

Integración de Sistema de Digitalización 3D Usando Matlab

Rodríguez-Morales Gustavo¹, Guzmán-Ramos Valentin², Martínez Hilariona², López López Prudencio² y Garza Álvarez Adriana Guadalupe²

¹Facultad de ingeniería Mecánica y eléctrica, Universidad Autónoma de Nuevo León

²Depto. Mecatrónica, Universidad Tecnológica "Gral. Mariano Escobedo"
gustavo.rodriguezml@uanl.edu.mx

Resumen

Se presenta la implementación de un escáner 3D utilizando Matlab y el micro controlador ATmega328 mejor conocido como Arduino. El objetivo de este trabajo es el llamar la atención de los estudiantes de ingeniería para que el proceso de aprendizaje tenga mejor eficiencia. El escáner 3D tiene como objetivo el extraer información de los relieves de los objetos, información que se utiliza después en los procesos de manufactura a través de la maquinaria CNC o la impresora conocida como prototipadora rápida para reproducir el objeto digitalizado. El sistema laser 3D requiere de pocos componentes para integrarlo, estos son un láser, una lente cilíndrica, una cámara y un motor a pasos con sus controladores. Para realizar el escaneo de forma automática se utiliza un micro controlador el cual montado en la tarjeta llamada arduino permite la comunicación vía USB con matlab para controlar todas las funciones del escáner. El uso de la tarjeta Arduino se hace debido a su bajo costo y la fácil implementación de controles a través de esta.

Palabras clave: integración de sistemas, luz estructurada, control, arduino.

1. Introducción

El integrar un escáner para la digitalización de objetos en tres dimensiones es un tópico muy atractivo para estudiantes del área de mecatrónica, electrónica y control. Por otro lado como parte de la llamada reingeniería el escáner es ampliamente usado industrialmente [1][2]. En el proceso de digitalización en 3D existen tres técnicas principales [2][3], 1) por triangulación, esta técnica utiliza dos puntos de referencia para calcular el tercer vértice de un triángulo, el cual corresponde a la altura del objeto en digitalización [1] [4]. 2) la técnica de tiempo de vuelo utiliza pulsos de luz para determinar la altura del objeto. 3) por interferometría [5], esta es una técnica de medición de alta precisión, pero requiere de un

interferómetro. Esta técnica produce un patrón de franjas paralelas. El objeto a digitalizar deforma estas líneas lo que nos permite cuantificar la altura del objeto. Esta técnica puede ser simplificada al generar las franjas con un proyector en lugar del interferómetro. La primera y tercera técnicas se conocen como de luz estructurada [1][2].

En este trabajo, utilizamos la primera técnica que consiste en una línea laser la cual se deforma por la presencia del objeto y entonces se captura por una cámara CCD. Las imágenes capturadas se procesan con Matlab para extraer la información de la deformación de la línea laser y cuantificar la altura del objeto por la técnica de triangulación.

Cabe destacar que algunas de las aplicaciones importantes de la luz estructurada son en mediciones sin contacto con la muestra, o también conocidas como no invasivas y no destructivas [6]. Las nuevas tecnologías en sensores inteligentes utilizan esta técnica de visualización en aplicaciones de robótica y de visión por computadora [3].

Por otra parte, los cursos de programación están basados en prácticas utilizando modelos de física para simular algún proceso, lo cual es poco atractivo para el estudiante debido a que no refleja la práctica de la ingeniería que ellos buscan. Una buena alternativa es la integración de un proyecto a lo largo del semestre que los anime a investigar e implementar las aplicaciones de su interés con el lenguaje de programación.

Otro factor que suele ser decisivo para la implementación de un proyecto por un estudiante es lo costoso que suelen ser los componentes, sobre todo las tarjetas de adquisición de datos y control. También el uso de electrónica compleja que desalienta al estudiante a llevar a cabo este tipo de proyectos. La existencia de tarjetas para prototipos de bajo costo y de fácil uso abre un gran número de

posibilidades para que el estudiante desarrolle habilidades de integración de proyectos aplicando el conocimiento teórico aprendido en clase.

La tarjeta para prototipos usada en este trabajo es la llamada Arduino cuya información de construcción y uso está abierta a todo público en la internet [7]. Otro componente electrónico que se utilizó en este trabajo es el controlador del motor a pasos, llamado EasyDriver V4. El cual solo requiere que se le indique cuando mover el eje y en qué dirección para tener movimiento en el motor.

