Face-tracking mount display for medical and engineering applications

Silviu Alin Jecu
Department of Product Design, Environment and Mechatronics
Transilvania University of Brasov
Brasov, Romania
silviu.jecu@student.unitbv.ro

Andreea Georgiana Demeter
Automation and Informational Technology
Transilvania University of Brasov
Brasov, Romania
georgiana.demeter@student.unitbv.ro

Ioana Corina Bogdan
Department of Electronics and Computers
Transilvania University of Brasov
Brasov, Romania
corina.bogdan@unitbv.ro

Abstract — **This paper presents a face-tracking mount display (FTMD) that, has applications, such as medicine (telerobotics surgery), home automation or research and development of integrated circuits and many more, by positioning the display to assure the best viewing angle. The working principle of the FTMD is designed to recognize and identify the relative position and number of persons that are watching the display. Through the Kanade-Lucas-Tomasi and Haar Cascade algorithms and synchronized actuators, the FTMD can change its orientation and inclination angles, ensuring the best-average view. The large number of medical devices that are using cameras in the non-invasive surgery or endoscopy is rising. The output of such devices is a sort of video that can be projected on the display. A face tracking/following mount display would facilitate the viewing angles for the medical personnel as they move during operations. Further, this device can be implemented on the "Vision Cart", which is part of the DaVinci Robot, transmitting the video output from the "Surgeon Console" and "Patient Cart" to the display. The DaVinci Robot is a great example of medical improvement. This system has as its main points the patient cart, surgeon console, and vision cart. The face-tracking mount display mode of operation is based on computer vision, where the camera detects how many people are in a room and, based on the Kanade-Lucas-Tomasi and Haar Cascade algorithm, moves himself according to it.**

Keywords — face-tracking, mount device, teleoperation, telerobotics

I. Introduction

These days, robotics has found its place in the most diverse fields, from aeronautics to electronic technologies, automation, transport, energy industry/industries, essential services, and even medicine. Various robots have been developed in the field of medicine, such as the mounting and manipulation equipment, the DaVinci robotic arms used in the noninvasive surgery, and which have been adapted from the point of view of technology to the surgeon's field of work or the precision of performing the noninvasive surgery, [1], [2].

Smart TVs are one of the equipment that assure the comfort of our houses, but also has a tremendous involvement in the medical devices, such as the surgical monitor display, the DaVinci vision cart, or used in the digital imaging and communication in medicine. On contrary with what many people believe, TV or the monitor display does not cause permanent eye damage, but it may cause a temporary headache, blurry vision, nausea, or eye pain [3]. These symptoms are caused by spending too much time in front of the screen by the inappropriate installation distance of the TV from the sitting place. The face-tracking mount display might be used to solve these inconveniences, and could be categorized as a "visual positioning system" (VPS) [4]. The VPS uses a technique that supports devices in pinpointing their accurate location within an indoor, and is utilized for indoor navigation and positioning [5]. It is known that GPS uses satellites to detect the right location, but the VPS system tracks the device's location within an indoor environment by using computer vision algorithms ([6]–[9]) and a database of 2D or 3D maps. To obtain visual data about the surroundings and the device's location, the VPS system often uses several cameras accompanied by depth sensors. Then, the system processes these data using specific algorithms in locating the device by comparing it to the stored map. This data can be utilized for many tasks related to enhancing the location-based services and augmented reality capabilities, to assist users in navigating indoor environments, or to aid in asset tracking and inventory management.

