table>
<thead>
<tr>
<th>Hyperparameters</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch division</td>
<td>320</td>
</tr>
<tr>
<td>Momentum</td>
<td>0.937</td>
</tr>
<tr>
<td>Weight decay</td>
<td>0.0005</td>
</tr>
<tr>
<td>Saturation</td>
<td>1.5</td>
</tr>
<tr>
<td>Exposure</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Evaluation metrics

In the approach, the root mean square error (RMSE) is used to measure the error between the predicted flow velocity values and the reference flow velocity labels, as follows.

\[
\text{RMSE}(V_{\text{pre}}, V_{\text{label}}) = \sqrt{\frac{1}{m-1} \sum_{i=1}^{m-1} (V^i_{\text{pre}} - V^i_{\text{label}})^2}
\]  

(19)

where \( m - 1 \) represents the number of test RGB optical flow images, \( m \) represents the number of test images, \( V^i_{\text{pre}} \) represents the predicted flow velocity value of the current \( i \)th RGB optical flow image, and \( V^i_{\text{label}} \) represents the actual flow velocity labels of the current \( i \)th RGB optical flow image.

Experiment

Experimental comparison of external alignment algorithms

The water flow video recorded at Beichuan mountain torrent (Sichuan, China) is used as the test set to assess the performance of the algorithm. The maximum width of the river is 3.2 m, the average water depth is about 0.46 m, and the slope is 3%. The pixel outer calibration performs the corner point detection on the first frame to obtain the corners of each velocity measurement region. In order to ensure that the number of corners is not too small, the approach compares a variety of corner detection algorithms (Shi & Tomasi 2002; Rosten et al. 2009, 2011) in the calmer water corner detection effect, as shown in the Figure 8.

In the case of setting the same corner threshold, the Oriented Fast and Rotated Brief (ORB) algorithm detects more corners than other corner detection algorithms, and the corners can reflect more texture features. The corner point of pixel (80, 236) in Figure 7(c) reflects the texture features of the image, but (a), (b) are not detected. After comparison, the method proposed in this paper uses ORB algorithm to detect corner points of pixel displacement.

When traversing the upper and lower frame images each time of the water flow video, the predicted flow velocity output by the EC part is stored in the flow velocity list. The out-of-pixel calibration part will perform the sparse optical flow tracking based on the corner points detected in the first frame and then store the corner pixel displacement obtained each time into the displacement list. As shown in the Figure 8(a), two velocity measurement areas in the flood time flow video are selected for out-of-pixel calibration. The number of detected corner points is 4 and 5, respectively. The trajectory obtained by sparse optical flow tracking of the corner points is shown in the Figure 8(b). The sparse optical flow tracking algorithm can judge the direction of water flow and assess the pixel displacement of water flow.

In the actual scene, the displacement of each position on the surface of the river water body is different. The above detection and tracking algorithm obtains the displacement of the key feature points on the river surface, which is used to reflect the
displacement of each position on the actual river surface, that is, the overall displacement of the river surface. After traversing multiple video frames, the obtained displacement list obtains the calibration pixel value in the EC through the calculation formula in Section 3.1.

It is important to test the effect of the number of traversals of the EC algorithm on the output calibrated flow velocity values and the pixel calibration values of the EC algorithm. Figure 9 shows the experimental results for a total of five flow measurement areas in two rivers with low- and high-flow velocity. The video duration is 10 min. On those graphs, the solid line represents the actual flow velocity values measured by the velocimeter, and different colors show the different flow measurement regions, and the dashed line indicates the output results when traversing the image up to 25 frames. When the traversal

Figure 7 | Comparison of corner detection algorithms.

