Hindawi
Scientific Programming
Volume 2022, Article ID 4862806, 10 pages
https://doi.org/10.1155/2022/4862806

*Research Article*

# Methods on Visual Positioning Based on Basketball Shooting Direction Standardisation

**Yiqun Huang and XiaoYun Liu**

*QuanZhou Medica College, QuanZhou, Fujian 362000, China*

Correspondence should be addressed to XiaoYun Liu; 1988005@qzmc.edu.cn

Received 16 January 2022; Accepted 14 February 2022; Published 15 March 2022

Academic Editor: Baiyuan Ding

The existing basketball training shooting direction correction method has the problems of low correction accuracy and poor self-adaptability, and proposes a basketball training shooting direction correction method based on visual perception. A visual localisation algorithm for tracking feature points of object targets is used as the basis for the process of visual robot localisation and its effects, from camera calibration, template matching, background modelling and foreground target separation to feature point extraction, motion estimation, and Kalman filtering. An in-depth understanding and analysis of the traditional corner point detection algorithm is presented, on the basis of which improvements are proposed. An accurate tracking method based on improved Harris corner point extraction is introduced, which builds on the traditional Harris feature point detection by using the changing relationship between the gradient of the grey value of the pixels near the corner point, using simple operations and analysis to exclude some pseudocorner points and noncorner points, and further processing the retained points to derive the correct feature points. The code of this algorithm is written to finally achieve its detection effect, and compared with the traditional algorithm, it is concluded that this algorithm can then extract more accurate corner points in a shorter time, which lays the foundation for the next step of accurate basketball tracking, reflecting the practicality of this algorithm.

## 1. Introduction

In the context of increasingly sophisticated computer vision and image processing technologies, machine vision is used to recognise images, analyse the video parameters of captured sports images, and feed back into human-computer interaction systems and expert systems to achieve guidance in sports training. According to this idea, the basketball training shooting angle correction method is studied, the shooting angle information feature quantity of basketball training is extracted, the basketball training shooting angle parameters and training movement characteristics are analysed, and the correction improvement of basketball training movements is guided. For the basketball training shooting angle visual features modelling difficulties and other problems, combined with machine vision analysis, to achieve basketball training shooting angle correction, related basketball training shooting angle correction method research in the field of sports and computer vision parameter analysis have been better applied [1–4].

Visual positioning technology is divided into monocular and multiocular in terms of the form of image acquisition. Single vision refers to the use of a single camera for the acquisition of environmental information, while multivision refers to the use of multiple cameras, and what we currently call multivision generally stands for dual vision [5–8]. Monocular vision positioning uses a single camera to collect information about the target and calculate its motion parameters, offering the advantages of simple computing, low cost, and ease of installation. However, the limitations of the monocular acquisition window result in its inability to effectively accomplish localisation in complex environments, and monocular systems do not have access to height information about the target environment, so they cannot perform the task of reconstructing three-dimensional space [9, 10]. Binocular stereo vision can simultaneously obtain two images about the target and compute and process them, and through triangulation can obtain three-dimensional information about the surrounding environment to reconstruct three-dimensional space [11–14]. The Zhurong exploration rover is China's Mars Exploration Robot for 2020, which uses vision processing and positioning technology to move around the surface of Mars and collect information about its surroundings. Visual localisation algorithms have been proposed since the 1980s. This method uses the feature points of the target being located to achieve object tracking, so the problem of extracting target feature points has become the key to visual localisation technology. Vision robots have been increasingly used in aerospace, industrial and agricultural production, military, and other fields, mainly due to the rapid development of image processing technology in recent years as well as the decreasing hardware cost of computers and the rapid increase in their computing speed. Vision robotics incorporates core elements from many fields and is the perfect combination of today's scientific and technological developments and advanced industrial operations [10, 11, 15, 16]. The ability to autonomously recognise its surroundings is now a mainstream direction in robotics research, as it has greater plasticity than traditional robots and can successfully perform the tasks we require in a variety of environments and situations. How to identify the target to be operated in the vision window is a prerequisite for a vision robot to work properly, so vision-based localisation and tracking technology is the core and key to robot systems. The study of its algorithms has therefore also become a problem for many researchers [17]. There are two general positioning methods: one is Global Positioning System (GPS) and the other is Voyage Position Projection. The first method is limited by the fact that positioning is not possible at all when GPS is not available, while the second method can cause the robot to be inaccurately positioned after a long period of operation due to environmental influences. Vision-based robot positioning systems are well placed to avoid the positioning problems caused by both of these methods and have therefore become a common international research priority. The advantages of low energy consumption, a wide field of application, and the small size of the equipment are also reasons why visual positioning systems have replaced the usual methods [18–20].

Therefore, this article will focus on the standardisation of basketball shooting direction based on visual localisation, analyse the traditional implementation equipment and algorithms, and design a visual localisation system with more real-time effect and accurate effect.

