3D Tracking of Facial Features

Ayham Alharbat, *a.alharbat@student.utwente.nl, 2379589, MSc EE-RAM*
Jeroen Ritmeester, *email address, student id, educational program*
Protik Banerji, *email address, student id, educational program*

***Abstract*—In this paper, a 3D facial features tracking system is developed to track facial features and measure the mobility of the tongue tip. The system uses Kanade-Lucas-Tomasi tracking algorithm to track a set of points of interest, and it uses Random Sample Consensus algorithm to eliminate the outliers and estimate parameters. Then these parameters are used to measure the tracked points in 3D space.**

***Keywords*—Point tracking, RANSAC, Stereovision**

I. Introduction

PATIENTS undergoing surgery and/or radiotherapy in the oral region, especially the tongue, have the risk of limited tongue mobility with serious deterioration of oral functions, such as speech, food transport, swallowing, and mastication. The mobility of the tongue is expressed in the so-called 'Range of Motion'. To study the statistical correlation between the range of motion and a given treatment, this range of motion should be measured in a population of patients. This is to be done before and after the treatment.

To measure the range of motion, patients are asked to move their tongue to standardized, extreme positions, left, right, forward, downward and upward. A triple camera system is used to measure the 3D positions of the tongue tip. Of course, to compare these positions, pre- and post-treatment, the positions should be expressed with reference to a fixed, well-defined coordinate system that is attached to the head. The developed system is able to track the motion of facial features and the tongue tip and then measure the 3D positions of tongue tip to establish a Range of Motion for the tongue tip.

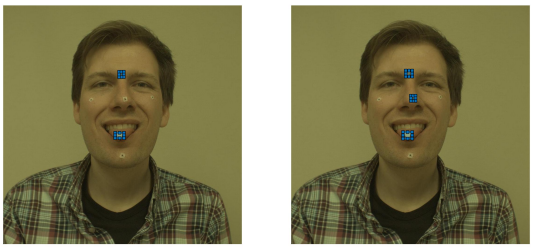
A brief introduction to the topic is presented first, then the methodology which involves the three main topics (2D tracking, 3D tracking, and 3D measurements) is presented, then the results of the experiments will be presented and discussed.

II. METHODOLOGY

In this section, we will present the workflow of our system, from calibrating the cameras to get the camera parameters, selecting and tracking Region Of Interest ROI, triangulating the tracked path of the ROIs to get the 3D positions of these ROIs, until we end up with a Range of Motion for the tongue tip with respect to a coordinate system that is fixed on the face.

A. 2D Tracking

For our final purpose, measuring the Range of Motion for the tongue tip, we need at least 1 point that is fixed on the face to work as an origin to our coordinate system that is fixed on the face. Therefore, the tongue tip and another point will be tracked in 2D. To track these points we start by selecting a small ROI around these two points as shown in Fig. 1. After selecting those ROIs, the minimum eigenvalue algorithm[1] is used to detect corners within the ROI and define them as interest points to track.



(a) Two ROIs (b) Three ROIs

Fig. 1: Selecting ROIs: two ROIs are selected in (a) and three ROIs in (b)

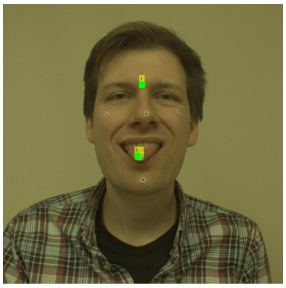


Fig. 2: Two selected ROI with the selected points inside them

However, the ROIs are relatively small and there is a very limited number of interest points that can be found using this algorithm. And because large displacements are expected at the tongue tip we will definitely lose many points along the way, random points inside the ROIs will be selected and also tracked to increase the overall number of the points that are tracked in each ROI as shown in Fig. 2. This set of points will be tracked using the Kanade-Lucas-Tomasi (KLT) tracking algorithm.

[1]This algorithm was chosen after simply comparing the number of points it can detect against other corner detectors that are available on Matlab like Harris–Stephens, BRISK and FAST algorithms.

