Design and Implementation of a Computer Vision Tracking System

Abdulrahman Nasereldin Bilal, Mohamed Siddig Altyeb, Sharief F. Babiker
Department of Electrical & Electronic Engineering
University of Khartoum
Khartoum, Sudan

***ABSTRACT*: *Visual serving, also known as Vision-Based Robot Control and abbreviated VS, is a technique which uses feedback information extracted from a vision sensor to control the motion of a robot. In the recent past, research on image-based visual serving has been concentrated on potential problems of stability and on robustness with respect to camera calibration errors and feature extracting errors. We present two image-based visual serving strategies for driving a robot equipped with a camera toward a desired configuration. Since the topic spans many disciplines our goal was to implement complete integrated robot by reviewing the prerequisite topics from robotics and computer vision. Since any visual servo system must be capable of tracking image features in a sequence of images, we also include an overview of feature-based and correlation-based methods for tracking without human intervention – only GUI for calibration purposes and manual control. Moreover we have done a complete control driver for the robot using embedded systems. Simulation and experimental results show the effectiveness of the proposed system.*

Keywords: Subtraction Techniques, Image Motion Analysis, Machine Vision, Image Matching, Real Time Systems, Open Source Software, Servosystems, Servomotors, Position Control, Microcontrollers

Received: 3 November 2012, Revised 20 December 2012, Accepted 26 December 2012

1. Introduction

The aim of motion tracking is to detect and track moving objects through a sequence of images. Motion tracking is not only useful for monitoring activity in public places, but it is becoming a key ingredient for further analysis of video sequences. Information about the location and identity of objects at different points in time is the basis of detecting unusual object movements (e.g. someone being mugged at an ATM) or coordinated activities (e.g. strategic plays in a football game).

The use of visual data for robot control has been a goal for many years. It provides very dynamic information about the environment, but is passive unlike sonar or laser range finding systems which measure a reflected output signal. It also enjoys the advantage of providing sensory data that humans are naturally familiar with [1].

The field of computer vision is concerned with problems that involve interfacing computers with their surrounding environment through visual means. One such problem, object recognition, involves detecting the presence of a known object in an image, given some knowledge about what that object should looks like. As humans, we take this ability for granted, as our brains are extraordinarily proficient at both learning new objects and recognizing them later. However, in computer vision, this same problem has proven to be one of the most difficult and computationally intensive of the field. Given the current state of the art, a successful algorithm for object recognition requires one to define the problem with a more specific focus.

The ability to reliably detect and track human motion is a useful tool for higher-level applications that rely on visual input. Interacting with humans and understanding their activities are at the core of many problems in intelligent systems, such as human-computer interaction and robotics. An algorithm for human motion detection digests high-bandwidth video into a compact description of the human presence in that scene. This high-level description can then be put to use in other applications.

Some examples of applications that could be realized with reliable human motion detection and tracking are:

• Automated surveillance for security-conscious venues such as airports, casinos, museums, and government installations: Intelligent software could monitor security cameras and detect suspicious behavior. Furthermore, human operators could search archived video for classes of activity that they specify without requiring manual viewing of each sequence. Having automated surveillance vastly increases the productivity of the human operator and increases coverage of the surveillance.

• Human interaction for mobile robotics: Autonomous mobile robots in the workplace or home could interact more seamlessly with the humans in their environment if they could reliably detect their presence. For example, robots to assist the elderly would know when assistance is needed based on the motion of a person.

• Safety devices for pedestrian detection on motor vehicles: Intelligent software on a camera-equipped car could detect pedestrians and warn the driver.

• Automatic motion capture for film and television: Producing computer-generated imagery of realistic motion currently requires the use of a motion-capture system that stores the exact 2-D or 3-D motion of a human body using visual or radio markers attached to each limb of an actor. With accurate algorithms for human motion tracking.

One of the most common approaches is to compare the current frame with the previous one. It's useful in video compression when you need to estimate changes and to write only the changes, not the whole frame. But it is not the best one for motion detection applications.