## 2. Image Processing for Visual Positioning

*2.1. Overall System Structure.* The system includes a camera, decoder, field-programmable gate array (FPGA), synchronous dynamic random memory (SDRAM), digital signal processor (DSP), and display, as shown in the schematic diagram in Figure 1. In order to reduce the workload of the FPGA, the system is designed to use a division of labour between the FPGA and the DSP, and the use of multiple core chips in conjunction with each other makes the system more scalable and more powerful. This system not only meets the requirements of real-time image processing but also allocates the various functional modules of the FPGA effectively. A single core chip allows the FPGA to complete the function of carrying on and processing image data in the whole system [21–23].

*2.2. Image Acquisition.* This system focuses on the automatic positioning of vision robots in a complex and changing application environment, which requires improved accuracy and stability of data acquisition. Charge-coupled device (CCD) cameras are superior to complementary metal-oxide semiconductor (CMOS) cameras in terms of sensitivity, resolution, and noise immunity, so CCD cameras are used. The visual positioning of the basketball does not require high colour resolution of the image, and in terms of sensitivity to infrared light, black and white cameras are superior to colour cameras and have a certain night vision effect, so based on the above advantages, choose black and white CCD cameras.

The design of the system focuses on the automatic positioning of basketballs, which involves many external environments and is difficult to achieve good results without further processing of the general image acquisition. The image information captured is optimised by selecting the most suitable processing algorithm according to the interference of the external environment. In order to adapt the system to different working environments, two distinct image processing algorithms, smoothing and sharpening, have been chosen for this design. The image sharpening algorithm takes into account the fact that the positioning system does not need to focus too much on the details of the acquisition screen, but more on obtaining the contour information of the scene in order to extract the correct corner points; thus, the Sobel sharpening operator is chosen as the edge detection algorithm. The image smoothing process is based on the general median filtering algorithm, which is improved according to the FPGA processing characteristics, in order to better adapt to the FPGA design requirements and reflect the real-time.

The focus of image processing is to achieve a better representation of the real scene and to enhance the efficiency of image transmission and storage. Image processing consists of image enhancement and restoration, image transformation, image coding and compression, and image segmentation. Image enhancement and restoration is used to restore or enhance image information to the maximum. By enhancing the high-frequency and low-frequency parts of the image to improve the clarity of the contours of the objects in the image and to remove noise, the image is able to reflect its true structure more completely. Image transforms are a fundamental tool in the study of complex algorithms and have become an extremely important part of the research process. The use of mathematical mapping methods such as the Fourier transform, wavelet transform, and discrete cosine transform to solve the problem of large amounts of information in image processing facilitates the extraction and understanding of further image information.

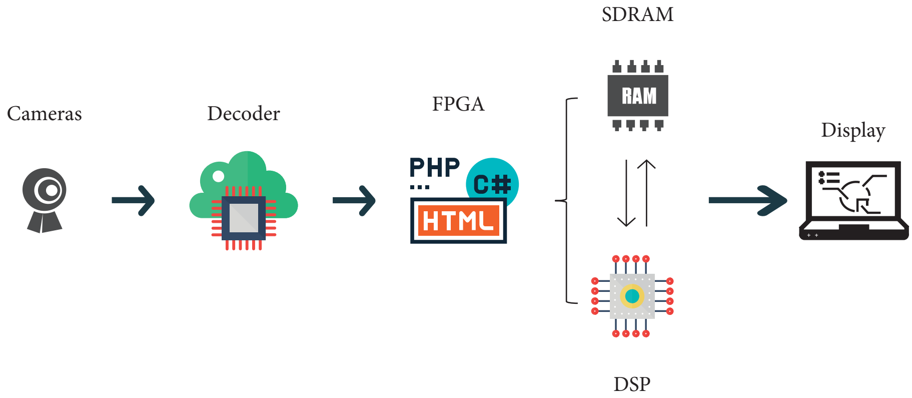

Figure 1: Overall system structure.

Image encoding and compression is the conversion of an image into a certain format and compression to increase the transmission speed and reduce the storage capacity of the image, while ensuring the quality of the image. Image segmentation extracts important parts of the image information such as edges, regions, and other characteristic parts.

*2.3. Improved Median Filtering.* The linear filtering method is low-pass, which leads to the removal of noise but also to the loss of image information. Median filtering, on the other hand, is a nonlinear filtering algorithm, so this algorithm can remove noise while preserving the original effect of the image. The algorithm for median filtering is to select a scan window $X$ with a noneven number of pixel points, read the individual pixels of the image on the subwindow, arrange the scanned pixel points in this window by the size of the grey value, and finally replace the grey value in the centre of the window with the middlemost grey value in the arrangement. Any odd set of data is arranged in the descending order.

$$
\begin{aligned}
& x_{i1} \leq x_{i2} \leq x_{i3} \leq \cdots \leq x_{\text{in}}, \\
& y = \operatorname{Med}\left(x_1, x_2, x_3, \cdots, x_n\right) \\
& = \begin{cases} x_{i((n+1)/2)}, & n \text{ is odd}, \\ \dfrac{1}{2}\left[x_{i(n/2)} + x_{i((n+1)/2)}\right], & n \text{ is even}, \end{cases}
\end{aligned}
\tag{1}
$$

Generally, windows acquire image pixel values in a left-to-right, top-to-bottom sequence. Processing a 256 × 256 image with a 3 ∗ 3 window using traditional median filtering would result in 65536 pixel values being calculated, and each pixel value would need to be compared 36 times. Such a large amount of computation is obviously unsatisfiable in some areas with high real-time requirements [24, 25]. The use of FPGAs for image processing makes use of their parallel data processing characteristics to better perform median filtering. Instead of finding the median by ranking the magnitude of a pixel and its surrounding grey values in order of magnitude, the improved median filtering algorithm takes full advantage of the essence of the sorting algorithm to find the median and exclude other nonmedian values as quickly as possible in order to achieve greater efficiency.

