Scientific Research Publishing

Journal of Computer and Communications, 2018, 6, 265-274
http://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Motion Localization with Optic Flow for Autonomous Robot Teams and Swarms

Andrew K. Massimino, Donald A. Sofge*

U.S. Naval Research Laboratory, Washington DC, USA
Email: *donald.sofge@nrl.navy.mil

How to cite this paper: Massimino, A.K. and Sofge, D.A. (2018) Motion Localization with Optic Flow for Autonomous Robot Teams and Swarms. *Journal of Computer and Communications*, **6**, 265-274.
https://doi.org/10.4236/jcc.2018.61026

Received: October 29, 2017
Accepted: January 5, 2018
Published: January 8, 2018

Abstract

The ability to localize moving objects within the environment is critical for autonomous robotic systems. This paper describes a moving object detection and localization system using multiple robots equipped with inexpensive optic flow sensors. We demonstrate an architecture capable of detecting motion along a plane by collecting three sets of one-dimensional optic flow data. The detected object is then localized with respect to each of the robots in the system.

Keywords

Localization, Optic Flow, Robot Team, Swarm, Situational Awareness

1. Introduction

Sensor-equipped mobile swarm robots often have limited sensing and communication capabilities. In particular, sensors are sometimes constrained by the amount of data they can acquire or transmit to each other or to a base station. As a result, it is desirable to focus expensive sensors on only important targets and not waste sensing resources on empty space.

We introduce a system for inexpensive moving object detection among swarm robots. Coarse optic flow information is computed from sensor data quickly on special hardware and is used to perform a rough localization on a plane. In this scheme, the addition of sensors scales well as optic flow is suited to computation on specialized hardware. The initial detection and localization given in this paper may be used for further investigation by more expensive sensors and computation.

2. Background

Optic flow is the perceived two-dimensional motion of pixels in an image as ob-

served by a camera. This effect is caused by the motion of objects in a scene and relative to the camera [1]. Although optic flow often refers to a dense vector field of motion of each pixel, averages over an entire scene can be very useful in many applications and is often much easier to compute. Optic flow has been used extensively in robotic applications such as egomotion estimation, navigation, and obstacle detection [2] [3].

Analogous to optic flow, the motion of points in three- dimensional space is sometimes called scene flow. There has been work in reconstructing the scene flow from optical flow data from multiple camera views. This problem is ill-posed in general because flow tangent to the direction of the image intensity gradient is indeterminate [1].

3. Image Interpolation Algorithm

There are a large number of popular algorithms for computing optic flow between two images. They exhibit varying computational complexities and accuracy. Most algorithms assume what is called the *brightness constancy*, which can be stated for an appropriate neighborhood around pixel coordinates (x, y) as

$$I(x, y, t) = I(x + u, y + v, t + 1) \tag{1}$$

where I denotes the image intensity at a given position and $(u, v) = (u(x, y, t), v(x, y, t))$ is the optical flow [4]. This assumption implies that an optical flow vector (u, v) exists whenever when the intensity of the image matches an image at a later time shifted by that flow vector. This works best when scenes are consistently illuminated and the reflectance of object surfaces is independent of viewing orientation.

Since Equation (1) imposes a single constraint with two unknowns, in general it is impossible to solve for the optic flow. For example, consider a pattern consisting of horizontal lines. While vertical motion is easily discerned, a horizontal displacement does not cause a change in the image. To solve this issue, called the *aperture problem*, algorithms impose other constraints such as smoothness of the optic flow vectors [4].

Many algorithms operate by matching features among a set of images, which can involve costly computation. The image interpolation algorithm (IIA) proposed by Srinivasan instead works directly on the image gradient in a single pass [5]. IIA aims to estimate global optical flow and therefore arrives at a solution which best explains motion of an entire image instead of at each pixel. In particular it considers a number of versions of the image shifted by reference amounts and finds the (x, y) vector that produces the best interpolating image.

