

MEDICIÓN DE PÉRDIDAS DE PRESIÓN USANDO VISION POR COMPUTADOR

MEASUREMENT OF PRESSURE LOSSES USING COMPUTER VISION

Jorge Andrés Cerquera-Yacumal
Ingeniero en Automática Industrial
Universidad del Cauca
Popayán, Colombia
jcerquera@unicauca.edu.co

John Edwin Córdoba-Papamija
Ingeniero en Automática Industrial
Universidad del Cauca
Popayán, Colombia
jcordoba@unicauca.edu.co

Judy Cristina Realpe-Chamorro
M. Sc. Electrónica y Telecomunicaciones
Universidad del Cauca
Popayán, Colombia
jcrealpe@unicauca.edu.co

Juan Fernando Flórez-Marulanda
M. Sc. Electrónica y Telecomunicaciones
Universidad del Cauca
Popayán, Colombia
jflórez@unicauca.edu.co

Resumen- En un circuito hidráulico, el fluido sufre pérdidas de presión al atravesar tramos o diferentes accesorios, estas pueden ser inspeccionadas observando el nivel del fluido en piezómetros. Este trabajo presenta un desarrollo en visión artificial, que permite la medición en línea de los niveles de presión registrados en dieciséis columnas piezométricas. La técnica de extracción de fondo basada en el algoritmo de *Codebook*, combinada con el filtrado morfológico y unión de blobs, es usada para obtener los niveles piezométricos. El algoritmo de procesamiento está diseñado en *Open CV* y soportado en *Linux*. Se obtuvieron coeficientes de variación inferiores al 0,1% y errores máximos del 1,4 %.

Palabras clave- *Codebook*, extracción de fondo, instrumentación y medición, piezómetro, visión por computador.

Abstract- In a hydraulic circuit, the fluid pressure loss when passing through different sections or accessories, and it can be inspected by observing the fluid level in piezometers. This paper presents a development in artificial vision, which allows on-line measurement of pressure levels recorded in sixteen piezometric columns. The technique of background subtraction based in algorithm on *Codebook*, combined with morphological filtering and junction blobs, are used to obtain piezometrics levels. The processing algorithm is designed in *Open CV* and supported on *Linux*. Coefficients of variation of less than 0.1% and maximum error of 1.4% were obtained.

Keywords- *Codebook*, computer vision, background subtraction, instrumentation and measurement, Piezometer.

1. INTRODUCCIÓN

La medición de una misma variable en diferentes puntos o tramos de una tubería representa un desafío en términos de costo, instrumentación e instalación. Se han realizado investigaciones que buscan reemplazar instrumentos físicos por otros que posean características especiales, brindando igual o mayor información y confiabilidad en las variables medidas [1], [2]. Los sensores o instrumentos virtuales son resultado de algunas de estas investigaciones, emulan el comportamiento de instrumentos reales y su operación se basa en la relación que existe entre las variables medibles y métodos para estimación de otras [3], [4]. En la actualidad los sensores virtuales son empleados en una amplia variedad de campos: aviación [5], robótica [6], control de tráfico [7], entre otros [8], [9], [10]. Tal es el caso [11], que se usan sensores virtuales en técnicas de visión artificial para la lectura de transmisores indicadores, y [12], [13] se presentan sensores *software*, basados en téc-

nicas de visión artificial, que permiten en tiempo real estimar la interfaz entre el petróleo crudo y la arena.

En el monitoreo de pérdida de presión en circuitos hidráulicos, más precisamente en los bancos de pérdidas de carga en conductos a presión, usados en los laboratorios de hidráulica, se precisa de numerosos piezómetros de lectura visual de la presión, 16 o más, observándose el nivel de agua alcanzado, instalados en segmentos o aditamentos a lo largo de la tubería y concentrados en un tablero. La lectura de pérdida de presión en los mismos es compleja, se hace midiendo la diferencia de nivel por parejas, para varios caudales y en condiciones de estado estable, esto consume valioso tiempo del operador, por lo que se podría optar por métodos alternativos de medición que brinden mejor desempeño y automatismo del proceso, uno de estos es técnicas de visión artificial [14].

