



ARDUINO BASED CAR BOT CONTROL VIA HAND GESTURE IMAGE RECOGNITION

Srihari Mandava, Abhishek Gudipalli and Vidhya Sagar G.

School of Electrical Engineering, VIT University, India

E-Mail: mandavasrihari@vit.ac.in

ABSTRACT

This paper presents a novel technique for recognizing the gestures as well as symbols made by hand via human-computer interaction. The main objective is to explore the power of image processing techniques of MATLAB to recognize the gestures made by a human hand. The image of the gesture of hand is captured and processes through stages like image processing, gesture extraction and identification. The first stage extracts the image of hand and separates it from the background. In the next step, the gesture is processed and other noises are filtered and in the third step the gesture is matched with the set of data which is predefined and stored in the database. After the hand gestures are processed and recognized using various techniques and functions, the command that the gesture represents is send to the Arduino based bot which moves accordingly. In addition to the left, right, move and stop commands, the Arduino bot is capable of recognizing the obstacles that comes in the way or come very near to the bot. If the obstacle is detected, the Arduino bot stops moving even if the command is given to it in order to avoid accidents.

Keywords: hand gesture recognition, arduino Uno, MATLAB, image processing.

1. INTRODUCTION

There has been a vast advancement of technology in the field of automobiles. During the early stages, communication to automobile was done manually using the body parts of human like legs and hands. This type of communication is now being replaced by speech, gesture and combination of both [RA Bolt 1980, S Oviatt., *et al* 2000]. Gesture recognition is becoming more popular as it is easy and versatile. Speech recognition is that in which the words and phrases of the sentence spoken are identified and converted them into machine language. The commands which are in machine language are now used by the bot to take action accordingly. Many proven methods like Hidden Markov Models (HMM) [E Trentin., *et al* 2000], Artificial Neural Network (ANN) [H Sakoe., *et al* 1989], Support Vector Machine (SVM) [M. Kavakli., *et al* 2007] exists in literature for speech signal processing technologies and most of them are based on the methodology of stochastic process as the speech is a non-stable characteristic in nature.

Later on the hand gesture recognition is found to be better one as the hands are more effective in functioning and object manipulation than with speech and other body parts [A Erol., *et al* 2007]. Christopher Lee *et al* [LiuYucheng., *et al* 2010] proposed a glove-based gesture recognition system that can recognize 14 of the letters from the hand gestures, learn new gestures updating the model of existing gesture in the system at a rate of 10Hz. Hyeon-Kyu Lee *et al*. [Hyeon-Kyu Lee., *et al* 1999] worked on real-time hand-gesture recognition using HMM. Kjeldsen *et al*. [Rick Kjeldsen., *et al* 1996] made use of back-propagation neural network for recognizing the gestures from the segmented hand images. Etsuko Ueda and Yoshio Matsumoto *et al*. discussed a novel technique in [Etsuko Ueda., *et al* 2003] on hand-pose estimation using multiple images taken by Multiview camera which can be applied for vision-based human interfaces. Chan Wah Ng *et al* proposed another hand

gesture recognition method [Chan Wah Ng., *et al* 2002], in which image furrier descriptor was used as their prime feature and classified using RBF network. Claudia Nölker *et al*. [Claudia Nölker., *et al* 2001] worked on hand gesture recognition using fingertip. In this finger joint angles are all identified using finger tips and hand gesture is prepared by neural network with the help of 3D modal. When hand gesture recognition is considered, a common way is to use hands with special gloves which can capture the movement of hand and fingers through special sensors [S. S. Fels., *et al* 1998]. The alternative method is to make use of vision based technologies of computer in which one or more cameras are used to collect images of the user's hands. These images are then processed by image processing techniques to recognise the hand gesture [Y Kuno., *et al* 1998]. Various literature works like [J Yoon., *et al* 2002, H Kang., *et al* 2004, S Carbin., *et al* 2006, HS Park., *et al* 2006] adopt cameras as gesture input devices. However, data glove technology had become more popular with the advancement of computing and fabrication technology. They are becoming more precise and lighter and some studies have applied data gloves in computer games successfully [Lee Kue-Bum., *et al* 2007]. Hand gestures are of two types: the static and the dynamic [Bastien Marcel Agnes 2009, A Erol., *et al* 2007]. Static gestures are the configurations or the poses of hands in an image. Dynamic gestures can be either as a series of hand postures in a sequence of images or the trajectory of hands. The stochastic characteristics however, to some extent limit techniques used to process hand gestures. The same techniques used for speech recognition like ANNs and HMMs are predominating ones over others in the field of hand gesture recognition also. Hong *et al* [P Hong., *et al* 2000] proposed an approach for 2D gesture recognition which models each gesture as a Finite State Machine (FSM) in spatial-temporal space.