There are many approaches for motion detection in a continuous video stream. All of them are based on comparing of the current video frame with one from the previous frames or with something that we will call background. In this article, we will try to describe some of the most common approaches.

2. System Requirements

Our design is intended to be a low cost solution that may be adopted in developing counties. This system should make use of the capabilities provided by existing sophisticated image processing software therefore we have used:

• OpenCV (Open Source Computer Vision) which is a great library for creating Computer Vision software using state-of-the-art techniques, and is freely available for Windows, Linux, Mac and even the iPhone. OpenCV is released under the liberal BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java (Android) interfaces and supports Windows, Linux, Android and Mac OS. The library has more than 2500 optimized algorithms. Adopted all around the world, OpenCV was originally designed by Intel in 1999 to show how fast Intel CPUs can run. So most of OpenCV runs very fast on Intel CPUs, However, OpenCV is mainly used for tasks that are complex in nature, often requiring post-grad experience in the fields of Computer Vision or Artificial Intelligence (AI). [2]

• BLDC (BrushLess DC) motor which is one of the motor types rapidly gaining popularity. BLDC motors are used in industries such as Appliances, Automotive, Aerospace, Consumer, Medical, Industrial Automation Equipment and Instrumentation. As the name implies, BLDC motors do not use brushes for commutation; instead, they are electronically commutated. BLDC motors have many advantages over brushed DC motors and induction motors [3]. The main objectives of using this type of motors in the system are the required torque for the robotic module and the feedback capability that accomplishes the required servo mechanism.

• Control circuitry: usually these kinds of motors need expensive special drivers to control the operation of the motor and supply it with the required power, but we have developed our own driver or controller that performs the same functions of the ordinary motor driver in addition to the advantages of low cost and serial communication capability (RS232 serial interface).

• System platform which is a mechanical prototype that we have designed and manufactured to show the reliability of the integrated system.

• GUI: to simplify the use of the system for an ordinary user and support complete access to calibrate the system parameters.

3. Design

3.1 Image Processing

In the Figure 1, we discussed the choice of algorithms for each stage of our code. Although this provides a good theoretical overview, it does not give enough informations to implement the system and hence we discuss the practical details of how each stage and how they come to form a complete system [4].

3.1.1 Motion Detection

Is done using frame subtraction commonly referred to as background subtraction. The video is converted to monochrome and each pixel in the previous frame is subtracted from the current frame. If nothing changed between frames, the result of all the pixel subtractions will be 0 [4].

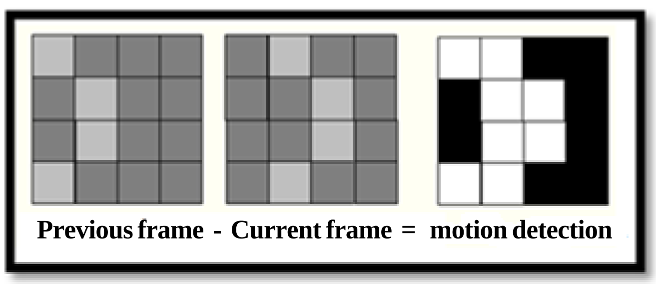

Figure 1. Explanation of Motion detection

Learn Rate: Regulates the update speed (how fast the accumulator forgets about earlier images).

Threshold: The minimum value for a pixel difference to be considered moving.

3.1.2 Filters

After background subtraction is completed, the morphological operation is applied twice to the foreground mask. This operation has the effect of enlarging the area of each connected region in the mask and will close any small gaps. Such gaps are closed to maximize the likelihood that each moving object is contained within a single region, rather than two smaller ones [5].

3.1.3 Contour Finder

The foreground mask provided by the background subtraction is processed to build a data structure that enumerates the boundaries for each distinct foreground object.

This enumeration is achieved by the contour finder, a simple algorithm that finds each "*connected component*" of the foreground mask. A connected component is defined as a region of an image which its pixels have the same value and are adjacent to other pixels in the same connected component [6].

