

# International Journal of Logistics Research and Applications

A Leading Journal of Supply Chain Management

ISSN: 1367-5567 (Print) 1469-848X (Online) Journal homepage: <https://www.tandfonline.com/loi/cjol20>

## Automatic extraction of 1D barcodes from video scans for drone-assisted inventory management in warehousing applications

Lichao Xu, Vineet R. Kamat & Carol C. Menassa

To cite this article: Lichao Xu, Vineet R. Kamat & Carol C. Menassa (2018) Automatic extraction of 1D barcodes from video scans for drone-assisted inventory management in warehousing applications, International Journal of Logistics Research and Applications, 21:3, 243-258, DOI: [10.1080/13675567.2017.1393505](https://doi.org/10.1080/13675567.2017.1393505)

To link to this article: <https://doi.org/10.1080/13675567.2017.1393505>



Published online: 23 Oct 2017.



Submit your article to this journal [↗](#)



Article views: 403



View related articles [↗](#)



View Crossmark data [↗](#)



Citing articles: 1 View citing articles [↗](#)



# Automatic extraction of 1D barcodes from video scans for drone-assisted inventory management in warehousing applications

Lichao Xu, Vineet R. Kamat and Carol C. Menassa

Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI, USA

## ABSTRACT

The widespread use of barcodes has significantly contributed to accurate, efficient and economic inventory management in warehouses and distribution centres. However, its efficiency has always been limited by the primary method of reading barcodes with a handheld laser scanner. Compared with this reading by line-of-sight at close proximity, vision-based barcode reading algorithms can further improve efficiency, particularly if accompanied by automated data collection platforms such as drones. This paper introduces algorithms that are able to automatically extract barcodes from video data, and verifies their feasibility and promise for inventory management in warehousing applications. Three key techniques corresponding to different recognition levels are proposed: For a known barcode region, a Harris corner detector and Hough transform-based algorithm is applied to quickly estimate the angle by which the frame area needs to rotate to orient the bars vertically for information extraction. Then, the idea of exploiting connectivity and geometry property of barcode areas is proposed to directly recognise multiple barcode regions in a single video frame to eliminate reading difficulties resulting from interactive influence of multiple juxtaposed barcodes, and to save computation time by only processing frame areas of interest for valid barcodes. In addition, a histogram difference-based fast extraction strategy is designed to further improve efficiency by reducing duplicate information processing. Finally, the performance of each technique is evaluated by analysing video data from a large logistics warehouse, demonstrating satisfactory performance in inventory management applications.

## ARTICLE HISTORY

Received 24 January 2017

Accepted 10 October 2017

## KEYWORDS

: 1D barcode; Hough transform and corner detector; connectivity; histogram; key frame; logistics

## 1. Introduction

One-dimensional (1D) barcodes are widely used for product identification and inventory management in supply chains and retail transactions. Compared to 2D barcodes (e.g. Quick Response codes), even though 1D barcodes can only contain basic information, their redundant design provides improved readability in situations of partial tear or abrasion, making them robust and reliable in harsh industrial environments (Kato, Tan, and Chai 2010). Utilisation of 1D barcodes has thus represented a significant milestone in automated stock and inventory management. Notwithstanding, barcode scanning is largely a human effort intensive process since a worker typically has to manually focus a barcode scanner (handheld or equipped on a forklift, Figure 1) on all codes to be read one by one, and from close proximity. This makes their application suitable to situations where relatively small numbers of barcodes must be scanned such as store checkout lanes, but



**Figure 1.** Manual barcode scanning in typical warehouse environments.

not in situations where large numbers of laterally distributed barcodes have to be regularly scanned for inventory management or stock-keeping in warehouses or distribution centres.

Long-range barcode scanners offer a potential solution in such industrial environments. However, their applicability is limited due to several practical issues that include small viewing angle (i.e. closely spaced racks result in too small viewing angles for reading barcodes at high places), and sight occlusion (i.e. product barcodes are occluded by other products or shelves and rack components). Thus, even with long-range barcode scanners, a barcode scanner has to get within close vicinity of all codes that need to be scanned, resulting in more practical use of standard-range barcode scanners having a range of 6–24 inches (Semicron Systems). In addition to significant scanning workloads, workers in warehouse-like environments face several other challenges. For instance, for all products stored above ground level on racks or shelves, workers have to use ladders, lifts, or forklifts to visually access and scan barcodes (Figure 1), significantly increasing risks of falls or other injuries and causing general waste of energy in operating forklifts or other lift platforms.

Besides such issues, the large scale of effort involved in barcode scanning in warehouses also presents a strong case for automation. For example, a typical warehouse supporting a manufacturing supply chain has hundreds of sections and thousands of racks, most of which hold high turnover products (i.e. products come in and go out quickly over a matter of hours or days). In this situation, inventory has to be scanned multiple times in a week or sometimes at least once a day, which is a very laborious and time-consuming job demanding a team of employees. A promising idea towards automation of such inventory management is to mount a barcode scanner on a drone and manually fly the drone to scan barcodes.

As is estimated in (Pons 2014) in a warehouse environment, a drone operator can scan 119 times faster than a person using a handheld barcode scanner. This solution can not only greatly improve operation efficiency, but can also liberate workers from this laborious and dangerous work while also conserving energy (the energy consumed by a flying drone carrying a barcode scanner is much less than that needed for lifting a heavy forklift platform). However, the idea to scan barcodes with a drone-mounted barcode scanner is in essence still a line-of-sight scan, which requires the drone to pause momentarily in front of each barcode for reading (Pons 2014). On the one hand, this stop-and-go scan pattern dictates that the drone has to fly at a very low speed making the scan

process very time-consuming. In addition, the high positioning accuracy requirement for drone hovering presents a major challenge for current self-navigation algorithms and further limits its application in completing automatic scans.

2. Technical approach and related work

To mitigate these issues, the proposed method scans barcodes with a video camera that can both enable area-of-sight and reduce the requirement for positioning accuracy, making it suitable for a completely automatic scan at a relatively high speed. With the help of vision-based barcode reading and drone navigation algorithms, our overall solution is to automatically scan a warehouse with a drone-mounted camera and extract barcode information from the obtained video, while requiring little human assistance for monitoring, verification, and maintenance. Figure 2 presents an overview of the whole system. In this overall automatic scan solution, to automate the whole process, the barcode scanning task is divided into two low-level tasks of automatic video data collection and automatic barcode extraction, which make up the task layer. These two tasks are further implemented and supported by the underlying algorithms listed in algorithm layer. In addition, right above the task layer, humans are only responsible for high-level tasks of monitoring, evaluating and maintaining the two low-level sub-tasks, such as drone state monitoring, barcode verification and system maintenance, which make up the human layer.

This paper primarily focuses on techniques for extracting barcodes from arbitrary sequences of scanned video data (enclosed by the dashed box in Figure 2), which is a key component of our overall solution. By building on existing well-developed barcode decoding methods, our algorithms focus on improving recognition rate and efficiency by developing methods for preparing easy-to-decode barcode regions. In particular, our method efficiently processes video sequences with thousands of frames containing an unspecified number of barcodes oriented in arbitrary directions and located in any part of the frames.