En la siguiente sección se analiza teóricamente la el funcionamiento del láser. En la sección 3 se describe la implementación desde el punto de vista de programación. En la sección 4 se analizan resultados obtenidos con el escáner.

2. Luz estructurada

2.1 Consideraciones teóricas

El sistema óptico consiste en un láser y una lente cilíndrica para generar la línea laser. Una cámara para capturar la escena del objeto con la línea laser. El láser y la cama deberán estar en planos diferentes los cuales hagan un ángulo para que se pueda visualizar la deformación de la línea laser (ver Fig. 1.)

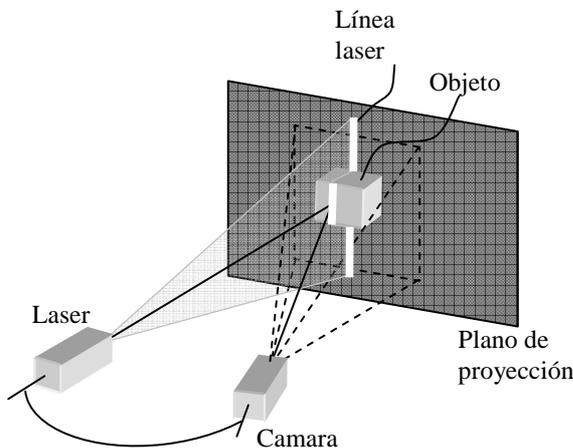


Fig. 1. Esquema del sistema para luz estructurada

2.2 Extracción de la trayectoria laser por correlación

El método consiste en calcular la correlación cruzada entre una función gaussiana de dos dimensiones y la información de la imagen. La

función Gaussiana está dada por

$$g(r) = \exp\left(-\frac{x^2}{w_x} - \frac{y^2}{w_y}\right), \quad (1)$$

Donde w_x y w_y son el ancho de la gaussiana en cada dimensión.

La correlación cruzada entre dos funciones $f(r)$ y $g(r)$ esta dad por

$$f(r) \otimes g(r) = \int_0^{\infty} f^*(\rho) g(r + \rho) d\rho, \quad (2)$$

Donde f^* es el conjugado complejo de la función. Por otro lado, podemos usar el teorema de la correlación y calcularla de la forma

$$f(r) \otimes g(r) = \mathcal{F}\left\{\mathcal{F}\{f(r)\}^* \mathcal{F}\{g(r)\}\right\}, \quad (3)$$

Donde \mathcal{F} es la transformada de Fourier.

El cálculo de la correlación resalta la intensidad de los pixeles que pertenecen a la línea laser, y disminuyen la intensidad de la información de fondo. Este proceso discrimina la luz del ambiente y hace más fácil el localizar el máximo de la línea laser.

En este método el valor del ancho de la Gaussiana (w_x y w_y) tiene un efecto un importante en resultado de la correlación. En la figura 2 se muestra los resultados del cálculo para $w_x = 1, 10, 20, 40$, y $w_y = 1$. Esta serie de imágenes muestra la importancia del ancho de la gaussiana para obtener una línea continua de los máximos de intensidad.

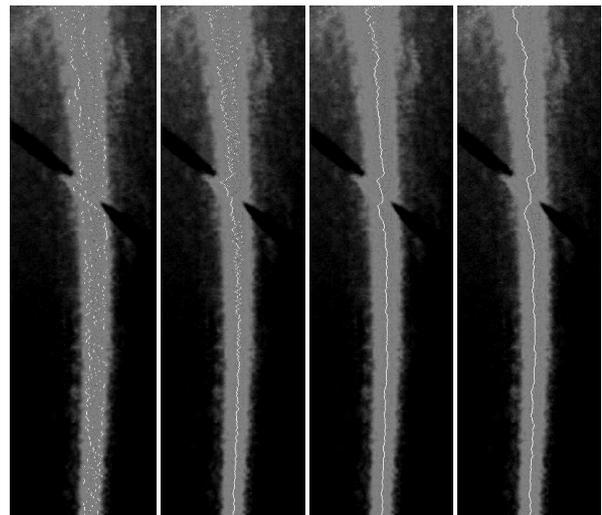


Fig. 2. Serie de imágenes mostrando los efectos del ancho de la Gaussiana en el resultado de la correlación. De izquierda a derecha $w_x = 1, 10, 20, 40$, y $w_y = 1$.

En las figuras 2 y 3 el ancho de la línea laser es variable, una medición directa nos indica que varía desde 10 a 40 pixeles desde la parte inferior a la parte superior.