3.1.4 Blob Finder

For our contour finder, once these contours are located, those with small geometric area are ignored. Such small contours are likely to be noise or small image disturbances. The second step is to link a sequence of object blobs together across image frames (blob tracking). To achieve this, it is useful to describe each object blob by a set of attributes, for example its position, velocity and size, collectively called its state.

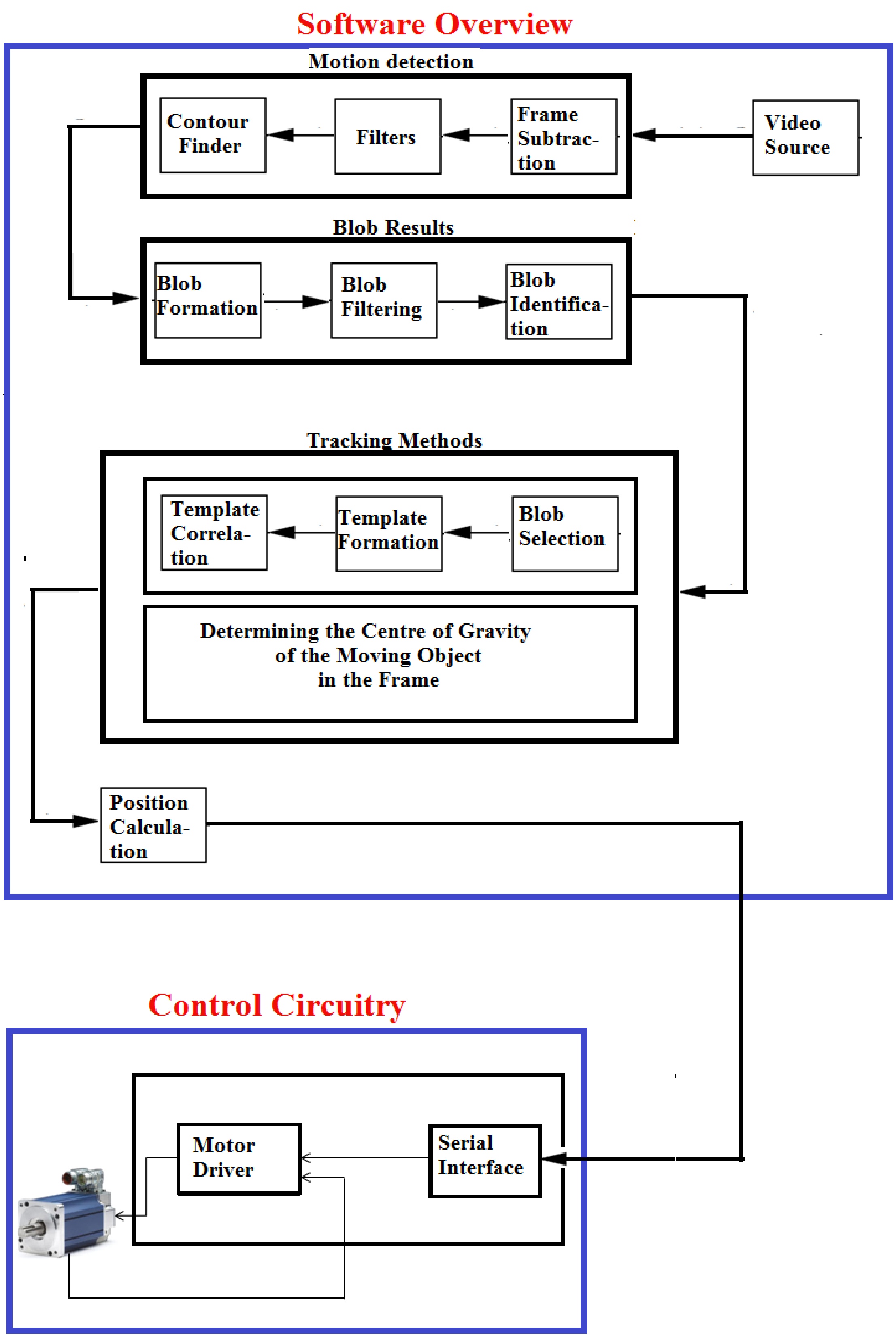

Figure 2. Schematic overview of the integrated system software and hardware

3.1.5 Data association

In circumstances where several objects are being tracked simultaneously, it is paramount to avoid confusion of distinct objects each circle signifies a blob which is tracked through frames from left to right. At the new input frame *t* +1, the identities of blobs are initially unknown and must be extrapolated from observed object dynamics of previous frames. Additional complications arise if objects can temporarily disappear (e.g. behind walls or cars) or if non-existent foreground objects are observed (as in the previous example).

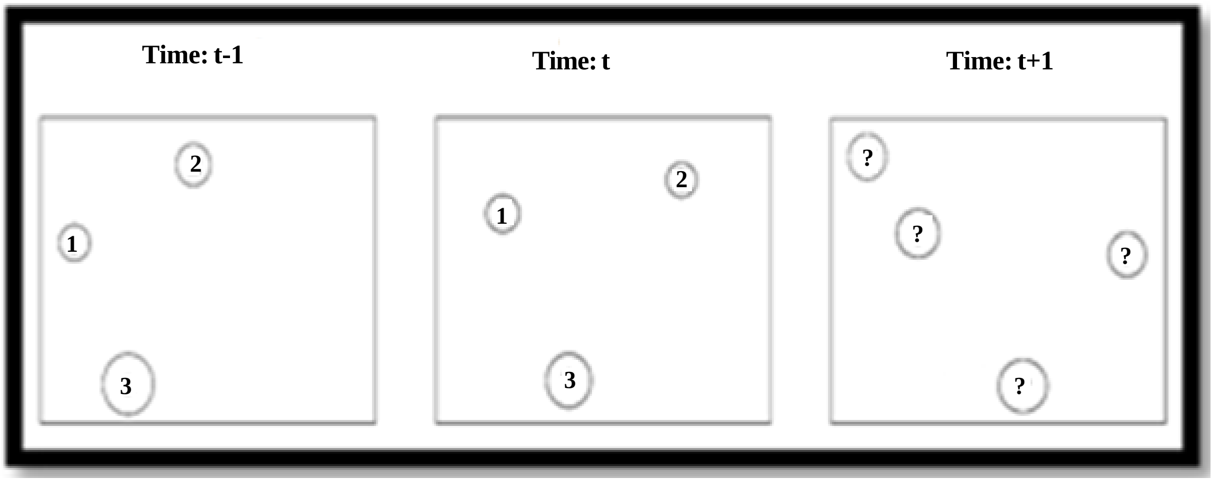

Figure 3. Blob identification

3.2 Tracking Methods

3.2.1 Centre of Gravity

Here, for better results, you need a bare image from camera, with no persons in it. It is the background. (It can also be done, even if you don't have this background image. But if you have it, it will be better. I will tell at end what to do if no background image). Finding the centroid of the wanted moving object to be tracked [7] [8].

3.2.2 Template Matching

Template matching works by sliding the template across the original image. As it slides, it compares or matches the template to the portion of the image directly under it. The output is an image that shows the degree of equality or correlation between the template and the portion under the template. Well, the first disadvantage is that you need to know what you're looking for. If you're looking for dynamic features, you'll be better off using some other techniques. However we overcome this problem by dynamically looking for new template in order to make it robust for moving objects even if the object is hidden suddenly the system can remain saving its last position till it appears again and hence scaling problem solution is achieved which was not originally supported by OpenCV [5].

3.3 GUI

Resource consumption may be controlled via a number of tuning parameters. In general, the results obtained do not depend too critically upon the parameter settings; the casual user can (and probably should) simply leave them at their default settings. So we made a simple GUI for calibration purposes.