This work is to control objects via gestures or without any kind of physical contact. Many of us are



aware of controlling cars with the help of a glove that responds when we tilt our hand and moves the car in the same direction. In this work, there is no need of any accelerometer based glove to control the car, but could perform tasks just via gestures. However, the project is based on controlling a car with gestures, but we can also make use of this technique to control other gadgets as well. The gesture recognition code could be used in any microcontroller based software and then can be linked with other embedded projects as well for control purpose. The objective of the project work is to focus on developing a new MATLAB code that is capable of recognizing the figures or gestures formed in-front of the camera that is being used as input device. The figures are then used to control an Arduino based bot car that runs, stops and changes directions based upon the gestures made by hand. The robust algorithm can also be used to identify other gestures made by hand and then based upon the code, perform the task that various gestures represent.

In addition to that, the Arduino based bot is equipped with sensor that are capable of detecting any kind of obstacle that comes in the way or very near to the bot. Based upon these sensor inputs, the bot will automatically stops so that there is no collision even if the user wants the bot to move. We can also say that the bot is going to be capable of avoiding obstacles and prevent accidents in case the user gives wrong gesture that might cause the car to move in an unexpected direction and crash. The project uses the power of MATLAB and its image processing functions that provide a method of capturing, extracting and recognizing the hand gesture in real time. A red coloured glove is used for easy recognition and processing as the red colour helps the software to track the palm more accurately. After the hand gesture is extracted from the frame, various functions are used to create surf or focussed points that the software creates in order to compare the current frame with the predefined database of gestures. To make the recognition technique more accurate, histograms of various figures are created, matched with the database and based upon that the output of the bot is determined. After the gesture is known, the Arduino toolbox is used to transfer the result to the Arduino Uno module. The Arduino Uno module initially checks the inputs send by various sensors so as to identify any obstacle that might be present nearby. If obstacle is present, the car does not move in that direction, otherwise it is controlled via the input send to it by MATLAB.

2. PROPOSED SCHEME

The Project is to design a code that can recognize the hand gestures made by a person and control a car bot respectively.

The microcontroller that is used to achieve the functions and run the motors is Arduino UNO. The work is divided into five broad areas in order to achieve the target functionality:

a) Real time image capture

The part of the code involves capturing the hand images and then saving them as single frames. We have used the basic Laptop webcam for this purpose. However, a better auto-focus web camera can also be used for better images and focusing on the hand even when it moves back and forth in the camera field.

b) Creating a database of pre-defined gesture images

In order to compare the captured images with other predefined gestures, we need a database that contains all the images that can be counted as a gesture. Since the size and design of palm varies from person to person, a database of many gestures is required. These gestures to be included in database must vary from one another and must cover the entire camera field so that the recognition process becomes more accurate.

c) Gesture extraction

After the image is received as a Frame, we make use of various functions of image subtraction and region props to extract the gesture or hand from the background. In the end we get a white patch which is hand gesture and rest of the area is black. Histogram and surf points are created in this extracted image so that it can be compared with the database images.

d) Image comparison and gesture recognition

The extracted image's histogram and surfpoints are compared with the database image's surfpoints and histograms using various functions in order to recognize the gesture.

e) Arduino uno bot

The Arduino Uno Car bot is connected to MATLAB which receives the gesture from the software. But, before moving in the required direction, the bot checks for the obstacles that might be present somewhere near. If the obstacles are not present, the bot moves in the direction as made by the person.

3. DESIGN APPROACH AND DETAILS

The very first thing that was required for completing the project was a device that could capture the gestures made by hand in-front. For this, the web camera installed in the laptop is used as it could be easily connected to the MATLAB 2013 as well. Thus, the camera is used to capture the live image of hand in the form of frames which are later processed in order to extract the figure of hand from the entire captured picture.