El resultado de la correlación con $w_x = 1$ produce una curva discontinua prácticamente en todos sus puntos debido a que estamos detectando el “moteo” característica inherente de la radiación laser. En este caso una alta resolución en dirección horizontal no es efectiva para aplicar en este experimento. Conforme se incrementa el ancho de la gaussiana la curva de máxima intensidad se vuelve continua en las zonas donde el ancho de la gaussiana es aproximadamente el ancho de la línea laser. Con lo cual estamos detectando la línea laser y no el “moteo” de la misma. El efecto se ilustra claramente en la imagen correspondiente a $w_x = 10$, esta muestra continuidad solo en la parte inferior que es donde la línea laser tiene aproximadamente 10 pixeles de ancho. Conforme se incrementa el ancho de la gaussiana la curva de máximos se vuelve continua como se muestra en las imágenes que corresponden a $w_x = 20$ y 40. Con $w_x = 40$ se alcanzado el máximo de ancho de la línea laser y la curva de máximos es continua en su totalidad mostrando la variación espacial de la línea laser proporcionando la información real del perfil del objeto.

El ancho de la gaussiana deberá tener un límite pues el hacer el ancho muy grande provocara que la correlación fusione la línea laser con el fondo de la imagen lo cual genera incertidumbre en las mediciones.

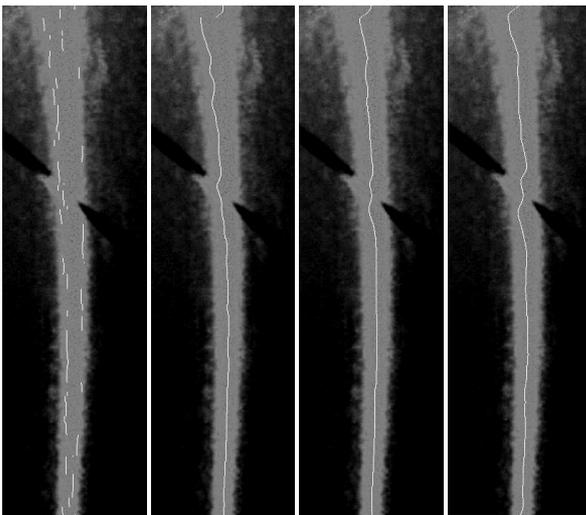


Fig. 3. Serie de imágenes mostrando los efectos del ancho de la Gaussiana en el resultado de la correlación. De izquierda a derecha $w_x = 1, 10, 20, 40$, y $w_y = 10$.

Realmente no hay una condición para el ancho de la gaussiana, pero recomendamos mantenerla en algún valor cercano al ancho de la línea laser.

La figura 3 muestra el mismo proceso que en la figura 2 solo que para un ancho diferente en la dirección vertical. El incremento en el ancho vertical reduce la dependencia del ancho horizontal para alcanzar una curva continua de máximos. Con $w_y=1$ la correlación se calcula prácticamente para cada renglón de la imagen. Mayores anchos verticales implican el involucrar pixeles de los renglones vecinos en el cálculo reduciendo la resolución vertical de la medición.

2.3 Cálculo geométrico de la altura del objeto

La reconstrucción digital de la superficie del objeto inicia con el cálculo de la altura del objeto determinada a través de la deformación de la línea laser. Este cálculo se realiza con algebra vectorial, los vectores se muestran en la figura 4. El plano que contiene la línea laser esta descrito por la ecuación

$$(\mathbf{r} - \mathbf{r}_0) \cdot \hat{\mathbf{n}} = 0, \quad (4)$$

Donde \mathbf{r}_0 es un punto conocido en el plano. El punto \mathbf{r} es un punto arbitrario en el plano., en este caso seleccionamos el punto en la intersección entre el plano y el eje óptico de la cámara.

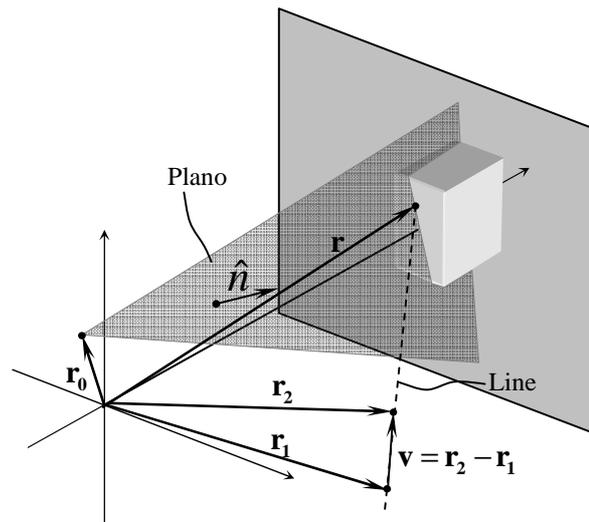


Fig. 4. Definición de los vectores usados para el cálculo de la altura del objeto.