In this paper, we propose a face-tracking mount device (FTMD) that facilitates the appropriate viewing angles and orientations. The FTMD can solve many ergonomic issues preventing inconvenient situations for the viewer, but also can have beneficial applications in the medical field, particularly in the noninvasive surgery and DaVinci robots. The FTMD can be used in telerobotic surgery, home automation, or the development of integrated circuits. The purpose of it is to detect the number of people sitting into a room using the Kanade-Lucas-Tomasi (KLT) algorithm combined with additional libraries developed for data processing and face detection. Based on the results, the FTMD is able to create an arithmetic average and move itself according to it. First it is given an introduction about current advancements. Secondly, it is described the FTMD with the experimental set up used for tests. Thirdly, it is presented the method based on the KLT and Haar Cascade (HC) algorithms and the extended algorithm for this work. Before presenting the conclusion and the future works, there are given results obtained during several tests involving one single person or multiple persons in the room space, reactions of the FTMD device at sudden/brusque motions or involving random facial expressions for the FTMD to count and distinguish the face profiles. The particularity of this work is given by the idea of adjusting a video output, by tilting or rotating the system so that all people in a room have a perfect view of it. The uses are unlimited because the algorithm at the base of this paper can be applied in almost any field.

II. Experimental Set Up

The FTMD needs to follow several rules related to distances and placement of the display. According to [10], it should be noted that the size of the display is directly proportional to the length at which it is needed to be placed.

In Fig. 1 is given the diagram of the FTMD and by equations (1) and (2) it is expressed the minimum and maximum distances between the viewers/persons and the display. The viewers are followed by a camera, and the display will adjust its inclination and orientation based on the face-tracking and recognition (xOy axes). The steps in proceeding with the FTMD system are given in Fig. 2. Some related works to face recognition, motion and detection can be found in [11]-[14]. In this work we can identify a number of viewers that are in the room, and distinguish the facial expression such that one person is not counted many times.

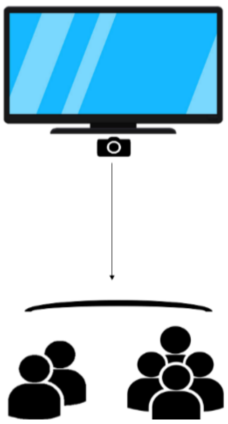

Fig. 1. Diagram of the FTMD

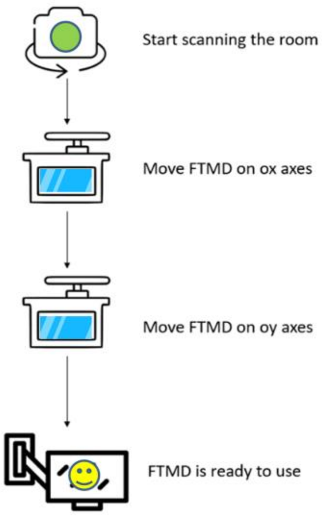

Fig. 2. Steps in proceeding with the FTMD system

$$Distance_{min} = screen_{size}(inch) \cdot 2.54 \cdot 2 \quad (1)$$

$$Distance_{max} = screen_{size}(inch) \cdot 2.54 \cdot 3 \quad (2)$$

For demonstrating purposes, the electronic components used in the experimental setup are: one Arduino Uno board, two TianKongRC 7120MG servo-motors, and a USB camera (WOWSTEP S500 with a maximum resolution of 1920x1080 pixels). The system chassis was 3D modeled and printed to fit the components described and to assure the tilting and rotation movements (or the movement around the x and y axes). The servo motor is designed for applications including robots or remote toys, characterized by a voltage range between 4.8V and 8.4V, with a maximum operating speed of 0.14s/60 deg, and controlled through a PWM signal with widths between 500÷2500s. The Arduino Uno is a development board based on the ATmega328 microcontroller designed with 14 input and output pins (6 pins used as PWM outputs and 6 pins as analog inputs), a 16 MHz oscillator, a USB connection, a power plug, and a reset button. It can be powered directly from the computer, from a the USB port, via a 9-volt battery, or via a 9-volt power supply.