3.1 Capturing image

In order to capture the images appearing in-front of the camera, *webcamlist* and *imread* function of MATLAB are used. 'Webcamlist' is the function that is used in order to return the list of available UVC compliant web cameras connected to the laptop or PC. The webcam list also allows the user to use the webcam preview option using which one can see the real time captured images as shown in Figure-1. In addition to that, the images can be



saved as well as processed if the user wishes. `IMREAD` is the function that is required to read the images that are saved in a directory. The function reads the image that has been specified by a filename, inferring the format of the file from the available contents. If there are more than one file with the similar filename, the 'imread' reads the first file with that name. The user can also provide the format of the file that the function should read.



Figure-1.

The `imread` function is flexible as many other functions that are offered by MATLAB and this is the reason why one can convert images to RG or can add

transparency etc. directly to the image while reading it from the directory.

3.2 Creating database of images

The very first thing that is to be done is to make a database that could be used for comparing the real time frames with the already saved images. A set of pictures as shown in Table-1 of both hands as well as at various positions is required so that the recognition procedure is accurate. Since the field of the camera is large and it is really hard to predict the position of the hand of the person, having an image at similar position in database is going to provide exact results. It is quite clear that more the number of images in the database, more accurate is the output of the code. However, it won't be right to add too many images in the database since it will make the process too slow and the comparison process slow as well.

3.3 Image processing

In this, the images from the webcam are taken and then process it using various functions available in MATLAB. Some of the functions that have been used in the code include region props, surf, blob extraction etc. Here is a step by step processing procedure of frames captured in real time that includes:

- Converting the Image into Gray Image
- Image Subtraction
- Median Filtered Image
- Blob Extraction and Surf point creation
- Histogram Creation

Table-1. Sample set of images saved in database.



A) Converting the Image into Gray Image



In this, the image is converted into gray image. It is the second stage which prepares the image for further image processing functions. RGB2GRAY is the function that is used to convert a coloured RGB image or changing a colormap to a grayscale image as shown in Figure-2. The function basically eliminates the hue as well as the saturation information and retains the luminance of the image at the same time.



Figure-2. Gray image.

B. Image subtraction

The next step is to subtract the image in the current frame from its background. `Imsubtract` is a function that is used to subtract one image from another similar or dissimilar image. It can also be used to subtract any constant from an image. The function subtracts each and every element present in the array Y from the corresponding element present in X and then returns the difference between both the elements in the output array z. The image with no hand as shown in Figure-3 is taken as the initial background image from which the current image has to be subtracted. Since the current frame includes hand gesture and background image does not contain any hand gesture, subtracting them gives us almost clear image of gesture as shown in Figure-4 that is further processes in order to get a blob of the current image.

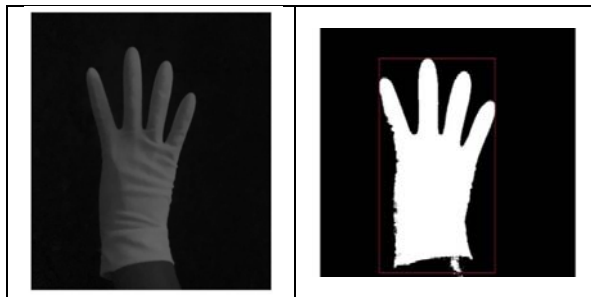


Figure-3. Images without hand and after subtraction.

D. Median filtered image

Median filter is a function that is used to perform median filtering of any matrix A which is available in two dimension. Each and every output pixel possesses a median value present in the 3 x 3 neighbourhood as well as around the corresponding pixel in the image that has been taken as an input.



Figure-4. Median filter image.

The function 'medfilt2' is used to pad the image with value '0' on the edges, so the median values of the points within one and a half the width of the neighborhood of the edges might or might not appear a bit distorted as shown in Figure-4.

C. Blob extraction and surf point creation

Blob Extraction is the final process in which the gesture is extracted perfectly from the image which is shown in Figure-5. Only the glove along with the gesture is extracted in the form of a black and white image which makes histogram production easy as well. In addition to that surf point function is also used to create hundreds focus points as shown in Figure-6 in current image which contains the extracted blob. These focus points are also present in the database images as well and help in detecting the gestures as well.