La ecuación de la línea recta en forma vectorial está dada por

$$\mathbf{r} = \mathbf{r}_1 + t\mathbf{v}, \quad (5)$$

Donde \mathbf{r}_1 es un punto conocido en la línea, y \mathbf{v} es un vector que tiene la dirección de la línea, en este caso construido por $\mathbf{r}_2 - \mathbf{r}_1$. En la cámara estos puntos pueden ser el centro del sensor y un punto en el vértice exterior de la lente de la cámara.

Sustituyendo la ec. (1.7) en (1.6) y resolviendo para t obtenemos

$$t = \frac{(\mathbf{r}_0 - \mathbf{r}_1) \cdot \hat{\mathbf{n}}}{\mathbf{v} \cdot \hat{\mathbf{n}}}, \quad (6)$$

Sustituyendo este parámetro en la ec. (5) obtenemos las coordenadas del punto de intersección.

3. Implementación

La implementación consiste principalmente de la adquisición de datos, el procesado de imágenes y el control del sistema.

3.1 Hardware

El sistema consiste de un microcontrolador para el control del movimiento de la plataforma donde se contiene el objeto que se desea digitalizar. También el control del láser se realiza a través de este microcontrolador. Comúnmente el programar estos dispositivos es una tarea difícil si no se es familiar con esta. Sin embargo la opción de la electrónica de prototipos de fuente abierta (open source prototyping electronics) resulta muy atractiva en cuanto a contar con una tarjeta de adquisición de datos accesible a los estudiantes por su bajo costo (aproximada mente \$400.00 pesos) y facilidad de uso. La tarjeta utilizada aquí es conocida como arduino la cual utiliza el microcontrolador ATMEGA328 de ATMEL. El lenguaje de programación es llamado Wiring [8] es un lenguaje de fácil aprendizaje muy parecido al lenguaje C.

Manejar un motor eléctrico siempre requiere de una cantidad grande de energía, y es común el usar componentes electrónicos independientes de la tarjeta de control para suministrar esta energía sin poner en riesgo el microcontrolador. En este caso utilizamos el controlador llamado EASYDRIVER V4.2. Este controlador para motores a pasos es de fácil uso, el cual, además puede controlar el motor hasta en 1/8 de

paso, es decir si utilizamos un motor de 200 pasos por revolución tendremos una resolución de 1600 micro-pasos por revolución. La resolución vertical la proporciona la cámara. En este caso las posibles resoluciones son las que tenga disponibles la cámara. Para este trabajo utilizamos la cámara Logitech C300 que cuenta con un sensor de 1.3 megapíxeles.

3.2 Software

El software para el funcionamiento del sistema se divide en dos partes, una respecto a la programación del microcontrolador para la comunicación y la segunda respecto a la programación en Matlab para el control y análisis de mediciones. El programa del microcontrolador se realizó en el lenguaje de programación de Arduino, que, como se mencionó antes este está basado en el lenguaje Wiring. La rutina tiene el objetivo de establecer comunicación serial con Matlab para recibir órdenes para el control del motor y el láser.

3.3 Arduino

El microcontrolador tiene un número pequeño de instrucciones entre las que encontramos instrucciones para la estructura del programa, para manejar variables, y funciones. En esta ocasión, usamos solamente cinco funciones para el control del motor y el láser. La adquisición de datos se realiza a través de la cámara la cual se controla directamente desde Matlab.

La comunicación a través del puerto USB se establece usando las funciones del puerto serie. Las funciones usadas fueron begin(), available(), read(). Para el control del motor y el láser se utilizan las funciones pinMode() y digitalWrite() para establecer los puertos y enviar información respectivamente. Un ejemplo esquemático del programa es al siguiente

```
void setup() {
  Serial.begin(9600); /*inicia serial*/
}
void loop() {
  /* ciclo principal*/
  static int pm = -1; /* estado */
  if (Serial.available() > 0) {
    /*lee dato serial en ascii */
    val = Serial.read();
    /* aqui van las instrucciones
    para las acciones a efectuar */
    /* establece el puerto de salida */
    pinMode(pin,OUTPUT);

    /* Envia datos digitales al Puerto */
    digitalWrite(pin,0);
```

```

        digitalWrite(pin,1);
    } /* end if */
} /* end loop */
    
```

3.4 Matlab

Las rutinas en Matlab pueden ser divididas en comunicación y captura, cálculo y la interfaz gráfica (GUI).