The programming environment is based on Python under Windows operating system. The specification of the computer on which the program ran, are Intel I5-11600K processor, having 16 GB of RAM and an NVIDIA GeForce RTX 3060 graphics card. For data and image processing, a computer running PyCharm was used. Through this software, the image from the camera is acquired, filtered and passed through the HC algorithm that outputs the position of the face(s). Knowing the position and the current orientation of the screen, the program can tell the development board, with the help of the serial bus, to either, actuate the servo-motors for the new best-average position, or to maintain the current one.

III. Methods

The FTMD adjustments related to the face detection, the image processing, the number of frames and viewers intervening in the webcam, the motion and positioning are developed using a combination of two algorithms: the KLT algorithm and the HC algorithm. As a result of this association of algorithms with additional loops or conditions to obtain a fine motion and detection of the device was developed the FTMD algorithm.

A. Kanade-Lucas-Tomasi Algorithm

The face detection uses the KLT algorithm, which is a feature point's tracker and a method for feature extraction in computer vision, [15]. This algorithm is primarily suggested to address and solve the issue that conventional image registration methods are quite expensive. The KLT algorithm could be a fast alternative and inexpensive either that has a number of years since was included in the computer vision field. It is characterized by a high speed and reasonable accuracy for a large variety of real-life applications (e.g., video stabilization, image mosaicking, 3D reconstruction, or recognition), using spatial intensity data to focus its search on the location that will produce the best match, [16]. It examines much fewer potential matches between the photos than conventional approaches, which make it faster, [15].

Generally, the KLT algorithm deals with a number of sequences related to: image edge detection, geometrical transformations (translations, affine, or projections), and computation of the image motion through consecutive frames, correlate the motion of vectors from consecutive frames, it is able to include new points for frames and follow the old and new frames. A mathematical description of the KLT algorithm can be found in [7]-[9] or [16]. During years, many extensive works were addressed to this algorithm with results related to its improvement, [16].

B. Haar Cascade Algorithm

The face detection uses the HC algorithm. This algorithm was first proposed by Paul Viola and Michael Jones [17], [18], and still of a real interest and impact today in computer vision applications. Haar Cascade is a machine learning algorithm, based on a trained cascade function with positive and negative images for a better detection quality.

For face detection, this algorithm is first trained for positive images, then with negative ones. In this case positive images are images with faces in it and the negative ones are images without faces. The sum of the pixels under a white rectangle and the sum of the pixels under a black rectangle are subtracted to provide a single value for each feature. A wide range of characteristics are calculated using all feasible sizes and positions for each kernel. It must determine the total number of pixels under the white and black rectangles for each feature computation.

If, before applying Cascade of Classifier we needed to apply 6000+ features on a window, for an accuracy greater than 95% and then checking if the image is positive or negative, was quite inefficient and time consuming. In a frame the largest area is taken by non-face region, it is better to check if the frame is only non-face area, then if it is to discard the image and not process it to the end, only to be a negative one. The Cascade Classifier focus only on the images that can be positives with a total of 38 stages and 6000+ features, first 5 preliminary stages containing 1, 10, 25, 25 and 50 features. A negative image is discarded in a single shoot, and never processed again. This method is faster and way more efficient.

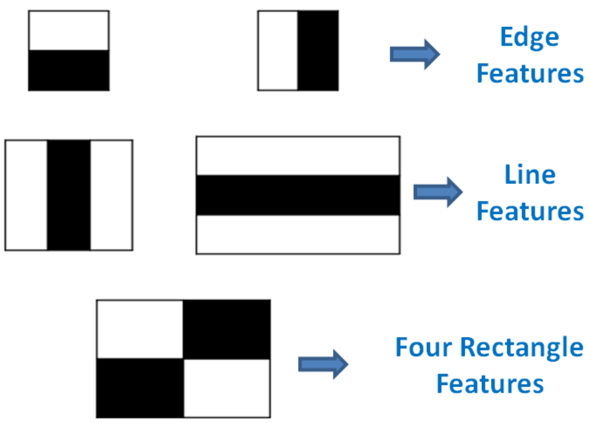

Fig. 3. Haar Cascade Features in a image

C. Face-Traching Mounting Device Algorithm

The opperational process is described in the following steps:

