Design of an Infusion Monitoring System Based On Image Processing

Shan’ao Song, Shirui Yan*, Sheng Zhang, Yongxian Jiang, Hao Luo, Linmin Ma

School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430080, China

*Corresponding author e-mail: 201804163036@wust.edu.cn

Abstract. At present, infusion monitoring heavily relies on nursing rounds or supervision. In many cases, nursing staff fail to remove patients' cannula in a timely manner after intravenous infusion due to negligence, which leads to serious swelling and blood backflow at the chosen venipuncture area over time, causing pain and even endangering patient lives. This study designed a smart real-time liquid level detection system based on image processing to solve this difficulty. By running the canny edge detection algorithm and Hough Transform (HT) algorithm on a Raspberry Pi computer with an industrial camera, the system extracted and calculated the image's pixels for judgment. As the number of pixels in the detected area reaches the alarm value, the system shall issue an alarm. In this experiment, a 62mm drip chamber was selected, and the system could achieve a high success rate of 98% with a set detection width of 5mm. The experimental results showed that whether the liquid level reached the alarm level could be accurately and effectively-identified utilizing the system. The information could be transmitted to the receiving end promptly with a high success rate, which verified the system's effectiveness. Given its real-time performance and high accuracy, the system has excellent application prospects.

Keywords: Smart Real-Time Liquid Level Detection System, High Accuracy, Image Processing, Industrial Camera, Infusion Monitoring

1. Introduction

Infusion is the most commonly used treatment in clinical medicine[1]. Except for the ICU, the current infusion basically uses an infusion bottle or an infusion bag plus a disposable infusion line with a Murphy's tube, and the drug is injected into the patient's body with the help of the liquid's own weight and atmospheric pressure. Although this method is simple and easy to implement, the status during the infusion process is completely dependent on the patient's own observation and inspection by the nurse. There is no self-alarm and protection mechanism, and abnormalities in the infusion process are prone to occur, but the failure to discover in time may lead to medical risks[2]. In recent years, with the rapid development of computer and wireless communication and other technologies, a variety of infusion monitoring devices have been introduced for clinical applications. These devices usually achieve
infusion-assisted monitoring using the weight method, photoelectric detection method, ultrasonic echo detection method, and capacitive detection method[4,7].

In this paper, a smart real-time liquid level detection system based on image recognition was designed as follows. A drip chamber for infusion was fixed in a closed box. Through lighting the box and collecting images using a camera, real-time liquid level change data can be automatically captured in real-time. The data were then processed locally and uploaded to a computer for data processing and information identification via the network to judge whether the liquid level in the chamber reached the alarm level, thereby achieving smart remote monitoring. The system's advantage is that the impacts of external conditions such as light conditions and drip chamber movement on the detection results during the image detection process can be ignored, and the detection success rate is high. Another advantage is that remote and real-time monitoring can be achieved to monitor the system remotely from the computer at the nurses' station outside the ward. When the drip chamber's liquid level in the ward reaches the alarm value, the system can automatically send a message to remind nursing staff, which avoids wasting human resources and resources and has real-time performance.

2. Liquid Level Image Processing
Image recognition relies on the powerful computing power of the computer, through the automatic processing of massive physical information, completes the recognition of various target types in the image, thereby replacing human mental work. Image recognition integrates the thinking methods of multiple disciplines, and is one of the most active content in the current computer vision field[9]. In order to design the smart real-time liquid level detection system, the first difficulty that needs to be addressed with is the algorithm for recognizing and processing the images returned by the camera. The liquid level image processing process mainly consists of four steps: sharpness improvement, straight line detection, liquid level height calculation, and threshold value setting, among which straight line detection and liquid level height calculation are the most important[9].

2.1 Sharpness Improvement of Original Images
In general, since the captured images are blurred, some details are difficult to be recognized by the computer. In this image processing, in order to facilitate straight line detection, the sharpness of the images was improved so that the contrast of the details on the images was enhanced, making the images look clearer[10]. Here MICRO Z301P USB Linux camera was used, which can improve image sharpness.