3.4 The Controller

The controller in the system is ATMEGA 32 microcontroller and it has two functions:

3.4.1 Serial Interface

It works as a serial interface between the PC and the motors of the robot, it receives the control commands from the PC after determining the exact current position of the object to be tracked by the tracking system in the PC, and from the coordinates of the center of the object to be tracked the required motion or displacement of the robot is determined in terms of steps in both the vertical or horizontal directions, these commands are transmitted to the controller via RS232 serial interface, the tracking system in the PC has a serial communication capability using a special header file that is developed to interface the software systems to the serial ports that implements the RS232 protocol, the signal is transmitted from the PC to the MAX232 chip (it is a voltage level converter used in the interface between RS232 signal from the PC and the TTL level that is used or compatible with ATMEGA 32 microcontroller) and then the signal is passed to the controller.

3.4.2 Motor Driver

The motor used in the system to control the robotic module is the BLDC servo motor with encoder (the encoder provides additional accuracy by obtaining the shaft position in smaller scale than the zones obtained from the hall sensors but it isn't used in our system). The driver develops the appropriate sequence to drive the motor and supplies the motor with the power it needs to operate properly. The methodology used in the control of the motor is the electronic commutation.

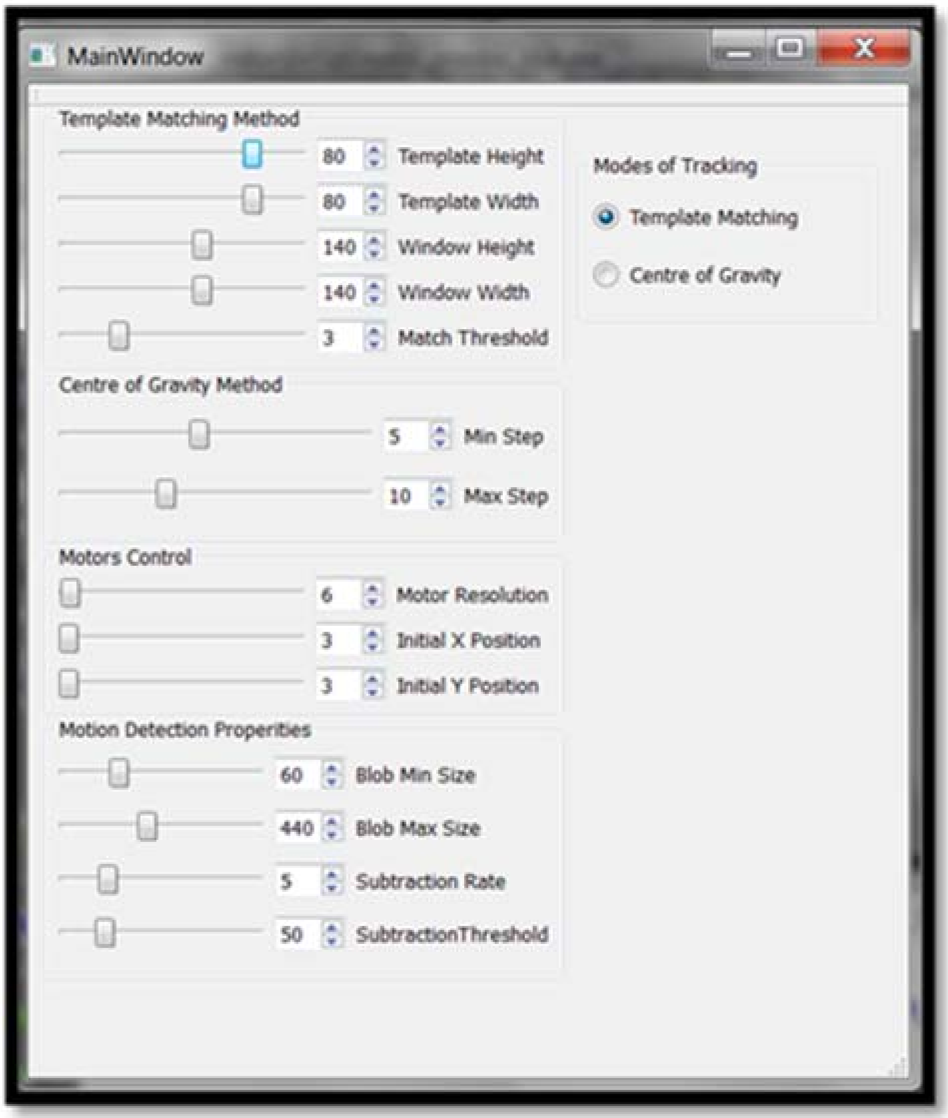

Figure 4. GUI

Because the controller can't provide the motor with the required current, power MOSFETs are used to supply the motors from an external supply rather than the one used to supply the controller, so the control signals or commands just commutate or switch the supply on and off in order determine the appropriate sequence to keep the motor moving in the correct direction, another function of the power MOSFETs is the protection from the induced back EMF (electromotive force) generated by the coils of the motor at the start of the motor rotation because these MOSFETs used have built in free-wheeling diodes.

