FlumeNet: A neural network model for generating videos of flume experiments

Júlio Hoffimann Mendes (juliohm@stanford.edu)

CS230 conference, Stanford, CA, March 2018

Category: Physical Sciences

Abstract

In geomorphology research, flume experiments are used to study patterns of landscape evolution, and to understand the physical processes by which these patterns are created on the surface of the Earth. Understanding these processes is important for assessing risk of environmental disasters (e.g. floods in urban areas) and for modeling natural resources such as oil & gas and groundwater. Although various numerical models were proposed in the literature for approximating flow and sediment transport captured in flume experiment videos, these models often show limited resemblance to the records and/or are quite expensive to calibrate and run. In this work, neural network models are proposed for generating new videos of the flume, which are laborious to obtain otherwise, but that are important for geomodeling and uncertainty quantification studies.

Background

- Surface processes are constantly reworking the landscape of our planet with perhaps the most diverse patterns of sediment displacement known to humanity. Capturing this diversity is important for advancing our knowledge of systems [Murray et al. 2009], and for sustainable exploitation of natural resources such as oil & gas and groundwater.
- From a modeler's perspective, great diversity comes with great uncertainty. Modeling this uncertainty explicitly is crucial for assessing the risk of floods in urban areas, and for estimating reserves of fluids stored in the ground, among other reasons.
- In order to model uncertainty, statistical methods require hundreds or thousands of observations, yet it is only recently that the geomorphology community started to collect high-resolution image data from flume experiments.

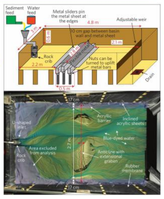

Fig. 1: Setup of the experiment (top) and overhead shot of the tank during flow and transport (bottom). Images by Bufe et al. 2016

Research Objective

Develop neural network models for generating new videos of flume experiments, and assess their performance in terms of return level and autocorrelation statistics.

Methodology

- Seven high-resolution videos of flume experiments recorded under different boundary conditions (e.g. uplift rate, sediment discharge) are investigated in this work [Bufe et al. 2016]. The frames are cropped to include only the sandbox, and the resulting images are upscaled to a more manageable resolution with 150x100 pixels.

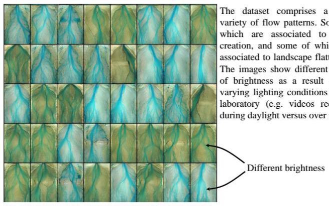

Fig. 2: Randomly selected frames from different experiments after cropping and upscaling to 150x100 pixels.

The dataset comprises a large variety of flow patterns. Some of which are associated to relief creation, and some of which are associated to landscape flattening. The images show different levels of brightness as a result of the varying lighting conditions in the laboratory (e.g. videos recorded during daylight versus over night).

Methodology (contd.)

- The now cropped and upscaled video frames extracted at a rate of 0.5 frames per minute are further organized into separate folders as summarized in Table 1. All experiment runs in the table are included in the training set, except for run3.1, which is saved for validation.

Table 1: Number of frames for each flow regime

run1	run2.1	run2.2	run3.1	run3.2	run4
1638	171	1339	113	1256	1256
run5	**run6.1**	**run6.2**	**run7.1**	**run7.2**	**TOTAL**
2404	141	1033	59	2106	11516

- Three versions of the dataset are investigated by converting between different color spaces: the original dataset in RGB, a grayscale version (GRAY), and a black & white version (BW). The BW dataset is obtained by first converting the RGB images to HSV space and then picking the hue value corresponding to the blueish color (flow streams) with a given tolerance range.

As noted in Table 1, the total number of frames in each of the three datasets is 11516, which is small for today's neural network standards. Patterns embedded in lower dimensional spaces such as GRAY and BW, however; are the most relevant for the goals of this project.

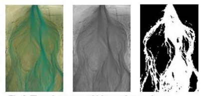

Fig. 3: Three datasets on which neural networks are trained: RGB (left), GRAY (middle), BW (right).

- Optical flow images are computed for each experiment run. Besides attempting to predict video frames directly, neural networks are trained to predict optical flow that is later used to synthesize new videos.

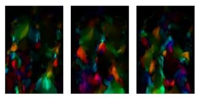

Fig. 4: Optical flow images computed from RGB dataset. Colors represent direction of flow.

Given an initial frame of the experiment, one can repeatedly warp the image with the predicted sequence of optical flow to synthesize a new video. The optical flow is an approximation of the true flow velocities, which is a concept of major interest in this project.

- All datasets are augmented by flipping the frames horizontally at random upon loading. The pixel values are normalized to the range [0,1] with a straightforward division by the maximum integer value of 255.

Regression problem:

$$\boldsymbol{x}_{t,p} = \left(I_{t-p+1}, \dots, I_{t-1}, I_t\right)$$
$$\boldsymbol{y}_{t,f} = \left(I_{t+1}, I_{t+2}, \dots, I_{t+f}\right)$$

From p frames of the past, predict f frames into the future. In this project, the focus on 1-step prediction: $p = 3, f = 1$.

- Inspired by one of Torricelli's equations of motion from classical mechanics $x_{t+\Delta t} = x_t + v\Delta t + a\frac{\Delta t^2}{2}$, a convolutional neural network model is proposed with three modules:

TorricelliNet

1. velocity module = 2. acceleration module

Repeat L times:

$$\boldsymbol{z}_p = Conv(\boldsymbol{x}_p, kern = 5, pad = 2)$$
$$\boldsymbol{a}_p = ReLU\left(BatchNorm(\boldsymbol{z}_p)\right)$$

Output layer:

$$\boldsymbol{v}(\boldsymbol{x}_p) = Conv(\boldsymbol{a}_p)$$

$$\boldsymbol{a}(\boldsymbol{x}_p) = same\ as\ velocity$$

3. prediction module

$$\boldsymbol{z}_p = Conv\left(\boldsymbol{x}_p + \boldsymbol{v}(\boldsymbol{x}_p) + \frac{\boldsymbol{a}(\boldsymbol{x}_p)}{2}\right)$$
$$\overline{\boldsymbol{y}}_f = Sigmoid\left(BatchNorm(\boldsymbol{z}_p)\right)$$

Methodology (contd.)