The steps followed to obtain such ideal barcode regions from a video scan are shown in Figure 3. To efficiently process multiple frames with overlapping scenes, in the first step, fewer frames (called key frames here) that do not miss any barcode information need to be selected for further processing. Then the problem that remains is how to read multiple barcodes from a single key frame. This can be further solved by the following two steps: recognising potential barcode regions in a frame, and

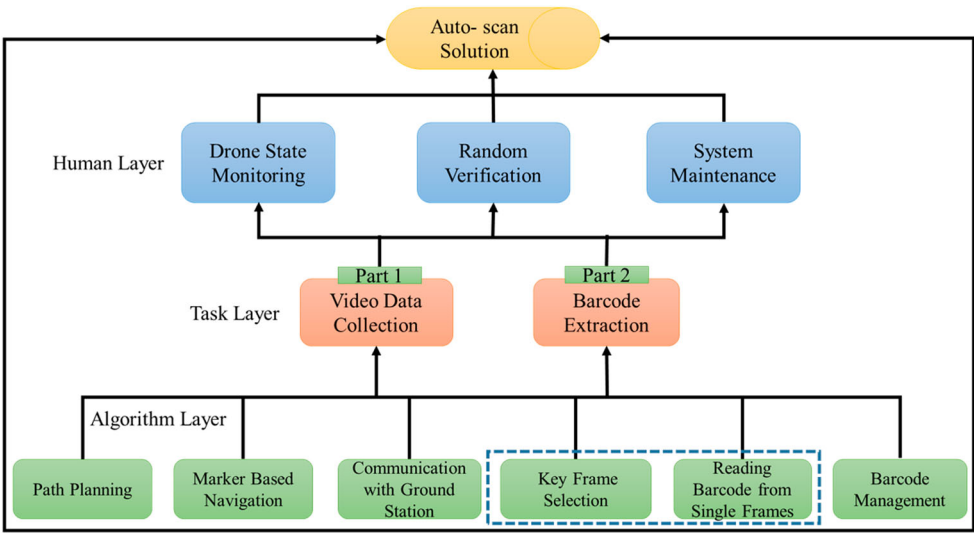
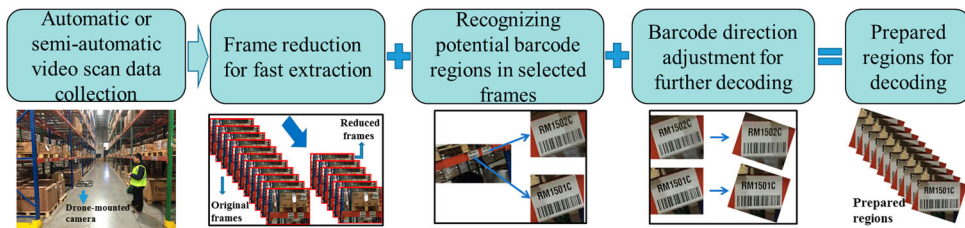


Figure 2. Automatic scan solution overview.



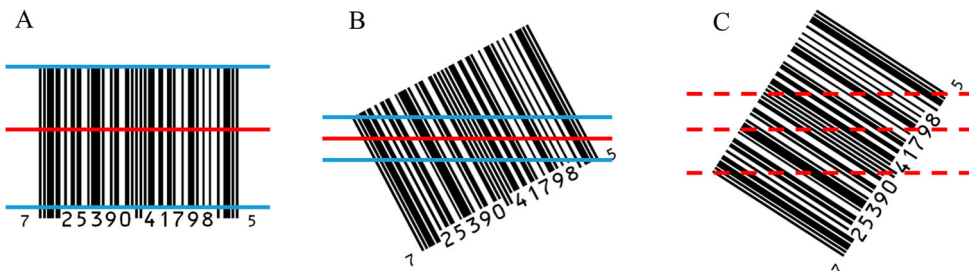
**Figure 3.** Process to prepare barcode regions for existing decoding algorithms.

adjusting the direction of each of these barcode regions for decoding. In an effort to provide a clear description of this paper's contributions, these three steps will be discussed in reverse order (also the order in which they were developed) compared to the sequence shown in [Figure 3](#).

With the popularity of barcodes as a tagging system, significant prior work has been done on reading barcodes using computer vision-based methods. Initially, barcode reading algorithms were mainly implemented on desktop computers based on domain transformation, such as the Fourier transformation or the Hough transformation as proposed in (Muniz, Junco, and Otero 1999). Compared with domain transformation, reading algorithms using scanlines need less computational resources and can effectively run on mobile devices, which has resulted in their rapid development in recent times (Ohbuchi, Hanaizumi, and Hock 2004; Adelmann, Langheinrich, and Floerkemeier 2006; Gallo and Manduchi 2011). In addition, there already exist some algorithms to address challenging barcodes like low resolution and blurring from motion or being out of focus (Liyana 2007; Gallo and Manduchi 2009).

However, most of these algorithms are only applicable to vertical or approximately vertical barcodes ([Figure 4\(A\)](#)), which greatly limits their wide application in practice. In addition, even though some commercialised algorithms such as the ClearImage Barcode Reader SDK (referred to as ClearImage hereinafter) ('ClearImage SDK' 2005) already provide certain abilities to read rotated barcodes ([Figure 4\(B,C\)](#)) from an image, their performance is significantly limited for blurred images. Instead of focusing on decoding a barcode itself, this component of our proposed solution focuses on estimating barcode orientation in an image in an effort to make existing decoding algorithms more effective.

Many methods have been developed for this problem. In (Adelmann, Langheinrich, and Floerkemeier 2006; Wachenfeld, Terlunen, and Jiang 2010), barcode direction is determined by the intersection of scan lines and bars. In (Zhang et al. 2006), the main direction is estimated by using an orientation filter in four directions. Besides, Hough transformation has also been used (Zamberletti et al. 2015; Wang et al. 2016). However, these methods are either not robust to detect arbitrarily rotated barcodes, or are not time efficient, or are too complicated to be implemented. Taking into consideration that Hough transformation alone does not work well in situations of complex spatial



**Figure 4.** Readable regions in different angular states.



context or high image noise, we propose to use corner detection and Hough transform together to implement a robust, efficient and easier solution.

With the development of barcode reading techniques, barcode localisation algorithms have also experienced significant progress. Compared with finding a single barcode in an image (Juett and Qi 2005; Bodnár and Nyúl 2012; Katona and Nyúl 2012, 2013), we are more interested in the ability to simultaneously recognise multiple barcodes of any size and orientation, which is more suitable for the motivated application in warehouse settings. Based on morphological operations, Lin, Lin, and Huang (2011) realised their barcode detection algorithm by background small cluster reduction. Other work such as Bodnár and Nyúl (2013) used image primitive operations and detected barcodes relying on distance transformation. Such algorithms rely on basic image operation, and their performance is sensitive to threshold parameters which are not easy to find. Besides, methods using machine learning (Zamberletti, Gallo, and Albertini 2013) or Maximal Stable Extremal Region (Creusot and Munawar 2015) detection have also been proposed for this problem.

All of these methods have either been tested with non-public image datasets or public image datasets where the barcodes take up a large portion of the whole image in each frame. In addition, the images used typically have a simple background and appear in specific patterns, thereby providing few insights about these methods' performance in complex practical environments. In order to address this, we propose a barcode region detection algorithm based on connectivity and geometry properties of barcode areas, which can work effectively and efficiently on real warehouse videos as well as find potential barcode regions beyond the reading ability of subsequent decoding algorithms such as ClearImage that is chosen in this work for decoding barcodes. It should be noted that some primitive image operations are used in our method, but in our case, finding appropriate thresholds is easier for extracting barcodes from consecutive frames under similar illumination conditions. The difficulty is how to get rid of a large number of redundant frames to improve efficiency. This problem is addressed by the last technique introduced in this paper.

