Memorias del 2023 Congreso Nacional de Control Automático
25-27 de Octubre, 2023. Acapulco, Guerrero, México.

Image Reconstruction using an Array of Photoresistors and Kalman Filter

Bastida M. Valentina * **Camacho L. Rubén** ** **Ruiz F. Ivan** ***

* *Institute of Applied Science and Technology, UNAM, Coyoacan, CDMX 04510 Mexico. zotaquitabm@ciencias.unam.mx*
** *ruben_ezequiel@ciencias.unam.mx*
*** *eivan.rf@gmail.com*

Abstract: On this paper we present a mechano-electrical system capable of scanning and acquiring images that are enhanced using Kalman filter. As photosensitive element we chose photoresistors that restrict the resolution of our system to their diameter. The whole system consists of three main parts: the scanning device, the acquiring circuit and finally an image processing stage. To process the image, we designed a Kalman Filter, that allows us to save two images, the one acquired directly from the mechano-electrical circuit and the one enhanced by the filter. Our system proposes a simple, small, low cost scanning solution that can be used as a camera for images that have little curves on their morphology.

Keywords: Image Reconstruction, Kalman Filter, Photorresistor, Arduino, Digital Signal Process.

1. INTRODUCTION

For a long time, image processing was a complidated task due the equipment and software needed to succesfully process data from pictures, then only a small group of people, scientist and engineers, were able to work on this field. Currently, we have many potential accesories such as digital cameras or scanners along with much more capable processors and computers that make this labor easier.

Hardware and software have evolved in a way where the usage of common photosensitive devices such as photorresistors can sense the intensity of light and translate it, helped by a microcontroller, to a digital signal in order to generate representative pixels of a surface or even a complete picture, working as a digital camera. To complete the data processing, the algorithms have evolved too, making the mathematical process more efficient and powerful.

1.1 Photoresistors

We must talk about photoresistors or Light Dependent Resistor (LDR), which are semiconductor devices that vary their electrical resistance in response to the intensity of light that falls on their sensitive area. This is due to the fact that they are composed of semiconductor materials, generally cadmium sulfide (CdS) or lead selenide (PbSe) [1]. In such a way that when the photons of light fall on the surface of the semiconductor material, some of them are absorbed by the atoms of the material, which causes the release of electrons in the semiconductor, these free electrons contribute to electrical conduction and, as a result, , the resistance of the photoresistor decreases; on the other hand, when there is little or no light falling on the photoresistor, the number of electrons released is less, which causes the resistance of the device to increase [2].

1.2 Kalman Filter

Kalman Filter is an estimation algorithm developed by Rudolf E. Kalman in the 1960s, used in signal processing and control systems to estimate the state of a dynamic systems from noisy or incomplete measurements [3]. The main objective of the Kalman Filter is to obtain an accurate estimate of the current state of a system, based on past measurements and the mathematical model of the system, so the filter can be divided into two stages: the prediction stage and the correction stage [4].

In the prediction stage, the mathematical model of the system is used to predict the future state of the system, which implies making an extrapolation of the current state using the laws of evolution of the system and taking into account previous measurements; so that an estimate of the future state and an estimate of the uncertainty associated with that prediction are obtained.

In the correction stage, the most recent measurements are incorporated to improve the estimate of the current state, so the prediction made in the prediction stage is compared with the real measurement of the system and then the difference or residual error is calculated. Through a weighted calculation, the prediction is combined with the measurement to obtain a more precise estimate and the associated uncertainty is updated.

1.3 Image processing and its application with photosensitive arrays

Image processing, in general, involves a series of stages, including image acquisition, pre-processing to correct artifacts or noise, feature extraction, image manipulation and transformation, and finally, analysis and interpretation of the data. results obtained. To reach a good result, techniques and algorithms are used, such as filtering, spatial and frequency transformations, edge detection, segmentation, pattern recognition, and machine learning.

But speaking specifically of what has been done in terms of using photosensitive arrays, there are two great examples of systems previously done. In the first of them [5], although it was possible to build an array of NxM photoresistors, the results are not the best since repeated information may be received, or even interference from the environment would be expected, but it is something to highlight. which is a fixed arrangement that can only be handled with Arduino.

While in the second case found [6],they used an eigth phototransistor array, and at the same time better images were acquired after a series of data interpolations within Matlab, but the system displaced the array, so that a sweep was with two axes movement made over a sheet of paper with letters on it, the results of this system are show in Fig. 1.