The USB camera's working principle can be described as follows: the optical picture information that is converted through the lens of the USB camera is first projected onto the surface of the USB camera sensor and then converted into electrical signals[11]. After analog-to-digital (AID) conversion, the signals shall be converted into digital image signals and then sent to the Raspberry Pi for further digital signal processing. In the end, the processed image signals are transmitted to the computer for processing, and the image information captured by the USB camera can be seen through a liquid crystal display (LCD).

The parameters are as follows:
◆ Rate: 30fps/s
◆ USB interface with a cable of about 1.2 meters, which is easy to use
◆ Metal case and all-glass lens
◆ Appearance size: 36mm*36mm
◆ Signal-to-noise ratio: greater than 48dB
◆ Video data format: 8/16bit
◆ Working temperature: 0°-40°

The advantages are as follows:
◆ Adopts the most classic and stable MICRO solution with realistic colours;
Adopts high-quality non-deforming lenses to restore colours and make images more straightforward;
 Ease-of-use USB interface;
 Photosensitive devices with high-quality CMOS 1/3inch, super CCD photosensitive effect, and VGACIF format;
 Supports moving and still image capture and Avi video recording;
 High-quality 64-bit true-colour;
 Fast compression with the compression ratio of 1:4 to 1:8.

2.2 Straight Line Detection
The flow of the conventional straight-line detection algorithm is shown as follows.

**Figure 1.** Straight line detection algorithm flowchart

**Figure 2.** Images of drip chamber
1. To effectively eliminate sharp noise and smooth the image, the image was blurred by a Gaussian function (as shown in Figure 2(b)).
2. Grayscale image (as shown in Figure 2(c)).
3. To reduce the interference of other lines in the edge detection to some extent before calculating the gradient, the grayscale image's contrast was reduced adequately in this paper (as shown in Figure 2(d)).
4. Then, edge detection was performed using the canny edge detection algorithm, and the binary image was obtained (as shown in Figure 2(e). Figure 2(f) showed the image without proper contrast reduction).
5. To remove the small disturbing blocks in the image, a simple morphological operation, that is, erosion operation was conducted on the image in this paper (as shown in Figure 2(g) and Figure 2(h))
   Note: The kernel of (3, 3) was chosen because the image was not large. The horizontal lines (Figure 2(g)) and vertical lines (Figure 2(h)) were eliminated by changing the operator, making necessary preparations for straight line detection later.
6. Probabilistic Hough Transform was used for straight-line detection, i.e., the HoughLinesP function in OpenCV was used to implement straight line detection (as shown in Figure 2(i) and Figure 2(j)).

2.3 Height Calculation and Threshold Value Setting
The specific algorithm steps are as follows.
1: Extract the XY coordinates of the line.
2: Calculate the approximate length of the drip chamber: \( L = y(\text{max}) - y(\text{min}) \)
   Suppose that the Y coordinates in the two-dimensional coordinate from top to bottom are y[0], y[1] ……y[n], where y(max) is y[0] and y(min) is y[n]. To reduce error, we can take
   \[
   L = \frac{1}{10} \left( \sum_{i=0}^{10} y[i] - \sum_{j=0}^{n-10} y[j] \right)
   \]
3: Calculate the height of the current liquid level.
   Assume that the value of y in the vicinity of the X coordinate \( x[t] \) is y[x0], y[x-1], y[x1], y[x-2], y[x2] ……y[x-i], y[x_i].
   Then the approximate height of the current liquid level is
   \[
   H = \sum_{x-x}^{x} y[x] - y[\text{min}]
   \]
4: Set the threshold value:
   In the conventional process, the doctor shall think the infusion is over when the drip chamber is one-third full. This paper, in order to improve image recognition accuracy, indicates the completion of the infusion.

3. Optimized Liquid Level Detection Algorithm
The above algorithm is very demanding for images. Some of the demanding requirements are listed as follows.
1) The image shall have a very high resolution.
2) The pixels at the edges of the drip chamber in the image shall have varying gradients.
3) The noise interference at the liquid surface demarcation shall not be too blatant.
   However, it is difficult to obtain the images satisfying the above conditions in the experiment, which results in a low success rate of the above scheme. Therefore, the following optimized algorithm was proposed in this paper for liquid level detection. (As shown in Figure 3)
Figure 3. Optimized algorithm flowchart

Step 1: Perform morphological operations.

