

# On the Problem about Optical Flow Prediction for Silhouette Image

Bedy Purnama<sup>1,3</sup>, Mera Kartika Delimayanti<sup>1,4</sup>, Ngoc Giang Nguyen<sup>1</sup>, Kunti Robiatul Mahmudah<sup>1</sup>, Mamoru Kubo<sup>2</sup>, Makiko Kakikawa<sup>2</sup>, Yoichi Yamada<sup>2</sup> and Kenji Satou<sup>2</sup>

<sup>1</sup>Graduate School of Natural Science and Technology, Kanazawa University, Kanazawa, Japan

<sup>2</sup>Institute of Science and Engineering, Kanazawa University, Kanazawa, Japan

<sup>3</sup>Telkom School of Computing, TELKOM University, Bandung, Indonesia

<sup>4</sup>Department of Computer and Informatics Engineering, Politeknik Negeri Jakarta, Jakarta, Indonesia

**Keywords:** Optical Flow, Silhouette Image, Homography, TV-L1 Optical Flow.

**Abstract:** We address a problem of finding an algorithm to handle unclear flows in the inner area of silhouette image. In this study, after getting the region of interest from an implemented algorithm of dual TV-L1 optical flow, we conducted experiments to predict the new flow in the unclear inner areas. The experiments include perspective transform methods. Five experiments were performed using these methods. As a result, an algorithm that uses double refining of perspective transform method obtained the optimal result. It might be useful for analyzing the motion of a unicolor animal (e.g. a black cat).

## 1 INTRODUCTION

This study aims at analyzing the motions of animals by tracking some points of the animal surfaces. Some of the surfaces are eyes, ears, hands, tail etc. The animals with clear characteristics and patterns are easier to be tracked, while the animals with unicolor are quite difficult to be tracked. For example, in the case of tracking a black cat, tracking its eyes, ears, or hands is relatively easier than tracking its particular body points. Meanwhile, tracking the animal surface brings some problems on the calculation of the optical flow to the moving unicolor animals. To the animals with characteristic patterns or rich textures on its body surface, the traditional state of the art of optical flow calculation algorithm will work well. In contrast, to the unicolor animals, the optical flow for the center of the body is usually not calculated because it has no flow to detect the movement of some points in the same color area. Therefore, to solve this problem, we presented the study on the homography and perspective transform. Finally, since the unicolor animals spread all over the world, it is important to calculate the optical flow of the unicolor animals correctly in order to analyze its motions.

In recent years, the development of object tracking methods using optical flow is growing rapidly. Many new studies have emerged to develop this method. Site provider for the optical flow datasets

such as MPI Sintel benchmark, KITTI Flow 2012 evaluation, KITTI Flow 2015 evaluations and Middlebury are becoming the leading reference to the extent of current research on optical flow.

To strengthen this, Horn and Schunck (1981) defined optical flow as the dissemination of clear velocities of an image brightness pattern movement that can arise from a relative motion of objects and the viewer. By knowing the optical flow from an image sequence, the movement of objects in that image can be analyzed. The case raised in this study was looking for an optical flow calculation at the center of the body of the unicolor animal image. This study uses a silhouette image in forming a model of solution that has been built. The silhouette image is a simplification of the unicolor animal image.

In considering a video of moving silhouette image as a series of images, when optical flows implemented into these series of images, the edge area of those images are showing clear flows while those inner areas are showing unclear flows. Therefore, attempts in encountering the unclear flows have been performed in this study to make the flows clear.

In section 2.5, an algorithm has been proposed to fill the unclear flows in the silhouette images. The algorithm does not only fill the unclear flow areas but also smooths and blends with the surrounding flows.

## 2 MATERIALS AND METHODS

This part elaborates several datasets and functions used in the built algorithm. The study uses dual TV-L1 optical flow, homography, and perspective transform. Those functions are available in the OpenCV library. Furthermore, the experimental design is also elaborated in this part.

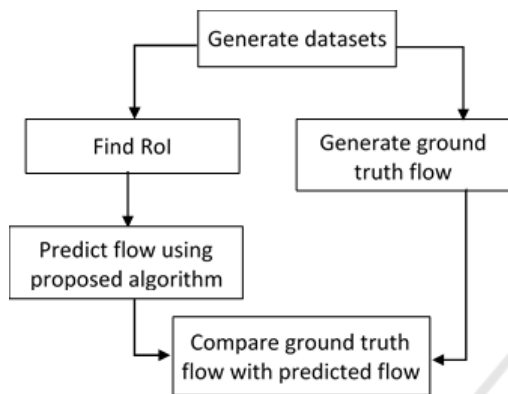


Figure 1: Optical flow prediction flowchart.

