

TRABAJO ESPECIAL DE GRADO

**DISEÑO E IMPLEMENTACIÓN DE UN INSTRUMENTO PARA  
LA DETECCIÓN DE DESPLAZAMIENTOS ESTRUCTURALES  
PERMANENTES O DERIVAS MEDIANTE USO DE CÁMARA Y  
PROCESAMIENTO DE IMÁGENES.**

Presentado ante la Ilustre  
Universidad Central de Venezuela  
Por el Br. Suarez C. Daniel A.  
Para optar al título de  
Ingeniero Electricista

Caracas, 2017

TRABAJO ESPECIAL DE GRADO

**DISEÑO E IMPLEMENTACIÓN DE UN INSTRUMENTO PARA  
LA DETECCIÓN DE DESPLAZAMIENTOS ESTRUCTURALES  
PERMANENTES O DERIVAS MEDIANTE USO DE CÁMARA Y  
PROCESAMIENTO DE IMÁGENES.**

Tutor: Ing. Rafael Rivero, M.Sc.

Presentado ante la Ilustre  
Universidad Central de Venezuela  
Por el Br. Suarez C. Daniel A.  
Para optar al título de  
Ingeniero Electricista

Caracas, 2017

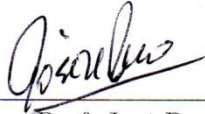
## CARTA DE APROBACIÓN

Caracas, 2017.

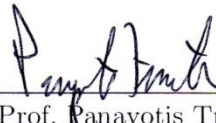
Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por el Bachiller Suarez C. Daniel A., titulado:

**“DISEÑO E IMPLEMENTACIÓN DE UN INSTRUMENTO PARA LA  
DETECCIÓN DE DESPLAZAMIENTOS ESTRUCTURALES  
PERMANENTES O DERIVAS MEDIANTE USO DE CÁMARA Y  
PROCESAMIENTO DE IMÁGENES.”**

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la mención de Electrónica, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.



Prof. José Romero  
Jurado



Prof. Panayotis Tremante  
Jurado



Prof. Rafael Rivero  
Tutor Académico

## **DEDICATORIA**

A mi familia, apoyo incondicional.

## **RECONOCIMIENTOS Y AGRADECIMIENTOS**

Agradecimiento especial a mi familia.

Agradezco también al profesorado de la Facultad de Ingeniería de la UCV y particularmente de la Escuela de Ingeniería Eléctrica, de manera destacada al Prof. Rafael Rivero, tutor de este trabajo y Prof. Servando Álvarez.

Suarez C. Daniel A.

## **DISEÑO E IMPLEMENTACIÓN DE UN INSTRUMENTO PARA LA DETECCIÓN DE DESPLAZAMIENTOS ESTRUCTURALES PERMANENTES O DERIVAS MEDIANTE USO DE CÁMARA Y PROCESAMIENTO DE IMÁGENES.**

Tutor Académico: Rafael Rivero. Tesis. Caracas U.C.V. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica. Ingeniero Electricista. Opción: Electrónica. Institución: LIDE. 2017. 59 h + anexos.

**Palabras Clave:** Procesamiento de Imágenes; Python; OpenCV.

**Resumen.** Los elementos que conforman una estructura tienen un rango natural de movimiento, cuando uno de estos elementos supera ese rango aumenta la vulnerabilidad de la estructura. Un indicador del movimiento de los elementos estructurales de una edificación lo constituyen las distancias que los separan. Este trabajo aborda la problemática de medir y monitorizar estas distancias para poder detectar tempranamente algún desplazamiento riesgoso. Con un hardware consistente de una cámara web y un computador embebido BeagleBoard-xM y usando como herramienta el procesamiento de imágenes, más concretamente el reconocimiento de marcadores presentes en la escena se llega a un programa con el que se obtuvieron resultados con desviaciones porcentuales menores al 3% para distancias mayores a 10 cm. El programa funciona de forma continua, es capaz de reconocer cuando una distancia supera un límite establecido por el usuario y reconoce la presencia de obstáculos en la imagen.

## INDICE GENERAL

CARTA DE APROBACIÓN	iii
DEDICATORIA	iv
RECONOCIMIENTOS Y AGRADECIMIENTOS	v
RESUMEN	vi
ÍNDICE GENERAL	vii
LISTA DE FIGURAS	xii
LISTA DE TABLAS	xiv
ACRÓNIMOS Y SIGLAS	xv
INTRODUCCIÓN	1
CAPÍTULO I	3
1 PROBLEMA	3
1.1 PLANTEAMIENTO DEL PROBLEMA	3
1.2 PROPUESTA DE SOLUCIÓN	4
1.3 OBJETIVOS	4
1.3.1 Objetivo General	4
1.3.2 Objetivos Específicos	4
1.4 ANTECEDENTES DE INVESTIGACIÓN	5
CAPÍTULO II	7
2 MARCO REFERENCIAL TEÓRICO	7
2.1 IMAGEN	7
2.1.1 Imagen en escala de grises	7
2.1.2 Imagen a color	7
2.2 IMAGEN DIGITAL	7
2.3 MATRIZ	8
2.4 ESPACIO DE COLOR	8
2.4.1 Espacio de color RGB	9
2.5 PROCESAMIENTO DE IMÁGENES	9

2.5.1 Segmentación	9
2.5.1.1 Thresholding	9
2.5.2 Momento	10
2.5.2.1 Momento Geométrico	10
2.5.2.2 Momento Central	11
2.5.2.3 Momento Central Normalizados	11
2.5.2.4 Momentos Hu	12
2.6 OPENCV	13
2.7 ESTRUCTURA	13
2.7.1 Deflexión	13
2.7.1.1 Deriva	13
CAPÍTULO III	14
3 DEFINICIÓN DEL SISTEMA	14
3.1 COMPONENTES FÍSICOS DEL SISTEMA	14
3.1.1 Computador	14
3.1.2 Cámara	15
3.2 SOFTWARE	15
3.2.1 Módulo de Configuración y Calibración	16
3.2.2 Módulo de Medición	17
CAPÍTULO IV	19
4 DEFINICIÓN DEL SISTEMA	19
4.1 COMPONENTES FÍSICOS DEL SISTEMA	19
4.1.1 Computador	19
4.1.2 Cámara	20
4.2 SOFTWARE	21
4.2.1 Módulo de Configuración y Calibración	25
4.2.1.1 Componente de Calibración	27
4.2.1.2 Componente de Configuración	31
4.2.2 Módulo de Medición y Comunicación	34



4.2.2.1 Componente de Medición de Distancias	36
4.2.2.2 Componente de Servicio de Peticiones	41
CAPÍTULO V	42
5 PRUEBAS DE VALIDACIÓN. RESULTADOS	42
5.1 MEDICIÓN DE DISTANCIAS	44
5.1.1 Resultados de las pruebas de Medición de Distancias	45
5.2 DETECCIÓN DE OBSTÁCULOS	47
5.2.1 Resultados de las pruebas de Detección de Obstáculos	47
5.3 DETECCIÓN DE DERIVA PERMANENTE	48
5.3.1 Resultados de las pruebas de Detección de Deriva Permanente	49
5.4 LIMITACIONES POR CONDICIONES DE ILUMINACIÓN	49
5.4.1 Resultados de las pruebas de Limitaciones por Condiciones de Iluminación	51
5.5 Pruebas en condiciones reales	51
5.5.1 Resultados de las pruebas en condiciones reales	53
CONCLUSIONES	55
RECOMENDACIONES	56
REFERENCIAS BIBLIOGRÁFICAS	57
BIBLIOGRAFÍA	59
ANEXOS	62

## LISTA DE FIGURAS

Figura 1	Matriz o arreglo bidimensional	8
Figura 2	Conjunto Cámara - Computador	14
Figura 3	Casos de uso del software propuesto	16
Figura 4	Computador BeagleBoard xM	19
Figura 5	Cámara Genius WideCam 1050	20
Figura 6	Marcador usado para demarcar distancias a medir	21
Figura 7	Actividades para el reconocimiento de marcadores en la imagen	24
Figura 8	Módulos del Software	25
Figura 9	Clases del programa de configuración y calibración	26
Figura 10	Actividades para la calibración y configuración	27
Figura 11	Desplazamiento del marcador	28
Figura 12	Diámetro de marcador	29
Figura 13	Actividades de calibración	30
Figura 14	Actividades de configuración	33
Figura 15	Archivo de configuración	34
Figura 16	Clases del módulo de medición y servicio de peticiones	35
Figura 17	Actividades de medición de distancias y servicio de peticiones	36
Figura 18	Distancia que separan los marcadores	37
Figura 19	Estructura del archivo de texto de salida	39
Figura 20	Actividades de medición de distancias	40
Figura 21	Los tipos de pieza fabricados	42
Figura 22	Esquema del montaje	43
Figura 23	Vista lateral del montaje	44
Figura 24	Vista superior del montaje	44
Figura 25	Reporte de obstáculo en la escena en archivo de texto de salida	48
Figura 26	Reporte de detección de distancia máxima admisible en archivo de texto de salida	49
Figura 27	Topología eléctrica para controlar iluminación	50
Figura 28	Esquema de montaje con iluminación controlada	50
Figura 29	Estacionamiento Estructural de la Ciudad Universitaria de Caracas	52
Figura 30	Edificios del I.P.P. y estacionamiento estructural de la Ciudad Universitaria de Caracas	52
Figura 31	Estacionamiento estructural, fotograma modificado	54

## LISTA DE TABLAS

Tabla 1	Especificaciones del computador BeagleBoard xM	20
Tabla 2	Mediciones luego de desplazamiento vertical	45
Tabla 3	Mediciones luego de desplazamiento horizontal	46
Tabla 4	Resultados para edificio estructural	53

## ACRÓNIMOS Y SIGLAS

APUCV:	Asociación de Profesores de la Universidad Central de Venezuela.
ASCII:	American Standard Code for Information Interchange, en español: Código Estándar Estadounidense para el Intercambio de Información
HSL:	Hue Saturation and Luminance, en español: Tinte Saturación y Luminancia
I.P.P.:	Instituto de Previsión del Profesorado
MDDR SDRAM:	Mobile Double Data Rate Synchronous Dynamic Random Access Memory, en español: Memoria dinámica de acceso aleatorio síncrona de doble velocidad de datos móvil
MDF:	Medium Density Fibreboard, en español: Tablero de Densidad Media
PC:	Personal Computer, en español: Computador Personal
PCB:	Polychlorinated Biphenyls, en español Policlorobifenilos
RGB:	Red Green Blue, en español: Rojo Verde Azul
RPyC:	Remote Python Call, en español: Llamada Remota en Python
UML:	Unified Modeling Language, en español Lenguaje Unificado de Modelado.
USB:	Universal Serial Bus, en español: Bus Universal en Serie.

## INTRODUCCIÓN

Cuando una afectación de la integridad física de una estructura ocurre, ésta puede acarrear daños y pérdidas humanas y materiales. Resalta la importancia de establecer una monitorización constante sobre el estatus de dicha estructura y de la gestión de la información producto de este proceso. Esta temática se aborda en la línea de investigación iniciada por la tesis doctoral Sistemas de Supervisión de Integridad Estructural en Edificaciones Esenciales en Zonas Metropolitanas [1], particularmente este trabajo es un avance en la tarea de la monitorización de la estructura.

Un factor de riesgo importante para la salud de una edificación lo constituye un evento sísmico, y la condición geográfica de nuestro país exige especial atención a la ocurrencia de estos eventos, demanda tomar medidas antes, durante y después de los mismos. Un proceso que contribuye a la toma de todas estas medidas es la monitorización constante de elementos de interés de una estructura puesto que la información que evidencia el estatus de su integridad física permite un análisis que daría pie a medidas correctivas necesarias.

Este trabajo presenta el desarrollo de una herramienta que permite la monitorización automática de las distancias que separan los distintos elementos de una estructura ya que los desplazamientos relativos de los elementos de una estructura son uno de los indicadores del estado de salud de la edificación. El enfoque para la medición de tales distancias es hacerlo mediante procesamiento de imágenes, esto es, con un software procesar fotografías de los elementos estructurales y determinar qué distancias los separan.

El presente trabajo escrito, dividido en cinco capítulos, es un registro de todo el proceso desde la problemática inicial hasta la solución propuesta y los resultados obtenidos con la misma luego de su implementación.