FTMD Algorithm

Input: KLT, Import libraries (*OpenCV, Serial, NumPy and MatPlotLib*)
Output: Servomotor positions and faces detected

1. *KLT algoritm* steps

    Generate a *set_res* function for camera resolution;
    Estabish the hardware communication through a serial object;
    Get a video capture object and sett the resolution using the *set_res* function;

2. Load *Haar Cascade* (HC) classifier for face detection;
3. Create an empty list to store: the servo positions and the number of faces detected;
4. Start the reading and frames processing;

    Read frames

    Flip frames horizontally to counter the mirror effect of the webcam (not always neccesary depending on the webcam model)

    Convert frames to grayscale for face detection

    Detect faces in the frane using the *detectMultiScale* method, from the face cascade in order

    A rectangle is drawn around the detected faces

    The resulted frame is displayed

    If the *q* key is pressed, the loop is ended and the program exits.

5. ***If***

    One face is detected, ***then*** calculate the error between the face center and the frame center;

    Multiple faces are detected, ***then*** calculate the mean center of all the rectangles and calculate the error between the mean center;

        Number of faces = n;
        Mean(x) = [sum x(n)]/n, where x is the center of the face on the x axis;
        Mean(y) = [sum y(n)]/n, where y is the center of the face on the y axis;

        Send the error values as signals to the hardware using the serial object;

        Add the servo positions and the number of faces to their according lists;

    No faces detected, ***then*** send a stop signal to the hardware and append 0 to the servo position and the number of faces lists;

6. Release the camera and serial objects;
7. Plot the servo positions and the number of faces using Matplotlib.

The servo motors can change their orientation with up to one degree at every 3 milliseconds, for both rotation and tilt.

IV. Results

The FTMD algorithm, which includes an association of two algorithms (the KLT and the HC algorithms), is evaluated by letting the webcam to follow for the begging of the tests one single person in the room which stays in front of it, then can move out from the range of the webcam. With servomotors help/information, the webcam/display is moving in real-time following the face/viewer in the room.

In Fig. 4, it is given the sequence on one single person tracking where it is observed the tilt (orange curve) and rotation (blue curve) motions of the two servomotors. In Fig. 5, it can be observed the motion of the servomotors following one single person, than counting a second person who is intervening sometime (randomly) in the range of the webcam at frames 300-500 and 3500 (represented by a blue color).

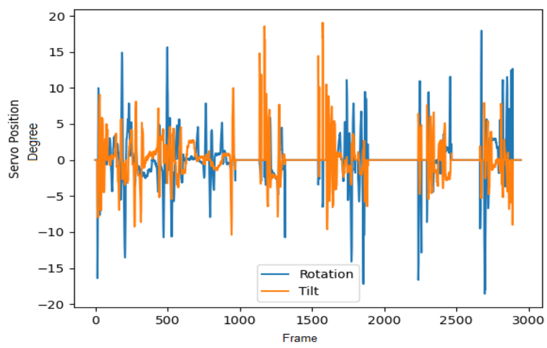

Fig. 4: Motion of the servomotors following one single face

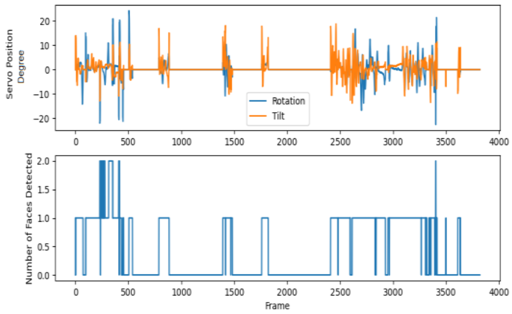

Fig. 5: Motion of the servomotors following one single face, and then adding an additional person (face)

In Fig. 6 and Fig. 7 can be followed the servomotors motions, and the tracking of the faces of two viewers that are moving randomly in the room. Then, the webcam can detect a third viewer entering in its range of detection to the frames 700-1200. It can be observed how the servomotors are adjusting the tilt and rotation motions as well.