Where f denotes an image intensity, we linearize Equation (1) about the first reference image f_0,

$$\hat{f} = f_0 + (\Delta x / 2\Delta x_{ref})(f_2 - f_1) + (\Delta y / 2\Delta y_{ref})(f_4 - f_3) \tag{2}$$

where

$$f_1(x, y) = f_0(x + \Delta x_{ref}, y)$$

$$f_2(x,y) = f_0(x - \Delta x_{ref}, y)$$

$$f_3(x,y) = f_0(x, y + \Delta y_{ref})$$

$$f_4(x,y) = f_0(x, y - \Delta y_{ref})$$

IIA finds the $(\Delta x, \Delta y)$ which minimizes the mean square error between the second image f and its estimate $\hat{f}$:

$$\hat{f} = argmin \iint \left(f - \hat{f}\right)^2 dxdy \tag{3}$$

We assume that the displacement consists of a vertical and horizontal component with no rotation. From substituting (2) into (3),

$$F_{21} = \iint (f_2 - f_2)^2 dxdy$$

$$F_{43} = \iint (f_4 - f_3)^2 dxdy$$

$$F_{4321} = \iint (f_4 - f_3)(f_2 - f_1) dxdy$$

$$F_h = 2\iint (f - f_0)(f_2 - f_1) dxdy$$

$$F_v = 2\iint (f_4 - f_{40})(f_2 - f_1) dxdy$$

$$\begin{bmatrix} F_h \\ F_v \end{bmatrix} = \begin{bmatrix} F_{21} & F_{4321} \\ F_{4321} & F_{43} \end{bmatrix} \begin{bmatrix} \dfrac{\Delta x}{\Delta x_{ref}} \\ \dfrac{\Delta y}{\Delta y_{ref}} \end{bmatrix}$$

Then the final equation can be solved as

$$\begin{bmatrix} \dfrac{\Delta x}{\Delta x_{ref}} \\ \dfrac{\Delta y}{\Delta y_{ref}} \end{bmatrix} = \frac{1}{F_{21}F_{43} - F_{4321}^2} \begin{bmatrix} F_{43} & -F_{4321} \\ -F_{4321} & F_{21} \end{bmatrix} \begin{bmatrix} F_h \\ F_v \end{bmatrix}$$

The IIA algorithm is simple to compute in a single pass and is robust to local failures of the assumption (1) as it averages over the entire image [5].

4. Approach

Optical flow is related to scene flow by the relative position of the object as well as the camera projection matrix. Assume an object in 3D space has a position represented in homogeneous coordinates by $\boldsymbol{x} = [x, y, z, 1]^T$. The projected point on camera i is given by $u_i = [u_i, v_i, 1]^T$. Then u_i is related to $\boldsymbol{x}$ by the projection matrix $P_i \in \mathbb{R}^{3\times 4}$ as follows:

$$u_i \sim P_i \boldsymbol{x} \tag{4}$$

where ~ denotes equality up to a scaling factor. The coordinates of u_i and v_i are given by

$$u_i = \frac{P_i^{11}x + P_i^{12}y + P_i^{13}z + P_i^{14}}{P_i^{31}x + P_i^{32}y + P_i^{33}z + P_i^{34}} \tag{5}$$

$$v_i = \frac{P_i^{21}x + P_i^{22}y + P_i^{23}z + P_i^{24}}{P_i^{31}x + P_i^{32}y + P_i^{33}z + P_i^{34}} \tag{6}$$

We can state the constraints (Equations (5) and (6)) for each view $1 \le i \le N/2$ as the following matrix equation:

$$\begin{bmatrix} P_1^{31}u_1 - P_1^{11} & P_1^{32}u_1 - P_1^{12} & P_1^{33}u_1 - P_1^{13} \\ P_1^{31}v_1 - P_1^{21} & P_1^{32}v_1 - P_1^{22} & P_1^{33}v_1 - P_1^{23} \\ P_2^{31}u_2 - P_2^{21} & P_2^{32}u_2 - P_2^{22} & P_2^{33}u_2 - P_2^{13} \\ P_2^{31}v_2 - P_2^{21} & P_2^{32}v_2 - P_2^{22} & P_2^{33}v_2 - P_2^{23} \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} P_1^{14} - P_1^{34}u_1 \\ P_1^{24} - P_1^{22}v_1 \\ P_2^{14} - P_2^{34}u_2 \\ P_2^{24} - P_2^{22}v_2 \\ \vdots \end{bmatrix} \tag{7}$$

Or where $Q \in \mathbb{R}^{3\times 4N\times 3}$ and $q \in \mathbb{R}^N$,

$$Q_{x_{1:3}} = \boldsymbol{q} \tag{8}$$

Equation (8) can be solved using the pseudo-inverse. This is called the *direct linear transformation algorithm* [6].