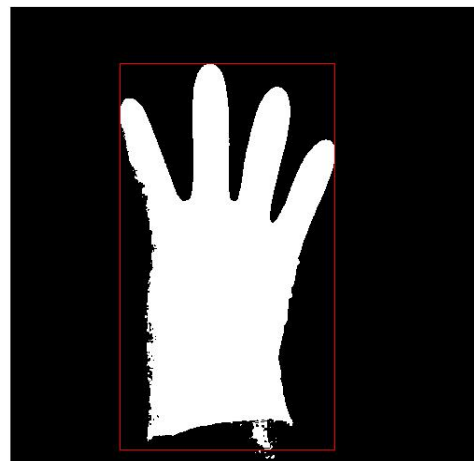


Figure-5. Blob extracted image.

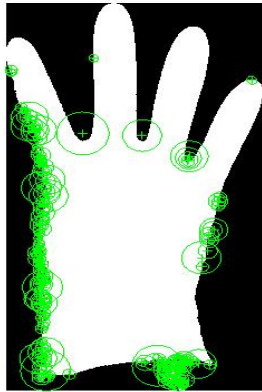


Figure-6. Surf points image.

E. Histogram creation

After getting the blob and getting surf points, a histogram of the current extracted image is created. The histograms of images already saved in the database have the histograms already created which are saved in an array. This is shown in Figure-7. The histogram of the current frame is obtained and compares it with histograms of other images and find out which matches the best. As an example, the histogram of the left turn gesture is shown in Figure-8.

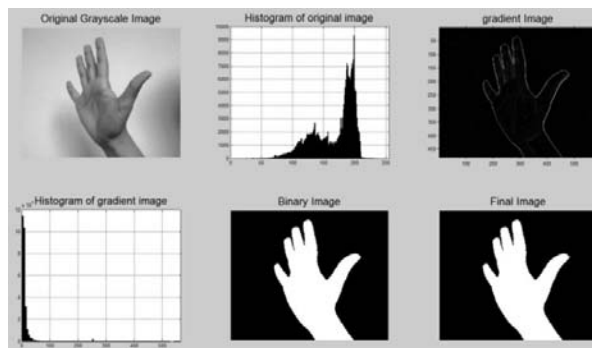


Figure-7. Histogram formation.

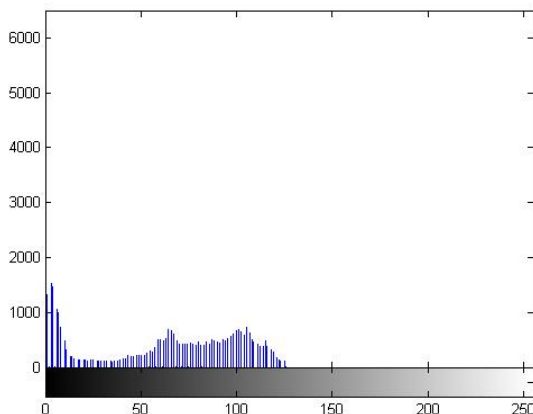


Figure-8. Histogram of left turn gesture.

F. Obstacle avoidance

Obstacle Avoidance in Arduino bot is made possible with the help of the ultrasonic sensor. A threshold value is set that determined when the obstacle is near and when the bot needs to stop. The ultrasonic sensor returns a value depending upon the distance. If the value returned by the sensors goes below the threshold value, the bot takes over and refuses to follow the command given by the user via hand gesture.

This obstacle avoidance method prevents the bot from various kinds of accidents that might occur due to miscommunication between the software and Arduino. Since human reactions are not hundred percent accurate and could include errors, the bot also takes care of its surrounding and movement with the help of the obstacle avoidance code. In addition to the obstacle avoidance, various methods are included that help both the motors to move forward at the same time for forward movement. The turn functions are so defined that only one of the motors move in order to change direction. Once the bot turns to the desired direction successfully, one can give the command to move forward in order to proceed in that particular path. The stop function disconnects power from both the motors.

G. Gear system

The gear system is something determines the speed of the bot by varying the voltage provided to both the motors. The voltage can also be controlled by Arduino and MATLAB. For example, for 1st Gear, minimum voltage can be provided to move the bot slowly. For 2nd gear, the voltage limitation is increased and the bot moves faster. Thus, one can add as many gears by dividing the voltage range into various parts.