El problema que dispara todo el proceso se expone en el primer capítulo junto con las metas que debe alcanzarse para considerar satisfactoria la solución del problema.

En el segundo capítulo están los basamentos teóricos que permiten la formulación de una solución que puede resolver el problema que se aborda.

En el tercer capítulo se establecen los requerimientos que se tienen para poder implementar la solución ideada y así alcanzar las metas propuestas al inicio del proceso.

Luego, en el cuarto capítulo, se exponen todos los detalles de la solución implementada tanto en hardware como en software.

Finalmente, la validación de los métodos implementados y los resultados obtenidos están en el quinto y último capítulo.

# **CAPÍTULO I**

## **1 PROBLEMA**

### **1.1. PLANTEAMIENTO DEL PROBLEMA**

Uno de los indicadores de la integridad física de una estructura lo forman los desplazamientos relativos de los elementos de una estructura que separan a los distintos elementos que la conforman (columnas, vigas o pilas y tableros en el caso de un puente). Las distancias que separan los elementos, por diseño, tienen un rango normal de variación y es cuando una de éstas excede el rango normal que aumenta la vulnerabilidad de la estructura y de todo lo contenido en ella.

Cuando uno de los elementos estructurales excede y permanece en una posición fuera de su rango se habla de la ocurrencia de una deriva.

La posibilidad de que una estructura sufra una deriva de uno de sus elementos plantea la necesidad de poder detectar este hecho la más pronto posible luego de que ocurra. Hasta ahora las inspecciones a una estructura solamente son realizadas por personal calificado que no cuenta, además, con información previa sobre la estructura y el estatus de su integridad.

Para detectar una deriva apenas ocurra debe hacerse una monitorización constante sobre la estructura y reportar cualquier novedad en el menor tiempo posible, minimizando las probabilidades de riesgo para lo contenido en tal estructura. Debido a la naturaleza del problema la solución debe actuar en campo, ser capaz de detectar una deriva y reportar su ocurrencia con mínima participación humana.

### **1.2. PROPUESTA DE SOLUCIÓN**

El enfoque que se propone es la determinación de las distancias que separan los elementos estructurales mediante procesamiento de imágenes. Un

software deberá ser capaz de medir tales distancias y detectar la ocurrencia de una deriva a partir de una fotografía de la estructura. La fuente de la fotografía debe ser una cámara integrada a la plataforma donde se ejecuta el software.

El software deberá ser de funcionamiento continuo, monitorizando la estructura permanentemente, procesar tantas fotografías como sea posible y arrojar resultados en unidades de longitud.

Como esta solución formará parte de un sistema mayor que genera y gestiona información sobre el estatus de la estructura el software debe aceptar órdenes del exterior y reportarle los resultados obtenidos. Así como de la ocurrencia de una deriva.

### **1.3. OBJETIVOS**

#### **1.3.1. OBJETIVO GENERAL**

Detectar la ocurrencia de desplazamientos estructurales permanentes o derivas mediante el uso de una cámara digital de video y procesamiento de imágenes.

#### **1.3.2. OBJETIVOS ESPECÍFICOS**

- Desarrollar un dispositivo sobre plataforma de mínimo hardware, en particular cámara web de 320 x 240 píxeles de resolución y computador embebido Beagle xM para la adquisición de imágenes consecutivas y discretas en el tiempo con una tasa menor o igual a 24 cuadros por segundo además de imágenes bajo demanda del usuario.
- Resolver cambios de posición en dos dimensiones de objetivos presentes en la escena. Descartar cambios en la escena causados por obstáculos temporales y permanentes.



- Determinar la ocurrencia de un desplazamiento estructural permanente.
- Analizar las imágenes mediante un programa sin interfaz que se ejecute en segundo plano y produzca un archivo de texto como salida con los desplazamientos detectados fechados y listados.
- Caracterizar las limitaciones de detección a consecuencia de las condiciones de iluminación.
- Mantener un registro fotográfico fijo y actualizado de imágenes para su análisis fuera de línea.
- Proponer la metodología de validación y efectuar la calibración de las medidas de los desplazamientos.
- Asentar (sic) el funcionamiento del dispositivo, en hardware y software, y el mantenimiento necesario en un manual explicativo.

#### **1.4. ANTECEDENTES DE INVESTIGACIÓN**

El procesamiento de imágenes aplicado a mediciones de desplazamientos en estructuras ha sido ensayado por varias investigaciones entre las que pueden citarse la de Jong-Woong Parka y col. [2], en 2010, quienes obtuvieron una diferencia menor al 0,5% entre la medida real y la hecha mediante cámara del desplazamiento de una columna de acero como modelo de prueba.

Otro estudio, hecho por Yunfeng Ji [3] en 2010 también, presenta resultados con excelente precisión al hacer seguimiento de marcas colocadas sobre una estructura vibrante modelo.

Por otra parte, Jong-Jae Lee y col. [4], en 2011, lograron validar la efectividad de las mediciones de desplazamientos hechas por medio de una cámara digital al realizar ensayos sobre una estructura a escala real de 5 pisos.

Todas estas investigaciones previas dan luces sobre los resultados que pueden obtenerse al desarrollar un sistema que mida desplazamientos en

estructuras mediante una cámara y un posterior procesamiento de las imágenes y constituyen un buen norte para seguir y lograr integrar este proyecto al catálogo de investigaciones en ésta área, que está ganando terreno por sus numerosas ventajas.

## CAPÍTULO II

### 2 MARCO REFERENCIAL TEÓRICO

A continuación se presentan los conceptos y definiciones sobre los que se fundamenta la implementación de la solución al problema. Se inicia con los conceptos ligados a la imagen sus tipos, formas de representación entre otros atributos.

#### 2.1 Imagen.

Reproducción de la cantidad de luz que incide en un punto. Una función de dos variables,  $f(x, y)$ , puede representar una imagen; el valor de la función en un punto es el brillo o intensidad de luz en dicho punto [5].

##### 2.1.1 Imagen en escala de grises.

Reproducción de la cantidad de luz blanca que incide en un punto. Es suficiente con una función de dos variables,  $f(x, y)$ , para representarla [5].

##### 2.1.2 Imagen a color.

Imagen en la que la luz que incide en un punto es descrita por tres parámetros distintos para poder representar todo el espectro de luz visible para el humano. Se necesitan tres funciones de dos variables para representar una imagen a color [5].

#### 2.2 Imagen Digital.

Imagen representada por una función de dos variables,  $f(x, y)$ , en la que dominio y rango son discretos y finitos. El valor de la función en un punto representa el brillo o cantidad de luz en dicho punto [5].

#### 2.3 Matriz.

Arreglo bidimensional que representa a una imagen digital, contiene los valores de la función para todos los puntos donde está definida [5]. En la figura 1 se muestra un ejemplo de matriz, cada coeficiente  $b_{mn}$  representa la intensidad de luz en ese punto.

$$\begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{pmatrix}$$

Figura 1: Matriz o arreglo bidimensional

En el caso de una imagen digital a color es un arreglo tridimensional y contiene los valores de todos los parámetros que describen la luz en todos los puntos donde está definida la imagen. Los parámetros que describen la luz dependerán del sistema que se emplee para representar la imagen.

Ahora se examinan las definiciones que tratan el sistema de representación de imágenes usado en este trabajo.

## 2.4 Espacio de color.

Cuerpo tridimensional usado para representar un sistema de organización del color de acuerdo 3 coordenadas que describen el color. Algunos de los sistemas son: RGB que describe un color por la cantidad de rojo, azul y verde o el HSL que lo describe por matiz, saturación y brillo [6].

### 2.4.1 Espacio de color RGB.

Espacio de color cúbico en el que los colores están representados por 3 coordenadas que indican la cantidad de rojo, azul y verde de un color específico. Las imágenes a color, mediante este modelo, se representan por una matriz  $m \times n \times 3$ , donde  $m$  y  $n$  son las dimensiones de la imagen en píxeles y cada tupla de la forma  $((i, j, 0), (i, j, 1), (i, j, 2))$  describe completamente el color de un pixel de la imagen [6].

Seguidamente se exponen los conceptos relacionados con las operaciones que pueden aplicarse a la imagen con el fin de extraer información de ella.

## **2.5 Procesamiento de Imágenes.**

Algoritmos computacionales que se aplican a una imagen con el fin de obtener otra imagen o extraer información relevante que se encuentre en ella [7]. Los posibles procesos son numerosos: segmentación, cálculo de momentos, cambios de espacio de color, redimensionamiento. Particularmente son de interés los primeros dos.

### **2.5.1 Segmentación.**

Es la fragmentación de la imagen ya sea para aislar los distintos objetos que en ella aparezcan o retener áreas de la imagen que compartan una característica común como brillo o textura [7].

#### **2.5.1.1 “Thresholding”.**

Tipo de segmentación que se realiza estableciendo un valor umbral y todo pixel con valor de brillo por debajo se lleva al valor mínimo y todo pixel con valor de brillo por encima se lleva al valor máximo, el resultado es una imagen binaria (solamente están presentes píxeles negros y blancos) [7]. Este proceso solo puede aplicarse a imágenes en escala de grises.

El valor umbral es establecido de forma externa y a discreción de quien dirige el proceso.

### **2.5.2 Momento.**

Cantidad escalar que caracteriza un atributo de una función. El momento general  $M$  de una función  $f(x, y)$  definida en un dominio  $D$  queda definido como sigue:

$$M_{pq}^{(f)} = \iint_D p_{pq}(x, y) f(x, y) dx dy \quad (1)$$

En la expresión 1  $p_{pq}$  es una función polinómica definida en  $D$  y los subíndices  $p$  y  $q$  son enteros no negativos; el orden del momento  $M$  es  $r = p + q$  [8].

En el caso de una imagen digital, representada por  $g(x, y)$ , el momento general  $m_{pq}$  queda definido así:

$$m_{pq} = \sum_{x=0}^{\infty} \sum_{y=0}^{\infty} p_{pq} g(x, y) \quad (2)$$

La ecuación 2 muestra la definición de momento general aplicado a una función discreta. En adelante se usarán las definiciones para el caso de imágenes digitales.

#### 2.5.2.1 Momento Geométrico.

$$p_{pq} = x^p y^q \quad (3)$$

$$m_{pq} = \sum_{x=0}^{\infty} \sum_{y=0}^{\infty} x^p y^q g(x, y) \quad (4)$$

Cuando  $p_{pq}$  es una función que tiene la forma de la ecuación 3 el momento general pasa a ser un momento geométrico y queda definido como la ecuación 4 muestra [8].

El momento geométrico de orden cero,  $m_{00}$ , representa el área del objeto presente en una imagen binaria.

#### 2.5.2.2 Momento Central.

Al referir un momento geométrico al centroide  $(\bar{x}, \bar{y})$  de la función se obtiene el momento central que es invariante ante la traslación de la misma [8].

$$\mu_{pq} = \sum_{x=0}^{\infty} \sum_{y=0}^{\infty} (x - \bar{x})^p (y - \bar{y})^q g(x, y) \quad (5)$$

El momento central  $\mu_{pq}$  definido según la ecuación 5.

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad (6)$$

$$\bar{y} = \frac{m_{01}}{m_{00}} \quad (7)$$

Las ecuaciones 6 y 7 muestran el cálculo de las coordenadas del centroide de la función.

### 2.5.2.3 Momento central normalizado.

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}} \quad (8)$$

$$\gamma = \frac{p+q}{2} + 1 \quad (9)$$

Al normalizar el momento central de una función mediante una potencia del área de la función ( $\mu_{00}$ ) (ecuación 8) se logra la estabilidad de los valores ante cambios de escala [8]. El momento normalizado de una función  $\eta_{pq}$  viene dado por las expresión 9.

### 2.5.2.4 Momentos Hu.

$$\varphi_1 = \eta_{20} + \eta_{02} \quad (10)$$

$$\varphi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (11)$$

$$\varphi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (12)$$

$$\varphi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (13)$$

$$\begin{aligned} \varphi_5 = & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 \\ & - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (14)$$

$$\begin{aligned} \varphi_6 = & (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ & + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \end{aligned} \quad (15)$$

$$\begin{aligned} \varphi_7 = & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \quad (16) \\ & - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 \\ & - (\eta_{21} + \eta_{03})^2] \end{aligned}$$

Basándose en los momentos centrales normalizados el investigador Ming-Kuei Hu [9] propone un conjunto de 7 momentos ( $\varphi_i$ ) que son invariantes ante la traslación, escalamiento y rotación de una función y por lo tanto funcionan como descriptores de la misma. Los 7 momentos propuestos por Hu son los expuestos en las ecuaciones 10 a 16.