Figure 8 | Sparse optical flow tracking in two velocity measurement regions: (a) corner of initial detection and (b) corner trajectory tracked after 10 frames.
number is around 25 frames, the calibrated flow velocity value reaches the nearest inflection point to the actual flow velocity value in the five velocity measurement regions, and the calibrated flow velocity value after the inflection point fluctuates around the inflection point. In particular, as shown in Figure 9(b) for the high-flow velocity region, the calibrated flow velocity fluctuates more in the interval of the number of calibration frames from 10 to 20, but the amplitude of fluctuation decreases with the increase of the number of calibrations. The experimental results of the pixel EC are similar to the experimental results of the velocity EC, and its pixel calibration also reaches stability around 25 frames. So the traversal number of the EC algorithm is set to 25 frames. The calibration flow velocity values and pixel calibration values of the five velocity measurement regions were obtained by the EC algorithm and then used in the internal calibration algorithm for calibration.

Table 3 shows the flow velocity calibration values and pixel calibration values of the five measurement areas when the EC algorithm traverses 10, 25 and 50 frames, the RMSE between the predicted flow velocity value and the corresponding actual value when traversing 50 frames without the EC algorithm. In Figure 10, the velocity measurement regions 1–5 correspond sequentially to the five line segments in Figure 9, where the reference flow velocity values range from 0.1 to 0.8 m/s in low to high intervals. According to Table 3, when traversing 10 frames, the RMSE of the flow velocity calibration value and the pixel calibration value in the five measurement areas are mostly greater than 0.1, and the errors between the flow calibration values and the reference values are larger. When the traversal reaches 25 frames, the historical values are gradually averaged according to the post-processing, which is closer to the true value of the measurement, and the RMSE is below 0.07. The RMSE of the flow velocity calibration value of the velocity measurement area 2 and the pixel calibration value of the velocity measurement area 1 reached the minimum at 50 frames, but the calibration time was too long. However, when the EC is not used, the RMSE corresponding to the traversal of 50 frames reaches the maximum in the five velocity measurement regions. In terms of comprehensive performance, the EC traversal of 25 frames provides the best performance.

![Figure 9](https://example.com/figure9.png)

**Figure 9** | The influence of image frame traversal times on EC parameters: (a) changes of the calibration parameters at low flow rates and (b) changes of the calibration parameters at high flow rates.
Table 3 | The influence of the traversal times of the EC algorithm on the calibration parameter RMSE

<table>
<thead>
<tr>
<th>The velocity measurement area</th>
<th>Error RMSE (10 frame m/s)</th>
<th>Error RMSE (10 frame pix)</th>
<th>Error RMSE (25 frame m/s)</th>
<th>Error RMSE (25 frame pix)</th>
<th>Error RMSE (50 frame m/s)</th>
<th>Error RMSE (50 frame pix)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement area 1</td>
<td>0.1351</td>
<td>0.1118</td>
<td>0.0535</td>
<td>0.0621</td>
<td>0.0602</td>
<td>0.0593</td>
</tr>
<tr>
<td>Measurement area 2</td>
<td>0.1203</td>
<td>0.0753</td>
<td>0.0619</td>
<td>0.0475</td>
<td>0.0576</td>
<td>0.1033</td>
</tr>
<tr>
<td>Measurement area 3</td>
<td>0.1146</td>
<td>0.0935</td>
<td>0.0514</td>
<td>0.0693</td>
<td>0.0741</td>
<td>0.0724</td>
</tr>
<tr>
<td>Measurement area 4</td>
<td>0.1609</td>
<td>0.1590</td>
<td>0.0504</td>
<td>0.0468</td>
<td>0.0536</td>
<td>0.0492</td>
</tr>
<tr>
<td>Measurement area 5</td>
<td>0.1832</td>
<td>0.1647</td>
<td>0.0621</td>
<td>0.0515</td>
<td>0.0679</td>
<td>0.0524</td>
</tr>
</tbody>
</table>

Figure 10 | The influence of abnormal probability factor and abnormal constant on the change of flow velocity: (a) influence of different values of $\beta$ and (b) influence of different values of $\delta$. 
Experimental comparison of internal calibration algorithms

The internal calibration algorithm allows to calibrate the predicted output flow velocity value according to the calibration parameter output by the EC algorithm. The internal calibration algorithm has influence on the prediction results and should be analyzed. When the flow velocity changes drastically, the internal calibration algorithm needs to compare the predicted flow velocity value with the calibration flow velocity value according to the current frame. The Figure 10 shows the influence of the value of the abnormal probability factor and the abnormal constant on the internal calibration results.

