Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-16 (2016) 359–364

Feature-based visual tracking for agricultural implements

Trevor P. Stanhope*, Viacheslav I. Adamchuk**

*McGill University, Montreal, QC (Tel: 514-398-8712; e-mail: trevor.stanhope@mail.mcgill.ca).
**McGill University, Montreal, QC (Tel: 514-398-7657; e-mail: viacheslav.adamchuk@mcgill.ca)

Abstract: Systems which utilize implement-mounted cameras for machinery feedback, such as row crop cultivators and sectional sprayers, can be upgraded to provide high-accuracy ground speed and tracking data using visual tracking algorithms. Vector data produced by visual tracking can be incorporated into control systems to compensate for implement dynamics in complement with RTK-GNSS receivers and other sensors. Variations of the SURF, SIFT, and ORB feature-descriptor algorithms were evaluated using a dataset of 640×480 pixel videos on six surfaces (gravel, asphalt, grass, seedlings, residue, and pasture) for speeds from 1 to 5 m/s. Feature-descriptor matching of consecutive video frames was tested using two methods of the k-Nearest Neighbors (kNN) algorithm: (1) 1NN with cross-checking, and (2) 2NN with the ratio-test. Ground speed and tracking direction were calculated using a fast histogram filter to reject outliers between frames. Compared to RTK-GNSS, ORB with CLAHE pre-processing (CLORB) and 1NN cross-checking was found to be the most robust with respect to real-time applications. For 95% of measurements, CLORB achieved an error of 0.23 m/s. Similar accuracy was achieved with SURF, U-SURF, and SIFT, but CLORB was capable of producing vector data in real-time (approximately 25 Hz), whereas SURF, U-SURF, and SIFT were only capable of 15 Hz or less.

Keywords: Computer-vision, feature description, visual tracking, control systems.

1. INTRODUCTION

An essential sensor feedback required by many agricultural control systems is real-time estimation of the vector data for the machinery. Several different techniques exist for acquiring vector data, each of which have advantages and disadvantages depending on the environment and required level of precision and accuracy.

For most applications, real-time kinematic (RTK) global navigation satellite system (GNSS) receivers are the preferred method for determining both vector and positional data of the vehicle. Field operations which require high accuracy and precision, such as tractor auto-steering, utilize RTK-GNSS receivers which are capable of up to 1 cm precision (Gan-Mor, 2007). With respect to implements, the position of the toolbar relative to the tractor is dynamic, e.g. for pull-type and articulated-hitch implements, and therefore a secondary receiver is required on the implement itself, increasing the total cost. Another common method for determining ground speed is mechanical detection with rotary encoders in contact with the soil surface, commonly known as fifth-wheels. Fifth-wheels are advantageous because they are inexpensive and provide a sufficient degree of accuracy in many applications, e.g. seed metering. However, fifth-wheels are limited in that they only provide 2 degrees-of-freedom, i.e. acceleration and velocity along a single axis. Additionally, as contact sensors, fifth-wheels are prone to errors due to slippage on some agricultural surfaces, such as tilled soil (Tompkins, 1988).

Visual tracking using cameras is an alternative, non-contact approach which is unaffected by slip (Thansandote, 1977) and provides 4 degrees-of-freedom with a single camera (monocular) system. Visual tracking refers to the process of determining the movement of a vehicle by analyzing consecutive images captured by an on-board camera. Visual tracking is valuable for many robotics applications, such as simultaneous localization and mapping (Nistér, 2004), and can be incorporated into systems which utilize cameras for plant detection, such as inter-row cultivators or sectional sprayers. However, visual tracking is computationally intensive, and with respect to real-time applications, i.e. embedded systems, it is important that the tracking algorithm be accurate yet computationally efficient.

The **goal** of this study was to compare several visual tracking algorithms in the context of agricultural applications. Vector data of an agricultural vehicle was calculated for each algorithm for speeds from 1 to 5 m/s on six surfaces to assess its reliability with respect to computational efficiency, accuracy and robustness.

2. MATERIALS AND METHODS

2.1 Feature Detection and Description

A robust approach to visual tracking is to identify distinctive features which can be matched between consecutive images (Gauglitz, 2011). A feature, also known as a keypoint, is a sub-region of an image which exhibits maximum variation within its neighborhood. Keypoints are identified within an image using algorithms known as feature detectors. Once features have been located, each feature and its neighborhood must be described mathematically by a process known as

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.10.066

feature description. After features and descriptors have been generated for both images, a classification algorithm such as k-Nearest Neighbors can be employed to find matches between the two sets of descriptors (Boiman, 2008).

2.2 SIFT

The Scale-Invariant Feature Transform (SIFT), proposed by Lowe et al. (2004), is a highly effective algorithm which incorporates both feature detection and description. SIFT is based on the Laplacian of Gaussian (LoG) which acts as a blob detector for features of different sizes by varying a scaling parameter (σ). However, LoG is computationally expensive, so the SIFT algorithm detects local extremas of the image by applying a Difference of Gaussians (DoG) filter as an approximation of LoG. SIFT is a very popular algorithm and performs with high accuracy compared to other feature-descriptors (Rosten, 2010).