The median filtering algorithm requires only 7 calls to the 3-value comparator to perform 21 comparisons, which significantly reduces the amount of computation, and the parallel processing of this algorithm using an FPGA will result in a significant increase in processing speed. The specific process of the algorithm is shown in Figure 2.

The improved median filter hardware circuit (Figure 3) is divided into four main parts: the 3 ∗ 3 filter window generation module, the row counter module, the median filter module, and the implementation module. The execution module is divided into a reset signal (rst) and a clock signal (clk), which together control the input of the image information. din (7 : 0) is the 8-bit image data input to the scan window, DOUT (7 : 0) is the image data after processing by median filtering, and DV is the output signal valid flag. (Figure 4)

Taking the 3 ∗ 3 filter window as an example, in order to ensure that the 9 captured pixels are output at the same time, 2 FIFO memories are used, with each FIFO memory storing the pixel data of the first captured row, waiting for the last row of data to be captured, and then outputting it at the same time as the data of the first two rows to form the 3 ∗ 3 template.

Once the filter hardware design is complete, the effect of the improved median filtering is verified using FPGA-related software. The improved median filter code is written in verilog and simulated jointly on MATLAB and ModelSim to compare the results and draw conclusions. MATLAB first reads the image data, processes it appropriately, and saves it as a data file, and then, ModelSim reads the data file, performs median filtering, and writes the results to a new data file.

*2.4. Edge Detection Algorithms.* Image processing is generally carried out through image scan windows, so in the edge extraction algorithm in this section, detection extraction can be achieved using the 3 ∗ 3 scan window in the median filtering described above. Through the analysis of the algorithm of edge detection, it can be concluded that the process is divided into three parts, namely, the calculation of

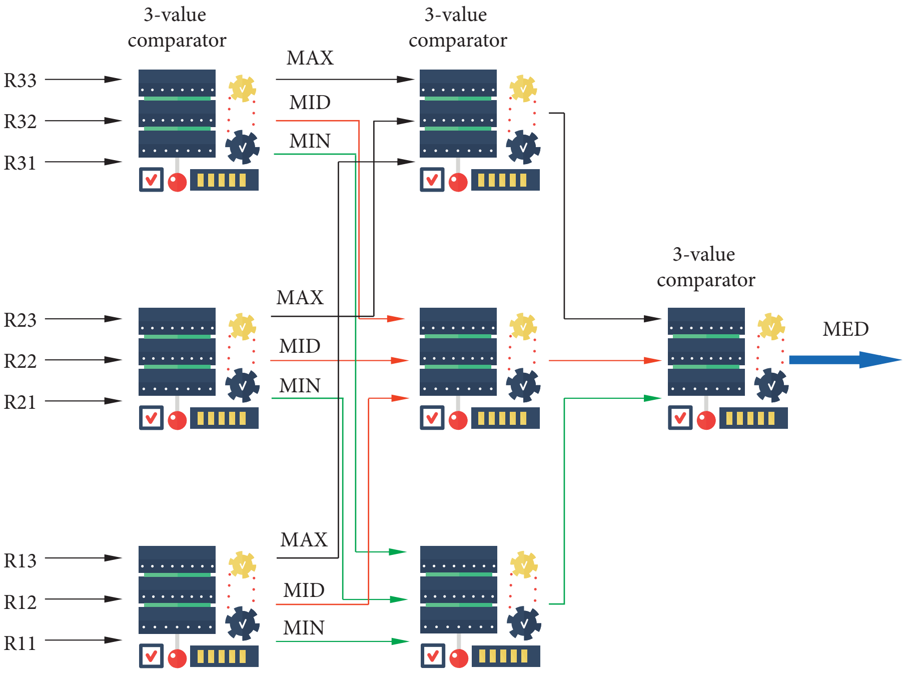

Figure 2: Median filtering algorithm.