Step 2: Since the relative positions of the drip chamber and the camera remain unchanged during the experiment; we only need to analyze the images at a critical position. Here the critical position refers to the position where the liquid level reaches the set border height, signaling the end of infusion. Then we segment the images and obtain the result, as shown in Figure 2(k).

Step 3: Count the number of pixels in the images.

Step 4: As we can see from the pixel histogram (Figure 4), when the liquid level does not reach the specified position shown in Figure 2(k), the number of white pixels (256) in the binary images remains constant. Therefore, we assume that the number of white pixels is \((H^+ - S)\), where \(S\) can be used as a threshold value for the liquid level judgement.

Step 5: When the liquid level reaches the segmentation position, the number of white pixels (256) in the binary images shown in Figure 2(l) will increase significantly. Assuming that the increment is \(X\), as shown in Figure 5. \(X>>S\) indicates that the liquid level reaches the border height and the infusion is complete.

Figure 4. Liquid level not reaching the specified position
4. Hardware

Hardware: Raspberry Pi 3 B+ development board, 2 million pixel industrial camera, Windows10 PC
Software: python 3.0 in PyCharm, OpenCV, Raspberry Pi 3 B+ Linux mirror system, remote desktop software XRDP for Raspberry Pi 3 B+, camera software webcam for Raspberry Pi 3B+

Download Raspberry Pi Linux mirror system to an SD card, and configure Wi-Fi connection and SSH file in the SD card. Insert the card into the card slot of Raspberry Pi. Query the IP port of Raspberry Pi and use the remote desktop software to log in to Raspberry Pi through IP and configure the development environment. Install webcam through Linux terminal. Connect the USB industrial camera. Use python compiler to create a command script, and run the host's script through remote control. The camera can take a liquid level image of the drip chamber every 3 seconds, and the images can be sent to the computer for processing through Wi-Fi. Then use canny, Hough Lines, and other algorithms to binarize the images. Perform straight-line detection and image segmentation. The technical flow of measuring the drip chamber's liquid level employing Raspberry Pi is shown as follows.
Figure 7. Workflow diagram 1

Figure 8. Workflow diagram 2
5. Experimental Data
The experimental device is shown as follows:
**Figure 12.** Simple experimental model

**Table 1.** Experimental data of using the algorithm before optimization

| Trial No. | Predicted alarm position (from the bottom) / mm | Actual alarm position (from the bottom)/mm | Error/mm | Success or not |
|-----------|-----------------------------------------------|-------------------------------------------|----------|----------------|
| 1         | 10.2                                          | 0.2                                       | Yes      |
| 2         | 11.1                                          | 1.1                                       | Yes      |
| 3         | 9.9                                           | 0.1                                       | Yes      |
| 4         | 10.9                                          | 0.9                                       | Yes      |
| 5         | 9.3                                           | 0.7                                       | Yes      |
| 6         | 13.5                                          | 3.5                                       | No       |
| 7         | 11.2                                          | 1.2                                       | Yes      |
| 8         | 8.9                                           | 1.1                                       | Yes      |
| 9         | 8.6                                           | 1.4                                       | Yes      |
| 10        | 6.8                                           | 3.2                                       | No       |

The algorithm of the system before optimization was tested first. In this experiment, a 62mm drip chamber was chosen. The alarm liquid level was set at 10mm from the bottom with an error range of 3mm. The liquid level detection system was tested, and the results were shown in Table 1. After several trials, the detection success rate was only 75%. Through research, it was found that the reason for the low success rate was that the detection algorithm before optimization required high accuracy of the images collected by the camera.

Trials were conducted on the optimized algorithm. The camera captured an image every 0.1 seconds. The number of white pixels in the 20 images captured in two seconds was summed up, and the maximum value of the cumulative number in the process was taken as the peak value. Then the peak value was averaged to eliminate the jitter error. When the average peak number of white pixels exceeded the threshold value, the trial was considered to be successful. The experimental data is shown in Table 2. After several trials, the detection success rate reached 96%, which was a high success rate, showing that the system was feasible.