2.3 SURF

Speeded-Up Robust Features (SURF) was developed as an improvement upon the SIFT algorithm (Bay, 2008). Similar to SIFT, SURF is a scale-invariant and rotation-invariant feature detector-descriptor based on LoG. Convolution with Gaussian second order derivatives in LoG is computationally expensive, so SURF improves computation speed by approximating LoG with box filters and using pre-computed integral images.

2.4 ORB

Oriented FAST and Rotated BRIEF (ORB) was created by Rublee et al. (2011) as a binary feature-descriptor algorithm with comparable performance to SIFT and SURF but faster computation speed. ORB uses the FAST feature detector to find keypoints, and then the Harris corner measure is applied to filter for the n-best points. For rotation-invariance, ORB assigns orientations to each FAST keypoint, a process known as Oriented FAST (oFAST). A second improvement in ORB was the inclusion of the Rotation-Aware BRIEF (rBRIEF) algorithm, a binary-descriptor algorithm based on BRIEF which adds integrates orientation into BRIEF keypoints by computing the intensity weighted centroid of a 15×15 region around each keypoint where the direction of the vector from each keypoint to its centroid gives its orientation.

2.5 CLAHE

Histogram equalization is a pre-processing technique for adjusting image intensity to enhance contrast and edge definition. However, global histogram equalization can cause degradation of some features. To address this, adaptive histogram equalization (AHE) partitions the image into equally-sized tiles, e.g. 8×8 tiles (Zuiderveld, 1994). For each tile, the sub-histograms are calculated and used to equalize each tile independently. However, AHE is still prone to amplification of noise in homogenous regions.

Contrast limited adaptive histogram equalization (CLAHE) was proposed by Pizer et al. in 1998 to mitigate noise amplification. CLAHE uses the slope of the transformation function in the neighborhood of a given pixel to perform contrast amplification whereby the amplification is proportional to the slope of the neighborhood's cumulative distribution function (CDF). CLAHE limits amplification by clipping the histogram at a predefined value, known as the clip limit, before computing the CDF. As a pre-processing technique, CLAHE significantly improves edge definition (Figure 1), but adds computational expense. CLAHE has been demonstrated to be an effective technique for real-time contrast enhancement (Yadav, 2014), and has been implemented in the histogram-binary combined corner enhancement (HBCCE) algorithm which was shown to improve the repeatability of feature detectors from 10% to 40% with negligible reduction in speed (El Harraj, 2015).

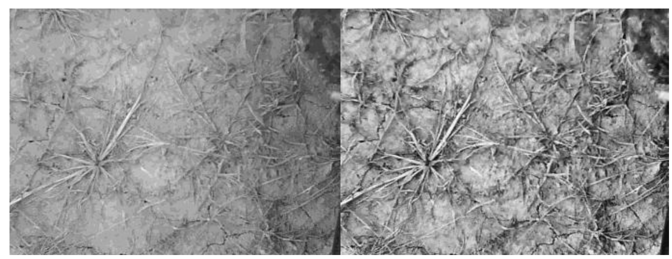

Fig. 1. Comparison of grayscale (left) and CLAHE (right).

2.6 kNN Matching

k-Nearest Neighbors (kNN) is a pattern matching algorithm commonly used in computer-vision. With respect to visual tracking, kNN is used to match keypoints based on the sets of descriptor vectors of two images. Applications of the kNN algorithm in computer vision commonly search for either one or two neighbors, referred to as 1NN and 2NN, respectively. Lowe et al. (2004) proposed the ratio-test to filter matches produced by 2NN where the distance metric of each neighbor is compared, and the match is accepted if the ratio between the two is less than a threshold (a ratio of 0.7 is typical). With respect to 1NN, a method known as cross-checking is often used to eliminate poor matches. Cross-checking is based on the principle that the distance from a point in a set (A) may be nearest to a point another in set (B), but when starting from the point in B, another point in A may be a nearer neighbor. Cross-checking computes 1NN for both directions, A-to-B and B-to-A, and a match is only accepted if points in both sets are nearest neighbors for both directions (Figure 2).

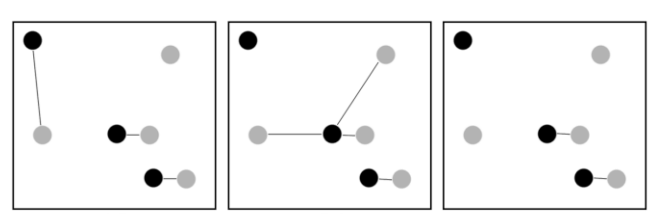

Fig. 2. Demonstration of 1NN matching with cross checking.