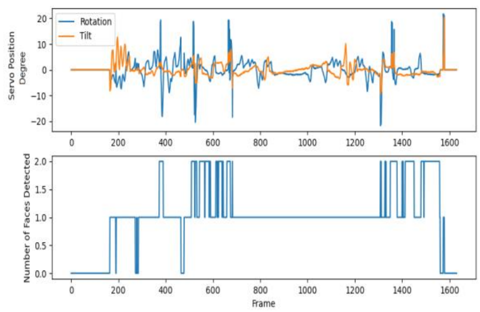

Fig. 6: Motion of the servomotors following two faces with random motion

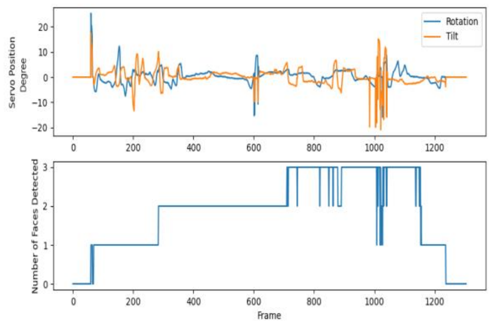

Fig. 7: Motion of the servomotors following three faces

In Fig. 8 there are included three viewers in the room that executes brusque motions. It can be seen in the regions of frames 0-200 and 600-850 how the two servomotors are adjusting their position so that the webcam will track and find the display position correctly after the viewer's locations.

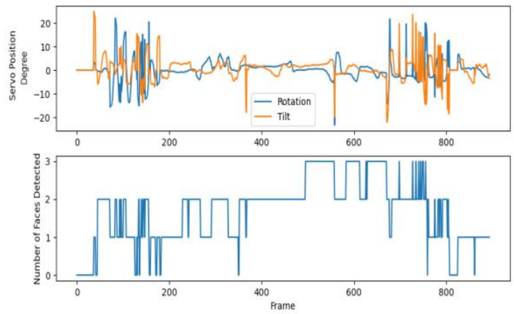

Fig. 8: Motion of servomotors and adjustment of the FTMD, when three viewers are executing brusque motions

V. Conclusions and Future Works

In this paper was presented a face-tracking mounting display that can be used for various applications, from industrial engineering, robotics to medical field including telerobotics surgery (the DaVinci robots). The complexity of this work resides in obtaining a fine and smooth motion of the servomotors of the device so that the display/webcam can accurately track human faces and to be placed in the right position where the viewer's/persons are located into a room. Combining two algorithms based on KLT and HC, and including PyCharm the new FTMD algorithm was able to detect properly, accurate and fast, the viewers in a room. The FTMD algorithm assures an accurate synchronization of the servomotors, and is able to detect the number of viewers watching the display and adjusting its position based on the viewer's location. Further works will include the detection of the face(s) following specific symptoms of work-related, such as fatigue, stress, burnout, or even more severe ones such as fainting, frequent coughing, and eye irritation or (facial) radiation burns caused by harsh work environments, in order to alert specialized medical personnel and prevent further damages to the user's health. Also, as further improvements, for some applications, the mounting system will be articulated in order to make possible the automation of movements that would be done manually, to bring or disperse the display in/out of focus for the users, such as the articulation of the display found on a dental unit/dental chair.

References

[1] M. Azizian, M. Liu, I. Khalaji, S. DiMaio, "The Da Vinci Surgical system", The Enciclopedia of Medical Robotics, Chapter 1, 2019.