**Table 2.** Experimental data of using the algorithm after optimization

| Trial No. | The threshold number of white pixels | The peak number of white pixels (2s) | The average peak number of white pixels (0.1s)/pc | Success or not |
|-----------|-------------------------------------|--------------------------------------|--------------------------------------------------|----------------|
| 1         | 180                                 | 4886                                 | 244.3                                            | Yes            |
| 2         | 4826                                | 5098                                 | 254.9                                            | Yes            |
| 3         | 4609                                | 4897                                 | 230.45                                           | Yes            |
| 4         | 4234                                | 4897                                 | 244.85                                           | Yes            |
| 5         | 4962                                | 4234                                 | 211.7                                            | Yes            |
| 6         | 4742                                | 4962                                 | 248.1                                            | Yes            |
| 7         | 5065                                | 4742                                 | 237.1                                            | Yes            |
| 8         | 3995                                | 5065                                 | 199.6                                            | Yes            |
| 9         | 4826                                | 3995                                 | 241.3                                            | Yes            |
| 10        | 241.3                               | 4826                                 | 241.3                                            | Yes            |
6. Summary
This paper describes the implementation of a liquid level monitoring system. The system can be used to monitor the liquid level and obtain relevant data for real-time monitoring and information transmission of the liquid level of medical appliances in medical systems. It not only improves the work efficiency of medical personnel but also reduces labor costs. In this system, smart real-time liquid level monitoring is achieved based on the image recognition algorithm. Also, by automatically capturing real-time liquid level change data and uploading them to the computer locally or through the network for smart data processing and information identification, the liquid level can be remotely controlled in a smart manner. With the advantages of small size, good stability, real-time and straightforward operation, the system can better benefit the medical system. Suppose the smart liquid level monitoring system is used. In that case, the need to change a hanging bottle can be automatically identified, and an alarm can be sent out when the remaining medicine liquid in the hanging bottle is insufficient and a new bottle is required. In other words, nurses can be automatically notified to replace the hanging bottle or perform other operations. In the future, this system can be integrated with the hospital information system to improve nursing work efficiency.

References
[1] Zhu Duojia, Hu Xinyu. Intelligent transfusion alarm system based on infrared sensor. Internet of Things Technologies, 2018, 8(05): 72-4.
[2] Liu Bo Feng Fang. Design of Intelligent Infusion Monitoring System. Electronic Test, 2020, 12): 9-12.
[3] WU S, WANG G, HU M, et al. [Design and Implementation of Infusion Monitor Software Based on C# Language]. Zhongguo yi liao qi xie za zhi = Chinese journal of medical instrumentation, 2019, 43(6): 429-31.
[4] Zhang Ai-hua, Zhang Ke-ping, and Zhu Liang. Study on measurement of medical infusion liquid level using capacitive sensor. Journal of Transducer Technology, 2004, 08): 9-11.
[5] Peng Duanyun, Cheng Zifeng, Chen Hongbo, et al. Development of infusion remote-control system based on wireless data-transfer and ultrasonic acquisition. Chinese Medical Equipment Journal. 2005, 08): 7-8.
[6] Deng Zhenjin. Discussion on the testing methods for the accuracy of the infusion pump flow rate. Information of Medical Equipment. 2015, 30(04): 110-1+09.
[7] Wang Yong, Zhang Zhengzhong. Design and research of a photoelectric equipment capable of detecting blood return from transfusion. Light Industry Science and Technology. 2018, 34(06): 88-9.
[8] Cao Min. The Research on Application of Real-time Image Recognition Algorithm in Mobile Devices. Journal of Natural Science of Harbin Normal University, 2018, 34(06): 22-5.
[9] Han Wei. Numerical Study on Multiphase Flow Characteristics and Flotation Dynamics in Flotation Machine; Lanzhou University of Technology, 2009.
[10] Zhan Yibing. Research on Image Quality Evaluation Method Based on Visual Perception Characteristics; University of Science and Technology of China, 2018.
[11] Liu Dongdong. Research on Monocular Vision Ranging Technology Based on USB Camera; Dalian Jiaotong University, 2014.