Selecting fewer key frames that can represent the content of a video can not only help improve barcode reading efficiency but can also assist human verification. Such techniques are usually used for movie abstraction (Li and Jay Kuo 2003; Ott et al. 2007). The difference and difficulty of our case are that there are lots of similar and repetitive scenes in a warehouse which makes a selection using features very difficult, thereby rendering feature-based algorithms ineffective even if they work well for traditional movie abstraction purposes (Steadly, Pal, and Szeliski 2005; Brown and Lowe 2007). For application in this challenging environment, we propose to choose key frames based on histogram difference. This algorithm enables the use of colour information from the whole region of a frame which makes it more robust compared with extracted features. In the next section, each of these three algorithms that help improve barcode extraction from video frames are discussed in detail.

### 3. Technical approach details

In this section, three algorithms are proposed to improve the process of extracting barcodes from arbitrary video frames corresponding to the three steps in Figure 3, that is, barcode direction estimation, barcode region detection, and key frame selection.

#### 3.1. Barcode direction estimation

General methods of decoding barcodes from images are based on the encoding rule to find the best representation of the binary patterns sampled along scanlines, which usually move from top to bottom of barcode areas. To be able to read out barcode information, there has to exist at least a readable region in which a horizontal scanline intersects with all bars. In addition, since the scanline usually

moves down with a fixed distance at each step, the larger the readable region is, the more chance that the barcode can be successfully read. For this reason, algorithms proposed in (Chai and Hock 2005; Wachenfeld, Terlunen, and Jiang 2008) are only limited to processing situations when the bars of barcodes are close to the vertical direction. But similar algorithms would become more valuable, and their application would significantly broaden if, prior to decoding barcodes, the images could be preprocessed by an angle-aware rotation through which they can be adjusted to bring them to the ideal state shown in Figure 4(A) from prior states such as Figure 4(B or C).

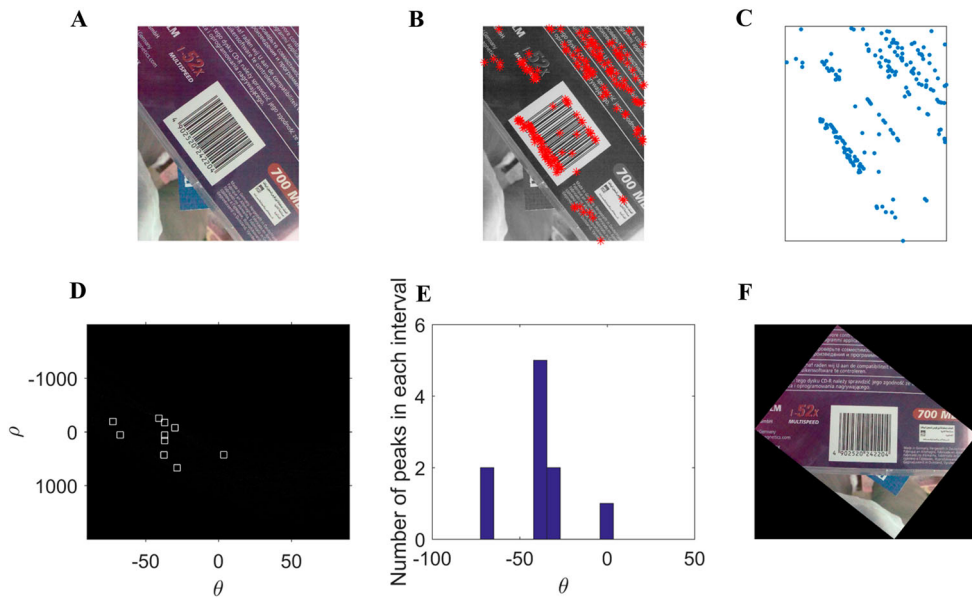
In Figure 4, where solid blue lines represent the margin of readable regions, solid red lines represent valid scanlines and red dash lines mean invalid scanlines. A is the ideal state where the barcode can be read from any scanline between the top and the bottom of the barcode. B is a suboptimal state where the readable region still exists but is really small. C represents the worst situation since there is no readable region anymore and the barcode cannot be read out from any horizontal scanline.

To estimate barcode direction, Hough transform is generally used to recognise bar features (straight lines) in the image (Muniz, Junco, and Otero 1999; Wang et al. 2016). Instead of traditional representation of straight lines, it uses the Hesse normal form  $r = x \cos \theta + y \sin \theta$  (Duda and Hart 1972) and thus associates each straight line with a parameter pair  $(r, \theta)$ , where  $r$  is the distance from origin to the straight line to be represented and  $\theta$  is the angle between  $x$  axis and the line passing through the origin as well as perpendicular to the line. It follows that in the  $(r, \theta)$  space, representation of all straight lines passing through point  $(x, y)$  forms a sinusoidal curve and the intersection of such curves gives the  $(r, \theta)$  parameter of the straight line connecting the points corresponding to the intersected curves. Intersection multiplicity values at different  $(r, \theta)$  parameters form a parameter space matrix (also called Hough space) whose rows and columns correspond to  $r$  and  $\theta$  values, which describes the voting scores for all  $(r, \theta)$  values in the space (Duda and Hart 1972).

With this benefit, straight lines can be found by selecting the parameter points in Hough space with big intersection multiplicity values. Since such intersection multiplicity values are found using a voting strategy, Hough transformation enables discontinuous lines (due to noise, reflection, etc.) to be recognised. However, it was found in our experimentation that Hough transform alone cannot work robustly if image noise is relatively large or repetitive patterns appear in a barcode's background. In such situations, the barcode direction is usually drowned by the noisy directions, making it hard to be distinguished. Some researchers recently proposed to identify a characteristic pattern in  $(r, \theta)$  space using machine learning (Zamberletti, Gallo, and Albertini 2013; Zamberletti et al. 2015), but such solutions need significant training data preparation effort.

Noticing that a large number of corners exist at the bar ends, the idea here is to first recognise these corners and use Hough transformation on such corner features instead of on the original image. The main reason why this works is that the corners extracted at bar ends in most cases are arranged perfectly in straight lines with high density, which makes the straight lines passing through these points have the largest votes in Hough transform and can be easily and robustly found. From a computer vision perspective, a corner point should be easily recognised by looking at intensity values within a small window, and small shift of the window in any direction should yield a large change in appearance. To find those corners at the end of the bars, Harris corner detector is applied, which finds corner points by evaluating weighted squared sum of intensity change in a small window and approximating the intensity change in the first order (Harris and Stephens 1988).

The detailed results of estimating barcode direction are shown in Figure 5. It first converts an original RGB image to greyscale image and finds corners with Harris corner detector (Figure 5(B)). In Figure 5(B), it is clear that a large number of corner points at bar ends are detected as shown in the visualisation in Figure 5(C). Then Hough transform is applied on these corner points, and Hough peaks (limited to at most 20) are found in Hough space (Figure 5(D)). After that, the peaks are put into 10 even spaced bins between the minimum and maximum value of the  $\theta$  coordinate of the peaks, and the centre of bins including maximum number of peak points is considered as the direction perpendicular to bar direction (Figure 5(E)). In addition, the barcode is also rotated to the ideal state by the corresponding angle (Figure 5(F)).