## 2.6 OpenCV.

Librería de código abierto dirigida a la visión computarizada. Se enfoca en el procesamiento de imágenes a alta velocidad para poder implementar aplicaciones de tiempo real. Escrita en C/C++ tiene interfaces para python y java y soporta los sistemas operativos Windows, Mac OS, Linux, Android y iOS [10].

Ahora, unas definiciones básicas sobre estructuras y los desplazamientos en los que pueden incurrir sus elementos.

## 2.7 Estructura.

Elementos contruidos, y por tanto interconectados, de una forma organizada para proveerles de rigidez [11].

### 2.7.1 Deflexión.

Desplazamiento de una estructura o uno de sus elementos de su posición estática o de su dirección o plano debido a la aplicación de una fuerza [11].

#### 2.7.1.1 Deriva.

Deflexión horizontal de una estructura o uno de sus elementos debido a la actuación de una fuerza [11].



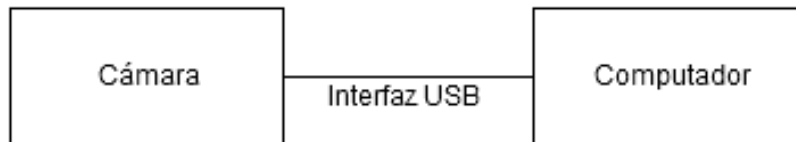
## CAPÍTULO III

### 3 DEFINICIÓN DEL SISTEMA

#### 3.1 Componentes Físicos del Sistema.

Los componentes físicos necesarios para la implementación del dispositivo propuesto son: una cámara y un computador. Debido a la naturaleza del proyecto, se requiere que el conjunto cámara-computador sea un dispositivo de campo por lo que ambos deben ser del tamaño más reducido posible para facilitar la instalación y la operación que debe ser continua en el tiempo.

En la figura 2 se muestra un esquema:



**Figura 2:** Conjunto Cámara – Computador.

#### 3.1.1 Computador.

Uno de las condiciones que determinan las características que deben tener los componentes físicos es que es una aplicación de campo por lo que debe procurarse facilitar su instalación y minimizar los riesgos a los que se expone el computador debe ser lo más reducido posible en tamaño, así mismo debe ofrecer por lo menos 1 puerto USB host y 1 puerto RJ45 para poder integrarse a una red de área local

#### 3.1.2 Cámara.

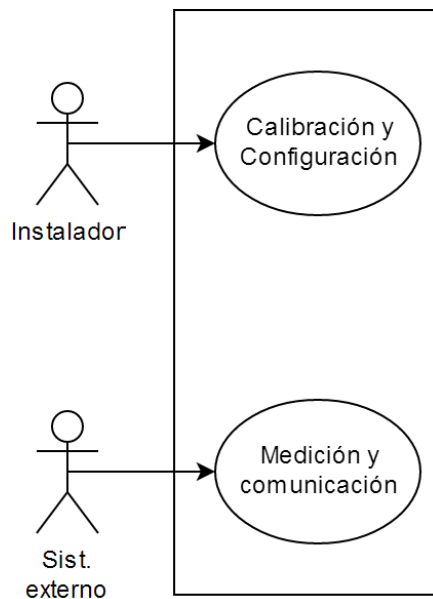
Para alcanzar los objetivos del trabajo, la cámara debe proveer imágenes de por lo menos 320x240 píxeles de resolución. Debe admitir suministro de energía a través de una conexión USB.

### 3.2 Software.

El propósito principal del software es monitorizar la distancia que separa dos puntos físicos a través del procesamiento de una imagen donde se visualicen ambos puntos, para esto el programa debe ser capaz de reconocer dichos puntos en la imagen y lograr convertir una cantidad de píxeles en unidades de distancia.

El software se compone de dos programas separados, uno para el proceso de configuración y calibración y otro para el proceso de medición y comunicación con un sistema externo. Dicho sistema externo es el propuesto en la tesis doctoral Sistema de supervisión de integridad estructural en edificaciones esenciales de zonas metropolitanas de Ing. Rafael Rivero, M. Sc. [1].

En la figura 3 se muestran los casos de uso del software de este proyecto:



**Figura 3:** Casos de uso del software propuesto.

Todo el software usado en este proyecto debe ser de código abierto, de esta forma se evita cualquier compromiso económico además de que el programa propuesto queda completamente a disposición de futuros investigadores que lo podrán tomar como base para proponer mejoras y expandir su funcionalidad.

### **3.2.1 Módulo de Configuración y Calibración.**

El programa de calibración y configuración tiene por finalidad establecer parámetros necesarios para el funcionamiento del programa de medición.

El producto de la calibración debe ser el factor de conversión pixel/longitud que se usa posteriormente para arrojar resultados en unidades de longitud.

Luego del proceso de configuración debe quedar fija la siguiente información: cuáles distancias medir, qué número  $n$  de mediciones recientes deben ser devueltas bajo petición de un sistema externo, el mismo número aplica para los formatos de texto plano y diccionario, cuál es el aumento máximo admisible de la derivas y luego de cuántas detecciones continuas de ese valor debe reportarse la ocurrencia de una deriva, la cadena de caracteres que se usará para nombrar las fotografías tomadas bajo demanda y finalmente, el tamaño del registro fotográfico que será almacenado en el computador.

Toda la información debe quedar registrada en texto plano, código ASCII, y guardada en memoria del computador.

### **3.2.2 Módulo de Medición.**

Este programa debe encargarse de la adquisición y procesamiento de las imágenes, realizar mediciones de distancia en dos dimensiones y arrojar un resultado en unidades de longitud.

Las mediciones deben realizarse lo más rápidamente posible, estableciendo en 24 medidas por segundo como la tasa deseable. Se toma esa tasa porque es la cantidad de cuadros por segundo para una señal de video por lo que al alcanzarla se trataría de un proceso bastante cercano a uno en tiempo real.

Todas las medidas realizadas deben quedar listadas y fechadas en texto plano, código ASCII (reporte de medidas). También deben estar disponibles en un formato que pueda ser usado por paquetes matemáticos externos. Todo debe estar guardado en memoria del computador.

Es importante que el programa reconozca la ocurrencia de una deriva en la estructura, esto es, un desplazamiento permanente de alguno de los elementos estructurales y lo reporte.

Para permitir una inspección visual rápida de la estructura a monitorizar el programa debe salvar y mantener las últimas imágenes tomadas en la memoria del computador. El número máximo de fotografías a almacenar es de 20, ya que la memoria del PC también debe almacenar el registro en texto plano de todas las mediciones.

Como es un dispositivo de campo, la visual de la cámara hacia la estructura puede verse momentánea o permanentemente obstruida, el programa debe reconocer esta situación y hacerlo saber en el reporte de medidas.

Este dispositivo está previsto que pueda integrarse a un sistema más grande de gestión de la información de riesgo de una estructura por lo que es importante que acepte y sirva las peticiones de dicho sistema. En particular las peticiones que deben ser servidas: consulta de las últimas mediciones, tanto en texto plano como en formato de diccionario y la toma de una foto en un instante determinado.

Es requerimiento del sistema externo que para la comunicación debe emplearse la librería RPyC para el lenguaje Python.

Es importante que las tareas de medición y comunicación se realicen de forma paralela: se debe medir las distancias que hayan sido configuradas de forma constante y atender cualquiera de las peticiones que reciba desde el sistema externo.

## CAPÍTULO IV

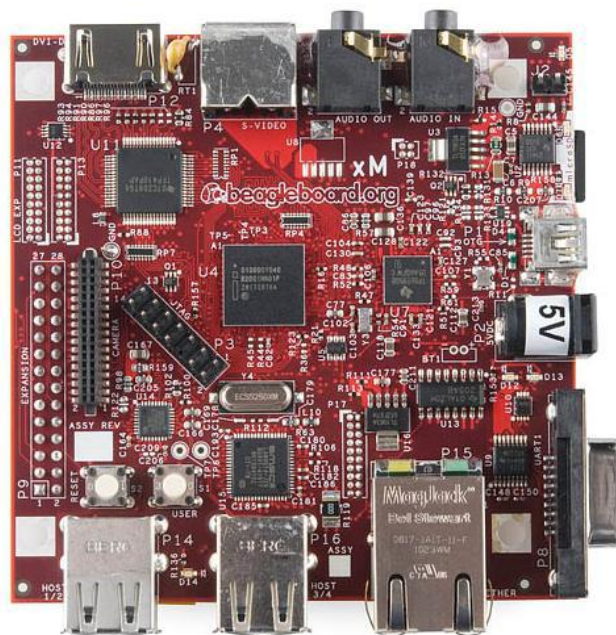
### 4 DESCRIPCIÓN DEL SISTEMA

#### 4.1 Componentes Físicos del Sistema.

Los componentes físicos del sistema son: un computador para la ejecución del software y una cámara para la adquisición de las imágenes.

##### 4.1.1 Computador.

El computador es una plataforma de tarjeta única BeagleBoard xM revisión C1. El sistema operativo es Ubuntu 14.04. Las prestaciones de este dispositivo y el hecho de que uno estuviese disponible de inmediato para empezar el proyecto prevalecieron en la elección del mismo. La figura 4 muestra el computador:



**Figura 4:** Computador BeagleBoard xM

Algunas de las prestaciones y características de este computador se exponen en la tabla 1.

**Tabla 1:** Especificaciones del computador BeagleBoard xM.

Alimentación	5 V DC
Procesador	Texas Instruments Cortex A8 DM3730CBP 1GHz
Memoria	Micron 4Gb MDDR SDRAM (512MB) 200MHz

Puertos USB	4 puertos
Interfaz de red	10/100 Mbps
Dimensiones PCB	3.1" x 3.0" (78.74 x 76.2mm)

#### 4.1.2 Cámara.

La cámara es tipo web marca Genius modelo Widecam 1050, provee una imagen con resolución de hasta 720p. Alimentación vía conexión USB. La existencia de este equipo y su disponibilidad inmediata para la ejecución del proyecto prevalecieron en la elección de ésta cámara. En la figura 5 se muestra la cámara usada:



**Figura 5:** Cámara Genius WideCam 1050.

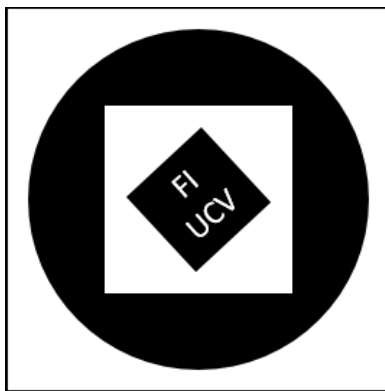
#### 4.2 Software.

Para la programación se usa el lenguaje Python, es de código abierto, permite la programación orientada a objetos y es multiplataforma. Además, el sistema externo exige que la comunicación con él se programe usando la librería RPyC para Python.

Para el procesamiento de las imágenes se usa la librería OpenCV, que ha sido desarrollada en C/C++ con interface para Python y es compatible con

sistemas operativos linux. Esta librería está orientada a la visión computarizada y es de código abierto.

Como el trabajo consiste en la medición de distancias entre puntos específicos mediante procesamiento de imágenes, es necesario que el software sea capaz de reconocer dichos puntos por lo que se usan marcadores. Un marcador es una imagen patrón que debe situarse sobre un elemento estructural de la edificación; para delimitar una distancia a medir se usa un par de marcadores. En la figura 6 se muestra un marcador.



**Figura 6:** Marcador usado para demarcar distancias a medir.

La elección de los colores, blanco y negro, obedece a la intención de maximizar el contraste entre el marcador y la estructura además de facilitar la elección del valor umbral que se usa en el proceso de segmentación de la imagen.

La forma a detectar, el círculo negro, es poco frecuente sobre todo en estructuras y edificaciones donde abundan las líneas rectas y ángulos por lo que se elige ésta como marcador ideal. Para identificar el marcador a simple vista se coloca el texto “FI UCV”.

