**Analogue Clock**

**Project Proposal**

****

**Spring 2021**

**CSE301L-Signal $ System**

Submitted by: **Ashfaq Ahmad (19pwcse1795)**

**Osama Marwat (19pwcse1777)**

**Ataullah Khan (19pwcse1746)**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Engr Durr-e-Nayab**

July 12, 2021

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**Objective:**

* To design the **Analogue clock** using Matlab Software.

**Introduction:**

* Clocks are instruments that measure and show the time. An analogue clock is a clock or watches that has moving hands and (usually) hours marked from 1 to 12 to show you the time.
* Some have Roman Numerals (I, II, III, etc) instead, or no numbers at all, instead only relying on the positioning of the hands and what angle they’re at to indicate the time.
* An analogue clock contains two (or possibly three) “hands" that go around in circles to show you the current hour and minute (and possibly second).
* Each of these hands can point anywhere along the circle. That means that the minute hand can point to every numerical value that exists between 0 and 60.
* For example, when the minute hand is halfway between the 11 and 12 minute marks (both of which are between the 2 and 3 hour marks), the time must be 11.5 minutes past some hour. Halfway between that point and the 12 minute mark, the time must be 11.75 minutes past some hour.
* As each hand moves continuously over a portion of its circular path, it moves through an infinite number of real numbers that represent the continuous moments of time.
* Something that changes continuously like this—meaning it moves smoothly from one point to another without any sudden jumps or breaks—is called an analogue quantity. And that’s why this type of clock is called an analogue clock.

Methodology:

A circle with a radius of 10 units is drawn or plotted. Hours are marked from 1 to 12, 30° apart. First, the numbers are converted to string format by using an inbuilt function in MATLAB, i.e.**‘num2str’**and then by using inbuilt ‘**text**’ function in MATLAB 1 to 12 is written as text in the plot. **System time such as our PC’s is read by the command ‘clock’ which is an inbuilt function in MATLAB.** According to the current time of your PC, the hours, minutes, and seconds hands are displayed. As the clock function returns time in 24-hour format, the first step is to convert time to 12-hour format.

The minute hand moves 360 degrees or 2 π radians in 60 minutes (or 6 degrees in one minute) and hour hand moves 360 degrees in 12 hours (or 0.5 degrees in 1 minute).

In h hours and m minutes, the minute hand and hour hand would move by :

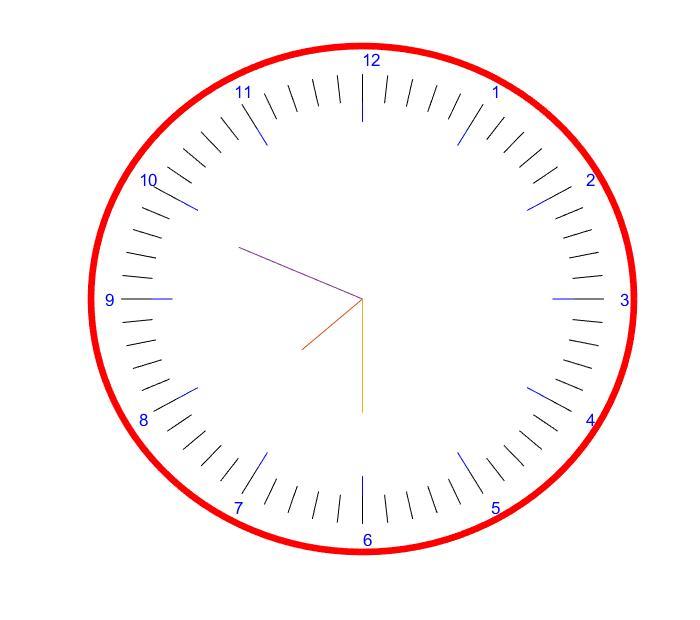
Minute\_hand= (h\*60 + m)\*6 degrees

Hour\_hand= (h\*60 + m)\*0.5 degrees

The second-hand moves 360 degrees or 2 π radians in 60 seconds (6 degrees in one second), in‘s’ seconds, the second hand would move by **(s\*6) degrees**.

As for every one second, the time changes, the plotted figure pauses for one second and the current figures, i.e. the hour hand, minute hand, and second hand are deleted and plotted again. After every one second, the cycle repeats. In order to pause and delete the figures, the inbuilt functions**‘*pause*’** and **‘*delete*’** are used.

After running the code in MATLAB, we can see an analog clock running in the screen and the time shown is synchronized with that of the system.



**The above figure is the result that we got after running the MATLAB code and the time shown by the clock will be system current time.**

**Conclusions:**

* Our conclusions from this program, is that in computer vision, there is no easy solution for anything. Each problem must have a specifically-designed solution based on trial and error, using different approaches, and combining different methods. In Our program, we attempted to apply a multi-stage solution for analog clock construction. Our goal was to correctly define all stages of construction, and reach a working implementation which has an acceptable success rate and can be applied to many image variations. We feel that we managed to correctly identify the stages needed to solve the problem. Regarding the implementation, our program has identical results with a wide variety of pictures, regardless of clock background, style, color and shape. Our success rate overall is acceptable. Most failures occur during the final stage, and we have yet to find a foolproof way of classifying clock hands. We feel that we have learned a lot and gained valuable experience during this project, and have managed to implement some of the Matlab vision theory studied in class.