$$\mathrm{x}_{1:3} = Q^{+}\boldsymbol{q}$$

Given a set of optical flow measurements, $\left\{u_i^0, u_i^1 \mid 1 \le i \le N/2\right\}$, we would like to recover x_0 and x_1. The scene flow $\dot{\boldsymbol{x}}$ and optical flow $\dot{\boldsymbol{u}}$ over time Δt satisfy

$$\boldsymbol{x}_1 \approx \boldsymbol{x}_0 + \Delta t\dot{\boldsymbol{x}}$$

$$\boldsymbol{u}_i^1 \approx \boldsymbol{u}_i^0 + \Delta t\dot{\boldsymbol{u}}$$

As in (4), $\boldsymbol{x}_{\{0,1\}}$ and $\boldsymbol{u}_i^{\{0,1\}}$ are related by a scaling factor:

$$\boldsymbol{u}_i^0 \sim P_i\boldsymbol{x}_0\boldsymbol{u}_i^1 \sim P_i\boldsymbol{x}_1$$

Both x_0 and x_1 can be computed using Equation (8) for each set of points. This gives us our estimate of the location of the object and its velocity.

5. Implementation

In this work we focus on coarse localization in sparse scenes. We assume that objects are in the foreground and move rigidly. Optical flow sensors are arranged on the robot agents as in **Figure 1**. We assume the positions and orientations of the robots are known.

The optical flow sensor we use is the Centeye *Stonyman* vision chip fitted with a cellphone-type lens. Among the advantages of this device are its low cost, low power consumption, and ease of interfacing with a micro-controller. The chip supports a resolution of up to 112 → 112 although data can be read asynchronously from any size pixel region. We determined the (horizontal) field of view of the sensor to be 40 by collecting images of a checkerboard pattern at various ranges and computing the angular extents. The projection matrix is assumed to be of the form

$$P = \begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R^T & -R^T T \end{bmatrix}$$

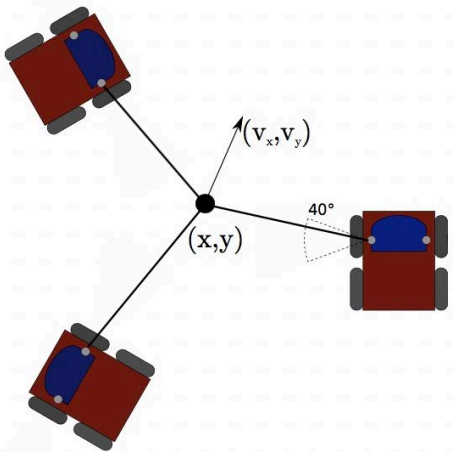

Figure 1. Multi-robot optical flow scenario.

where R and T are the orientation and translation of the sensor in global coordinates. The focal lengths α_x and α_y were estimated from the field of view by assuming a pin-hole camera response of the Centeye sensors. We set the skew, $\gamma = 0$. u_0 and v_0 are the coordinates of the center of the optical flow region.

We use the Mbed NXP LPC1768 microcontroller to interface with the Centeye sensor. This architecture is illustrated in **Figure 2**. Using a microcontroller allows the data to be processed with little expense and without introducing a burden to the CPU on board the robot. This device is fairly low power and inexpensive. This optic flow algorithm could also be implemented at even lower cost directly in hardware. From the robot's point of view, the microcontroller and Centeye configuration emulates a special purpose optical flow sensor. The robot communicates with the Mbed chip through a serial interface over USB. A simple message library was implemented to deal with initialization and collecting vertical and horizontal flow data from each region of the sensor.

Finally, optic flow data from each robot is relayed to a centralized processor which computes the target position and velocity estimate. It would also be easy to perform this calculation directly on one or all of the robots, given a communication link between them. As optic flow is computed for few large regions on each vehicle, there is little data to transmit compared to full images from the Centeye sensors.

We compute the optic flow using the image interpolation algorithm on blocks of size 24 × 24 pixels. Therefore, each window covers about 8.57˚ in the vertical and horizontal directions. Three such regions are used for each sensor: one in the center, and two 12.14˚ to the left and right. Optic flow is computed on each microcontroller at 6.67 Hz. A length 6 moving average filter is used to reduce sensor noise. This data is sent to the central computer for processing. Averaged optical flow data under a threshold is discarded to ignore sensor noise. For each region i, we assume $\boldsymbol{u}_i^0$ refers to the center of that cell. The final position $\boldsymbol{u}_i^1$ is then estimated from the computation of $\dot{\boldsymbol{u}}$, performed on the microcontroller.