On this document we present an Image Reconstruction System based on a photorresistors array that is working as a sensor and is connected to the Analog inputs of an Arduino board, the analog values are read in a MATLAB program which reconstructs the image pixel by pixel and includes a Kalman Filter for image processing and filttering. Previous works and projects have been inspiration for this work where the scanning process si done differently, also the algorithm used is Kalman Filter and is done directly in MATLAB, this one being the greatest difference.

2. METHODOLOGY

In the following subsections we will explain the different stages for the development of this work, our project is divided into four main sections, which are the circuit made with the photoresistors, the step motor function, the acquisition code and the processing of the image made.

2.1 Photorresistor array

The type of photoresistor array selected is based on a linear array similar to the one mentioned during section 1.3. Due to the amount of ADC channels available,

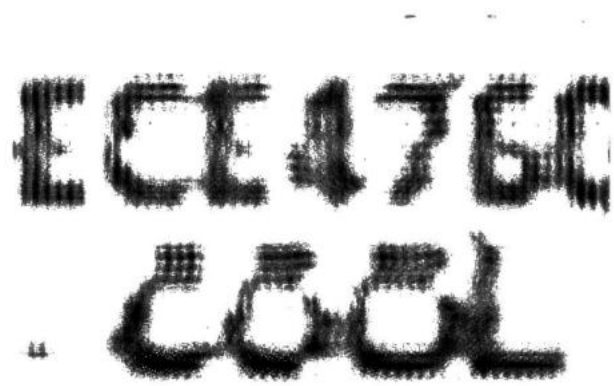

Fig. 1. Reconstructed image of the case study of a mobile linear array of phototransistors. Obtained from [6].

given by the hardware architecture (microcontroller ATmega328P), the final arrangement was a 6x1 arrangement, shown in Fig. 2; so that the change in voltage given by each one of the photoresistors was carried out by means of an Arduino UNO board, which is based on the ATmega328P microcontroller, that has a maximum working frequency of 16 [MHz], 6 A-D conversion channels, and 14 I/O digital channels, with this characteristics the development board is well suited for this project.

As considerations regarding this array, pieces of thermofit were placed around the photoresistors, in order that each one of them had an external light limiter. An effort was made to prevent the same information from being obtained on more than one photoresistors; finally the circuit was made with photoresistors in series with a fixed resistor of 1 KΩ, to obtain a voltage divider, were the output voltage is the voltage from the photoresistors.

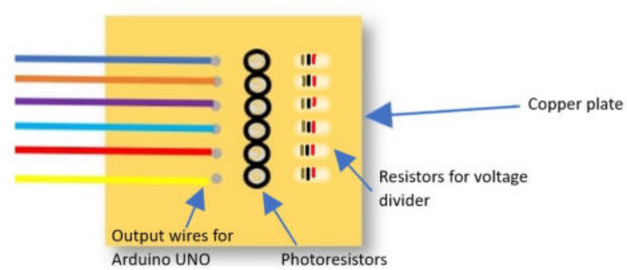

Fig. 2. Array of photoresistors with the thermofit placed, soldered on a copper plate.

2.2 Motor

The purpose of implementing a motor was given based on the case of the photoresistor sweep mentioned above, but since it was considered that the photoresistors could be affected by movement, and since they are the most important elements of the system for obtaining information, we opted for implementing this engine to move the piece of paper where the image to be registered was located.

In this way, a Nema 17 motor was used, while one end of the paper was adhered to a rod fixed to the motor, the other end was adhered to a second rod that would rotate due to the effect of the motor itself, thus generating a kind of roller, the mechanic system is represented in Fig. 3; with this a sweep over the image was carried out, with the distinction that what moves is the image, this is a differentiation with respect to previous image reconstruction systems that use photoresistor arrays.

It should be noted that, in initial tests, when acquiring data from the photoresistors by allowing the motor to take one step, the motor rotated 1.8^o, thus the reconstructed image presented data overlapping. In image terms the overlap caused the reconstructed figures to appear wider than they really were. For this reason through trial and error, a good number of steps was reached in order to eliminate data overlapping, but without greatly increasing the total data acquisition time, therefore every 4 steps a measurement was acquired.

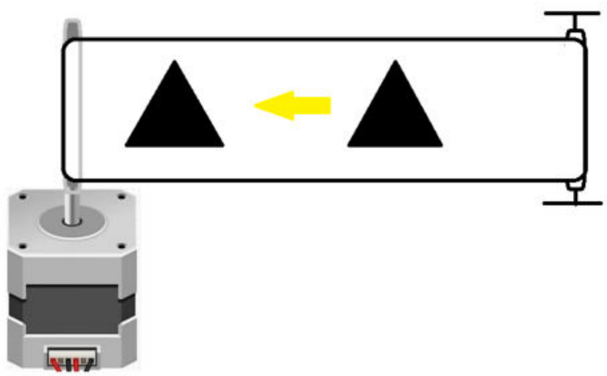

Fig. 3. Representation of the paper sheet movement system through the rotation of the Nema 17 motor.