**Figure 5.** Procedures to estimate barcode direction. (A) Original image containing a barcode (Zamberletti et al. 2010). (B) Corner points detected by Harris corner detector. (C) Visualisation of Harris corner points. (D) Hough peaks in  $(r, \theta)$  space ( $\rho$  represents  $r$ ). (E) Histogram of Hough peaks. (F) The result of barcode rotated to the ideal state.

This algorithm is straightforward to implement and works robustly with one single barcode. For images that include multiple barcodes, potential barcode regions have to be identified and selected first before this direction adjustment algorithm can be applied. This aspect of our proposed method is discussed next.

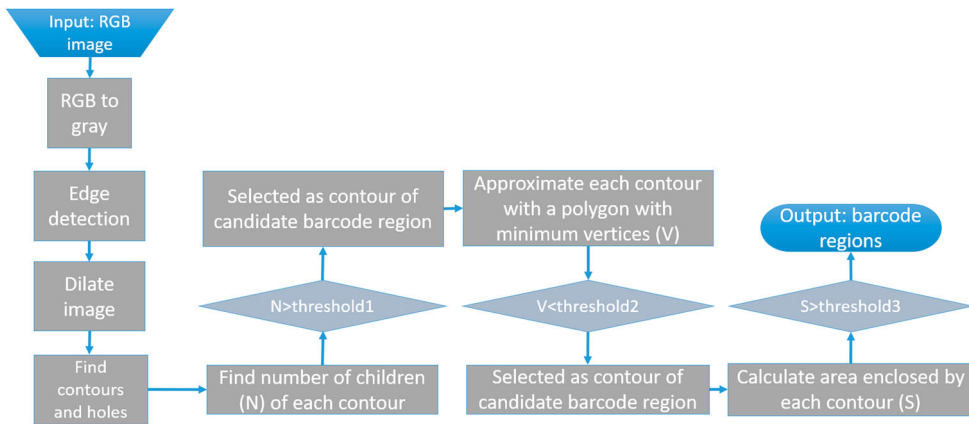
### 3.2. Barcode region detection

In the present time, there is little difficulty in recognising a barcode with a mobile phone camera or reading barcodes from most public barcode datasets when barcodes are usually intentionally focused on and occupy relatively large part of the whole image. Different from such situations, the difficulty in our situation arises mainly from multiple barcodes with unexpected direction existing in one frame and a much smaller portion of separate barcode regions. The fallout of this situation is that in the decoding phase, significant time has to be spent on searching recognisable barcodes in the whole image. Our proposed idea is to help find potential barcode regions for the decoding algorithm and thus save time by avoiding the processing of non-value-adding regions.

To identify barcode regions in an image, the most intuitive idea is to see whether a certain number of parallel straight lines come together in a local region. However, detecting bars is very sensitive to image noise and similar line structures in the background, which makes it unreliable to use in practice. Instead of detecting straight lines, we propose to recognise barcode regions through their following properties: connectivity, quadrilateral contour as well as least area to be decoded, which is more robust, scale-invariant and applicable to find multiple barcodes. In order to better articulate how this process works, the flowchart of this barcode region detection algorithm and its results after key steps on a given image stitched by four different images from (Zamberletti et al. 2010) are shown in Figures 6 and 7, respectively.

As is shown in the flowchart (Figure 6), the RGB image is first converted to a grey image and then edge detection (makes barcode regions convenient to be detected by highlighting their edges and bars included) and dilation (helps close some discontinuous parts in edges of barcode regions) are

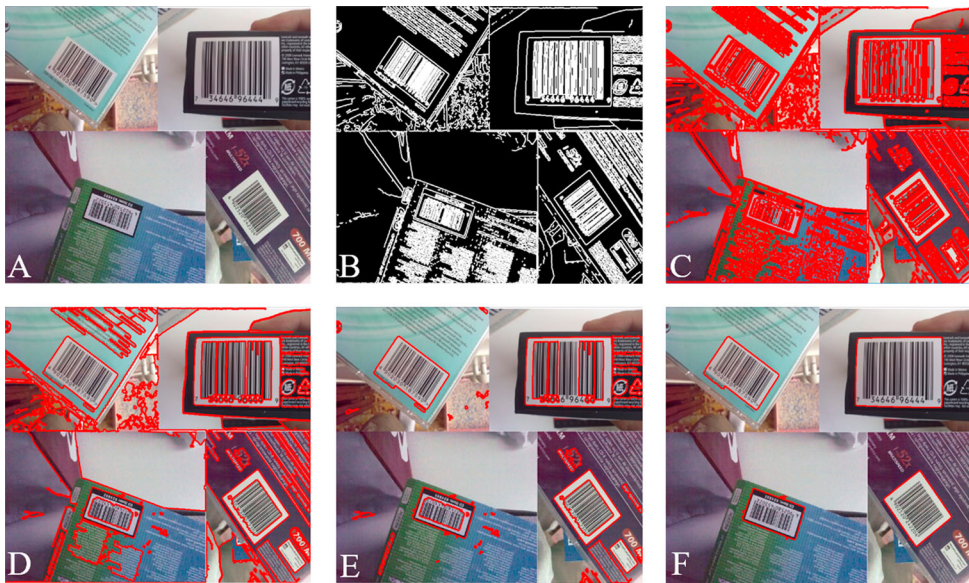




**Figure 6.** Algorithm for barcode region detection algorithm (implemented in OpenCV).

performed. This result is shown in Figure 7(B), from which it can be observed that edges of barcode regions approximately emerge because of grey change from the background to barcode regions. Based on Figure 7(B), all the contours and holes can be searched out (Figure 7(C)).

Before discussing the core of the algorithm, some terms have to be explained first. If one region is completely inside another region, this region is another region's child and another region is this region's parent. According to this definition, one region can have multiple children or/and multiple parents. In order to select the most possible barcode regions from these contours, three steps are needed. The first step is to eliminate contours with no or small number of children by setting a threshold of children number of each contour (Figure 7(D)). The primary reason is that barcode regions usually contain more children due to the multiple bars contained within. Then, considering that a barcode region is usually quadrilateral, if a polygon is used to approximate it with certain accuracy, the polygon should not have many vertices which is limited by threshold2. After this step, only



**Figure 7.** Visualisation of results after key steps. (A) An original image including barcodes with different backgrounds. (B) The result of edge detection and dilation. (C) All the contours and holes found. (D) The result of first selection by children number  $N$ . (E) Result of second selection by vertices number  $V$ . (F) Result of final selection by area  $S$ .

the contours that have relatively regular shapes or very small areas are left as shown in [Figure 7\(E\)](#). At last, the final result ([Figure 7\(F\)](#)) is given by eliminating invalid barcode regions with the area less than threshold3 which makes it difficult to read by the decoding program.

For the specific example above, this algorithm works well to recognise all the barcode regions, but several points still need to be emphasised. One observation is that edge detection has to be applied here because backgrounds of different barcodes in the given image result from the combination of four separate images. However, for real warehouse environments (e.g. [Figure 11](#)) where the background of barcode areas is relatively uniform, this operation can simply be replaced by a binarisation operation which uses less time. Another observation is that the processing above does not use any special features of barcodes and just identifies the regions meeting the three restrictions. As a result, the final regions may also include some redundant ones besides the real barcode regions ([Figure 11](#)).