One can also feed the gesture recognizing code in the MATLAB code so that the gear system can also be controlled via hand gestures. Similar to gear system, a lot of other features can be added to the bot as well like turning indicators and lights on and off etc. and could be controlled by hand gestures easily.

4. METHODOLOGY

The bot designed using Arduino is as shown in Figure-9. This is a portable one and the step for working with this bot is as follows:

- Starting the MATLAB and adding the files stored in the database to path.
- Start Arduino car bot which becomes ready to take the input from the MATLAB once the code is run on the PC.
- Connecting Arduino bot to PC and connecting it to MATLAB.
- Once the code is running, place the hand in front of the camera and make left, right, move and stop gestures.
- Based upon the gestures, the bot moves left, right, stops and moves.
- If there is any obstacle on the way or if the Arduino car bot comes close to any obstacle from any of the



sides, the bot stops taking the command from the user hand gesture and stops in order to prevent any kind of accident.

5. CONCLUSIONS

The bot performs as per the hand gesture commands given to it and stops when an obstacle is detected and does not move even if the command is given to move in that particular direction. The camera installed in personal computer is used for taking the pictures. The webcam is capable of serving the basic requirement of an image that could be used for the detection purpose. However, due to lack of auto focus and high resolution it is difficult to capture the hand gestures when the hand are moving closer or away fast which can be overcome by high resolution camera.

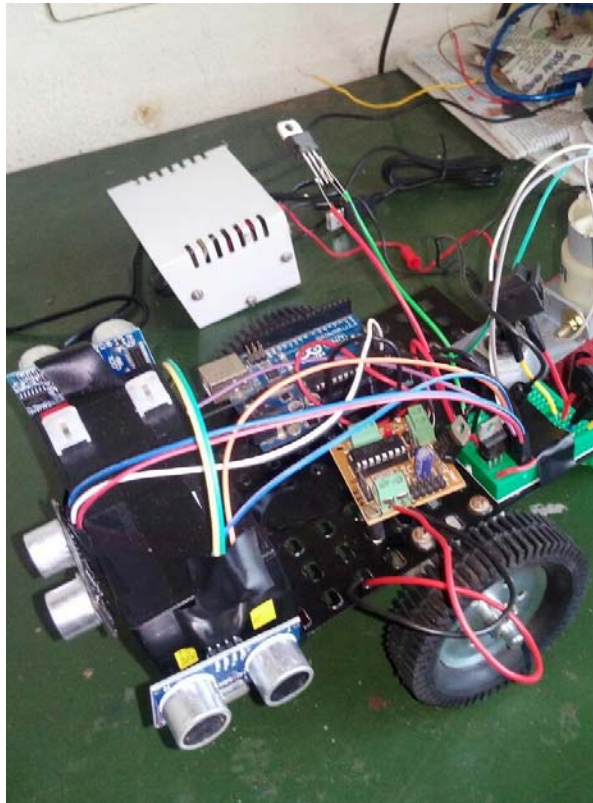


Figure-9. Bot designed using Arduino.

The captured images are then passes through a series of filters as well as functions that help in detecting the hand of the user in the entire frame and extract it. The histogram and surf points are created for the hand gesture alone and then compared with the images that are already there in the database. The histogram comparison gives us the most suitable or most matching gesture command accurately which is transferred to an Arduino bot as a command. The Arduino car bot works standalone as an obstacle avoider and follows the commands at the same time.

Initially the bot check the area around itself. If there is any obstacle in the path or sideways, the bot stops taking the command from the user and stops in order to prevent any type of collision. However, in the absence of obstacle the bot follows the commands that are being passed from the MATLAB and which are in turn given as a gesture at the same time from the user in front of the webcam. Thus, the project excludes the use of accelerometers in order to turn the car and reduces the chance of accidents to a lot extent. Since, it is really hard to maintain the orientation of your hand studded with accelerometer, one can control it in a better way via gesture by making use of this work in real time application. In addition to that, a lot more gestures can be fed into the code to perform various actions as desired.