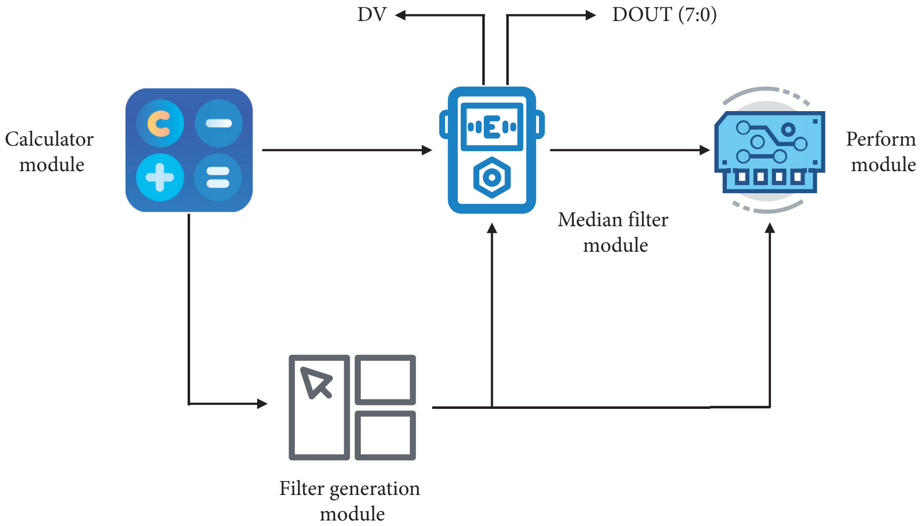

Figure 3: Improved median filter hardware circuit.

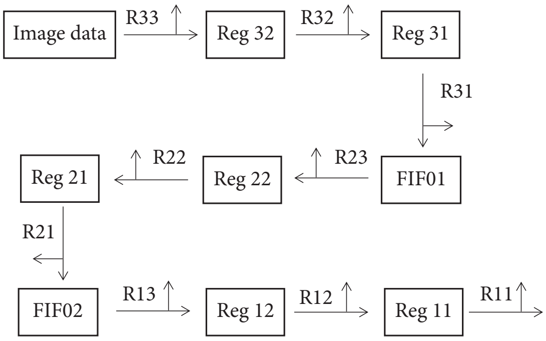

Figure 4: Algorithmic logic circuit diagram.

gradient amplitude, gradient intercomparison, and threshold comparison; the process of the edge detection algorithm is shown in Figure 5.

As can be seen from Figure 5, the gradient values in each direction are calculated separately and comparing the maximum value with the threshold value is the focus of edge extraction, so the computational analysis of the gradient is the main part of the algorithm implementation, and the gradient comparison structure can be divided into two parts, that is, comparing each other to arrive at the larger value and then comparing it to arrive at the maximum value.

## 3. Visual Positioning Algorithms

*3.1. Binocular Stereo Vision Cameras.* Digital imaging technology transforms the surrounding scene into an image and accurately reproduces the real 3D scene by means of specific algorithms, the interrelationships of which are shown in the following equation:

$$Z_C = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & \gamma & u_o & 0 \\ 0 & \alpha_v & v_o & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \tag{2}$$

where a point is defined as $(x_w, y_w, z_w, 1)$ in the world coordinate system, $(x_c, y_c, z_c, 1)$ in the camera coordinate system, and $(u, v, 1)^{\mathrm{T}}$ in the digital image coordinate system. The image centre coordinates are $u_o$ and $v_o$, and the information about the camera itself represented by the matrix $\mathrm{A} = \begin{bmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$ is called the camera internal reference matrix, where $\alpha_u$ and $\alpha_v$ represent the scale factor of the $u$ and $v$ axes of the image plane, respectively, and $\gamma$ is the tilt of the image coordinate axis factor. $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$, $T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$ represents the rotation and translation between the world coordinate system and the camera coordinate system during the entire imaging process. Define the matrix $[R|T]$ as the external parameters of the camera and the matrix $A[R|T]$ as the camera perspective projection matrix $P$. The calibration of the camera is the process of calculating the internal and external parameters by obtaining the surrounding information in a certain way.

Generally, in stereo vision systems, we need to calibrate two cameras, unlike monoculars where not only the internal and external parameters of each camera are required, but also the relative positions of the two cameras. By using the above method, the internal and external parameters of the individual camera are prepared for the next step of the calculation. The results of the external parameters are expressed in the matrices $[R_l|T_l]$ and $[R_r|T_r]$. Let $x_w, x_{cl}$, and $x_{cr}$ be the world coordinates and the left and right camera coordinates of a point, respectively:

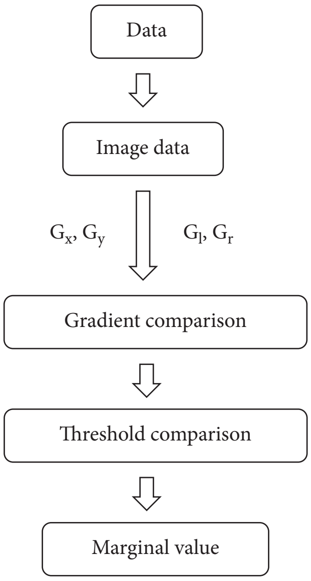

Figure 5: Edge detection algorithm.

$$\begin{aligned} x_{cl} &= R_l x_w + T_l, \\ x_{cr} &= R_r x_w + T_r. \end{aligned} \tag{3}$$

The relative positions of the binocular cameras are represented by $R_0$ and $T_{0,}$ and the coordinates are collated as follows:

$$\begin{aligned} R_o &= R_l R_r^{-1}, \\ R_o &= T_l - R_r^{-1} T_r. \end{aligned} \tag{4}$$

It can be concluded from the above equation that the external reference of the binocular camera can be calculated from the template of the same orientation, and the coordinates can be obtained by substituting the result into (4), where it should be noted that the position of both cameras cannot change when obtaining the template information.