Flowchart in Figure 1 describes the process built in this study. The process begins with the dataset construction and followed by the construction of data ground truth flow. The generated data serve as a comparison for the data generated by the build algorithm. In addition, those datasets are used to determine the Region of Interest (RoI). This RoI reveals the location of the unclear flows in the inner area stored in the mask\_RoI\_center and the location of the clear flows stored in the mask\_RoI\_edge. To fill in the empty flow in the mask\_RoI\_center area, we implemented several algorithms. The outputs of the algorithms are then compared with the data ground truth to observe its performance.

### 2.1 Dataset

In this study, we used deer silhouette image data downloaded from MarriArt (2013). First, we converted the image into 100 images by rotating 7 degrees counter clockwise 99 times within size 640x360.

Second, we created the ground truth flows from the above datasets. By knowing the rotation matrix obtained from the formation of the main dataset, we can identify the displacement of each point both on x-axis and y-axis. The point displacement on the x-axis is called  $u(x, y)$ , while the other is called  $v(x, y)$ . Speed magnitude of  $r(x, y)$  is obtained from Eq. (1) and the angle  $\theta(x, y)$  is obtained from Eq. (2), both

of which are spatial derivatives of  $u$  and  $v$  (Butler, 2012). From values  $r(x, y)$  and  $\theta(x, y)$  we can visualize flow as a color image. For flow color coding (Figure 2), we refer to the scheme built by Baker et al. (2010).

$$r(x, y) = \sqrt{u(x, y)^2 + v(x, y)^2} \quad (1)$$

$$\theta(x, y) = \tan^{-1} \frac{v(x, y)}{u(x, y)} \quad (2)$$

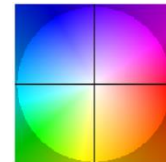


Figure 2: Color coding of the flow vectors.

### 2.2 Optical Flow Dual TV-L1

Research on optical flow was initiated by Horn-Schunck in 1981. Horn formulated optical flow problems by minimizing global energy functions in Eq. (3).

$$\min \left\{ \int_{\Omega} |\nabla_{u_1}|^2 + |\nabla_{u_2}|^2 d\Omega + \lambda \int_{\Omega} (I_1(x + u(x)) - I_0(x))^2 d\Omega \right\} \quad (3)$$

Where  $I_0$  and  $I_1$  represents the image pair,  $u$  represents two-dimensional displacement field, and  $\lambda$  represents a free parameter. The first term, known as the regularization term, caused the appearance of high variations in  $u$  to obtain smooth displacement fields. Meanwhile, the optical flow constraint reflected by the second term is known as data term. Therefore, it is assumed that there is no change in the intensity values of  $I_0$  in the process of its motion to  $I_1$  (Zach Pock and Bischof, 2007).

Dual TV-L1 commonly used in flow estimation of dense optical flow model. According to Zach, Pock and Bischof (2007), the TV-L1 formulation that was built has an efficient numerical scheme, based on the dual formulation of Total Variation energy and employed a useful point-wise thresholding step. While the implementation was developed by Sánchez Pérez, Meinhardt-Llopis and Facciolo, (2013).

### 2.3 Finding Region of Interest

Img\_flow, the output of optical flow, consists of

horizontal flow  $u(x,y)$  and vertical flow  $v(x,y)$ . The next step is to look for the RoI. The flowchart of the process is shown in Figure 3.

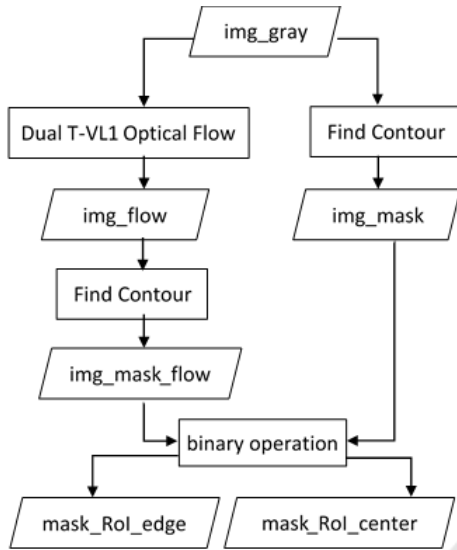


Figure 3: Flowchart for finding RoI.