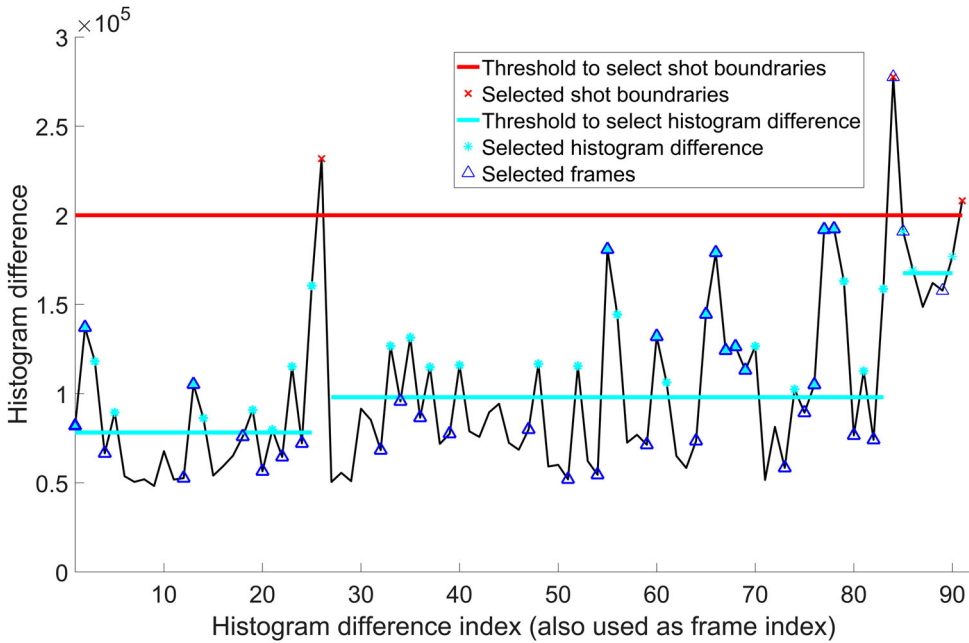
### 3.3. Fast extraction

The two parts introduced above together are sufficient to find multiple barcode areas in an image, adjust their direction, and read them one by one. The problem left here is that when they are applied to process large volumes of video data containing thousands of frames, it will take a long time to extract all the barcodes since each frame has to be processed separately. However, it is clear that not all frames can provide new barcode information, especially when two or more sequential frames generally have a big overlap which contains redundant information. The motivation of our fast extraction algorithm is to use fewer frames (key frames) to identify and extract all barcodes of interest in a shorter time.

Although from a human perspective, a warehouse is a simple repetitive environment that is well-organised for management operations, its repetitive pattern of shelves, boxes, labels and barcodes renders it difficult for algorithms to measure the difference between different or subsequent video frames. Therefore, instead of representing overlaps with a number of matching features, like SIFT (Brown and Lowe 2007) and MOPs (Steadly, Pal, and Szeliski 2005), histogram difference is used in our approach for frame change allowing the use of colour information from all parts of a frame. The procedure followed by our algorithm and the corresponding result on a video (same one from Section 3 but only some front frames are used to explain frame selection results) are shown in [Figures 8](#) and [9](#), respectively. This algorithm works effectively mainly depending on two strategies.

1. Calculate histogram for each frame (after converting to gray images from RGB ones) and histogram differences between two adjacent frames.
2. Find histogram differences which exceed certain value, and with them divide the video into different “shots”.
3. In each “shot”, calculate average of histogram difference and find frames with histogram differences which exceed certain percent of this average.
4. Select the frames which are immediately before the frames producing these founded differences as key frames.

**Figure 8.** Algorithm of histogram difference-based key frame selection.



**Figure 9.** Visualisation of key frame selection.

First considering different levels of histogram difference between sequential frames due to scene change or/and camera moving speed change, the concept of the virtual shot is introduced here to reflect this kind of frame change (even though the video is a one-take shot). Since generally the frames with larger histogram difference have less chance of being readable (due to more likelihood of being blurred), these different shots are considered to be divided by the frames with larger histogram difference. The threshold set here is usually determined by the camera moving patterns which can be easily measured using some consecutive frames.

Another strategy used in this approach is that the final frames selected are not exactly the same as those found in step 3 (Figure 8) but rather are the frames that are immediately before them. The direct effect of this is that frames with smaller, medium or larger histogram difference all have a likelihood of being selected albeit with different probabilities, which makes the final frames manifest enough frame change while keeping a certain number of clearer images to ensure recognition rate (Figure 9).

#### 4. Experimental results and analysis

The previous sections have discussed all proposed techniques – barcode direction estimation, barcode region detection, and fast extraction – that together work effectively to extract barcodes from an arbitrary video scan. In this section, we test our algorithm using video scan data obtained from an active logistics warehouse supporting an automobile manufacturing supply chain located in the metro Detroit area. It should be noted that for testing the algorithm's effectiveness and robustness, the video is taken by a handheld camera under normal illumination conditions (under which the warehouse is normally operated), and intentionally includes continuous left and right shaking of the camera, various shot angles, rapid change of camera moving speed as well as some re-visiting frames. All of these intentional artefacts help simulate the difficulties for barcode extraction which are likely to be bigger than that can be expected when a drone-mounted camera conducts automatic scans across the entire expanse of a warehouse (current commercial camera-equipped drones, such as DJI Phantom 4 (DJI 2017), can easily record video with much better frame stability).

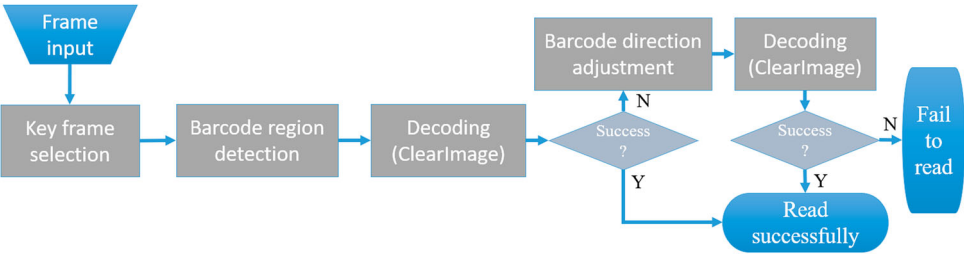


Figure 10. Complete algorithm to read barcode from video scan data.

The entire technical approach including all the components (the three techniques proposed above as well as a chosen barcode decoding algorithm, ClearImage) is shown in Figure 10, and an example of processing a key frame is given in Figure 11.

In the complete solution, with video frame input, key frame selection first helps select fewer number of frames necessary to process (the main parameter is key frame selection threshold). Then in each selected frame, potential barcode regions are picked out by the barcode region detection algorithm (the main parameter is the threshold of binarisation), as in Figure 11 (regions A, B, C and D). In the following decoding procedure, ClearImage is selected for use due to its partial ability to read multiple rotated barcodes from an image.