El reconocimiento se realiza mediante el procesamiento de la imagen. La imagen a color proveniente de la cámara se transforma a una imagen en escala de grises y luego se aplica una segmentación a 2 niveles, obteniéndose una imagen con pixeles únicamente negros y blancos. El valor umbral para este

proceso de segmentación es establecido empíricamente. Luego son extraídas las fronteras o bordes entre pixeles negros y blancos y son analizados matemáticamente calculando sus momentos Hu. Los momentos Hu de todas las fronteras son calculados y también son filtradas por su tamaño, los momentos de las fronteras que alcancen el tamaño mínimo establecido por software son comparados con los momentos Hu del marcador precargado en la memoria del computador como archivo de imagen (.jpg).

$$m_i^A = \text{sign}(h_i^A) \log(h_i^A) \quad (17)$$

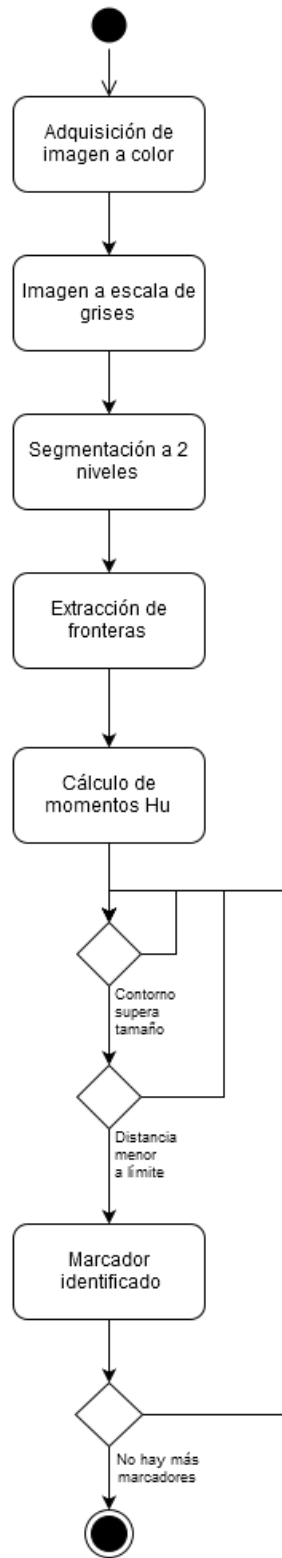
$$I_{(A,B)} = \max \frac{|m_i^A - m_i^B|}{|m_i^A|} \quad (18)$$

Para comparar una frontera A con otra frontera B y determinar si son la misma, se calculan los 7 momentos Hu ( $h_i$ ) de cada una y mediante las expresiones 17 y 18 se determinan si son la misma. Si el resultado de la expresión 18 ( $I_{(A,B)}$ ) es menor que un valor establecido empíricamente mediante ensayo y error, se concluye que se trata efectivamente de la misma frontera, en este caso particular de un marcador. Este proceso se realiza con todas las fronteras que hayan sido encontradas luego de la segmentación de la imagen.

Un aspecto importante sobre la ejecución del software es que debe hacerse una vez que el dispositivo haya sido ubicado en el lugar previsto para su operación. Los marcadores deben estar sobre los elementos estructurales de la edificación a monitorizar, todos en el mismo plano y éste debe ser paralelo al lente de la cámara (visual totalmente frontal).

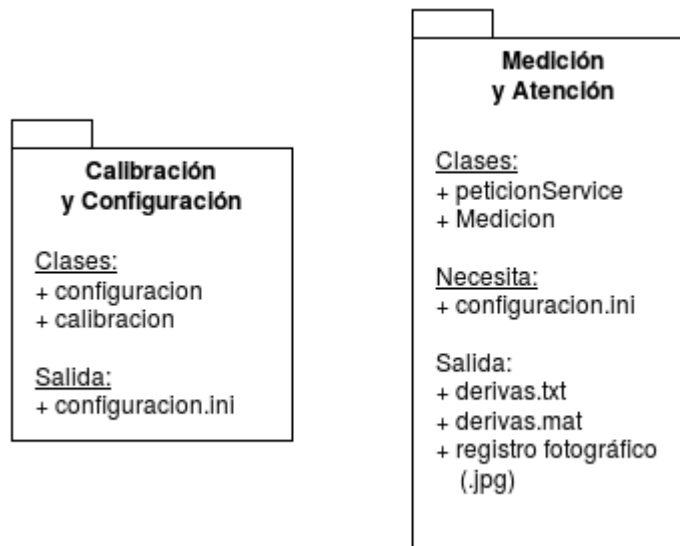
El diagrama UML de actividades que ilustra los pasos que sigue el proceso de reconocimiento de los marcadores se muestra en la figura 7.





**Figura 7:** Actividades para el reconocimiento de marcadores en la imagen.

En la figura 8 se muestra un esquema de los componentes del software.



**Figura 8:** Módulos del Software.

#### 4.2.1 Módulo de Configuración y Calibración.

Con este programa se fijan todos los parámetros necesarios para la ejecución del programa de medición. El usuario puede elegir cuál de los dos procesos realizar primero entre calibración y configuración, es indiferente el orden pero es esencial que ambos sean realizados.

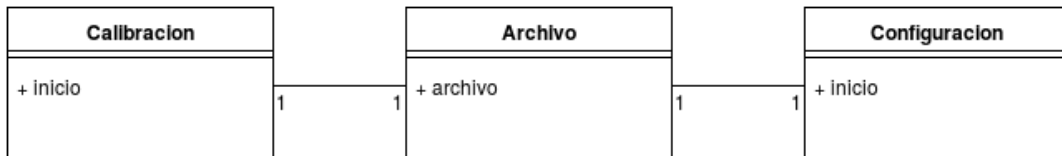
Debido a lo anteriormente mencionado, el programa chequea que ambos procesos han sido realizados, de lo contrario así lo advierte al usuario.

Este programa requiere que el usuario ingrese datos por lo que también debe existir una comunicación con él. La interfaz de este programa es una ventana de terminal.

El producto de estos procesos es un archivo de nombre configuracion.ini que contiene, en texto plano, todos los parámetros y que serán leídos una única vez y durante el arranque del programa de medición.

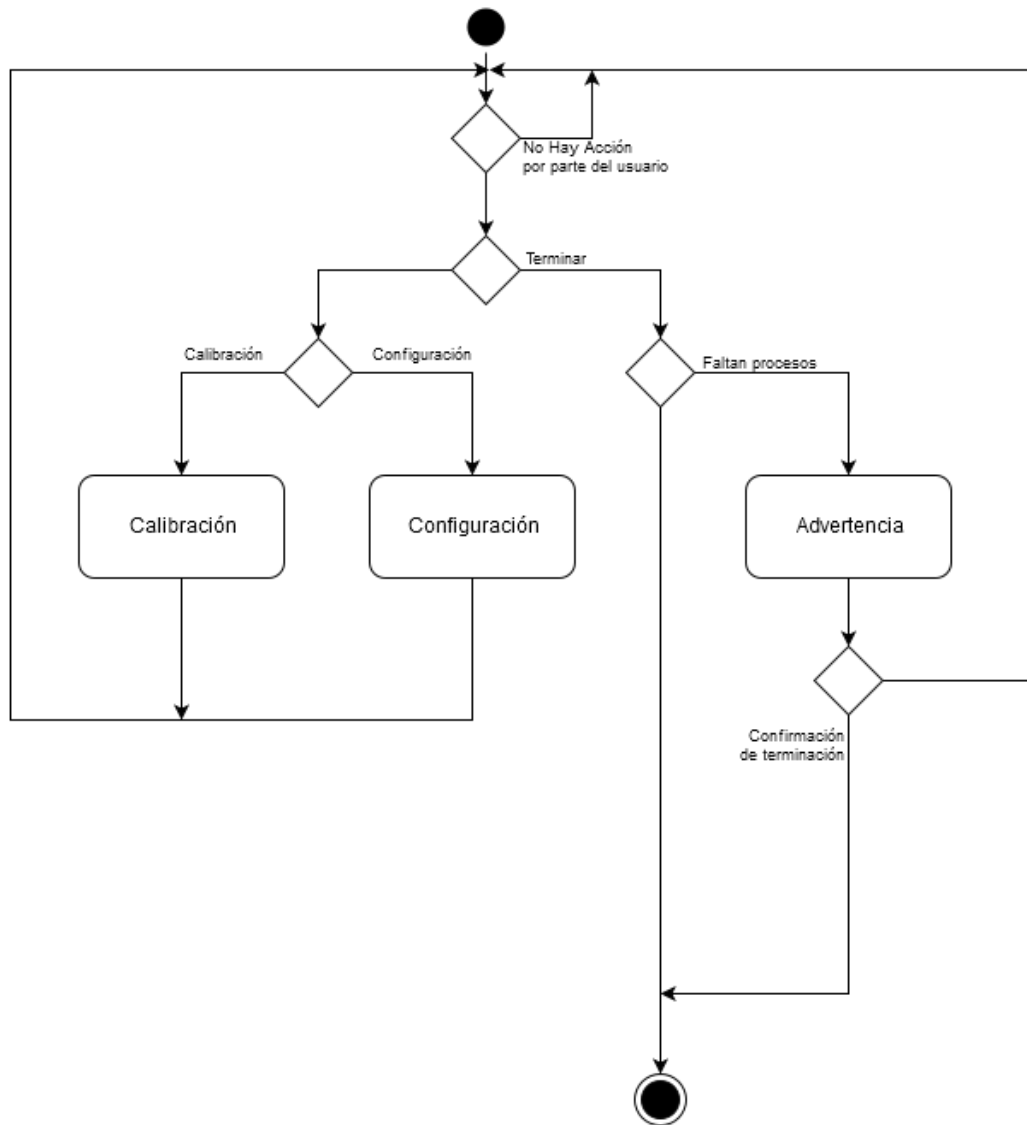
Para cada uno de los procesos se define e instancia una clase. La generación del archivo de texto que es común a ambos procesos amerita la

definición de una clase independiente, en la figura 9 se muestra cómo se relacionan dichas clases.



**Figura 9:** Clases del programa de configuración y calibración.

El diagrama general de actividades del programa de calibración y configuración se muestra en la figura 10, cada uno de los 2 procesos se explica más a fondo en secciones posteriores.



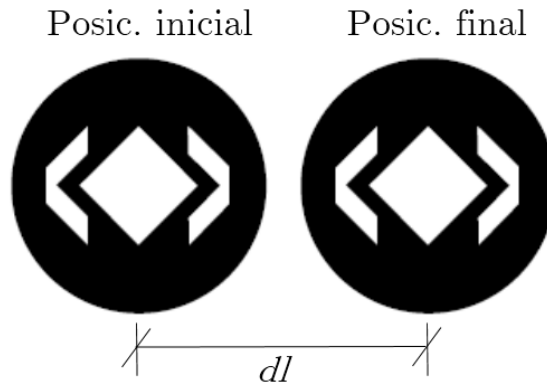
**Figura 10:** Actividades para la calibración y configuración.

#### 4.2.1.1 Componente de Calibración.

Durante esta fase únicamente se calcula el factor de conversión pixel-longitud ( $f$ ) y puede hacerse de dos formas: mediante desplazamientos controlados y mediante el tamaño del marcador, que es conocido.

El primer método consiste en reconocer un marcador en la imagen (posición inicial) y seguidamente volver a reconocerlo (posición final) luego de

que haya sido desplazado en una dimensión una medida de longitud conocida ( $dl$ ). La figura 11 muestra un esquema de las posiciones del marcador.



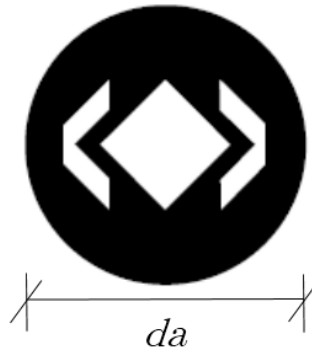
**Figura 11:** Desplazamiento del marcador.

El programa determina cuántos píxeles separan la primera posición de la segunda ( $dp$ ) y se pide al usuario que ingrese cuanto fue ese desplazamiento en centímetros ( $dl$ ).

$$f = \frac{dl}{dp} \quad (19)$$

El factor de conversión se calcula según la ecuación 19.

El segundo método consiste en la detección de un marcador en la imagen, se calcula cuántos píxeles representan el diámetro del marcador ( $da$ ) y se pide al usuario que ingrese cuál es el diámetro en centímetros ( $db$ ). La figura 12 muestra un esquema de la distancia  $da$ .