*3.2. Separation of Targets.* The similarity between the contours of two objects can be quickly obtained by calculating the contour moments, which are generally a feature of the object that provides an approximate representation of the target's contour:

$$mp, q = \sum_{t=1}^{n} I(x, y) x^p y^q, \tag{5}$$

where $p$ is the moment in the $x$-direction and $q$ is the moment in the $y$-direction. The algorithm is an integration operation for each point on the target contour. If $p$ and $q$ are both equal to 0, then $m_{00}$ is the sum of the number of pixel points on the target contour.

To solve the problem of degradation of matching accuracy due to size and rotation, a new method of moments, H-moment, is proposed. It is a normalised central moment with size and rotation invariance, ensuring matching accuracy.

Centre moment:

$$mp,q = \sum_{t=1}^{n} I\left(x - x_{\mathrm{avg}}\right)\left(y - y_{\mathrm{avg}}\right)x^{p}y^{q}. \tag{6}$$

Normalised matrix:

$$\eta_{p,q} = \frac{\mu_{p,q}}{m_{00}^{(p+q)/2+1}}. \tag{7}$$

The H-moment is a linear combination of the normalised centre distances.

In order to be able to avoid the inaccurate identification of targets due to the actual influence factor, a model is built for each point or group of points in the background with respect to time. It is effective in resolving time-dependent changes, but only at the cost of a large memory footprint. An analysis of the existing hardware equipment shows that the processing of the captured video information in the way described above is very computationally intensive and does not achieve the required processing speed, making it impractical to model the background in this way.

The analysis and study of video compression techniques has led to a background modelling algorithm that can perform similarly to the above method, that is, constructing a codebook (codebook) to represent the state in the background. The simplest way to do this is to compare the current value of a pixel with its past value; if the two values do not differ significantly, then it is defined as interference under the corresponding pixel. If the pixel difference is large, then it can generate a range of colours corresponding to it. To effectively address the problem of changing backgrounds, we have introduced codebook background modelling. It consists of a number of boxes containing pixels that are constant over time. The method models each pixel or group of pixels through changes in time, observing the value of this pixel at the corresponding position at different points in time and deriving a curve for each pixel with respect to time, which is then encoded and used to construct a background model.

The encoded pixel is defined as symbol code_elements, and then, all symbols are gathered together to form a codebook with the same scale as the background image. The extraction of information about changes in the background is achieved by using the function update_codebook() for all pixels. This process can be updated continuously, while the clear_stale_entries() function is used to train backgrounds that may be foreground targets. This is to remove those background variations caused by real foreground targets. When extracting foreground targets using the codebook method, it is first necessary to add an adjustment value maxMod and minMod to the two borders of each codebookbox. If the pixels in the box are made to add maxMod to the high part of each channel or subtract minMod from the low part, the value of matchChannel is added by one. When the matchChannel and the number of channels are the same, each dimension is searched and a match is known to have been made. If a pixel point is in the box for training at this point, then 255 is returned (deciding that this point is the foreground target), or 0 if it is not in the box (deciding that this point is the background).

The implementation of codebook foreground target separation using OpenCV is usually divided into the following steps:

(1) Using the function update_codebook() to construct an original background model within a certain time frame

(2) Call the clear_stale_entries() function to clear the stale index

(3) Setting the appropriate minMod and maxMod to achieve accurate separation of the identified foreground targets

(4) Maintaining a higher level scene model

(5) Using the function background_diff() to separate foreground targets

(6) Periodically update the learned background pixels

*3.3. Target Tracking and Identification.* Recognition refers to the extraction of the desired target from the observed environment. The moment mentioned in template matching can help identify the target object to be concerned. The most commonly used method of tracking unknown objects is to extract the visual feature points of the target, track these features, and then track the whole target. OpenCV contains two methods to track key points: Lucas Kanade and horn Schunk methods. These two methods represent the commonly mentioned sparse and dense optical flow.

Dense optical flow uses the relationship between each point in the captured image information and velocity, or the relative movement of the same point before and after the motion of the target object to estimate the trajectory of the object. This method of motion estimation is all achieved by the relationship between the point and the velocity of the motion. It is not feasible to use dense optical flow for motion estimation of the target, and it is quite computationally intensive, so we found an alternative method, sparse optical flow. Estimation using sparse optical flow is predicated on first providing a series of target-specific points. OpenCV can help us to find the most suitable tracking angle point. From the above presentation, it can be analysed that the sparse optical flow method is much superior to the dense optical flow in terms of computational speed and complexity.