3.5 Comunicación y captura

La comunicación en Matlab puede ser llevada a cabo como se muestra en el siguiente esquema de rutina

```

% define el objeto serial
aser=serial(comPort);

% Abre el puerto
try
    fopen(aser);
    catch ME,
        error(['No puedo abrir: 'comPort]);
    end
    
```

Mientras se desarrollaba este trabajos nos enteramos de las rutinas desarrolladas por Giampiero Campa del equipo de Mathworks[9] las cuales proveen la comunicación en el mismo estilo en que se presenta aquí. Estas rutinas son una opción para simplificar aún más el proceso de integración del sistema. Basta cargar estas rutinas en el microcontrolador y el resto de la programación es completamente hecha en Matlab.

El envío de datos al microcontrolador se realiza con la instrucción

```
fwrite(aser,[ascii values],'uchar');
```

Donde los valores indican el número de Puerto y los datos a enviar.

La cámara se caracteriza de la siguiente forma

```

% Información de las camaras activas
Cmr=imqhwinfo('winvideo');
% Informacion de la 1ª camara disponible
dev_info = imqhwinfo('winvideo',1);
% Informacion de la resolución
A=dev_info.SupportedFormats';
% extrae el nombre de la 1ª resolución
c=cell2mat(A(1));
% Establece la cámara y resolución a
% usar
obj=videoinput('winvideo',1,c);
% captura una imagen
I=getsnapshot(obj);
    
```

3.6 Análisis de datos

Para el cálculo de la correlación se genera la función gaussiana de la siguiente forma

```

[X,Y]=meshgrid(x,y);
wx=15;
wy=1;
G=exp(-((X-max(x)/2)/wx).^2 - ...
      ((Y-max(y)/2)/wy).^2);
    
```

Donde la longitud de los vectores x y y deben corresponder a las dimensiones de la imagen. Una vez establecida la función gaussiana la correlación la calculamos como

```

% TF de la imagen y de la gaussiana
FI=fft2(I(:,:,1)); % solo color rojo
FG=fft2(G);
% correlación cruzada
CC=fftshift(real(fft2(FI.*conj(FG))));
    
```

Aquí discriminamos la parte azul y verde de la imagen a manera de filtro debido a que usamos un láser color rojo.

Después de la correlación encontramos el máximo de intensidad usando la función find()

```
[row,col]=find(CC(n,:)==max(CC(n,:)));
```

Esta instrucción localiza el máximo del n-esimo renglón y proporciona su posición en términos de píxeles lo cual corresponde al índice del arreglo que tiene la información de las intensidades de los píxeles. Finalmente se compara la deformación de la línea laser con respecto a la línea laser cuando no se tiene objeto. La medición estará dada en unidades de píxeles y la conversión a unidades de longitud se hace a través de las ecuaciones (5) y (6)

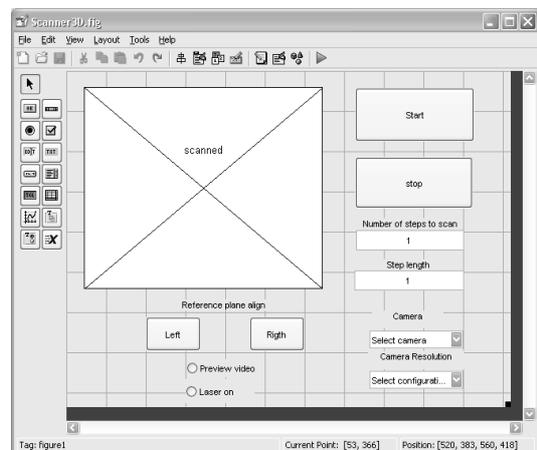


Fig. 5. Editor de GUIs de Matlab mostrando los controles para el sistema de digitalización 3D.

3.7 GUI

Otra característica del trabajo es elaborar una interface grafica para que el sistema pueda ser usado por usuarios que no tengan conocimiento de Matlab. Un objetivo [particular es el incluir la mayor cantidad de parámetros que tengan que ver con el control del sistema y su uso eficiente.