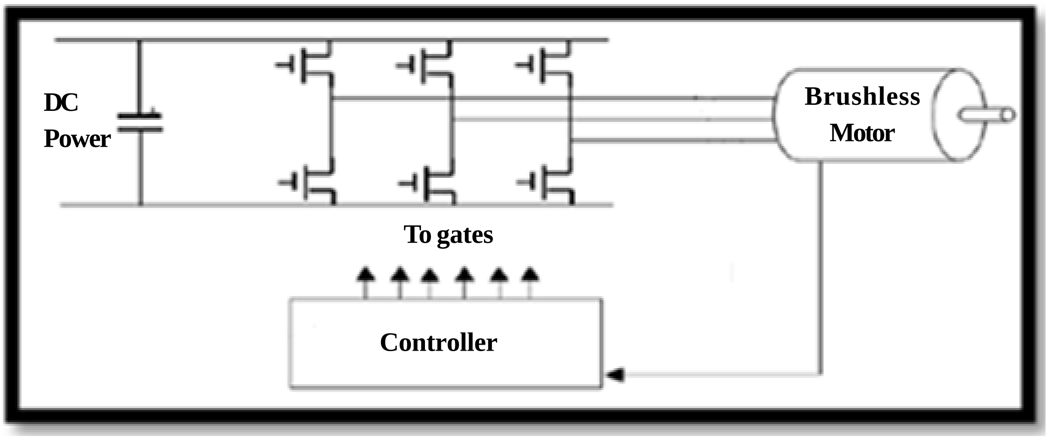

Figure 5. Circuit for the driver of a BLDC servo motor

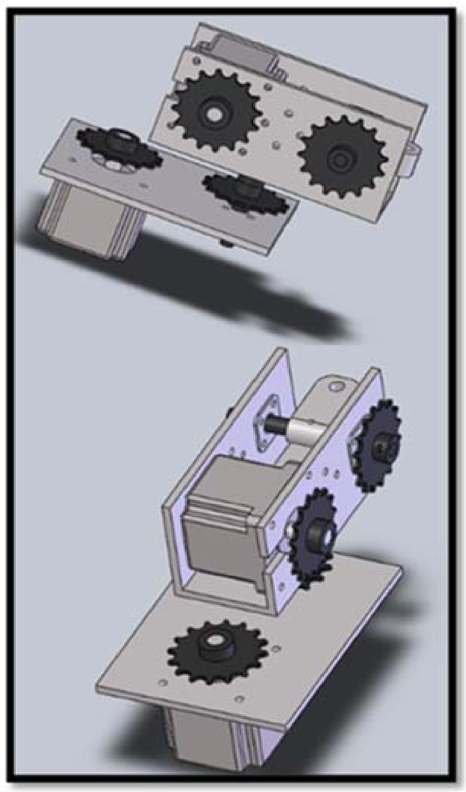

Figure 6. System Tracking

The robot consists of two BLDC servo motors one for the horizontal displacement and the other for the vertical motion, we also have a controller for each motor that works as a driver for that motor as discussed above, but the two controllers differ in the serial interface function; the positions or the command signals are sent by the PC via the RS232 serial interface and received by the first controller that analyzes the signal according to a predefined protocol by the software system in the PC and the microcontroller (this protocol attaches a specific number for the horizontal displacement and another number for the vertical one) and obtains the command that it needs, after that the first controller retransmit the command signal to the second controller that just specifies the number of the required steps that the second or vertical motor has to take in order to track the required object, it is a small network between two controllers each one of them controls the robot in one direction.

The servo mechanism (A servo mechanism or servo is an automatic device that uses error-sensing feedback to correct the performance of a mechanism. The term correctly applies only to systems where the feedback or error-correction signals help control mechanical position or other parameters.) is implemented in our system by using the feedback from the hall sensors of the motor that determine the exact position of the rotor in one of 12 predetermined zones available in the motors used, the controller of the system receives the feedback from the motor and acts accordingly by providing the appropriate sequence to drive the motor, the controller always compares the current position of the robotic module with the position specified by the PC and then determines the appropriate action to be taken in one of the three possible commands to the motor either to rotate in the clockwise or in the counter clockwise or just to hold in the current position (for the motor to move in clockwise or counter clockwise the motor has a specified sequence for each one of them and for the hold command the controller just remains the current sequence or order to the motor).

3.5 Mechanical platform

It is done using computer aided design software (Solid work) considering the required torque and speed. As shown in Figure 6. The moving object is detected and hence numbered. And it is obvious that the system is immune to ambient light intensity.

4. Concluding Remarks