**Figura 12:** Diámetro de marcador.

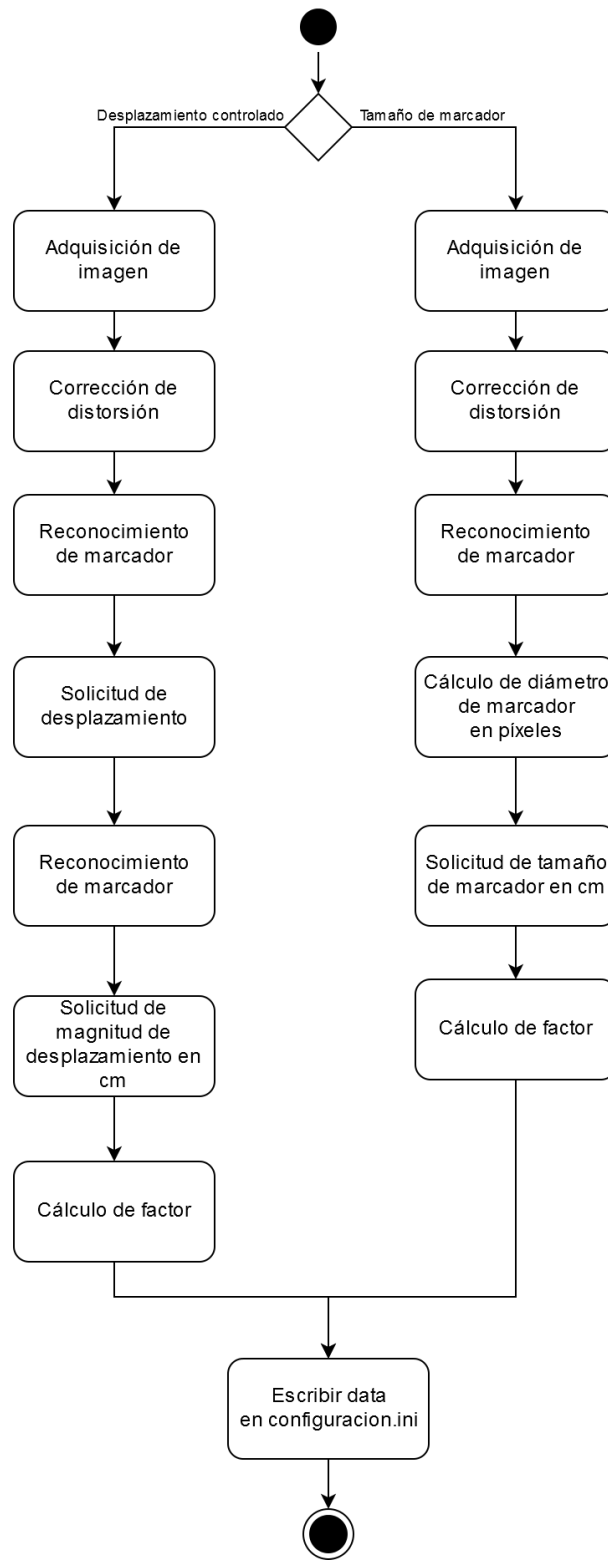
$$f = \frac{db}{da} \quad (20)$$

Finalmente, el factor de conversión se calcula con la ecuación 20.

En el caso del proceso de calibración se reconoce un solo marcador en la imagen, de todos los que puedan estar presentes porque se asume que todos están en un mismo plano y que la visual de la cámara a los marcadores es totalmente frontal (de no cumplirse éstas condiciones el factor calculado será erróneo y por lo tanto las mediciones que se realicen). De esta forma el factor  $f$  calculado con un solo marcador funciona para todas las regiones de la imagen.

El factor  $f$  obtenido se salva en el archivo configuracion.ini y queda en la memoria del computador.

La figura 13 muestra el diagrama UML de actividades del proceso de calibración.



**Figura 13:** Actividades de calibración.

#### 4.2.1.2 Componente de Configuración.

El resto de parámetros necesarios para el funcionamiento del programa de medición y que se fijan con el programa de configuración son: qué distancias se van a medir, la cantidad de fotos a tener almacenadas, el número de mediciones a devolver cuando se reciba una consulta externa, el aumento máximo admisible de las derivas y luego de cuantas detecciones de dicho valor debe reportarse la ocurrencia de deriva y finalmente la cadena de caracteres que se usará para el guardado de fotografías tomadas bajo demanda externa. También es información útil para el programa de medición saber cuántos marcadores existe en la imagen y la región donde se encuentran.

Las distancias a medir se delimitan por pares de marcadores y deben ser ingresadas por el usuario en forma de pares ordenados  $(A, B)$  una vez que hayan sido reconocidos en la imagen por el programa, numerados y presentados en pantalla.

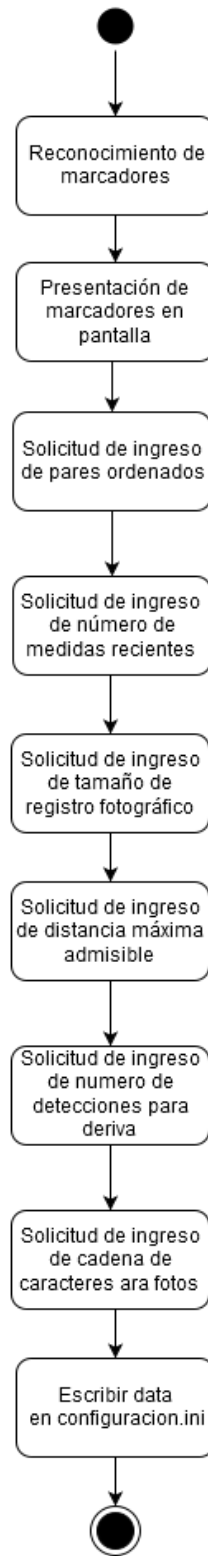
Los demás parámetros son números reales con excepción del nombre que se usa para las fotografías tomadas bajo demanda que es una cadena de caracteres. Todos estos parámetros deben ser ingresados por el usuario cuando se le requiera.

Una vez completo el proceso de configuración, toda la información ingresada por el usuario es escrita en el archivo `configuracion.ini` y salvada en la memoria del computador. Es el mismo archivo que se usa para el proceso de calibración y donde se escribe o escribirá el factor de conversión (calibración y configuración pueden hacerse en cualquier orden).

La información sobre cuántos marcadores y su posicionamiento en la imagen es guardada automáticamente de forma transparente para el usuario.

La figura 14 muestra el diagrama UML de actividades del proceso de configuración.





**Figura 14:** Actividades de configuración.

La figura 15 muestra un archivo ejemplo producto de haber realizado ambos procesos: calibración y configuración:

```
centroMasa_0 = 311.590909091, 167.364583333, 270
centroMasa_1 = 351.694470774, 165.303633491, 286
region_0 = 258, 115, 108, 106, 0
region_1 = 298, 113, 108, 106, 1
indiceRegiones = 2
par_0 = 0, 1
indicePares = 1
numeroMediciones = 10
bancoImagenes = 11
distanciaDeriva = 20
deteccionesDeriva = 20
fotoUsuario = fot
referencia_0 = 40.1035616832, 2.06094984202
segmento = 170
umbralSimil = 0.05
configHecho = True
contribdeltafactor = 1.27551020408e-05
factor = 1.07692307692
deltafactor = 0.00384615384615
calibHecho = True
```

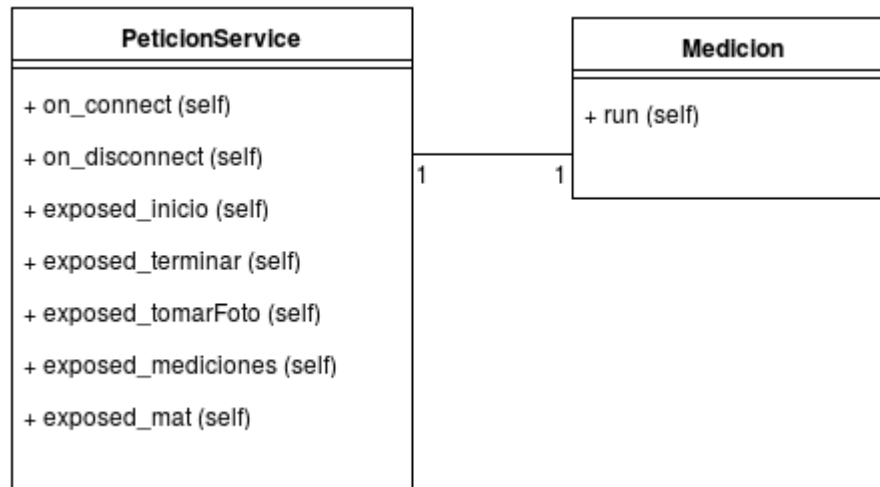
Figura 15 Archivo de configuración.

#### 4.2.2 Módulo de Medición y Comunicación.

Este módulo realiza dos procesos simultáneos: el de medición de distancias propiamente y el de servicio a las peticiones externas. Para alcanzar la simultaneidad de ambos procesos se recurre a la arquitectura de programación de hilos. El procesador del computador atiende por un número de instrucciones uno de los procesos y pasa al otro, sigue alternándose indefinidamente.

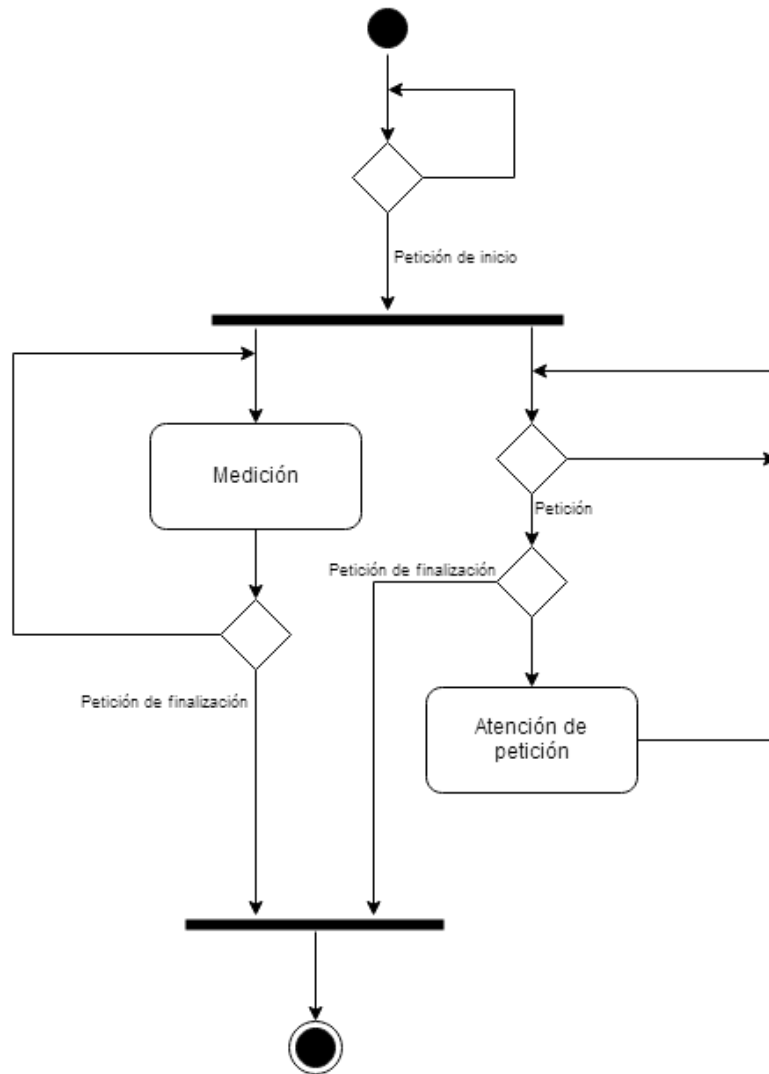
El proceso de servicio a las peticiones externas es el primero que se inicia, y está constantemente a la espera de una petición externa, de hecho el ciclo de mediciones no inicia hasta que se recibe esa orden y se detiene de la misma forma.

Para cada uno de los procesos se define una clase que se relacionan según se muestra en el diagrama de la figura 16.