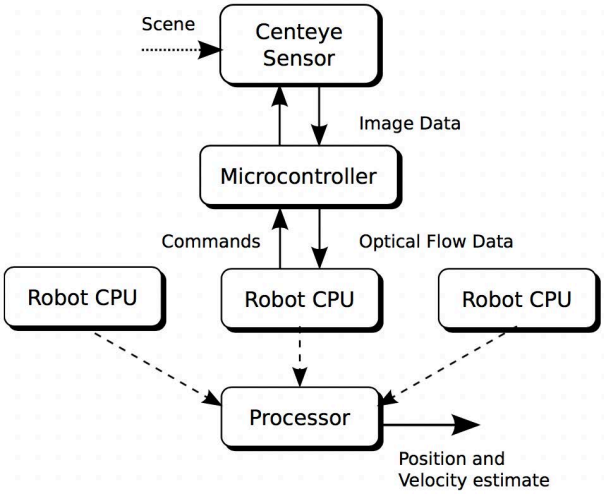

Figure 2. Multi-robot optic flow architecture. Processor collects and processes data from multiple robots.

From $\boldsymbol{u}_i^0$ and $\boldsymbol{u}_i^1$ we use the direct linear transform algorithm described above to compute $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$.

6. Results

The experiment was conducted in a Vicon arena approximately 10 meters by 12 meters in size. Three robots were situated in an approximate triangle. A human wearing a Vicon tracked hat walks into the scene between the robots. Each robot collected and averaged data, relaying it to a base station. The position was then estimated from the flow data and the positions and orientations of the robots. In our experiment, the human was alerted by an audible command (e.g. "Stop right there") and the position estimate is relayed to a camera system which performs facial detection and finer localization.

The averaged, thresholded flow data is shown in **Figure 3**. The threshold was chosen to eliminate most false positives when there was no scene activity. It is clear that as the target walks onto the scene, the sensors register hits in either the positive or negative direction.

An overview of the scene is given in **Figure 4**. The path followed by the person is given in green, as tracked using a Vicon marker helmet. Black points are the estimated coordinates. Red vectors give the estimated scene flow in arbitrary units. The black points are generally on intersections of the yellow sensitive regions of the Centeye sensors. This is a limitation of the block-based optical flow computation. However, the red motion estimates also track the position change of the object, providing more data about its path. Estimates with extremely low velocities such as the extraneous point in **Figure 4** can be discarded.

7. Discussion

Optic flow was successful in detecting motion about a scene. Although individual sensors are very noisy, thresholding and combining data from multiple