Generally, most of barcodes are already in the relatively ideal angular state for decoding and considering that an algorithm such as ClearImage can process some rotated barcodes, in order to save time by not rotating unnecessary barcodes, it is first directly used to decode barcodes from the identified regions (Figure 11, region B is successfully read). If this step fails (Figure 11, regions A, C and D), the direction adjustment algorithm is then applied to rotate the failed region to let ClearImage attempt

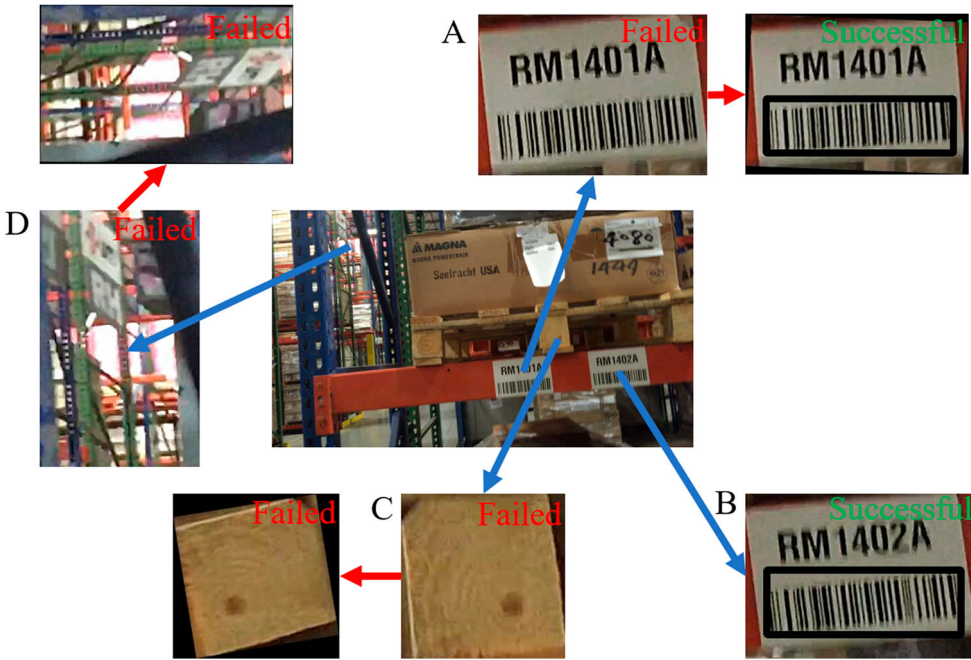


Figure 11. Illustration of processing a specific key frame.

Notes: Blue arrow represents the process of barcode region detection. The red arrow represents the process of barcode direction adjustment. Failed/Successful represents whether a barcode is read out from the current state.



the decode step again to determine if it can be successfully recognised (Figure 11, region A is finally successfully read, but regions C and D still fail).

In practice application, it is usually not necessary to use all the components indicated by the greyed boxes in Figure 10. With benefits of modular design, it is easy to just plug in different combinations of the components and they would be ready to work with other parts of the solution. The users would be expected to choose the best specific solution by testing different combinations of these components and different threshold parameters using some front frames of the video scan data to be processed.

In this experiment, we tested the performance of different combinations of the proposed techniques and analysed how well each technique discussed in Section 3 performs to contribute to better performance of a whole solution.

Different from the previous order used to describe the various components of the proposed algorithms, in this section, it is more convenient to test the barcode region detection algorithm first. For this purpose, only barcode region detection and barcode decoding algorithms are used without key frame selection and barcode direction adjustment. This implies that in this special case, all the input frames will be used for barcode region detection and ClearImage only reads each detected region once without direction adjustment for a second attempt.

The given video totally has 18 different location identifying barcodes recorded in 1968 frames. Such barcodes are usually attached to storage racks to identify the location of goods stored in each cell of the rack (Figure 11). The corresponding experimental result is shown in Figure 12, CImg represents ClearImage, reg\_dec represents our barcode region detection algorithm, and the number in the parentheses behind is the threshold chosen in binarisation. The number on the right of each barcode is how many times the barcode is successfully read from all frames. Successful reads are calculated by adding up all the successful reading numbers in the corresponding column. Recognition rate is the percentage of the barcodes which are read successfully at least once (the total number is 18), which is equivalent to calculating the percentage of storage cell positions that can be successfully located out of 18 different cell positions. Such position information is very important to automatically navigate a drone in a warehouse and provide location information for stored goods.

From Figure 12, it can be observed that in this illumination condition, region detection works best at binarisation threshold from 0.4 to 0.43, when it can help recognise two more barcodes than ClearImage alone and increase recognition rate from 77.78% to 89%. It also helps save time since ClearImage only needs to directly process useful areas of images instead of the whole images, which saves it about 40 s while processing this video. Besides, as binarisation threshold either decreases or increases, recognition rate would always decrease even though it uses lesser time. The reason behind

	CImg(only)	reg_dec(0.2)+CImg	reg_dec(0.3)+CImg	reg_dec(0.4)+CImg	reg_dec(0.43)+CImg	reg_dec(0.5)+CImg	reg_dec(0.6)+CImg	reg_dec(0.7)+CImg
Total frames	1968	1968	1968	1968	1968	1968	1968	1968
RM1401A	27	0	15	24	26	24	2	0
RM1402A	22	0	14	20	21	22	0	0
RM1401B	6	0	1	4	4	2	1	0
RM1402B	3	0	1	2	2	2	2	2
RM1401C	0	0	0	1	2	3	0	0
RM1402C	0	0	0	1	1	1	0	0
RM1501A	0	0	0	0	0	0	0	0
RM1502A	0	0	0	0	0	0	0	0
RM1501B	12	0	0	9	12	7	3	0
RM1502B	17	0	3	14	16	13	15	4
RM1501C	12	1	2	8	8	4	3	0
RM1502C	12	0	3	9	10	7	0	0
RM1601A	8	0	4	10	10	9	0	0
RM1602A	7	0	11	10	8	12	0	0
RM1601B	5	0	7	7	6	0	0	0
RM1602B	4	0	3	2	1	0	0	0
RM1601C	2	0	7	7	5	0	0	0
RM1602C	6	0	1	3	3	4	0	0
Successful reads	143	1	72	131	135	110	26	6
Recognition rate	77.78%	6%	72%	89%	89%	72%	33%	11%
Time cost(sec)	473	308	345	427	433	390	306	239

**Figure 12.** Recognition results of 18 barcodes in the video scan including 1968 frames, with CImg (only) and reg\_dec+CImg (under seven different binarisation thresholds).



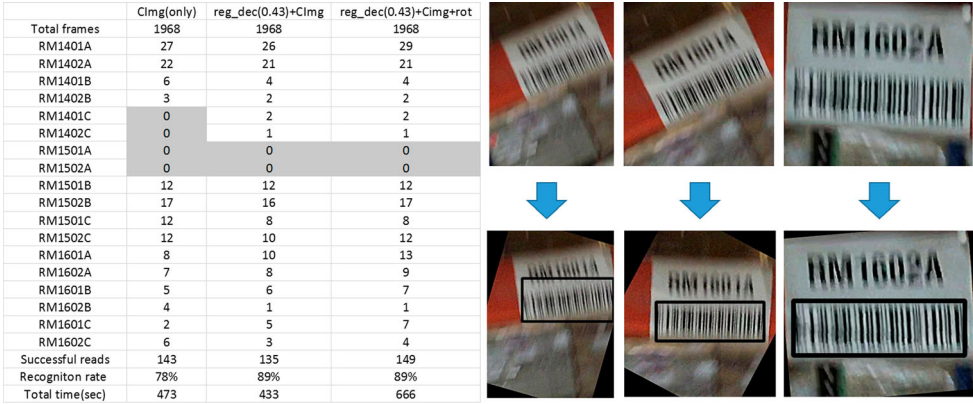
this is that for the specific illumination condition of the given video, the optimal binarisation threshold is around the range from 0.4 to 0.43, which can give richest counters. As the threshold goes up or down away from the optimal value, more and more counter details would be lost. Correspondingly, it takes more time to process richer counters and produces better recognition results, and vice versa.