To obtain a threshold value, we use the Otsu (1979) method. This value is the boundary between RoI and background in forming contour search. By using this method, we obtained `img_mask` and `img_mask_flow` which are the RoI binary images of flow and grayscale images. Furthermore, using binary operations on both binary images produces `mask_RoI_edge` and `mask_RoI_center`.

`Mask_RoI_edge` is an area on RoI that has a flow, whereas `mask_RoI_center` is an area on RoI that has no flow or has a very little flow. When all points on RoI have the flow, there is no `mask_RoI_center`. So, in this particular condition, there is no need for correcting the flow.

## 2.4 Perspective Transform

We analyzed the movement of objects captured by the image sequence find out how the objects move the use of homography or perspective transformation between two fields. Here it is assumed that the two analyzed images have the same object with relatively similar intensity.

$$s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4)$$

The transformation between two fields is formulated in eq (4). Matrix  $H$  has 9 components but we need only 8 Degree of Freedom (DoF) in its implementation. Therefore, the value of  $h_{33}$  is set to

1.  $H$  is the homography matrix which required us to search its components. It requires at least 4 pairs of points from the image source and target image to get the  $H$  value. (Docs.opencv.org, 2018).

To gain the matrix  $H$ , the position of the starting point of the flow as the source field and the position of the end point of the flow as the target field has been used in its implementation. After obtaining the homography matrix, the values in the masking area of the reference area can be determined.

## 2.5 Design of Experiment

There are 5 algorithms built in this study. As for algorithm 0, it is the basic process for other algorithms.

### 2.5.1 Algorithm 0

In algorithm 0, a new flow is generated by cutting `img_flow` which is an optical flow output with `mask_RoI_edge`. This new flow is named as `img_calculated1`. Then the `mask_RoI_center` is ignored so that the resulting flow will leave an empty area.

```

//Pseudo code algorithm 0
//Generate new flow
img_calculated1 ← img_flow AND
mask_RoI_edge
flow ← img_calculated1
  
```

### 2.5.2 Algorithm 1

Starting from algorithm 1 onwards, flow calculations are performed on `mask_RoI_center`. This step will enable the resulting flow to meet all points on the RoI.

By implementing the perspective transform on the pairs of points `img_calculated1`, matrix  $H$  is obtained. Then, the matrix  $H$  is used to generate flow in the `mask_RoI_center` named `img_predicted1`. The output of the algorithm 1 is the new flow combined result of `img_calculated1` and `img_predicted1`.

```

//Pseudo code algorithm 1
//Generate prediction flow
img_calculated1 ← img_flow AND
mask_RoI_edge
img_predicted1 ←
perspectiveTransform(img_calculated1,
mask_RoI_center)

//Generate new flow
flow ← img_calculated1 +
img_predicted1
  
```

### 2.5.3 Algorithm 2

There is an addition to the smoothing process in algorithm 2. This smoothing process changes the flow in mask\_RoI\_edge. The step is to use perspective transform in img\_predicted1 to produce a replacement flow in mask\_RoI\_edge. This replacement flow is named img\_predicted2. The output of this program is a combined result of img\_predicted1 and img\_predicted2.

```
//Pseudo code algorithm 2
//Generate prediction flow
img_calculated1 ← img_flow AND
mask_RoI_edge
img_predicted1 ←
perspectiveTransform(img_calculated1,
mask_RoI_center)

//Smoothing process
img_predicted2 ←
perspectiveTransform(img_predicted1,
mask_RoI_edge)

//Generate new flow
flow ← img_predicted1 +
img_predicted2
```

### 2.5.4 Algorithm 3

Modifications in the smoothing process are done by combining img\_predicted1 and img\_calculated1 and include it as inputs in the perspective transform. This process aims to replace flow in mask\_RoI\_edge. This replacement flow is labelled as img\_predicted2. The output of this algorithm is a combination of img\_predicted1 and img\_predicted2.

```
//Pseudo code algorithm 3
//Generate prediction flow
img_calculated1 ← img_flow AND
mask_RoI_edge
img_predicted1 ←
perspectiveTransform(img_calculated1,
mask_RoI_center)

//Smoothing process
img_predicted2 ←
perspectiveTransform(img_predicted1,
img_calculated1, mask_RoI_edge)

//Generate new flow
flow ← img_predicted1 + img_
predicted2
```