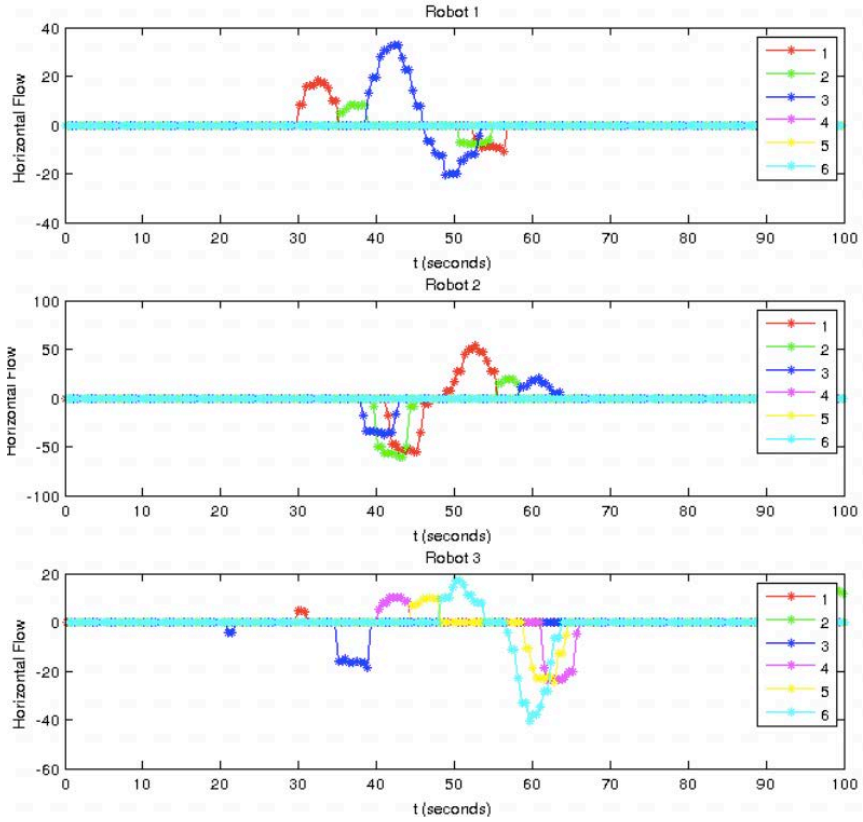

Figure 3. Optic flow scenario data collected on each robot.

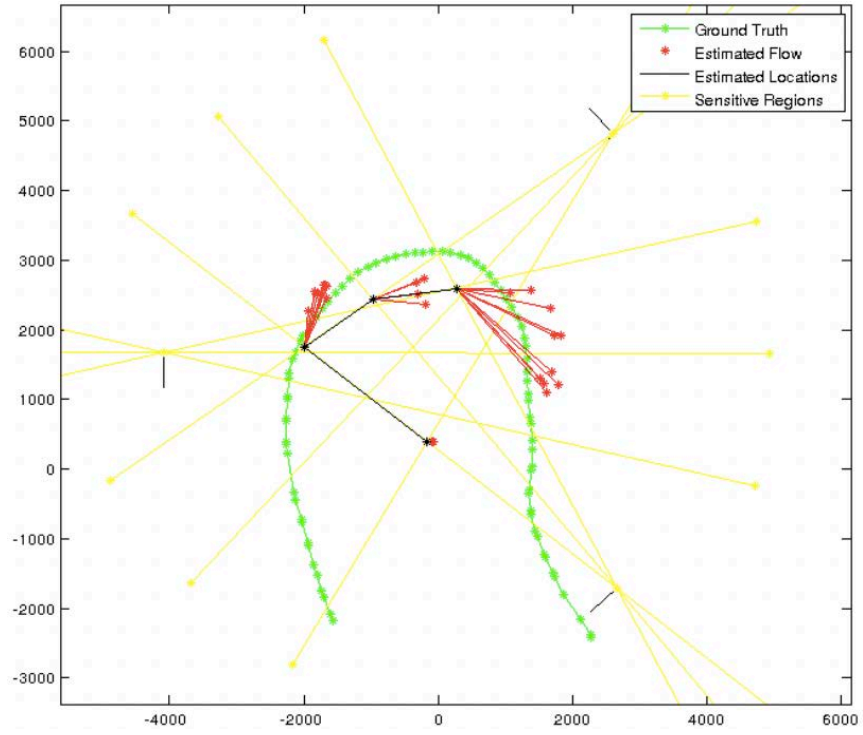

Figure 4. Scenario view from above. Dimensions are in millimeters. The three points where yellow lines emanate from are the locations of the three robots. A spurious point near (0, 500) could be excluded due to its low scene flow value.

sources allows false alarms to be reduced. As shown by the results, this method approximately tracks both the position and motion of a moving object.

Noise in the optic flow data presented one of the greatest difficulties encountered in this experiment. The following are some of sources of inaccuracy present in our experimental setup:

- **Noise on the CMOS sensor**—Although fixed-pattern CMOS noise is mitigated by an initial calibration step, there is still a good deal of noise on the imagery obtained from the Centeye sensors. This is from a variety of internal and environmental sources.
- **Interference from the Vicon motion capture system**—The Vicon motion capture environment consists of 8 infrared cameras each surrounded by an array of infrared emitters. IR-reflective markers on the robots allow precise position and orientation estimates to be made. The Centeye lenses ideally block infrared light but the largest source of leakage was from the lens mounts. The mounts were rapid prototyped in plastic which is translucent to IR. This leakage would be greatly improved with better lens mounts fabricated from other materials.
- **Failure of the projection approximation**—We assumed that the Centeye lenses act essentially as pinhole cameras in the determination of the projection matrix. This is not true in general. A more accurate camera calibration should be done to determine the correspondence from global coordinates to pixel coordinates (and therefore flow correspondence).
- **Inaccurate mounting of the lenses and sensors**—Due to difficulties encountered during fabrication, the lens mounts were imperfect and did not hold the lenses at the correct focal length and orientation. In particular, the lenses were threaded but the precision of the rapid prototyping machine was insufficient to preserve the threaded receptacle for the lens. The lack of lens precise mounting breaks down some of the assumptions of the theoretical work. The field of view was determined from just one sensor, and the assumption was made that all were identical. Again, better lens mounting would solve this major issue.