Another observation is that in most frames barcode regions can be identified correctly, but some of them would likely be omitted when the barcode labels do not have approximately uniform intensity especially due to shadow from surrounding objects (like the square wood beam in Figure 11). However, ClearImage searches for all the valid barcodes in the whole image, such that using ClearImage alone recognised one barcode more times in the frames it appeared compared to other methods which did barcode region detection first (Figure 12).

For most clear images, ClearImage can work well to recognise barcodes with different directions; however, it is much better at reading barcodes in the ideal rotated state, especially for blurred frames which frequently exist in a video scan (Figure 13, right side). The results in Figure 12 are thus expected to improve by adding an extra barcode direction adjustment step to rotate the region to the near ideal state for another read (as shown in Figure 10, the only difference is that all the frames are used here). The left side of Figure 13 shows that the direction adjustment operation can enable 14 more successful reads and makes its total number higher than using ClearImage alone. However, it needs significant more time and does not further help increase recognition rate compared to reg\_dec (0.43)+CImg. In theory, this is a trade-off between how thoroughly barcodes need to be read and how much time can be afforded.

Finally, the key frame selection component is evaluated with different parameter settings. As shown in Figure 14, as the selection threshold parameter increases, the number of frames selected and time cost both keep decreasing. Initially, for parameter 0.5mean, even though fewer frames (1540 out of 1968) are used for further processing, the recognition rate is maintained but the time cost is even greater (449 s > 433 s) than the case without using key frame selection (Figure 12), since in this case time spent on selecting frames is more than the time saving it provides.

Subsequently, when the parameter increases to 0.6mean, the algorithm can obtain almost the same performance in recognition rate and time cost as the case when key frame selection is not used. As the parameter further goes up to 0.7mean, much fewer frames (1350 < 1968) and time cost (409 s < 433 s) are needed while still maintaining the original recognition rate (89%). With this video, the recognition rate starts to decrease as the parameter increases to 0.8mean, which means that accuracy has to be sacrificed if more time is desired to be saved. This is however unique to this specific video.



**Figure 13.** Left side: Recognition results of all 18 barcodes in the video scan including 1968 frames, with Cimg(only), reg\_dec (0.43)+Cimg and reg\_dec+Cimg+rot. rot represents additional rotation as shown in Figure 10. All other abbreviations represent the same as in Figure 12. Right side: several examples of barcodes whose directions have to be adjusted before they can be read. That is, they cannot be read out directly using ClearImage.

Key frame+reg_dec(0.43)+Clmg	Key frame(0.5mean)	Key frame(0.6mean)	Key frame(0.7mean)	Key frame(0.8mean)	Key frame(0.9mean)	Key frame(1mean)
Selected frames	1540	1481	1350	1158	945	669
RM1401A	23	20	15	13	10	7
RM1402A	19	16	11	10	7	4
RM1401B	4	3	2	2	1	1
RM1402B	2	2	1	1	0	0
RM1401C	1	1	1	0	0	0
RM1402C	1	1	1	0	0	0
RM1501A	0	0	0	0	0	0
RM1502A	0	0	0	0	0	0
RM1501B	10	8	5	4	4	3
RM1502B	12	10	6	5	5	4
RM1501C	8	8	7	7	7	5
RM1502C	10	10	9	7	7	5
RM1601A	8	8	8	6	6	4
RM1602A	6	6	6	5	4	3
RM1601B	4	4	4	4	4	2
RM1602B	1	1	1	1	1	0
RM1601C	2	2	2	2	1	1
RM1602C	2	2	2	2	1	1
Successful reads	113	102	81	69	58	40
Recognition rate	89%	89%	89%	78%	72%	67%
Time cost(sec)	449.048	433.745	409.315	384.01	327.127	273.407

**Figure 14.** Recognition results of all 18 barcodes in the video scan including 1968 frames, with Key frame selection+reg\_dec (0.43)+Clmg. Key frame represents Key frame selection, and the number in the parentheses is the threshold to select histogram difference in procedure 3, Figure 8. All other abbreviations represent the same as in Figure 12.

In fact, compared with the case of not using key frame selection, the new four barcodes that cannot be read after the parameter goes up to 1mean, RM1402B, RM1401C, RM1402C and RM1602B only appear a few times in the video and have been poorly recognised (successful reads are 2, 2, 1 and 1 in Figure 12) even if all the frames are used. This observation suggests that these barcodes are very sensitive to key frame selection. In a real application, performance can be further improved if video scan data collection is carefully controlled to ensure that each barcode is captured sufficient number of times in the video frames.

In the experiment above, the optimised solution, keyframe(0.7mean)+barcode region detection (0.43)+ClearImage, can process video data including 18 barcodes in about 400 s, with the efficiency of about 22 s for each barcode, which is still relatively lower than manual scan. However, this comparison is based on the situation of scanning barcodes at lower position of storage racks within human reach. For those barcodes at higher places, this reading efficiency would be very competitive compared to manual scan, not to mention other benefits of automation, energy efficiency and worker safety. In addition, the efficiency of the optimised solution can be further greatly improved by breaking down an original scan video into shorter pieces and processing the shorter videos in parallel. From this perspective, the method is very promising for deployment in practice, even if we do not yet integrate drone platform and perform scan test for a whole storage rack or a whole warehouse in the paper.

## 5. Discussion and conclusions

Even though many algorithms have been developed to extract barcode information from images (as those listed in Section 2), they have primarily been tested only on non-public or public image datasets that were well prepared (with the barcodes being in the centre area and taking up a large portion of each image). These tests do not adequately reflect their effectiveness or robustness for video data collected under more challenging conditions with drones. In addition, none of them have any intentional design features to reduce redundant information in a video to improve efficiency.

In contrast, in an effort to enable drone-assisted inventory management in warehousing applications, we proposed three algorithms to correspondingly address the three key issues involved in automatic extraction of 1D barcodes from arbitrary video scan data. In barcode direction adjustment, Harris corner detector and Hough transform work together to enable a fast and robust estimation of the direction of one single barcode. In addition, based on connectivity and geometry properties, barcode region detection helps to find all the potential barcode regions in one frame.

Finally, to deal with a large number of frames in a video, a fast extraction algorithm using histogram difference to select key frames is discussed to exploit effective information efficiently.

Experiments conducted using video footage collected at an active warehouse show that the proposed algorithm components work effectively to read out and extract the majority of the location (i.e. cell) identifying barcodes robustly, given that the video was intentionally shot in challenging conditions. Another significance of this work is that each of the three techniques discussed above do not use specific information from other steps, which make them easy to combine with other algorithms or computational sequences.

These characteristics increase the prospects of their wide application, even though some technical challenges still remain for future work before their practical feasibility. The main limitation is that some thresholds, such as the threshold in binarisation and selecting histogram difference, have to be chosen by analysing a short part of the whole video first and needs human assistance. This step can benefit from automatically comparing the performance of different parameter settings and choosing the best combination of the threshold parameters. In order to further eliminate the step of choosing the binarisation threshold, we plan to use deep learning methods to recognise barcode regions automatically, in which the labour-intensive task of preparing labelled data can be significantly alleviated by using the processing results of our current solution.

Furthermore, the selection of the keyframe selection threshold can be conducted more effectively by integrating the pose estimation of the camera when it is available. Another limitation is that, besides location (i.e. cell) identifying barcodes, various other barcodes (e.g. manufacturer's barcode, shipper's barcode, recipient's barcode) present on stored inventory products must also be simultaneously extracted and sorted for overall warehouse management and inventory control. Our current algorithm has no difficulty in reading such barcodes if their size in the video is large enough to be readable. Since such barcode labels are usually significantly smaller compared to the location identifying barcodes, to guarantee their size, a drone has to go closer when capturing them and the drone's trajectory has to be carefully designed.