Taking velocity measurement area 1 as an example, the change of flow velocity is simulated by frame extraction, and the red line segment in Figure 11(a) represents the actual flow velocity. During the test, no frame extraction is performed within 10 frames, and the internal calibration algorithm is loaded with an abnormal element list to facilitate subsequent calibration. The flow velocity reaches the maximum at 140 frames. The weakly calibrated black line segment (δ = 0.3, β = 20) cannot track the change of the measured flow velocity in time and generates a large error with the reference measurement. However, the strongly calibrated green line segment can track the rapid change of the flow velocity in time by continuously calibrating the flow velocity that is closest to the instant value. With the decrease of flow velocity, the influence of parameters β is obviously weaker, and the error between strong calibration and weak calibration is relatively small. When the flow rate continues to increase rapidly (after 300 frames), the error increases significantly again compared to the reference measurement, and the influence of parameters β is enhanced. Appropriately reducing the value of the parameter β to increase the frequency of calibration within the algorithm will make the prediction result more accurate, which is generally set to 8.

The test environment of the sensitivity of the parameter δ is the same as that of the parameter β, and the error analysis is performed with reference measurements. As shown in Figure 11(b), the flow velocity reaches the maximum value at 140 frames. When the threshold δ is adjusted to 0.6 (δ = 0.6, β = 8), the error of the flow velocity curve reaches the maximum value relative to the curve of other thresholds, but it is smaller than the maximum error of the parameter β (δ = 0.3, β = 20). This is because the parameter δ is only responsible for judging the abnormal flow velocity in the algorithm stage. In fact, the calibration is caused by the parameter β, so the parameter δ has a weak effect on the flow velocity. Similarly, the relative error is small when the flow velocity decreases, and the error increases when the value increases rapidly. The value of δ parameter can be appropriately reduced to lower the threshold for screening abnormal flow velocity, and then the prediction results become more accurate by increasing the frequency of calibration. In summary, the parameter setting of δ = 0.3, β = 8 will make the prediction results more accurate.

The RMSE corresponding to the test flow velocity of β and δ parameter is shown in Table 4. The tests data of 450 frames are used after frame extraction to verify. It can be seen that the parameter setting of δ = 0.3, β = 8 minimizes the RMSE (represented by bold), and the parameter setting of δ = 0.3, β = 20 maximizes the error to 0.2134. Therefore, it can be

Figure 11 | Test results of intra-pixel calibration when the flow velocity changes.
concluded that with the increase of $\beta$ and $\delta$ parameters, the error of the predicted output velocity value will also increase. This can be explained by the unsteady flow conditions in river. Consequently, a high frequency calibration process is needed.

Based on the output prediction flow velocity results with $\delta = 0.3$, $\beta = 8$ parameter settings, the number of interval frames for pixel calibration needs to correspond to the anomaly constant. Here, intra-pixel calibration is performed every eight frames. The test results of the intra-pixel calibration are shown in the Figure 11. The initial calibration value after EC is shown in the blue point in the lower left corner, and then the pixel calibration at eight frame intervals is continuously performed to obtain the updated calibration value, as shown in the orange point. The flow velocity calibration value after pixel calibration is closer to the real measurement value. When the flow velocity changes rapidly within the river, the pixel calibration process ensures that the flow velocity calibration value can synchronize with the change of the instant value, and also lays a foundation for the accuracy of the predicted output.

In the internal calibration, the pixel calibration values and the corresponding real flow velocity values of seven different frames are selected as shown in the second and third lines of Table 5. Among them, the 10th frame represents the pixel calibration value of the EC output of 5.218. The last line is the internal calibration flow velocity calculated by Equation (13) according to the internal calibration pixel value. The visible error is small and can represent the relationship between the internal calibration pixel value and the internal calibration flow velocity value obtained in the internal calibration algorithm.