**Figura 16:** Clases del módulo de medición y servicio de peticiones.

La figura 17 muestra el diagrama UML de actividades general.



**Figura 17:** Actividades de medición de distancias y servicio de peticiones.

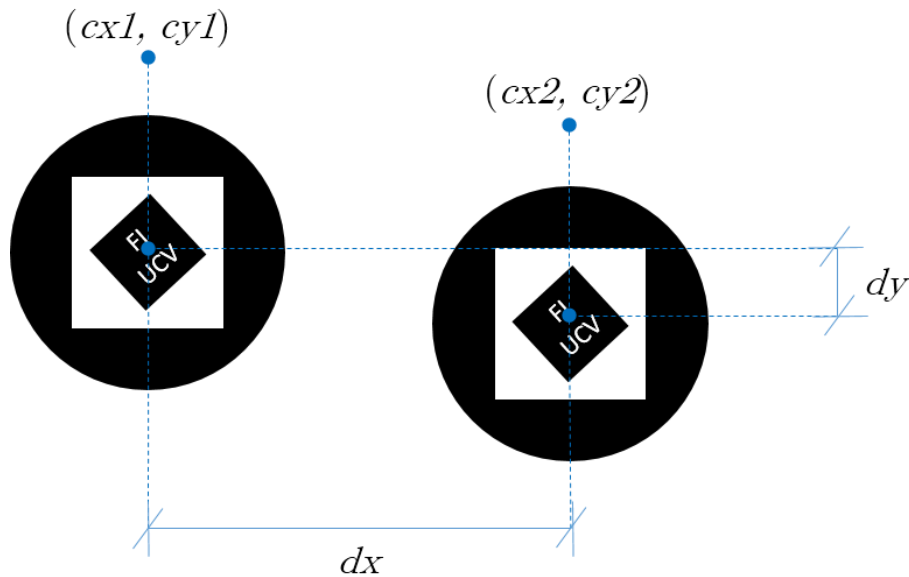
#### 4.2.2.1 Componente de Medición de Distancias.

Este programa funciona de forma continua midiendo las distancias que fueron especificadas en la etapa de configuración. Es un programa que no tiene interfaz y que se ejecuta en segundo plano. También sirve a las peticiones externas de consulta de mediciones recientes y toma de fotografías en un momento cualquiera.

Este programa inicia su ejecución y se detiene cuando recibe las peticiones para eso desde el exterior.

El programa reconoce en la imagen todos los marcadores y determina la posición del centro de masa de cada uno a través de los momentos  $H_u$ , al tratarse de un marcador circular su centro de masa es igual al centro geométrico. Las coordenadas del centro de masa de un marcador  $(c_x, c_y)$  están referidas al origen de coordenadas de la imagen, la esquina superior izquierda.

Habiendo obtenido las posiciones de los centros de masa de los marcadores, se calculan las distancias que los separan tanto en el eje horizontal como vertical ( $d_x$  y  $d_y$ ) según lo indicado por el usuario en la etapa de configuración. La figura 18 muestra un esquema de las distancias a medir.



**Figura 18:** Distancia que separan los marcadores.

$$d_x = c_{x1} - c_{x2} \quad (21)$$

$$d_y = c_{y1} - c_{y2} \quad (22)$$

Las distancias que separan a dos marcadores, 1 y 2, se calculan con las ecuaciones 21 y 22.

De no haber detección de alguno o ninguno de los marcadores se asume que hay un obstáculo en la visual de la cámara y se escribe el valor -999 en las distancias como señal de que no hubo lectura válida.

Luego de obtenidas las distancias el programa revisa si alguno de los valores sobrepasa la deriva máxima admisible y de ser así, lo reporta en los resultados.

Todos los resultados obtenidos se escriben en texto plano, código ASCII, en un archivo de texto de extensión .txt junto con la fecha en que se obtuvieron. También se escriben en formato de diccionario en un archivo de extensión .mat que sirve para su procesamiento en paquetes matemáticos externos y es el archivo que se manda a quien haga la petición de las medidas más recientes. En la figura 19 se observa la estructura del archivo de texto de salida y toda la información que contiene.

```
Derivas detectadas
Foto 1: 19-10-2016 16:51:15

Distancia (0,3)
X:32.400000
Y:5.300000

Distancia (1,3)
X:23.300000
Y:0.500000

Distancia (2,3)
X:14.000000
Y:4.900000

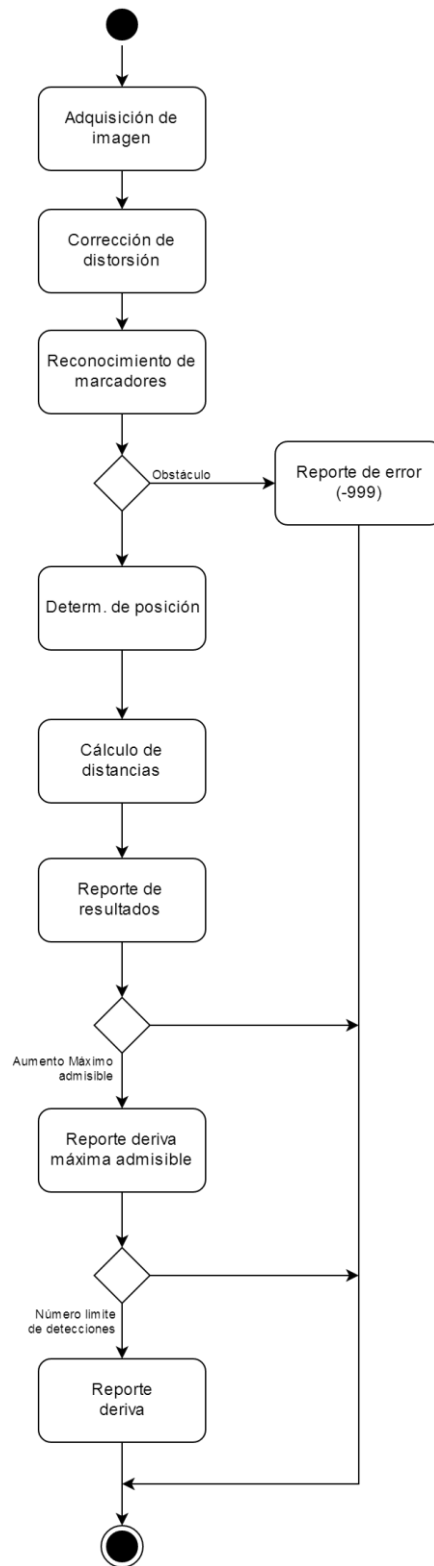
Distancia (4,3)
X:4.900000
Y:4.500000
```

**Figura 19:** Estructura del archivo de texto de salida.

Las fotografías sobre las que se reconocen los marcadores también son salvadas en memoria del computador en formato .jpg para efectos del registro fotográfico que debe existir. El tamaño de este registro es fijo y especificado por el usuario. Las fotografías se nombran con la siguiente cadena de caracteres

'fotox.jpg' donde x es un número entre 1 y tamaño del registro fotográfico establecido en el proceso de configuración. El valor de x inicia en 1 y va aumentando progresivamente hasta el máximo, momento en el que vuelve al valor 1 y la foto "foto1.jpg" sustituye a la anterior en el computador.

La figura 20 muestra las actividades que se ejecutan durante el proceso de medición.



**Figura 20:** Actividades de medición de distancias.



#### **4.2.2.2 Componente de Servicio de peticiones.**

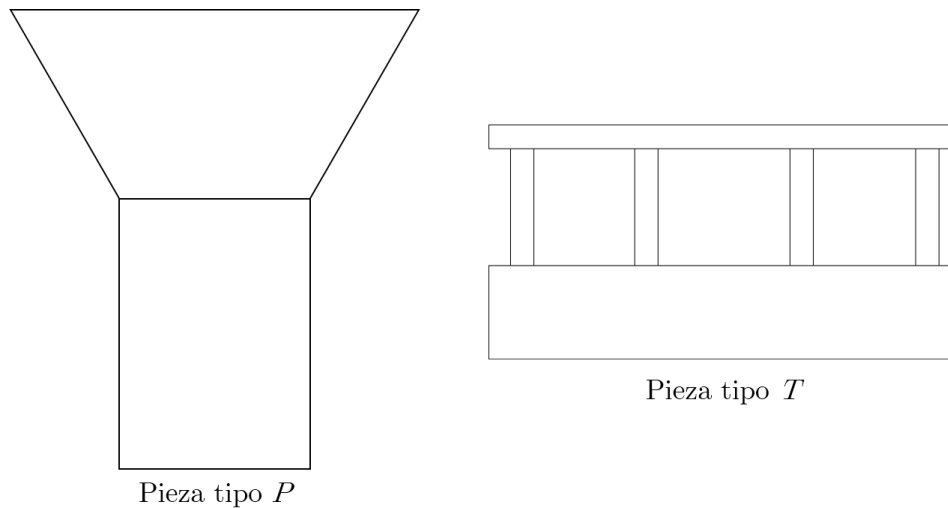
Este componente del programa está simplemente a la espera de recibir alguna petición desde el exterior. Las peticiones que sirve son: inicio y detención del ciclo de medición, consulta de las mediciones más recientes tanto en texto plano como en formato de diccionario y tomar una fotografía.

Para la programación de este componente se usó la librería RPyC para Python que permite la comunicación entre dos equipos a través de una red de área local y es requisito del sistema exterior al que este proyecto debe integrarse.

## CAPÍTULO V

### 5 PRUEBAS DE VALIDACIÓN. RESULTADOS

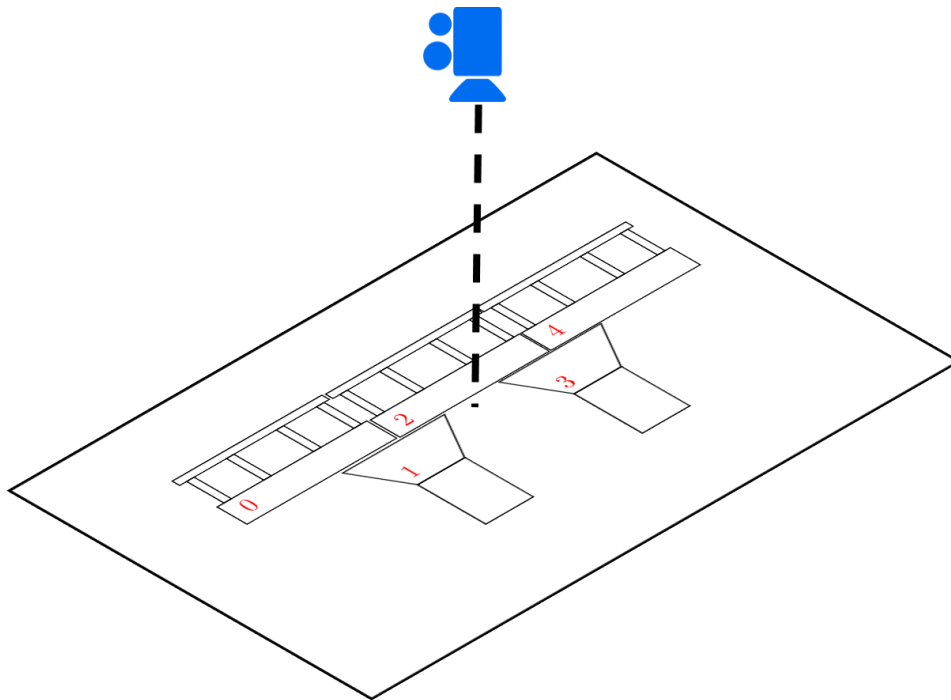
Para verificar el correcto funcionamiento del software se realizó un montaje con 2 tipos de piezas planas,  $P$  y  $T$ , que modelan un puente; las piezas fueron fabricadas en MDF de  $(1/8)$ " de espesor y se colocan sobre ellas los marcadores, tal como estarían en una estructura real. En la figura 21 se muestran ambos tipos de figura.