Vision in robotics in general, is a constantly evolving field. As innovations continue to be made, it becomes increasingly important to explore the different methods in order to gain insight into the characteristics, strengths, and weaknesses of each. Focusing on the field of partitioned image-based visual Servo systems, we have performed several standardized tests of robustness in the face of imaging error and system performance against difficult tasks. This data can be used to select appropriate visual servo systems for specific tasks and conditions or to provide direction for future research.

We were able to determine several key characteristics of each tracking method: Template matching and Centre of Gravity. In addition to the introduction of a new method to reduce the computation overhead, as in computer vision: matching problem has proven to be one of the most difficult and computationally intensive of the field. Furthermore we have obtained a successful algorithm for object detection after being hidden in a bounded region depending on the required performance which can be controlled easily by GUI which can be also used to for calibration and testing purposes for future studies to define the problems with a more specific focus.

Eembedded systems background beside industrial drivers manuals have been applied to implement motor drivers connected to the computer using serial communication.

5. Acknowledgment

The authors would like to thank Eng. Shakir Alrubatabi (Sudanese Telecom Company) for valuable assistance. Also, the authors would like to acknowledge the technical staff at the Electronics Laboratory (UofK, Sudan) for their hospitality and willingness to provide services and facilities under their control.

References

[1] Shirai, Y., Inoue, H. (1973). Guiding a robot by visual feedback in assembling tasks, *Pattern Recognition*, 5, 99-108.

[2] www.opencv.org., accessed 23/Aug/2012.

[3] An honours project carried out by Ramanjanyulu Challapalli and Utkarsh Gupta at the CVR College of Engineering: Design Of A Position Loop Servo Controller For A BLDC Motor Based Rotary Electro - Mechanical Actuation System, PIN: 501510.

[4] Richard Szeliski. (2010) .Computer Vision: Algorithms and Applications, March 24.

[5] Bradski, Kaehler. (2008). Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly.

[6] Robert Laganière. (2011). OpenCV 2 Computer Vision Application Programming Cookbook, Packt.

[7] Kim, K., Chalidabhongse, T. H., Harwood, D., Davis, L. Real-time foreground – background segmentation using codebook model.

[8] An honours project carried out by Fabian Wauthier at the School of Informatics at the University of Edinburgh. http://www.cs.berkeley.edu/~flw/tracker.