- In an attempt to capture very small changes in flow patterns, a recurrent neural network model is proposed. First, $S = 50$ evenly spaced rows of the frames are forward in time using different GRU units. Then, another set of GRU units is used to fill in the gaps between the predicted rows as illustrated in Fig. 5.

SliceNet

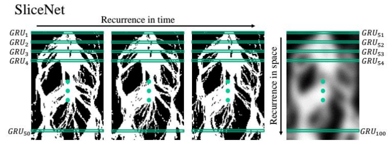

Fig. 5: SliceNet architecture. GRU units (1-50) are used to predict evenly spaced rows in the future frame, and another set of GRU units (51-100) is used to fill in the gaps.

The hidden states of the GRU units are fed into dense layers with sigmoid activation to produce frames with valid pixel values in [0,1]. To enforce continuity in between rows, a total variation term is added to the binary cross-entropy loss:

$$L(\boldsymbol{y}, \overline{\boldsymbol{y}}) = \underbrace{-\sum_{i,j} \boldsymbol{y}_{i,j} \log \overline{\boldsymbol{y}}_{i,j} + (1 - \boldsymbol{y}_{i,j}) \log\left(1 - \overline{\boldsymbol{y}}_{i,j}\right)}_{\text{Binary cross-entropy}} + \underbrace{\sum_i \left| \left\|\boldsymbol{y}_{i,:} - \boldsymbol{y}_{i+1,:}\right\|_1 - \left\|\overline{\boldsymbol{y}}_{i,:} - \overline{\boldsymbol{y}}_{i+1,:}\right\|_1 \right|}_{\text{Total variation}}$$

For color spaces other than BW, the L_1 and L_2 losses are used instead.

- The difference process is defined for the validation set (i.e. run3.1) $d_t = \|I_{t+1} - I_t\|_1$ as well as its normalized version $d_t^* = d_t / d_1$. Return levels and autocorrelation statistics are computed on the normalized process of the true video and the videos synthesized by the neural network models [Beirlant et al. 2005, Matheron 1971].

Results

TorricelliNet

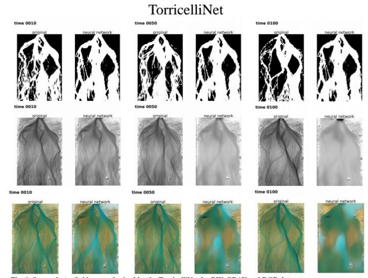

Fig. 6: Screenshots of videos synthesized by the TorricelliNet for BW, GRAY and RGB datasets.

- For the BW dataset, the neural network quickly looses its ability to mimic the flow dynamics and just copies the same frames forward in time.
- For the GRAY and RGB datasets, the frames quickly become blurry.

For full videos:

https://vimeo.com/album/5055962

Results (contd.)

SliceNet

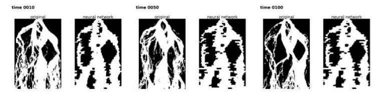

Fig. 7: Screenshots of videos synthesized by the SliceNet for the BW dataset.

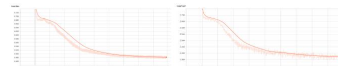

Fig. 8: Learning curves for SliceNet on validation set (left) and training set (right).

- Compared to the TorricelliNet, the SliceNet produces frames that are more varying.
- Artifacts are present at the interface between any two neighboring GRU units.
- Learning curves indicate a stable learning process.

- Proposed neural network models fail to reproduce the ciclicity of the flow patterns indicated by the empirical variogram.

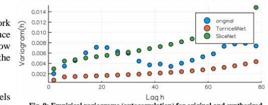

Fig. 9: Empirical variograms (autocorrelation) for original and synthesized videos of validation set.

- Neural network models produce videos with similar autocorrelation length. SliceNet has higher variance.
- Return levels computed on synthesized videos are far smaller than those present in original phenomena.

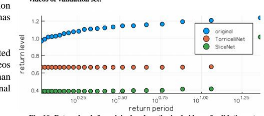

Fig. 10: Return levels for original and synthesized videos of validation set.

Conclusions

- The neural network models proposed in this work are far too simple to accommodate the complexity of flow and sediment transport recorded in flume experiments.
- Despite the various attempts to train the networks with different color spaces, architectures, and loss functions, all fail to reproduce statistics of interest such as return levels and autocorrelation.
- The SliceNet architecture together with the total variation loss is promising. Additional work is needed to eliminate artifacts in between neighboring GRU units.
- Training the networks on optical flow images instead of raw frames did not improve the results considerably. Future work should include a more careful investigation of video synthesis by means of warping frames with optical flow.

References

Murray, B. et al. (2009). "Geomorphology, complexity, and the emerging science of the Earth's surface". *Geomorphology Journal*.

Bufe, A. et al. (2016). "Fluvial bevelling of topography controlled by lateral channel mobility and uplift rate". *Nature Geoscience*.

Beirlant, G. et al. (2005). "Estimation of the extreme-value index and generalized quantile plots". *Bernoulli Journal*.

Matheron, G. (1971). "The theory of regionalized variables and its applications". Centre de Morphologie Mathemátique de Fontainebleau.