1) Kanade-Lucas-Tomasi Feature Tracker: Kanade-Lucas-Tomasi Feature Tracker, to be referred to later as KLT tracker, is a tracking method based on [1] and [2]. Where the algorithm tries to locate a set of points $P(x, y)$ in the image $(k+1)$ given its 2D coordinates in the image (k). This method only tracks selected points from the set $P(x, y)$ and discards the other points as untrackable. The selection is based on eigenvalues of the gradient matrix at a given point, if the two eigenvalues are larger than some threshold the point will be tracked, otherwise, it will be discarded.

Therefore, it is almost guaranteed to lose all the tracked points if the tracking was done over a long series of images. So it is important to start with a large set of points $P(x, y)$ and reacquire points periodically. Also, if a large set of points is tracked and since there are large displacements at the tongue tip, some of the tracked points might be tracked incorrectly. Which makes it important to eliminate those outliers, and keep track of only the points that are within the ROI. This can be done using Random Sampling and Consensus (RANSAC).

2) Random Sampling and Consensus: RANSAC is a method of estimating parameters from an observed dataset. This dataset might contain outliers, the algorithm is capable of detecting those outliers and eliminate them so that they do not influence the estimation of the parameters. In the context of this project, RANSAC was used to detect and eliminate the outliers from the tracked dataset, and also to estimate the (x, y) position of the ROIs, i.e. the reference point and tongue tip. After estimating the (x, y) position of the ROIs at a given time and from two cameras in the pixel coordinates, we will be able to estimate the 3D position of these ROIs in the world coordinates using the camera parameters and triangulation.

B. 3D Tracking

Estimating the (x, y) positions of the ROIs in the pixel coordinates for a series of n frames will provide us with the path of these ROIs along the series of n frames. If we have this path (series of (x, y) points for n frames) for two cameras, we can estimate the 3D position of this series of points with respect to a world frame, this is done using triangulation and camera parameters that can be estimated using Camera Calibration methods.

1) Camera Calibration: To be able to transform a certain point $a_p(x, y)$ expressed in pixels coordinates to the world coordinates $a_W(x, y)$ you need to transform first from pixel coordinates to camera coordinates, then to world coordinates. The first transform is done using, what is called, Intrinsics Parameters, and the second is done using the Extrinsics Parameters.

These parameters can be estimated in a process called camera calibration. In camera calbiration, the calibrator is provided with a set of pictures of a checkerboard that has squares of known side length that is also is passed to the calibrator. The calibrator will then detect the corners of the checkerboard's squares and then estimate the parameter by solving the closed-form equation that relates the pixel coordinates to the world coordinates in a pinhole camera model:

$$w\begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix} \begin{bmatrix} R \\ t \end{bmatrix} K \tag{1}$$

Where:

(X, Y, Z) are the world coordinates of the point.
(x, y) are the pixel coordinates of the point.
w is the homogeneous scale factor.
K is the intrinsic parameters matrix.
(R, t) are the rotation and translation of the camera with respect to the world coordinates, i.e. extrinsics parameters.

The calibrator can also estimate the lens distortion model, that can be used to undistort an image. After estimating the required parameters for the two cameras, we can estimate the (X, Y, Z) of a certain ROIs in the world coordinates given a pair of matching points that represent the ROI in the two cameras. This is called Triangulation.

2) Triangulation:

III. Results

Here, you give the results of what is described in Section II: tables, images and/or graphs. The accompanying text of these tables, images and/or graphs clarifies how they are related to the methods described in Section II.

If you have any remarkable observations, mention and describe them here, but without an interpretation, explanation, or meaning. Do not introduce new methods in this section. And do not give an interpretation or a judgement of these results.

IV. Discussion

Give an interpretation and judgement of the results. If you had any remarkable observations, discuss them here to give them meaning (interpretation, explanation, implication).

Describe limitations of the study. If applicable, compare your results with results from literature. If applicable, provide recommendations for further work.