**Figura 21:** Los tipos de pieza fabricados.

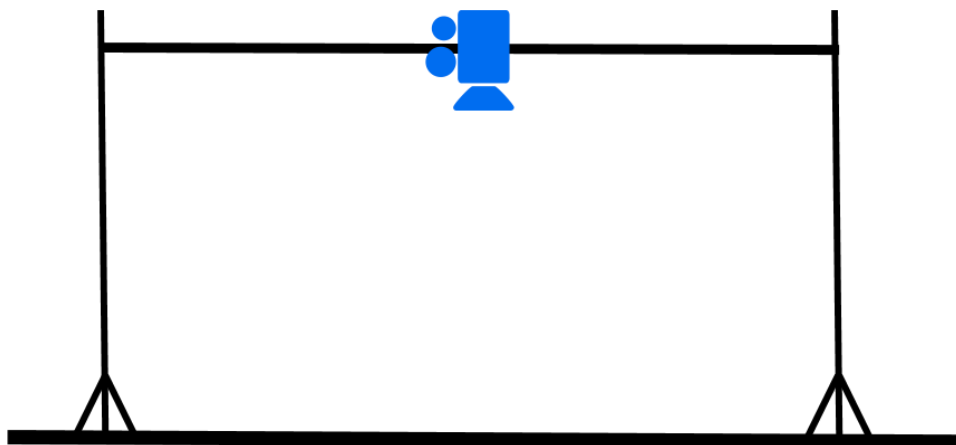
Las piezas se disponen en un plano horizontal para facilitar su movimiento en dos dimensiones además de garantizar que todos los marcadores estén en el mismo plano. La cámara se ubicó convenientemente de manera que tenga una visual totalmente frontal al puente modelado por las piezas, a una altura de 50 cm.

La figura 22 muestra un esquema general del montaje para la validación del software. La numeración de las piezas que se muestra es la misma que hace el software de calibración y configuración y se usará la misma para todos los efectos.

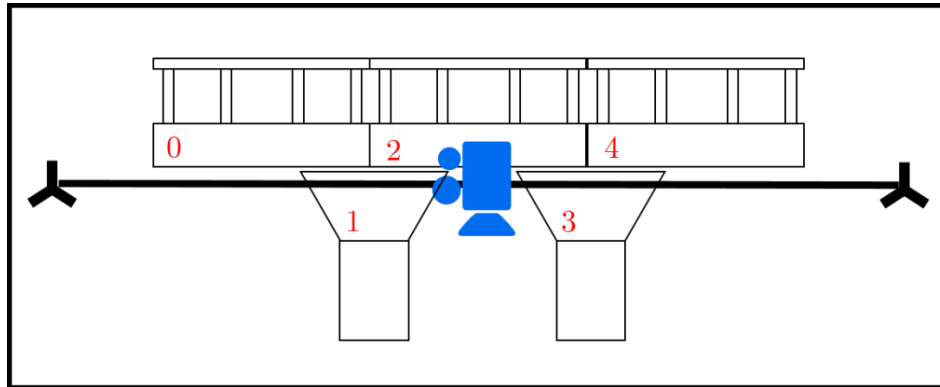


**Figura 22:** Esquema del montaje.

Para la ubicación de la cámara se emplearon un par de soportes verticales y uno más horizontal. Las vistas lateral y superior del montaje se muestran en las figuras 23 y 24 respectivamente.



**Figura 23:** Vista lateral del montaje



**Figura 24:** Vista superior del montaje.

También se contempla la realización de pruebas con imágenes adquiridas en condiciones reales con la finalidad de verificar el funcionamiento del programa bajo estas circunstancias.

### 5.1 Medición de Distancias.

Las distancias bajo estudio serán cuatro: (0,3), (1,3), (2,3) y (4,3) según la numeración de las piezas mostrada en la figura 22. Los valores iniciales de estas distancias serán medidos y tenidos como referencia.

La pieza número 3 será movida 5 cm en horizontal a partir de su posición de origen, y luego 5 cm en vertical a partir de su posición de origen. Se ejecutará el programa de medición y se obtendrán 30 medidas de cada distancia luego de cada uno de los movimientos. También serán medidas estas distancias con un sistema externo y se tendrán estos valores como referencia.

#### 5.1.1 Resultados de la prueba de medición de distancias.

Luego del desplazamiento de 5 cm en horizontal de la pieza 3 se obtuvieron los siguientes resultados (el renglón “determinación por software” refleja el promedio de las 30 medidas tomadas):

**Tabla 2:** Mediciones luego de desplazamiento horizontal.

Distancia	Medida	Determinación	Desviación	Desviación
-----------	--------	---------------	------------	------------

(A,B)	Referencia (cm)	por software (cm)	relativa	estándar
(0,3)	$35 \pm 0,1$	$35,1 \pm 0,9$	0,4 %	0,04
(1,3)	$25 \pm 0,1$	$25,1 \pm 0,7$	0,4 %	0
(2,3)	$15 \pm 0,1$	$15 \pm 0,4$	0 %	0,03
(4,3)	$5 \pm 0,1$	$5,3 \pm 0,2$	6 %	0

Luego del desplazamiento de 5 cm en vertical de la pieza 3 se obtuvieron los siguientes resultados (el renglón “determinación por software” refleja el promedio de las 30 medidas tomadas):

**Tabla 3:** Mediciones luego de desplazamiento vertical.

Distancia (A,B)	Medida Referencia (cm)	Determinación por software (cm)	Desviación relativa	Desviación estándar
(0,3)	$10 \pm 0,1$	$10,2 \pm 0,3$	2,1 %	0,03
(1,3)	$5 \pm 0,1$	$5,1 \pm 0,2$	2,3 %	0,04
(2,3)	$10 \pm 0,1$	$10,2 \pm 0,3$	1,9 %	0,03
(4,3)	$10 \pm 0,1$	$10,1 \pm 0,3$	1 %	0

Los resultados muestran que existe una desviación consistente de hasta 0,3 cm en las mediciones realizadas, esto confiere mayor confiabilidad a las mediciones de distancias mayores a 15 cm ya que porcentualmente la desviación es poco significativa. Las desviaciones en las medidas son atribuibles a la distorsión que introduce el mecanismo óptico de la cámara, aun cuando la librería OpenCV permite corregir esta distorsión esto se hace a partir de un modelo, los parámetros reales de la cámara no están a disposición de la investigación.

$$\delta(ms) = \sqrt{\left(\frac{\delta(f)}{f}\right)^2 + \left(\frac{\delta(p)}{p}\right)^2} * ms \quad (23)$$

La propagación del error hacia la determinación por software (*ms*) se calculó según la ecuación 23, donde *f* es el factor de conversión de píxeles a unidad de longitud  $\delta(f)$  es su respectivo error, *p* es la cantidad de píxeles captados por la cámara para el contorno que se mide y  $\delta(p)$  su error.

## **5.2 Detección de obstáculos.**

Partiendo del montaje inicial, se calibra y configura el programa para medir las distancias (2,4) y (2,3). Se realizarán 30 medidas de esas distancias con el programa de medición para verificar que efectivamente está obteniendo valores numéricos, luego la pieza 4 será obstruida de la visual de la cámara para que no pueda ser detectado por el software en la imagen adquirida, a partir del quinto intento de detección fallida el programa debe indicar la presencia de obstáculo permanente (este criterio se establece discrecionalmente como un parámetro en el código del programa) mientras que la distancia (2,3) deberá seguir siendo medida con normalidad. No es de interés en esta prueba la exactitud de la medida.

### **5.2.1 Resultados de la prueba de detección de obstáculos.**

En la figura 25 se exponen un extracto del archivo de texto donde se evidencia la detección del obstáculo y luego el reporte de obstáculo permanente.

```
N U E V O   C I C L O
Foto 1: 10-02-2017 12:18:48

Distancia (2,3)
10.000000
0.319800
5.100000
0.237180

Distancia (2,4)
X: --Obstaculo
Y: --Obstaculo

Foto 2: 10-02-2017 12:18:49

Recurrencia
Posible obstaculo permanente
Distancia (2,3)
10.000000
0.319700
5.100000
0.237020

Distancia (2,4)|
X: --Obstaculo
Y: --Obstaculo
```

**Figura 25:** Reporte de obstáculo en la escena en archivo de texto de salida.

El programa fue capaz de reconocer la ausencia de uno de los marcadores y luego, al cumplirse el criterio para considerar una ausencia permanente lo reporta correctamente en el archivo de texto. La otra distancia que se pide siguió siendo medida con normalidad.

### **5.3 Detección de deriva permanente.**

Partiendo del montaje inicial se calibra y se configura el software para medir la distancia (3,4) con un aumento máximo admisible de la deriva de 3 cm y que el número de detecciones continuas para reportar deriva sea de 5 cm. Luego a partir de su posición de origen, la pieza 4 se moverá 2 cm en vertical y se tomarán 30 medidas con el programa de determinación de distancias. Posteriormente se moverá la misma pieza 3 cm en vertical a partir de la posición anterior y se toman el mismo número de mediciones.

#### **5.3.1 Resultados de las pruebas de detección de derivas permanentes.**

La figura 26 muestra un extracto del archivo de texto de salida.

```
Foto 7: 10-02-2017 12:26:32

Distancia (3,4)
10.400000
0.328560
Distancia Maxima admisible superada
10.100000
0.322100

Foto 8: 10-02-2017 12:26:33

Distancia (3,4)
10.400000
0.328860
Distancia Maxima admisible superada
10.100000
0.322150

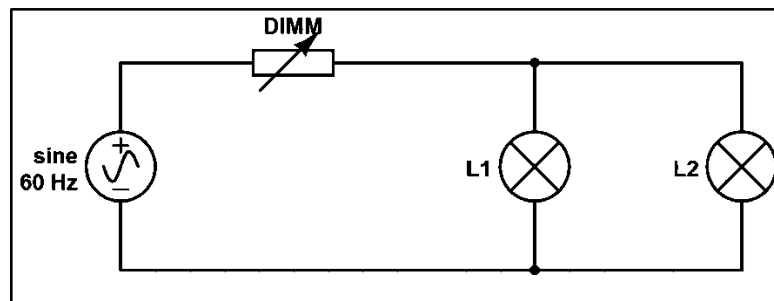
Detectado 5 veces
```

**Figura 26:** Reporte de detección de distancia máxima admisible en archivo de texto de salida.

Cuando el desplazamiento de la pieza 4 superó los 3 cm (deriva máxima admisible) el programa reconoció el evento y una vez alcanzado el criterio para considerarla permanente lo reportó en el archivo de texto correctamente.

#### 5.4 Limitaciones por condiciones de iluminación.

Además del montaje inicial se emplea una topología eléctrica para controlar la cantidad de luz que percibe la cámara. Para esta prueba se garantiza que la superficie sobre la que se encuentran las piezas es de acabado mate. Tanto el montaje de las piezas como la topología eléctrica se ubican en una habitación sin otras fuentes externas de iluminación. La topología eléctrica se muestra a continuación, en la figura 27.



**Figura 27:** Topología eléctrica para controlar iluminación.



En la figura 28 se muestra el esquema de la integración del montaje inicial con la iluminación controlada.

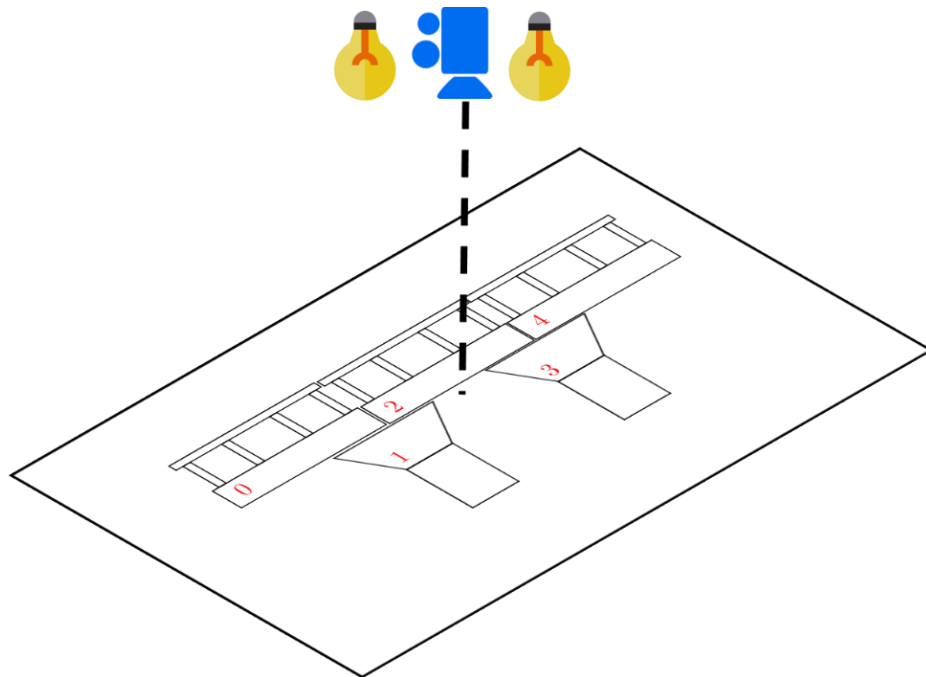


Figura 28: Esquema de montaje con iluminación controlada

Empezando desde una condición de oscuridad total y ubicando los bombillos a la misma altura de la cámara, se va incrementando la intensidad de luz emitida por los bombillos progresivamente hasta que el programa reconozca los 5 marcadores presentes en la escena, punto en el que la intensidad de luz que percibe la cámara se mide con un teléfono marca BLU modelo Life 8XL que ejecuta la aplicación “Light Lux Meter” que mide intensidad de luz. Luego sigue aumentándose la intensidad de luz emitida para encontrar un punto en el que la cámara deje de reconocer los marcadores por iluminación excesiva.