### 2.5.5 Algorithm 4

In this algorithm, the smoothing process is approximately similar to algorithm 3 where img\_predicted1 and img\_calculated1 become input in the perspective transform. However, img\_predicted2, the resulting replacement flow from smoothing process will replace the entire RoI. Consequently, img\_predicted2 becomes the output flow of this algorithm.

```
//Pseudo code algorithm 4
//Generate prediction flow
img_calculated1 ← img_flow AND
mask_RoI_edge
img_predicted1 ←
perspectiveTransform(img_calculated1,
mask_RoI_center)

//Smoothing process
img_predicted2 (RoI area) ←
perspectiveTransform(img_predicted1, img_
_calculated1, img_mask)

//Generate new flow
flow ← img_predicted2
```

## 3 EXPERIMENTAL RESULTS







Each built algorithm produced a flow which was then compared to ground truth flow. We used average end point error (EPE) as the evaluation metric. EPE is calculated through Eq.5.

$$EPE = \sqrt{(u - u_{GT})^2 + (v - v_{GT})^2} \quad (5)$$

where  $u_{GT}$  and  $v_{GT}$  refers to ground truth horizontal flow and ground truth vertical flow. Table 1 showed the EPE over all algorithms.

From the above table, we can see that algorithm 2 produces the lowest EPE. The lower the EPE, the closer the algorithm to the ground truth. This showed that algorithm 2 had better performance compared to other algorithms.

Table 1: EPE from several algorithms in comparison with Ground Truth.

Flow Image	Method	EPE
	Ground Truth	0.00
	Algorithm 0	6.28217
	Algorithm 1	5.28975
	Algorithm 2	5.28080
	Algorithm 3	5.31072
	Algorithm 4	5.28737

## 4 CONCLUSIONS

We proposed algorithms to correctly calculate optical flow in silhouette images. Five experiments with different algorithms have been carried out, then the output of each algorithm compared to ground truth data. We used EPE as the evaluation metric. As a result, algorithm 2 appears with the lowest EPE value. Algorithm 2 used two perspective transform processes to produce a flow that approaches the ground truth. As a suggestion of future work, real unicolor animal data could be used which will inevitably require more reliable preprocessing to identify its RoI.

## ACKNOWLEDGEMENTS

The first author would like to gratefully acknowledge the Indonesian Endowment Fund for Education

(LPDP) and The Directorate General of Higher Education (DIKTI) for providing BUDI-LN scholarship.

In this research, the super-computing resource was provided by Human Genome Center, the Institute of Medical Science, the University of Tokyo. Additional computation time was provided by the super computer system in Research Organization of Information and Systems (ROIS), National Institute of Genetics (NIG). This work was supported by JSPS KAKENHI Grant Number JP18K11525.

## REFERENCES

- Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M. and Szeliski, R., 2010. A Database and Evaluation Methodology for Optical Flow. *International Journal of Computer Vision*, 92(1), pp.1-31.
- Butler, J. D., Wulff, J., Stanley, G. B., Black, M. J., 2012. A Naturalistic Open Source Movie for Optical Flow Evaluation. 611-625. *10.1007/978-3-642-33783-3\_44*.
- Docs.opencv.org., 2018. *OpenCV: Basic concepts of the homography explained with code*. [online] Available at: [https://docs.opencv.org/3.4.1/d9/dab/tutorial\\_homography.html](https://docs.opencv.org/3.4.1/d9/dab/tutorial_homography.html) [Accessed 14 Nov. 2018].
- Horn, B. K. P., Schunck, B.G., 1981. Determining optical flow. *Artificial Intelligence*, 17:185-203
- MarriArt, 2013. *Silhouette animal runcycles*. [online] Available at: <https://www.youtube.com/watch?v=WQSVjxQGpf8> [Accessed 16 Jan. 2018].
- Otsu, N., 1979. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), pp.62-66.
- Sánchez, J., Meinhardt-Llopis, E., Facciolo, G., 2013. TV-L1 Optical Flow Estimation. *Image Processing On Line* 3, 137-150.
- Szeliski, R., 2010. *Computer vision: Algorithms and Applications*. New York: Springer., pp. 35-41
- Zach, C., Pock, T., Bischof, H., 2007. A duality based approach for realtime TV-L1 optical flow. *Proceeding of 29th DAGM Symposium of Pattern Recognition*