[2] C. D'Ettorre1, A. Mariani, A. Stilli, F. Baena , P. Valdastri, A. Deguet, P. Kazanzides, R. H. Taylor, G. S. Fischer, S. DiMaio, A. Menciassi, D. Stoyanov, " Accelerating Surgical Robotics Research: A Review of 10 Years with the da Vinci Research Kit," *(arxiv.org)*, 2021.

[3] https://www.thehealthyjournal.com/frequently-asked-questions/why-a-person-should-sit-away-from-a-tv

[4] M. Kloock, P. Scheffe, I. Tülleners, J. Maczijewski, S. Kowalewski, B. Alrifaee, "Vision-Based Real-Time Indoor Positioning System for Multiple Vehicles", Vol. 53, Issue 2, 2020, pp. 15446-15453.

[5] https://www.geospatialworld.net/prime/business-and-industry-trends/what-is-visual-positioning-system-vps/

[6] R. Szelinski, “Computer Vision – Algorithms and Applications”, eBook ISBN: 978-3-030-34372-9, 2022.

[7] N. H. Barnouti, M. H. N. Al-Mayyahi and S. S. M. Al-Dabbagh, “Real-Time Face Tracking and Recognition System Using Kanade-Lucas-Tomasi and Two-Dimensional Principal Component Analysis,” *2018 International Conference on Advanced Science and Engineering (ICOASE)*, 2018, pp. 24-29, DOI: 10.1109/ICOASE.2018.8548818.

[8] S. Sushma Sri, “Object detection and tracking using KLT algorithm”, *International Journal of Engineering Development and Research,* Volume 7, Issue 2, ISSN:2321-9939.

[9] Y. Han, C. Kim, Y. Jang and H. J. Kim, "Parametric analysis of KLT algorithm in autonomous driving," 2020 20th International Conference on Control, Automation and Systems (ICCAS), Busan, Korea (South), 2020, pp. 184-189, DOI: 10.23919/ICCAS50221.2020.9268239.

[10] https://engineeringdiscoveries.com/how-to-calculate-the-distance-to-view-tv-reasonably-and-safely-for-health/.

[11] U. D. Rosiani, P. Choirina, M.N. Shoumi, “Micro-expression recognition based on motion detection method,” IOP Conference Series, Materials Science and Engineering, vol. 1073, Issue 1, DOI: 10.1088/1757-899X/1073/1/012069, 2021.

[12] P. Pulc, M. Holena, “Towards Real-time Motion Estimation in High-Definition Video Based on Points of Interest,” Proceedings of the Federated Conference on Computer Science and Information Systems pp. 67–70, vol. 11, 2017, DOI: 10.15439/2017F417 ISSN 2300-5963.

[13] D. P. Chau, F. Bremond, M. Thonnat, “Automatic Tracker Selection w.r.t Object Detection Performance,” *2014 IEEE Winter Conference on Applications of Computer Vision (WACV)*, USA, 2014 pp. 870-876, doi: 10.1109/WACV.2014.6836012.

[14] D. Magalhaes, S. Mansoor, Y. Weng, E. Ghizoni, T. Barbosa, F. Silveira, R. Toledo, L.M Li, “ Multi-view autostereoscopic system for 3D visualization in anayomy”, Res. Biomedical Engineering, 34(3), 2018, https://doi.org/10.1590/2446-4740.180066.

[15] https://www.mathworks.com/help/vision/ug/face-detection-and-tracking-using-the-klt-algorithm.html

[16] M. Trummer and J. Denzler, “KLT tracking using intrinsec and extrinsec camera parameters in consideration of uncertainty”, Proc. Of the Third International Conference on Computer Vision: Theory and Applications, pages 346-361, Portugal, 2008.

[17] https://docs.opencv.org/3.4/d2/d99/tutorial_js_face_detection.html

[18] Viola Jones Algorithm and Haar Cascade Classifier | by Mrinal Tyagi | Towards Data Science