#### **5.4.1 Resultados de las pruebas de limitaciones por condiciones de iluminación.**

Para la cámara empleada y las condiciones a las que se sometió, el valor mínimo de iluminación al que el software reconoce los 5 marcadores en la imagen es de 8 lx. El software no dejó de reconocer los marcadores para valores

superiores hasta los 92 lx (el máximo valor medido para la topología y condiciones empleadas).

### **5.5 Pruebas en condiciones reales.**

En la edificación que constituye el estacionamiento estructural de la Ciudad Universitaria de Caracas se ubicará un par de marcadores de 28 cm de diámetro y se tomará un video de 10 s de duración, aproximadamente, en tres momentos del día: 1:00 pm, 3:00 pm y 5:00 pm. Para otro set de pruebas se ubicara un marcador en el estacionamiento estructural y otro en la fachada de la entrada principal del edificio del I.P.P. de la APUCV y se tomarán videos en las mismas condiciones anteriores.

El interés de esta prueba es verificar el reconocimiento de los marcadores en condiciones reales: iluminación natural en distintos momentos del día, marcadores con un tamaño factible para la escala de una edificación real. La exactitud de de las medidas obtenidas no es de interés en esta prueba.

Las ubicaciones reales con los marcadores se muestran en las figuras 29 y 30.



Figura 29 Estacionamiento Estructural de la Ciudad Universitaria de Caracas.



Figura 30 Edificios del I.P.P. y estacionamiento estructural de la Ciudad Universitaria de Caracas.

De cada video se extraerá un fotograma y será modificado para simular el movimiento de una parte de la estructura y ambos fotogramas, el original y el modificado, serán procesados.

### 5.5.1 Resultados de pruebas en condiciones reales.

Para el par de marcadores ubicado en el estacionamiento estructural se obtuvieron los siguientes resultados:

Tabla 4. Resultados para edificio estructural.

	Fotograma original (cm)	Fotograma modificado (cm)
1:00 pm	x: $43,2 \pm 1,0$ y: $2,2 \pm 1,0$	x: $70,1 \pm 1,1$ y: $2,2 \pm 1,0$
3:00 pm	x: $46,5 \pm 1,1$ y: $3,1 \pm 1,1$	x: $56,7 \pm 1,1$ y: $3,9 \pm 1,1$
5:00 pm	x: $48,6 \pm 1,1$ y: $0,1 \pm 1,1$	x: $54,8 \pm 1,1$ y: $6,1 \pm 1,1$

La figura 31 muestra el fotograma modificado tomado a la 1:00 pm:



Figura 31 Estacionamiento estructural, fotograma modificado.

En el caso de la prueba I.P.P.-estacionamiento estructural el programa no detectó satisfactoriamente ambos marcadores para ninguno de los fotogramas, únicamente fue reconocido el marcador ubicado en la fachada del I.P.P.

Los resultados obtenidos en estas pruebas en condiciones reales confirman las condiciones y limitaciones del conjunto cámara-computador para producir resultados satisfactorios: todos los marcadores deben estar ubicados en el mismo plano, esto reduce las posibles aplicaciones mayormente a medir distancias entre elementos de la misma estructura. En las pruebas realizadas en el I.P.P. el reconocimiento de los marcadores pudo haberse logrado si los mismos hubiesen sido de mayores dimensiones.

## CONCLUSIONES

El conjunto cámara-computador y un software que es capaz de determinar las distancias que separan 2 puntos en una imagen fue satisfactoriamente implementado y se obtuvieron los resultados anteriormente expuestos, la desviación porcentual de las medidas obtenidas es menor al 1% para distancias mayores a 15 cm, para una cámara y marcadores separados por 50 cm.

La programación orientada a objetos es un enfoque que permitió la claridad en el código del programa, pudiendo efectuar y seguir los cambios hechos con facilidad. Por otro lado, el uso de la arquitectura de hilos es una forma de aprovechar al máximo los recursos de la plataforma usada, BeagleBoard xM, la misma permite atender varios procesos al mismo tiempo. El proceso de medición de las distancias del programa propuesto se ejecuta en un hilo mientras que otro atiende las peticiones externas como la consulta de las  $n$  mediciones más recientes.

La librería OpenCV es una herramienta que permitió agilizar el trabajo práctico de programación del procesamiento de imagen. Luego de establecer teóricamente qué transformaciones e información extraer de la imagen y el algoritmo para hacerlo, se usaron instrucciones implementadas en OpenCV que cubrían las necesidades planteadas.

## RECOMENDACIONES

- Usar una mejor cámara en cuanto a la geometría de sus lentes y proceso de fabricación para reducir aún más la distorsión en la imagen, esto conduciría a resultados más exactos y confiables para distancias menores a 15 cm.
- Profundizar en el conocimiento de la librería OpenCV para incorporar otras funciones como el reconocimiento de caracteres con lo que puede aumentar la información que produce el software sobre la estructura que se monitoriza.
- Usar un sistema de alimentación ininterrumpida para lograr cierto grado de independencia de la red eléctrica.
- Buscar asesoramiento externo que oriente sobre toda la información que pueda extraerse de una imagen sobre el estatus de una edificación, extraer esa información y complementar la que ya se produce.
- Realizar pruebas sobre edificaciones reales.
- Incorporar iluminación infrarroja al conjunto cámara-computador para lograr la monitorización de las distancias de interés aún durante las horas de la noche.
- Aumentar la robustez de la medida de la distancia promediando los resultados de varias imágenes para producir un solo resultado.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] M.Sc. Ing. Rafael E. Rivero. Proyecto de tesis doctoral “Sistema de supervisión de integridad estructural en edificaciones esenciales de zonas metropolitanas”. Caracas. 2012.
- [2] Jong-Woong Parka, Jong-Jae Leeb, Hyung-Jo Junga and Hyun Myunga. *Vision-based displacement measurement method for high-rise building structures using partitioning approach*. 2010.
- [3] Yunfeng Ji. *A computer vision-based approach for structural displacement measurement*. CA, USA. 2010.
- [4] Hyung-Jo Jung, Jong-Jae Lee and Jeong-Su Park. *A Vision-based Displacement Measurement Method for Structural Health Monitoring of Large-scale infrastructures*. 2011.
- [5] González, R., Woods, R. & Eddins, S. *Image Processing Using Matlab*. 2007.
- [6] Levowitz, H. *Color Theory and Modeling for Computer Graphics, Visualization, and Multimedia Applications*. 1997
- [7] Young, I., Gerbrands, J. & Van Vliet, L. *Fundamentals of Image Processing Version 2.3*. 2007
- [8] Flusser, J., Suk, T. & Zitová, B. *Moments and Moment Invariants in Pattern Recognition*. 2009.
- [9] Hu, Ming-Kuei. *Visual Pattern Recognition by Moment Invariants*. Agosto, 1962.
- [11] Itseez. *OpenCV 2.4.10 Documentation*. 2015
- [12] Harris, Cyril. *Dictionary of Architecture & Construction*. McGraw-Hill, 2006.



## BIBLIOGRAFIA

Aleck Associates Ltd [en línea] <<http://www.aleckassociates.co.uk/glossary-of-structural-engineering-terms/>> [Consulta: 2016]

BRADSKI, G. & KAEHLER, A. (2008) Learning OpenCV. O'REILLY.

CHAUMETTE, F. (2004). Image Moments: A General and Useful Set of Features for Visual Servoing.

CHOPRA, A. (1995). Dynamics of Structures. Theory and Applications of Earthquakes Engineering. PRENTICE HALL.

FERRÉ, X. & SÁNCHEZ M. (s.f) Desarrollo Orientado a Objetos con UML.

FLUSSER, J., SUK, T. & ZITOVÁ, B. (2009). Moments and Moment Invariants in Pattern Recognition. JOHN WILEY & SONS.

Fundación BeagleBoard.org (2010). BeagleBoard-xM Rev. C System Reference Manual Rev. 1.0. Obtenido de: <http://beagleboard.org/>

Fundación Python Software (2015) Python 2.7 Documentation Obtenido de: <https://docs.python.org/2/download.html>

GONZÁLEZ, R., WOODS, R. & EDDINS, S. (2007) Image Processing Using Matlab. PRENTICE HALL.

GONZÁLEZ, R. (s.f) Python para Todos.

HARRIS, C. (2006). Dictionary of Architecture & Construction. McGRAW-HILL.

HARRIS, C. & STEPHENS M. (1998). A Combined Corner and Edge Detector.

HANCOCK, J. [en línea] <<http://www.hancockjoist.com/glossary.htm>> [Consulta: 2016]

HU, M. (1962). Visual Pattern Recognition by Moment Invariants.

HUANG, Z. & LENG, J. (2010). Analysis of Hu's Moment Invariants on Image Scaling and Rotation.

Isotope Eleven [en línea] <<https://isotope11.com/blog/storing-surf-sift-orb-keypoints-using-opencv-in-python>> [Consulta 2015]

- Itseez [en línea] <<http://docs.opencv.org/2.4.10/>> [Consulta: 2015]
- KILLIAN, J. (2001). Simple Image Analysis by Moments Version 0.2.
- LEVI, G. [en línea] <<https://gilscvblog.com/2015/01/02/adding-rotation-invariance-to-the-brief-descriptor/>> [Consulta: 2015]
- LEVOWITZ, H. (1997). Color Theory and Modeling for Computer Graphics, Visualization, and Multimedia Applications. KLUWER ACADEMIC PUBLISHERS.
- PRESSMAN, R. (2002). Ingeniería del Software, Un Enfoque Práctico. MCGRAW-HILL.
- RAHMAN, A. [en línea] <<http://opencvpython.blogspot.com/2012/06/hi-this-article-is-tutorial-which-try.html>> [Consulta: 2015]
- RIVERO, R. (2012). Proyecto de Tesis Doctoral, SISTEMA DE SUPERVISIÓN DE INTEGRIDAD ESTRUCTURAL EN EDIFICACIONES ESENCIALES DE ZONAS METROPOLITANAS. Caracas.
- ROSTEN, E., PORTER, R. & DRUMMOND, T. (s.f.). Faster and better: a machine learning approach to corner detection.
- ROSTEN, E. & DRUMMOND, T. (s.f.). Machine learning for high-speed corner detection.
- RUBLEE, E., RABAUD, V., KONOLIGE, K. & BRADSKI, G. (s.f.) ORB: an efficient alternative to SIFT or SURF.
- SCHMULLER, J. (s.f.). UML en 24 Horas. PRENTICE HALL.
- SOILLE, P. (2003). Morphological Image Analysis. SPRINGER.
- SOLOMON, C. & BRECKON, T. (2011). Fundamentals of Digital Image Processing. WILEY-BLACKWELL.
- Structural Engineers Cambridge Ltd [en línea] <<http://www.structuralengineerscambridge.co.uk/helpful-information/ctec-glossary-of-terms>> [Consulta: 2106]
- TKALČIČ, M. & TASIČ, J. (2001) Colour Spaces – perceptual, historical and applicational background.

Universidad de Harvard (2007). A Summary of Error Propagation. Obtenido de <http://ipl.physics.harvard.edu>

YOUNG, I., GERBRANDS, J. & VAN VLIET, L. (2007). Fundamentals of Image Processing  
Version 2.3