8. Future Work

There are a number of improvements that can be made to this experiment. First, better lens mounting would greatly improve the quality of the optical flow measurements. This would involve fabrication with a material that is opaque to infrared light. A higher precision prototyping machine could be used to create threads for the lenses. Better noise reduction and faster sampling of Centeye data would reduce the estimate error. Using wider angle lenses and more sensors would eliminate many missed detections by reducing the number of dead spots.

A missed detection often occurs in cases where a moving object is not seen simultaneously by three sensors, but rather one-by-one in quick succession. Such a situation may still provide enough information to estimate the target po-

sition by making forward projections of the limited flow data in spots where the object cannot be detected. Another way to mitigate this issue is to add more sensors to each robot and use lenses with a larger field of view.

As shown by the results, the estimated position alone does not give a full understanding of the path of the object. The position estimate could be combined with scene flow estimates to obtain a more accurate tracking. For instance, a Kalman filter could be used to leverage both sources of information and integrate incremental motion into path data.

In this experiment, optic flow data from each robot was relayed to a single base station. In scenarios with a larger number of robots that are dispersed farther, robots may use their collective on-board processing capability to compute target estimates. In this case, a communication scheme with internetworking between robots would be required. This experiment assumed knowledge of the position and orientation of the robots provided by Vicon motion tracking. Future work could involve estimating the locations of swarm members using optical flow for situations where this information is not precisely known. This would also require networking and coordination between swarm agents.

Finally, while motion in this experiment was constrained to the ground plane, the method can be extended to three- dimensional motion. The above derivations assumed the general case of 3D scene flow. The main reason for the experimental limitation was the small vertical field of view of the sensors. Due to the use of three square flow computation windows, the horizontal sensitivity covered about 25.7˚, while the vertical extent was only 8.57˚. The use of additional sensors would reduce this limitation.

Acknowledgements

This work was performed by Andrew K. Massimino at the U.S. Naval Research Laboratory (NRL) in the Laboratory for Autonomous Systems Research (LASR) under the mentorship of Donald A. Sofge through the Naval Research Enterprise Internship Program (NREIP) administered by American Society for Engineering Education (ASEE). The views expressed in this paper are strictly those of the authors and do not reflect the views of NRL, NREIP, or ASEE. We appreciate the support given by NREIP and numerous individuals in LASR at NRL who provided information and helpful discussions to help make this happen.

References

[1] Vedula, S., Rander, P., Collins, R. and Kanade, T. (2005) Three-Dimensional Scene Flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, 475-480. https://doi.org/10.1109/TPAMI.2005.63

[2] Low, T. and Wyeth, G. (2005) Obstacle Detection Using Optical Flow. *Australasian Conference on Robotics and Automation. Proceedings. ACRA* 2005, Australian Robotics & Automation Association (ARAA).

[3] Green, W.E., Oh, P.Y. and Barrows, G. (2004) Flying Insect Inspired Vision for Autonomous Aerial Robot Maneuvers in Near-Earth Environments. 2004 *IEEE Inter-*

national Conference on Robotics and Automation, **3**, 2347-2352. https://doi.org/10.1109/ROBOT.2004.1307412

[4] Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J. and Szeliski, R. (2011) A Database and Evaluation Methodology for Optical Flow. *International Journal of Computer Vision*, **92**, 1-31. https://doi.org/10.1007/s11263-010-0390-2

[5] Srinivasan, M.V. (1994) An Image-Interpolation Technique for the Computation of Optic Flow and Egomotion. *Biological Cybernetics*, **71**, 401-415. https://doi.org/10.1007/BF00198917

[6] Thompson, S. (2017) Direct Linear Transformation (DLT). BYU University Lecture Notes. https://me363.byu.edu/sites/me363.byu.edu/files/userfiles/5/DLTNotes.pdf