With the sheet of paper attached to the plastic rods, this is the final representation of the data acquisition system, without really talking about the entire digitization and information processing process.

2.3 Adquisition Code

The acquisition code for the photoresistor array data consists of 3 large items, the first is the communication beteween Arduino UNO and MATLAB 2022b, the second is the generation of a square signal with high and low values corresponding to 1 and 0, which establish the moments in which the Nema 17 motor moves, and finally the saving of the data of each analog singal once it is digitized in data vectors; since there were 6 photoresistors, the scanning was limited to 100 measurements, or 400 motors movements, at the end of this section of the code a 6x100 data matrix was obtained that contained the information for getting an image, following is the code to control de step motor and to save the data to reconstruct a whole image.

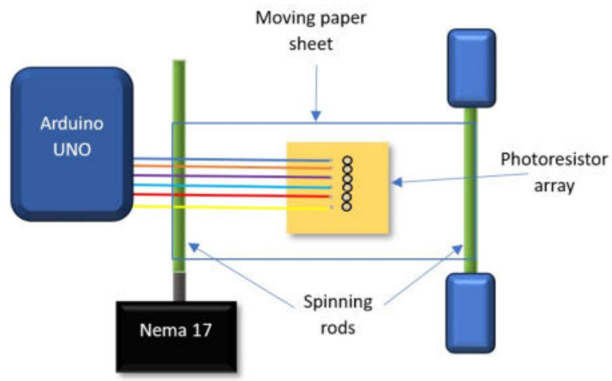

Fig. 4. Final design for the physical part of the system, which consists of the photoresistor arrangement, the motor and the printed sheet; the array is under the sheet.

```
clear;
close all;
clc;
a=arduino();

lecturas= 100

writeDigitalPin(a,'D5',0);
i=1;
while i<=lecturas
    writeDigitalPin(a,'D4',1); %Valor
       en Alto a la salida del pin52
    pause(0.001)                  %pausas
        de 10 ms que forman una senial
       cuadrada
    writeDigitalPin(a,'D4',0); %%Valor
       en Bajo a la salida del pin52
    pause(0.001)
    writeDigitalPin(a,'D4',1); %Valor
       en Alto a la salida del pin52
    pause(0.001)                  %pausas
        de 10 ms que forman una senial
       cuadrada
    writeDigitalPin(a,'D4',0); %%Valor
       en Bajo a la salida del pin52
    pause(0.001)
    writeDigitalPin(a,'D4',1); %Valor
       en Alto a la salida del pin52
    pause(0.001)                  %pausas
        de 10 ms que forman una senial
       cuadrada
    writeDigitalPin(a,'D4',0); %%Valor
       en Bajo a la salida del pin52
    pause(0.001)
    writeDigitalPin(a,'D4',1); %Valor
       en Alto a la salida del pin52
    pause(0.001)                  %pausas
        de 10 ms que forman una senial
       cuadrada
```

```
    writeDigitalPin(a,'D4',0); %%Valor
       en Bajo a la salida del pin52
    pause(0.001)%pausas de 10 ms que
       forman una senil cuadrada
    an1= readVoltage(a,'A0');
    an2= readVoltage(a,'A1');
    an3= readVoltage(a,'A2');
    an4= readVoltage(a,'A3');
    an5= readVoltage(a,'A4');
    an6= readVoltage(a,'A5');
    a1(i)=an1;
    a2(i)=an2;
    a3(i)=an3;
    a4(i)=an4;
    a5(i)=an5;
    a6(i)=an6;
    i= i+1;
end
% end
ima= [a1; a2; a3; a4; a5; a6];
figure(1)
imagesc(ima);
ima= rescale(ima, -1, 1);
figure(2)
imshow(ima, []);

imagesc(ima);

imwrite(ima, 'imagne.png');
```