The proposed method is scalable to video scans collected by any manual or automated means. Even though the overall methodology is proposed around video scans collected using drone-mounted cameras, the algorithms themselves work effectively with other sources of video data such as hard hat cameras, or forklift mounted cameras that are also easy to deploy in warehouse environments. The research presented in this paper is complementary to the authors' ongoing work on drone localisation and control in GPS-denied environments. Ongoing work is also focused on integrating the presented research results with warehouse inventory management systems.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## References

- Adelmann, Robert, Marc Langheinrich, and Christian Floerkemeier. 2006. "Toolkit for Bar Code Recognition and Resolving on Camera Phones-Jump Starting the Internet of Things." *GI Jahrestagung* 94 (2): 366–373.
- Bodnár, Péter, and László G Nyúl. 2012. "Improving Barcode Detection with Combination of Simple Detectors." 2012 Eighth international conference on signal image technology and internet based systems (SITIS), Naples, Italy, 300–306.
- Bodnár, Péter, and László G Nyúl. 2013. "Barcode Detection with Uniform Partitioning and Distance Transformation." IASTED international conference on computer graphics and imaging, Innsbruck, Austria, 48–53.
- Brown, Matthew, and David G Lowe. 2007. "Automatic Panoramic Image Stitching Using Invariant Features." *International Journal of Computer Vision* 74 (1): 59–73.
- Chai, Douglas, and Florian Hock. 2005. "Locating and Decoding EAN-13 Barcodes from Images Captured by Digital Cameras." 2005 Fifth international conference on information, communications and signal processing, Bangkok, Thailand, 1595–1599.
- "ClearImage SDK." 2005. <https://www.inliteresearch.com/>.

- Creusot, Clement, and Asim Munawar. 2015. "Real-Time Barcode Detection in the Wild." 2015 IEEE winter conference on applications of computer vision, Waikoloa, HI, USA, 239–245.
- DJI. 2017. "Phantom 4." <http://www.dji.com/phantom-4>.
- Duda, Richard O, and Peter E Hart. 1972. "Use of the Hough Transformation to Detect Lines and Curves in Pictures." *Communications of the ACM* 15 (1): 11–15.
- Gallo, Orazio, and Roberto Manduchi. 2009. "Reading Challenging Barcodes with Cameras." 2009 Workshop on applications of computer vision (WACV), 1–6.
- Gallo, Orazio, and Roberto Manduchi. 2011. "Reading 1D Barcodes with mobile Phones Using Deformable Templates." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33 (9): 1834–1843.
- Harris, Chris, and Mike Stephens. 1988. "A Combined Corner and Edge Detector." Alvey vision conference, Manchester, UK, 10.5244.
- Juett, XQ James, and Xiaojun Qi. 2005. "Barcode Localization Using Bottom-hat Filter." NSF research experience for undergraduates 19. <http://digital.cs.usu.edu/~xqi/Teaching/REU09/Website/James/finalPaper.pdf>
- Kato, Hiroko, Keng T Tan, and Douglas Chai. 2010. *Barcodes for Mobile Devices*. Cambridge: Cambridge University Press.
- Katona, Melinda, and László G Nyúl. 2012. "A Novel Method for Accurate and Efficient Barcode Detection with Morphological Operations." 2012 Eighth international conference on signal image technology and internet based systems (SITIS), Naples, Italy, 307–314.
- Katona, Melinda, and László G Nyúl. 2013. "Efficient 1D and 2D Barcode Detection Using Mathematical Morphology." International symposium on mathematical morphology and its applications to signal and image processing, Uppsala, Sweden, 464–475.
- Li, Ying, and C.-C. Jay Kuo. 2003. "A Robust Video Scene Extraction Approach to Movie Content Abstraction." *International Journal of Imaging Systems and Technology* 13 (5): 236–244.
- Lin, Daw-Tung, Min-Chueh Lin, and Kai-Yung Huang. 2011. "Real-Time Automatic Recognition of Omnidirectional Multiple Barcodes and DSP Implementation." *Machine Vision and Applications* 22 (2): 409–419.
- Liyanage, J. P. 2007. "Efficient Decoding of Blurred, Pitched, and Scratched Barcode Images." Proceedings of the 2nd international conference on industrial and information systems, Kandy, Sri Lanka.
- Muniz, Ruben, Luis Junco, and Adolfo Otero. 1999. "A Robust Software Barcode Reader Using the Hough Transform." 1999 International conference on information intelligence and systems, Bethesda, MD, USA, 313–319.
- Ohbuchi, Eisaku, Hiroshi Hanaizumi, and Lim Ah Hock. 2004. "Barcode Readers Using the Camera Device in Mobile Phones." 2004 International conference on cyberworlds, Tokyo, Japan, 260–265.
- Ott, L., P. Lambert, B. Ionescu, and D. Coquin. 2007. "Animation Movie Abstraction: Key Frame Adaptative Selection Based on Color Histogram Filtering." 2007 14th international conference on image analysis and processing workshops, Modena, Italy, 206–211.
- Pons, Jasper. 2014 "Drone Ready?" <http://www.scanman.co.za/downloads/whitepaperdronereadyscanman.pdf>.
- Semicon Systems. "Learn How to Select a Barcode Scanner or Bar Code Reader for Any Application." <http://semicon.com/scannertips.html>.
- Steadly, Drew, Chris Pal, and Richard Szeliski. 2005. "Efficiently Registering Video into Panoramic Mosaics." Tenth IEEE international conference on computer vision (ICCV'05), Beijing, China, Volume 1, 1300–1307.
- Wachenfeld, Steffen, Sebastian Terlunen, and Xiaoyi Jiang. 2008. "Robust Recognition of 1-d Barcodes Using Camera Phones." 2008 19th International conference on pattern recognition, Tampa, FL, USA, 1–4.
- Wachenfeld, Steffen, Sebastian Terlunen, and Xiaoyi Jiang. 2010. "Robust 1-D Barcode Recognition on Camera Phones and mobile Product Information Display." *Mobile Multimedia Processing* 5960: 53–69.
- Wang, Zhihui, Ai Chen, Jianjun Li, Ye Yao, and Zhongxuan Luo. 2016. "1D Barcode Region Detection Based on the Hough Transform and Support Vector Machine." *MultiMedia Modeling* 9517: 79–90.
- Zamberletti, Alessandro, Ignazio Gallo, and Simone Albertini. 2013. "Robust Angle Invariant 1D Barcode Detection." 2013 2nd IAPR Asian conference on pattern recognition, Naha, Japan, 160–164.
- Zamberletti, Alessandro, Ignazio Gallo, Simone Albertini, and Lucia Noce. 2015. "Neural 1D Barcode Detection Using the Hough Transform." *Information and Media Technologies* 10 (1): 157–165.
- Zamberletti, Alessandro, Ignazio Gallo, Moreno Carullo, and Elisabetta Binaghi. 2010. "Neural Image Restoration for Decoding 1-D Barcodes Using Common Camera Phones." Proceedings of fifth international conference on computer vision theory and applications, VISAPP 2010, Angers, France, 5–11.
- Zhang, Chunhui, Jian Wang, Shi Han, Mo Yi, and Zhengyou Zhang. 2006. "Automatic Real-Time Barcode Localization in Complex Scenes." 2006 International conference on image processing, Atlanta, GA, USA, 497–500.