La figura 5 muestra una opción de interface donde se controla la cámara que se usara, así como su resolución, el número de capturas en la digitalización, el número de pasos del motor entre cada captura, y la posición del objeto sin captura para efectos de alienación antes de la digitalización. La figura 5 muestra la configuración de los controles en el editor de GUIs de Matlab.

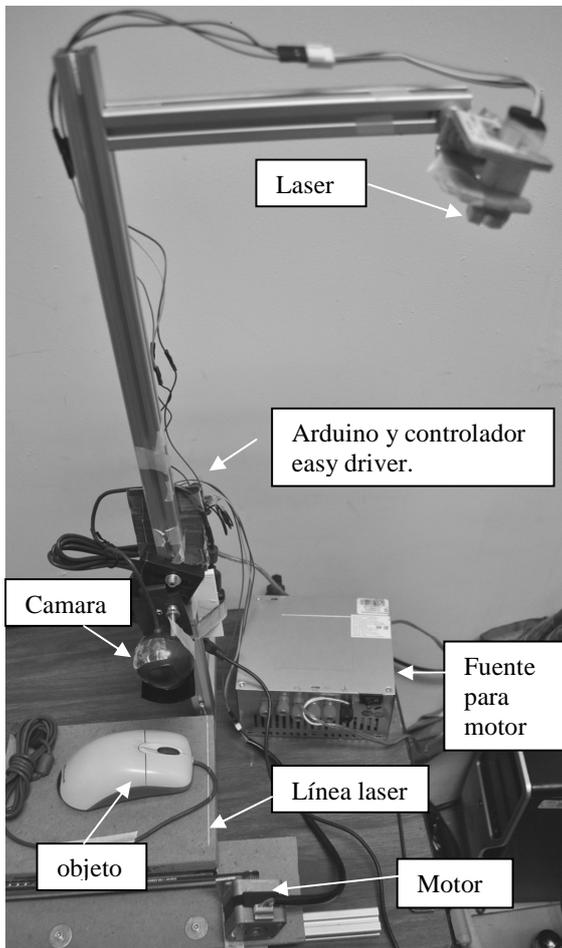


Fig. 6. Hardware del sistema integrado

4. Análisis de resultados

El hardware del sistema integrado se muestra en la figura 6. La figura 7 muestra un ratón de computadora digitalizado con el sistema desarrollado en este trabajo. Como es visible la digitalización es se realiza con buena alta fidelidad a reserva de las zonas de occlusión (o sombras) que producen información errónea en la digitalización. Las líneas la variación en las líneas transversales se debe a la vibración que tiene el objeto.

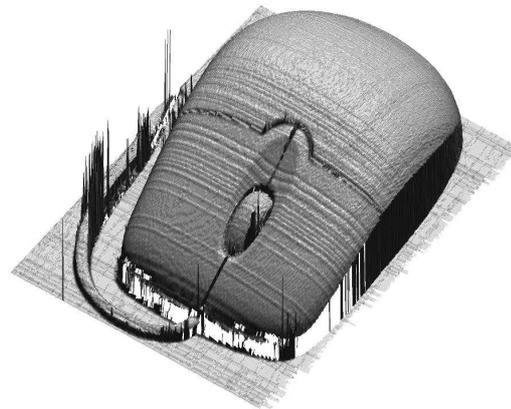


Fig. 7. Objeto digitalizado

Referencias

- [1] Raja V., Fernandes K., *Reverse Engineering, an industrial perspective*. London: Springer-Verlag 2008
- [2] Chen F., Brown G., Song M., "Overview of three-dimensional shape measurement using optical methods," *Opt. Eng.* vol. 39 no. 1pp 10-22, January 2000
- [3] Jähnd B., Haußecker H., *Computer vision and applications, a guide for students and practitioners*. San Diego, CA: Academic press, 2000
- [4] Gasvik K., *Optical metrology*. London: John Wiley and sons, 2002
- [5] Hariharan P., *Optical interferometry*. San Diego, CA: Academic Press, 2003
- [6] Demeyere M., Rurimunzu D., Eugéne C., "Diameter measurement of spherical objects by laser triangulation in an ambulatory context" *IEEE Trans. Instrum. Meas* vol 56 no. 3 pp 867-872 June 2007.
- [7] <http://www.arduino.cc>
- [8] <http://wiring.org.co/>
- [9] <http://www.mathworks.com/academia/arduino-software>