REFERENCES

- [1] RA Bolt. 1980. Put-that-there: Voice and gesture at the graphics interface. pp. 262-270, ACM New York, NY, USA.
- [2] S Oviatt, P Cohen, L Wu, L Duncan, B Suhm, J Bers, T Holzman, T Winograd, J Landay, and J Larson. 2000. Designing the user interface for multimodal speech and pen base gesture applications: State-of-the-art systems and future research directions. *Human-Computer Interaction*. 15(4): 263-322.
- [3] E Trentin and MGori. 2001. A survey of hybrid ann/hmm Mod-elsfor automatic speech recognition. *neuro-computing*. 37(1): 91-126.
- [4] H Sakoe, R Isotani, K Yoshida, KI Iso, and T Watanabe. 1989. Speaker-independent word recognition using dynamic programming neural networks. in *Acoustics, Speech, and Signal Processing*, 1989. ICASSP-89., 1989 International Conference on. pp. 29-32.
- [5] M. Kavakli, D. Richards, M. Dras, and J. Porte. 2007. An Immersive virtual reality training simulation for risk management. *SimTecT 2007: Simulation Conference: Simulation Improving Capability and Competitiveness*. pp. 1-6.
- [6] A Erol, G Bebis, MNicolescu, RD Boyle and X Twombly. 2007. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*. 108(1-2): 52-73.
- [7] LiuYucheng and Liu Yubin. 2010. Incremental Learning Method of Least Squares Support Vector Machine. *International Conference on Intelligent*



- Computation Technology and Automation" VCL-94-104.
- [8] Hyeon-Kyu Lee and Jin H. Kim. 1999. An HMM-Based Thres-hold Model Approach for Gesture Recognition. IEEE transactions on pattern analysis and machine intelligence. 21(10).
- [9] Rick Kjeldsen and John Kender. 1996. Finding skin in colour images. In Proc. IEEE Int. Conf. on autom. Face and Gesture Recognition. pp. 3 12-3 17.
- [10] Etsuko Ueda, Yoshio Matsumoto, Ma Imai, T Ogasawara. 2003. Hand Pose Estimation for Vision-based Human interface. IEEE Transactions on Industrial Electronics. 50(4): 676-684.
- [11] Chan Wah Ng, S Ranganath. 2002. Real-time gesture recogni-tion system and application. Image Vision Comput. 20(13-14): 993-1007.
- [12] Claudia Nölker and H Ritter. 2001. Visual Recognition of continuous Hand Postures. IEEE transactions on systems man, and cybernetics-part c: applications and reviews. 31(1).
- [13] S. S. Fels and G. E. Hinton. 1998. Glove-talkii-a Neural Net-work interface which maps gestures to parallel Formant speech synthesizer controls. Neural Networks, IEEE Transactions on. 9(1): 205-212.
- [14] Y Kuno, T Ishiyama, K Jo, N Shimada and Y Shirai. 1989. Vision-based human interface system: Selectively recognizing intentional hand gestures. pp. 219-223.
- [15] J Yoon, S Kim, J Ryu, and W Woo. 2002. Multimodal gumdo game: The whole body interaction with an intelligent cyberfencer. Lecture Notes in Computer Science. pp. 1088-1095.
- [16] H Kang, CWoo Lee, and K Jung. 2004. Recognition-based Gesture spotting in video games. Pattern Recognition Letters. 25(15): 1701-1714.
- [17] S Carbini, L Delphin-Poulat, L Perron and JE Viallet. 2006. From a wizard of oz experiment to a real time speech and gesture multimodal interface. Signal Processing. 86(12): 3559-3577.
- [18] HS Park, DJ Jung, and HJ Kim. 2006. Vision-based game Interface using human gesture. Lecture Notes in Computer Science. 4319: 662.
- [19] Lee Kue-Bum, Kim Jung-Hyun, and Hong Kwang-Seok. 2007. An implementation of multi-modal game interface basedon pdas. in Software Engineering Research, Management and Applications 2007. SERA 2007. 5th ACIS International Conference on, Busan, Korea. pp. 759-768.
- [20] Bastien Marcel Agnes Just. 2009. A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition. Comput.Vis. Image Underst. 113(4): 532-543.
- [21] A Erol, G Bebis, M Nicolescu, RD Boyle, and X Tombly. 2007. Vision-based hand pose estimation: A review. Compute Vision and Image Understanding. 108(1-2): 52-73.
- [22] P Hong, M Turk and T Huang. 2000. Constructing finitestate machines for fast gesture recognition. in international conference on pattern recognition, Barcelona, Spain. 15: 691-694.