After that the code focuses on the Kalman Filter, on this part we stablish the parameters of the image that we want to enhance.

```
%KALMAN FILTER FOR IMAGE RECONSTRUCTION
%Define system parameters
num_pixels = 600; % Total number of
   pixels in the image
A = eye(num_pixels); % System dynamics
   matrix
B = zeros(num_pixels,1);% Control input
    matrix
C = eye(num_pixels); % Measurement
   matrix
Q = 10*eye(num_pixels);% Process noise
   covariance
R = 1000*eye(num_pixels);% Measurement
   noise variance

% Initialize state and covariance
   matrices
x = zeros(num_pixels,1);% State vector
   (initially zero)
P = eye(num_pixels); % Covariance

% Load input image
input_image = imread('imagne.png')
input_image = imresize(input_image, [6
   100]); %Resize to 10x10 pixels
z = double(input_image(:)); %
   Measurement vector


% Run Kalman filter
num_frames = 1; % Number of frames to
   process
filtered_images = zeros(6,100,
   num_frames);
for i = 1:num_frames
    % Predict step
    x = A*x + B;
    P = A*P*A' + Q;
    % Update step
    K = P*C'/(C*P*C' + R);
    x = x + K*(z - C*x);
    P = (eye(num_pixels) - K*C)*P;
    % Reshape state vector into image
       matrix
    filtered_imagne = reshape(x, [6
       100]);
    % Store filtered image in output
       matrix
    filtered_images(:,:,i) =
       filtered_imagne;
    % Display filtered image
    imshow(filtered_imagne,[]);
    drawnow;
    filtered=imagesc(filtered_imagne)
    % Save filtered image as PNG file
    filename = sprintf('imagne.png_%d.
       png', i);
    imwrite(filtered_imagne, filename);
    % Capture new measurement (in this
       example, the same input image is
        used for simplicity)
    z = double(input_image(:));

end
```

2.4 Kalman Filter

For the image processing, we used a Kalman Filter based code that can be seen in section 2.3. The full code is divided in two segments the first part writes an image with the data previously saved in the matrix, the second part applies Kalman filter and writes a new figure with the filtered image.

For the first section the matrix of measurments, control, and noise are generated, all of this based in the number of pixels (600 pixel). Then, all the matrixes are operated to be filtered and processed.

3. RESULTS AND DISCUSSION

To verify the operation of both the physical system and the processing part of this work, 5 cases were taken as samples, Fig. 5, a circle, a square, a star, an inverted triangle and the word ”HOLA”, example images can be seen in Fig. 5.

The star was not ideal because of it own morphology, the peaks were lost in the resulting image. A similar case occurred with the triangle, but in this case it was possible the identification of its shape in the images.

Fig. 5. Figures taken for testing the implemented system and code.

On the other hand, the word ”HOLA” did not show a clear image, which was associated with the thickness of the letters, that was smaller than the resolution of the system, this can be anhancen by changing the number of steps the motor took to save one sample. But we wanted to have a program that would work with all the images to be analyzed, so it was discarded as proof of optimal performance.

The case of the square, due to its own geometry, did not represent a problem in being identified, it can be seen in Fig.6, but at the same time it represented the easiest figure to obtain, so it also did not function as an optimal case to verify operation; therefore we present only the cases, before and after the image processing of the circle and triangle. These images can be identified with the naked eye, but its geometry allowed to verify the change of intensities in the pixels and check the correct operation of the Kalman filter processing. The circle can be visualized in figures 7 y 8. On the other hand triangle images can be seen in images 9 and 10.

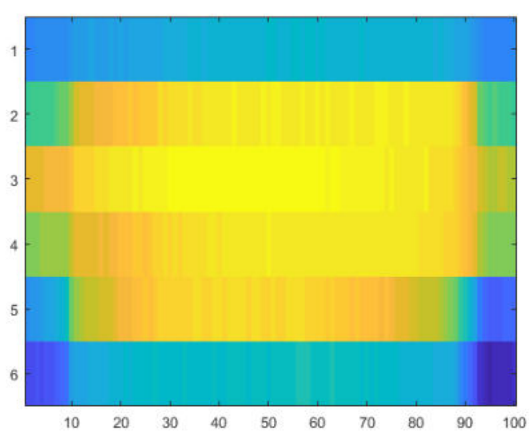

Fig. 6. Square image without filter.

An observation about the scanning system is that the images obtained presented a greater intensity when detecting the shadow of the figure to be reconstructed, so the printed figures on the sheet correspond to light-colored

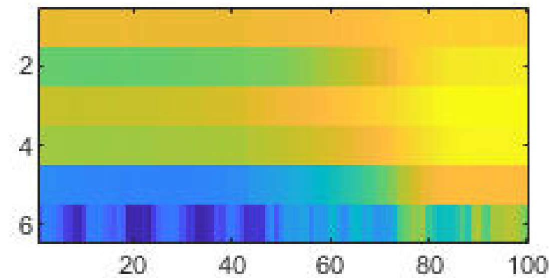

Fig. 7. Circle image without filter.