The use of corner point detection can quickly obtain the features of the acquired image and is widely used in target tracking, motion estimation, template matching, and other fields because of its fast and stable characteristics, also known as feature point detection. The pixels around a corner point should exist on at least two different boundaries, and the corner point can be said to be the intersection of two boundaries. However, in practice, the corner points extracted by the detection methods used are generally feature points that represent the target features and are not always just "corner points." The Harris feature point detection algorithm was obtained by improving the Moravec algorithm. It incorporates a Gaussian filter function in order to be able to avoid the effect of noise on the image during detection:

$$w(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}. \tag{8}$$

Moravec corner detection only calculates feature points 45 degrees apart, whereas the Harris algorithm uses the Taylor formula to extract feature points in each direction:

$$\begin{aligned} E(u, v) &= \sum_{x,y} w(x, y)[I(x+u, y+v) - I(x, y)]^2 \\ &= \sum_{x,y} w(x, y)\left[I_x u + I_y v + O\left(u^2, v^2\right)\right]^2. \end{aligned} \tag{9}$$

Matrix form:

$$\begin{aligned} M &= \sum_{x,y} w(x, y)\begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \\ &= w(x, y) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, \\ E(u, v) &= [u, v]M\begin{bmatrix} u \\ v \end{bmatrix}. \end{aligned} \tag{10}$$

where $I_x$ is the difference in the $x$-direction, $I_y$ is the difference in the $y$-direction, and $w(x, y)$ is a Gaussian function.

The Harris feature point determination method is not available in traditional algorithms. Since the eigenvectors $x_1$ and $x_2$ of the autocorrelation matrix $M$ are proportional to their main tendency to change, Harris uses $x_1$ and $x_2$ to indicate the orientation of the pixel values in terms of how quickly they change; that is, if $x_1$ and $x_2$ are close to each other and both are large, the pixel is a corner point; if $x_1$ and $x_2$ are very different, the point is an edge; and if $x_1$ and $x_2$ are both small, the point must not be a corner point.

Obtaining the eigenvectors involves a great deal of computation, and it is known from linear algebra that the trace of a matrix is the sum of the eigenvalues of the matrix and their product is equal to the determinant of the matrix. Therefore, thefollowing equation is used to select the most probable eigenpoints:

$$R = \Delta M - k(\text{trace}M)^2. \tag{11}$$

*3.4. Kalman Filters.* The Kalman filtering method was introduced in the 1960s and has since become an indispensable method in signal processing research. What the Kalman filter was originally intended to achieve was that if a set of convincing assumptions existed, when measurements were obtained for the entire target tracking process, then a model could be constructed to verify the probability of the current calculated value being correct. The previous measurements in this model do not need to be stored for a long time; that is, the content is always updated. The requirements for hardware equipment are therefore reduced, thus increasing the breadth of utilisation of this method.

The Kalman model is a linear function $F$ of the target state. The model will be related to the combination of the first- and second-order derivatives of the previous motion step. Controlling the processing of the control input $u_k$ in the model will result in a more realistic observed model $Z$ in which only a few model state variables need to be measured and there is no direct link between the measured values and the state variables. If the current estimate has a large jump, then the predicted value from the previous movement will be used instead of the current measurement. Conversely, if the previous prediction is not accurate, then a more accurate measurement needs to be obtained and the result considered accurate. If both the current measurement and the previous prediction are stable, the expectation of the current position must exist somewhere in between them. The above discussion is consistent with our expectations. Figure 6 represents how uncertainty changes over time in response to new observations.

Since updates are sensitive to uncertainty, some further notation needs to be introduced in order to solve this problem. The state at time $k$ is introduced as a function of the state at time $k-1$:

$$x_k = F x_{k-1} + B u_k + w_k, \tag{12}$$

where $x_k$ denotes an n-dimensional vector of state elements and the transfer matrix $F$ is an $n * n$ matrix multiplied by $x_k$. Vector $u_k$ is a new addition, which serves to allow external control to be applied to the system and consists of a c-dimensional vector representing the input control. $B$ is an $n * c$ matrix linking input control and state change. The variable $w_k$ is a random event or external force that directly affects the state of the system. The elements of $x_k$ are assumed to have a Gaussian distribution $N(0, Q_k)$ and $n * n$ covariance moments $Q_k$. It is often not entirely possible to determine whether $z_k$ is a direct measure of the state variable $x_k$:

$$z_k = H_k x_k + v_k, \tag{13}$$

where $H_k$ is the matrix of $m * n$ and $v_k$ is the measurement error, also assumed to have a Gaussian distribution $N(0, Q_k)$ and $m * m$ covariance matrix $R_k$.