The relationship between the internal calibration pixel values and the corresponding true flow velocity values is shown in the blue dots in Figure 12. The line segment represented by Equation (13) is shown in the orange line. The line fitted between the internal calibration pixel value and the true flow velocity value is close to the orange line, indicating that Equation 13 can be used to update the internal calibration value to make the predicted more accurate.

After traversing the first 50 frames of the five velocity measurement areas to obtain the corresponding calibrated flow velocity values by the EC algorithm, the flow velocity prediction is performed in real time for the images after 50 frames of the corresponding velocity measurement areas in the detection process, with the experimental parameters set to $\delta = 0.3$, $\beta = 8$.

Figure 13 shows the comparison between the EC flow velocity, the calibration flow velocity, and the deep learning prediction flow velocity for five velocity measurement areas with a duration of 200 s, with a flow rate sampling interval of 4 s and a total of 50 flow rate sampling points. The original deep learning algorithm predictions fluctuate widely, have poor accuracy, and are in error with the external calibrated flow velocity. The calibrated flow velocity after internal calibration fluctuates within the range of EC flow velocity, with a small amplitude and high accuracy. In disaster warning, it is necessary to maintain stable flow velocity values to prevent transient flow disturbances (Yang 2021).

### Comparison of different video velocimetry techniques

Table 6 compares the RMSE of this method with other commonly used video velocimetry techniques, including Optical tracking velocimetry (OTV) (Tauro et al. 2018) and conditional boundary equilibrium generative adversarial network (CBEGAN) (Wang et al. 2019). OTV uses ORB combined with sparse optical flow to calculate the pixel displacement in

<table>
<thead>
<tr>
<th>Frame</th>
<th>10 (EC)</th>
<th>50</th>
<th>90</th>
<th>140</th>
<th>210</th>
<th>300</th>
<th>370</th>
<th>450</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real flow velocity (m/s)</td>
<td>0.108</td>
<td>0.374</td>
<td>0.553</td>
<td>0.656</td>
<td>0.571</td>
<td>0.387</td>
<td>0.486</td>
<td>0.865</td>
</tr>
<tr>
<td>Flow velocity (m/s)</td>
<td>0.114</td>
<td>0.351</td>
<td>0.526</td>
<td>0.631</td>
<td>0.560</td>
<td>0.399</td>
<td>0.505</td>
<td>0.880</td>
</tr>
</tbody>
</table>

---

**Table 4 | RMSE of $\delta$ and $\beta$ parameters test flow velocity**

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>0.3</th>
<th>0.3</th>
<th>0.3</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta$</td>
<td>8</td>
<td>12</td>
<td>16</td>
<td>20</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>RMSE</td>
<td>0.0684</td>
<td>0.0875</td>
<td>0.1222</td>
<td>0.2134</td>
<td>0.0831</td>
<td>0.1024</td>
<td>0.1545</td>
</tr>
</tbody>
</table>
the same region, and the predicted flow velocity value is obtained according to the mapping relationship between the actual distance and the image distance (Hu et al. 2021). The classification model in CBEGAN adopts the EfficientNet used in this paper, and the training data are the same.

Compared with the other two algorithms, the proposed method achieves the smallest RMSE in the five velocity measurement regions. The highest RMSE of OTA may be due to the interference of video pixels and environment. The corner points

Figure 12 | Relationship between the internal calibration pixel values and the corresponding true values.