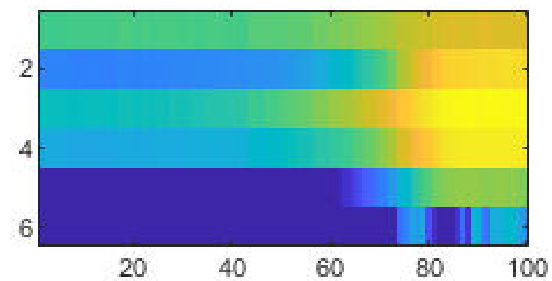

Fig. 8. Circle image with filter.

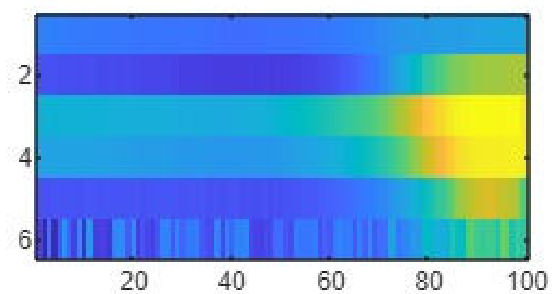

Fig. 9. Ttriangle image without filter.

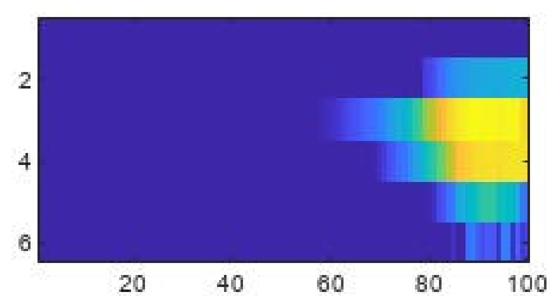

Fig. 10. Triangle image with filter.

pixels; contrary to what was observed in the example in Fig 1. This has to do with the circuit design and it could be changed, but digital image processing allows to obtain the negative of an image, so this would be unnecessary.

This results show that by making a simple scanning system that is very restricted in resolution, due to its phtosensitive element, an image can be obtained that also allows an enhancing process to take part. By making a simple implementation of a scanning device, a digital filter acquires a great weight within this work. Kalman filter was a good solution to the need of filtering a low quality image, because of its computational low cost, although the filter could be improved, the preliminar results show a very good performance, that can be further exploited for low cost, portable image systems.

4. CONCLUSION

It was possible to successfully carry out the implementation of a physical system that allowed the scanning of a series of basic geometric figures printed on a sheet of paper, by moving this sheet. Data was acquired by means of an Arduino UNO board; in this way, it was possible to process the data, through the use of a Kalman filter, demonstrating not only the usefulness of this mathematical method in image processing, but also the total operation of the system proposed in this work . Likewise, although factors were found that can hinder the system and that can be modified to improve it, such as the distance between samples of the sheet, or the same geometry and characteristics of the figures used as proof, it was possible to identify and improve the images corresponding to a square, a circle and an inverted triangle.

The system could be improved, for example by having a better band for the images, this may include a second motor to control the band and keep thense the paper, also placing a box around the system to isolate the sensing hardware from ambient light and external noise. And finally increasing the sensing elements, and making the array a matrix with more columns, this change in the quantity of pixels could lead to a better quality of the image.

ACKNOWLEDGEMENTS

We thank the Institute of Engineering of the National Autonomous University of Mexico (UNAM), and Dr. Giovanni Chavarría for providing the space to carry out this work, as well as the support shown during its development. Also, we appreciate the support from Dra. Flor Lizeth Torres Ortiz, she provided us the Kalman Filter Code used on this project, we could not finished this project without her contribuition to it.

REFERENCES

[1] Jeff Christenson. "Chapter Three - Sensors and Transducers". In: *Handbook of Biomechatronics*. Ed. by Jacob Segil. Academic Press, 2019, pp. 61–93. ISBN: 978-0-12-812539-7. DOI: https://doi.org/10.1016/B978-0-12-812539-7.00003-9. URL: https://www.sciencedirect.com/science/article/pii/B9780128125397000039.

[2] *Fotorresistencia (LDR) (1) - Introducción*. https://soloarduino.blogspot.com/2014/09/la-resistencia-variable.html. Accessed: 2023-06-10.

[3] Greg Welch and Gary. Bishop. *An Introduction to the Kalman Filter*. 2001.

[4] *Kalman Filter*. Accessed: 2023-06-10. URL: https://ece.montana.edu/seniordesign/archive/SP14/UnderwaterNavigation/kalman%5C_filter.html.

[5] Arduino Team. *Building the worst digital camera ever*.

[6] Yunchi Luo and Mengliang Yu. *ECE 4760: Final Project Image Scan using Phototransistor Array*.