*3.5. Model Testing.* Here is an example of measuring the motion of a basketball. The motion of the basketball is represented by two directions $x$ and $y$ and two velocities $v_k$ and $v_y$. From this information, the basketball's motion state vector $x_k$ is formed:

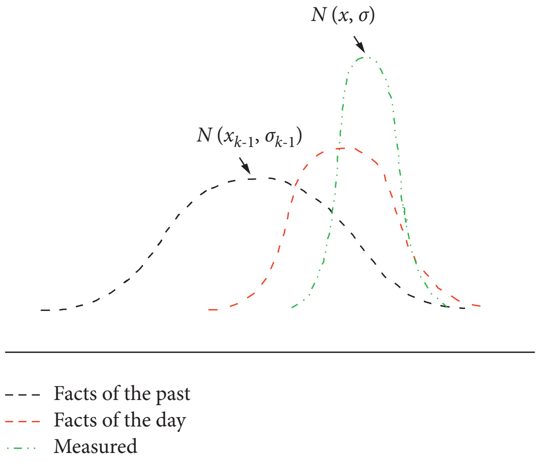

Figure 6: Curve of uncertain change.

$$x_k = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{14}$$

The change of relative position of the basketball movement can only be represented in the camera and only the position variable can be obtained:

$$z_k = \begin{bmatrix} z_x \\ z_y \end{bmatrix}_k. \tag{15}$$

Therefore, $H$ can be expressed as the following structure:

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{16}$$

The basketball is not moving at a uniform speed, so a value $Q_k$ is needed to reflect this. The current position of the basketball is estimated by the method mentioned earlier, and then, the choice of $R_k$ is made based on its accuracy.

Embedding the above expression into the broader update equation requires only the computation of an a priori estimate $x_k^-$ for the next state:

$$x_k^- = Fx_{k-1} + Bu_{k-1} + w_k. \tag{17}$$

The above formula gives an idea of the value to be expected in the next step from the results already obtained. This leads to the Kalman update rate or mixing ratio, which provides the most useful predictive information for the current measurement:

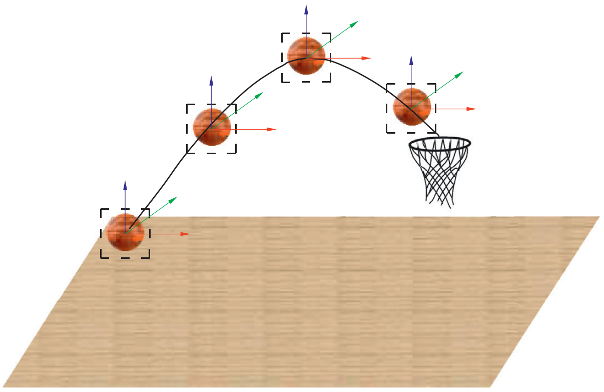

Figure 7: Basketball movement and actual movement predicted by the system.

$$K_k = P_k^- H_k^T \left( H_k P_k^- H_k^T + R_k \right)^{-1}. \tag{18}$$

If the one-dimensional case of a motion variable is obtained directly, $H_k$ is just an identity matrix with one dimension. So, if the measurement error is $\sigma_{2k+1}$, then $R_k$ can only be an identity matrix with one dimension. And similarly, $P_k$ is exactly this covariance, $\sigma_{2k}$:

$$K = \frac{\sigma_k^2}{\sigma_k^2 + \sigma_{k+1}^2}. \tag{19}$$

When the current measurement value is obtained through the update rate, the optimal current value of $x_k$ and $P_k$ can be obtained:

$$\begin{aligned} x_k &= x_k^- + K_k \left( z_k^- - H_k x_k^- \right), \\ P_k &= \left( I - K_k H_k x_k^- \right), \end{aligned} \tag{20}$$

where $I$ is the identity matrix; when the system enters the $k+1$ state, $P_k$ is $P_k^-$ in (18), so the algorithm can be repeated.

Taking the basketball moving in the shooting process as an example to observe the effect of Kalman filtering, Figure 7 is the graphical representation of basketball movement and actual movement predicted by the system. The box in the figure indicates the position predicted by the system.

It can be seen from the above experimental results that the Kalman filtering method can achieve better and more accurate tracking and positioning of moving objects.

## 4. Conclusion

Aiming at the shortcomings of shooting angle correction methods, this article proposes a shooting direction correction method based on visual perception in basketball training. From visual robot localisation, template matching, background modelling and foreground object separation to target point extraction, motion estimation, and Kalman filtering, the visual target localisation and its effect process are based. The traditional corner detection algorithm is deeply understood and analysed, and some improved methods are proposed. The experimental results show that this method has a good effect on shooting tracking in basketball training and can basically accurately reflect the position of basketball movement, which is helpful to basketball players' shooting training.

## Data Availability

The dataset can be accessed upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

[1] J. Xiong, Z. He, R. Lin et al., "Visual positioning technology of picking robots for dynamic litchi clusters with disturbance," *Computers and Electronics in Agriculture*, vol. 151, pp. 226–237, 2018.

[2] Y. Chen, S. Wang, J. Bi, and H. Ai, "Study on visual positioning and evaluation of automatic evisceration system of chicken," *Food and Bioproducts Processing*, vol. 124, pp. 222–232, 2020.

[3] V. Margarita, "Sotnikova, control system design for visual positioning of a ship based on NMPC and multi-objective structure," *IFAC-PapersOnLine*, vol. 51, no. 32, pp. 445–450, 2018.

[4] Y. Bian, X. Fang, M. Yang, and J. Yang, "Underwater visual position measurement system for high-accuracy beam installation," *Optik*, vol. 127, no. 8, pp. 964–3968, 2016.