Figure 13 | Comparison of the predicted values of the flow velocity under different flow measurement regions: (a) low-flow velocity regions and (b) high-flow velocity regions.
detected by OTA algorithm are missing as shown in Figure 7, which leads to inaccurate pixel displacement and affects the stability of the predicted flow velocity value. When the velocity is measured in the high-flow velocity region, the RMSE for reaches a maximum of 0.385, which may be due to the influence of the breakpoint problem during the sparse optical flow tracking process (Liuxia et al. 2019). CBEGAN directly uses the classification model to label and classify the flow velocity value. Because the training data set is difficult to cover all the environmental impacts, it will also cause inaccurate output results. In the framework of internal and EC in this paper, the optical flow estimation method improves the prediction accuracy of the classification model, and then uses the pixel displacement to correct the model to predict the flow velocity value, which is equivalent to the fusion of the above two algorithms, so that their respective advantages complement each other, and RMSE is minimized. At the same time, the EC stage of this method uses the velocity value of adjacent time periods to be equivalent to the velocity pretreatment, which can also be used to reduce RMSE in the other two methods. And, through the combined analysis of Table 6 and Figure 13, our method used in the high-flow river area also has high accuracy and stability.

CONCLUSIONS

In the proposed approach, a two-stage velocity measurement algorithm based on depth learning method is implemented. In the EC process, the flow velocity calibration value and pixel calibration value are obtained by preprocessing the output results of the two algorithms in a period of time. Then the predicted flow velocity output by the deep learning algorithm is internally calibrated to obtain the calibrated predicted flow velocity in the flow velocity detection stage. When the flow velocity changes (rain, confluence), the pixel calibration value can be used to predict the flow velocity update. The experimental results show that the flow velocity calibration values provided by the EC can reflect the instant flow velocity in the corresponding velocity measurement area. The subsequent detected flow velocities can also be based on the accurate predicted flow velocities around the flow velocity calibration values, and the internal calibration algorithm can output the corresponding more accurate changing flow velocity values when the flow velocity changes drastically. Those preliminary results are very promising. While maintaining accuracy, our method has an advantage over traditional methods in that it does not require knowledge of the internal and external parameters of the camera, thus reducing additional computational costs. Furthermore, any deep learning optical flow algorithm can be integrated into our two-stage framework.

Obviously and even if the flow conditions covered by the different recorded videos are covering a wide range of hydraulic situations, additional tests should be performed with additional examples collected on the field and especially for fast transient conditions that are observe during extreme hydrological events. The possibility to produce efficient and reliable flow velocity monitoring solutions based on video recording is a major target and could significantly contribute to assess extreme conditions and provide warning signals to exposed populations.

DATA AVAILABILITY STATEMENT

All relevant data are available from an online repository or repositories.

CONFLICT OF INTEREST

The authors declare there is no conflict.

Table 6 | Comparison of different video velocimetry techniques

<table>
<thead>
<tr>
<th>The velocity measurement area</th>
<th>RMSE</th>
<th>OTA</th>
<th>CBEGAN</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement area 1</td>
<td>0.117</td>
<td>0.085</td>
<td>0.0547</td>
<td></td>
</tr>
<tr>
<td>Measurement area 2</td>
<td>0.103</td>
<td>0.076</td>
<td>0.0413</td>
<td></td>
</tr>
<tr>
<td>Measurement area 3</td>
<td>0.095</td>
<td>0.092</td>
<td>0.0651</td>
<td></td>
</tr>
<tr>
<td>Measurement area 4</td>
<td>0.214</td>
<td>0.152</td>
<td>0.091</td>
<td></td>
</tr>
<tr>
<td>Measurement area 5</td>
<td>0.385</td>
<td>0.197</td>
<td>0.099</td>
<td></td>
</tr>
</tbody>
</table>
REFERENCES


Nohara, D., Gourbesville, P. & Ma, Q. 2018 Towards development of effective decision support systems for integrated water resources planning and management. *Disaster Prevention Research Institute Annuals* 61 (B), 702–710.


Yang, G. 2021 Study on River Flow Measurement Based on CNN and Image Processing. Shandong University, Shandong.
Yao, Q. 2022 Research on River Surface Velocimetry Based on Optical Flow Estimation. Zhejiang University, Zhejiang.

First received 16 August 2023; accepted in revised form 15 April 2024. Available online 26 April 2024