V. Conclusion

A conclusion reviews the main points of the paper. Describe the overall implication of the results to the original problem statement (or research questions). Do not replicate the abstract as the conclusion. A conclusion might also elaborate on the importance of the work, or suggest applications.

Appendix A
Guidelines for formatting Math

If you are using Word, use either the Microsoft Equation Editor or the MathType add-on (http://www.mathtype.com) for equations in your paper.

Number equations consecutively with equation numbers in parentheses flush with the right margin, as in (2). Punctuate equations when they are part of a sentence, as in

$$f_n = \sum_{m=0}^{N-1} F_m exp(2\pi \mathbf{j} \frac{nm}{N}). \tag{2}$$

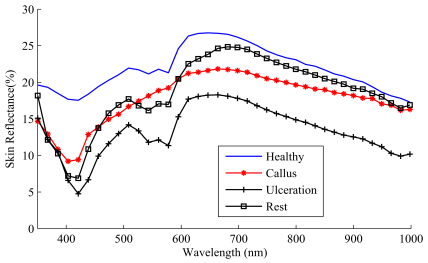


Fig. 3: Examples of spectrum from different classes

TABLE I: An Example of a Table

One	Two
Three	Four

Be sure that the symbols in your equation have been defined before the equation appears or immediately following. Italicize symbols (T might refer to temperature, but T is the unit tesla). Refer to (1), not eq. (1) or equation (1), except at the beginning of a sentence: Equation (1) is ... .

APPENDIX B
GUIDELINES FOR GRAPHS AND TABLES

A. Graphs and Images

Below each figure (graph or image) there must be a caption with a figure number and a figure title. See Fig 1. Figure titles should be legible, approximately 8 to 10 point type. Each figure should be referenced in the text. Each axis of a graph should have a label. Use words rather than symbols. As an example, write the quantity Wavelength, or Wavelength λ, not just λ. Put units in parentheses. Do not label axes only with units. As in Fig. 3, for example, write Wavelength (nm), not just (nm).

B. Tables

Tables should have a table caption on top. See Table I. Tables should be numbered with Roman Numerals (I, II, III, IV, and so on). Tables should also always be referenced in the text.

C. Videos

Videos should be uploaded as separate files. The preferred video format is mp4. Dont make the video files unnecessarily large. 20 Mbytes is acceptable, but 200 Mbyte is not. Appendix A contains a Matlab script that you can use to resize the frame size of a video. The parameter *quality* controls the amount of compression. Sometimes it is also useful to skip frames. For instance, only write the odd frames to the output video.

APPENDIX C
MATLAB SCRIPT FOR RESIZING A VIDEO

```
%% convert a video
clear variables
close all

inputname = 'input\_name.mp4';
outputname = 'output\_name.mp4';
profile = 'MPEG-4';
framerate = 25;
quality = 75;

resize = 1;     % resizing needed?
width = 640;    % if so, this the new width
height = 480;   % and this is the new height
crop = 0;       % cropping needed?
croprect = [ 142        36       563       672];

obj = VideoReader(inputname);
nFrames = obj.NumberOfFrames;
wobj = VideoWriter(outputname,profile);
wobj.FrameRate = framerate;
wobj.Quality = quality;
open(wobj);

% Read and write one frame at a time.
hwait = waitbar(0);
k = 1;
while hasFrame(obj)
    im = readframe(obj);
    if crop                          \% crop if wanted
        im = imcrop(im,croprect);
    end
    if resize                        \% resize if wanted
        im = imresize(im,[height width]);
    end
    writeVideo(wobj,im);
    if~mod(k,10)==1,waitbar(k/nFrames,hwait);end
    k = k+1;
end
delete(hwait);
close(wobj)
```

APPENDIX D
GUIDELINE FOR REFERENCES

In text, refer simply to the reference number. Do not use Ref. or reference except at the beginning of a sentence: Reference [3] shows ... .

REFERENCES

[1] Lucas, Bruce D., and Takeo Kanade. "An iterative image registration technique with an application to stereo vision." (1981): 674.

[2] Tomasi, Carlo, and Takeo Kanade. "Detection and tracking of point features." (1991).