[5] J. Fan, S. Bi, R. Xu, L. Wang, and L. Zhang, "Hybrid lightweight Deep-learning model for Sensor-fusion basketball Shooting-posture recognition," *Measurement*, vol. 189, Article ID 110595, 2022.

[6] Y. Wang, M. Sun, and L. Liu, "Basketball shooting angle calculation and analysis by deeply-learned vision model," *Future Generation Computer Systems*, vol. 125, pp. 949–953, 2021.

[7] R. R. Oudejans, R. W. van de Langenberg, and R. I. Hutter, "Aiming at a far target under different viewing conditions: visual control in basketball jump shooting," *Human Movement Science*, vol. 21, no. 4, pp. 457–480, 2002.

[8] J. Gun, "Basketball action recognition based on FPGA and particle image," *Microprocessors and Microsystems*, vol. 80, Article ID 103334, 2021.

[9] W. Ding, J. Gu, S. Tang, Z. Shang, E. A. Duodu, and C. Zheng, "Development of a calibrating algorithm for Delta Robot's visual positioning based on artificial neural network," *Optik*, vol. 127, no. 20, pp. 9095–9104, 2016.

[10] Z. Wang, H. Li, and X. Yang, "Vision-based robotic system for on-site construction and demolition waste sorting and recycling," *Journal of Building Engineering*, vol. 32, Article ID 101769, 2020.

[11] L. Bergamini, M. Sposato, M. Pellicciari, M. Peruzzini, S. Calderara, and J. Schmidt, "Deep learning-based method for vision-guided robotic grasping of unknown objects," *Advanced Engineering Informatics*, vol. 44, Article ID 101052, 2020.

[12] T. P. Nguyen and J. Yoon, "A novel vision-based method for 3D profile extraction of wire harness in robotized assembly process," *Journal of Manufacturing Systems*, vol. 61, pp. 365–374, 2021.

[13] Z. Lu, M. Zhao, J. Luo, G. Wang, and D. Wang, "Design of a winter-jujube grading robot based on machine vision," *Computers and Electronics in Agriculture*, vol. 186, Article ID 106170, 2021.

[14] S. Mishra and S. Jabin, "Recent trends in pedestrian detection for robotic vision using deep learning techniques," in *Artificial Intelligence for Future Generation Robotics*, N. S. Rabindra, G. Ankush, E. B. Valentina, and B. Monica, Eds., Elsevier, Amsterdam, Netherlands, pp. 137–157, 2021.

[15] H. Yang, L. Chen, Z. Ma et al., "Computer vision-based high-quality tea automatic plucking robot using Delta parallel manipulator," *Computers and Electronics in Agriculture*, vol. 181, Article ID 105946, 2021.

[16] W.-S. Kim, D.-H. Lee, Y.-J. Kim, T. Kim, W.-S. Lee, and C.-H. Choi, "Stereo-vision-based crop height estimation for agricultural robots," *Computers and Electronics in Agriculture*, vol. 181, Article ID 105937, 2021.

[17] R. Wang, A. Wu, X. Chen, and J. Wang, "A point and distance constraint based 6R robot calibration method through machine vision," *Robotics and Computer-Integrated Manufacturing*, vol. 65, Article ID 101959, 2020.

[18] A. Anna, "The use of vision systems in the autonomous control of mobile robots equipped with a manipulator," *Transportation Research Procedia*, vol. 40, pp. 132–135, 2019.

[19] W. Fang, F. Chao, L. Yang et al., "A recurrent emotional CMAC neural network controller for vision-based mobile robots," *Neurocomputing*, vol. 334, pp. 227–238, 2019.

[20] M. D. Hazrat Ali, K. Aizat, K. Yerkhan, T. Zhandos, and O. Anuar, "Vision-based robot manipulator for industrial applications," *Procedia Computer Science*, vol. 133, pp. 205–212, 2018.

[21] M. Badawy, H. Khalifa, and H. Arafat, "New approach to enhancing the performance of cloud-based vision system of mobile robots," *Computers & Electrical Engineering*, vol. 74, pp. 1–21, 2019.

[22] L. Xiao, Y. Zhao, L. Gong, C. Liu, and T. Wang, "Dual-arm cooperation and implementing for robotic harvesting tomato using binocular vision," *Robotics and Autonomous Systems*, vol. 114, pp. 134–143, 2019.

[23] S. V. Shavetov, I. I. Merkulova, A. A. Ekimenko, O. I. Borisov, and V. S. Gromov, "Computer vision in control and robotics for educational purposes," *Computer Vision in Control and Robotics for Educational Purposes*∗∗This work was supported by the Ministry of Education and Science of Russian Federation*, vol. 52, no. 9, pp. 127–132, 2019.

[24] K. Asadi, H. Ramshankar, H. Pullagurla et al., "Vision-based integrated mobile robotic system for real-time applications in

construction," *Automation in Construction*, vol. 96, pp. 470–482, 2018.

[25] R.-J. Halme, M. Lanz, J. Kämäräinen, R. Pieters, J. Latokartano, and A. Hietanen, "Review of vision-based safety systems for human-robot collaboration," *Procedia CIRP*, vol. 72, pp. 111–116, 2018.