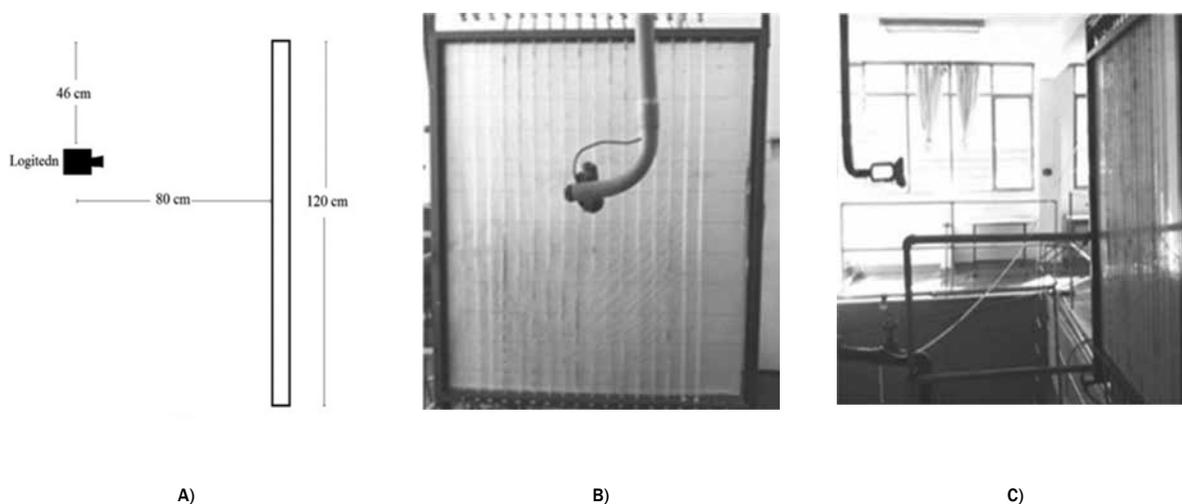
En este artículo se describe el desarrollo de un algoritmo de visión artificial, que permite la detección y medición simultánea de dieciséis variables de presión, indicadas por niveles de un fluido en un tablero piezométrico, el cual se integró como núcleo de un multi-transmisor virtual validado en un banco de pérdidas de carga en conductos a presión [14]. El algoritmo se basa en la técnica de extracción de fondo de *Codebook* combinada

con el filtrado morfológico y unión de blobs [15]. En la sección de experimentación se presenta el desarrollo del algoritmo basado en la técnica extracción de fondo; en la sección resultados se analizan las pruebas de desempeño del algoritmo y, por último, se consignan las conclusiones del trabajo.

2. EXPERIMENTACIÓN Y MATERIALES

El banco de pérdidas de carga en conductos a presión es un circuito hidráulico con tuberías flexible, instalado entre segmentos de diferente diámetro de tubería y aditamentos como codos, uniones, entre otros. Para medir simultáneamente los dieciséis piezómetros se propuso la instalación de una cámara webcam Logitech (Webcam Pro 9000, USB 2.0), ubicada al frente del tablero, Fig. 1.a, Fig. 1.b y Fig. 1.c; el procesamiento se hace con visión artificial usando la librería de procesamiento óptico digital de imágenes OpenCV, por medio de un algoritmo programado en lenguaje C [16], [17], [18], [19]. Este se diseñó concebido como un multitransmisor virtual y se ejecutó en un computador de escritorio (Pentium IV, 2.6 GHz, 433 MB, kernel Linux 2.6.34.5 Ubuntu 10.10 con RTAI). El algoritmo multitransmisor virtual de alturas piezométricas usa una metodología organizada en las fases: captura, pre-procesamiento, segmentación y reconocimiento.

Fig. 1. A) DIMENSIONES PIEZÓMETRO Y UBICACIÓN DE CÁMARA, EN B) IMAGEN FRONTAL Y EN C) IMAGEN LATERAL

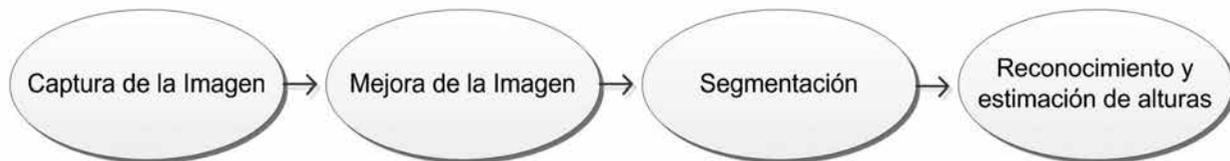


2.1 Descripción general del algoritmo

El algoritmo de procesamiento de imágenes consta de cuatro componentes asociados a cada fase de la metodología usada. *Captura de la imagen*, realiza la adquisición de la secuencia de imágenes del tablero piezométrico. *Mejora de*

la imagen, se usan técnicas de filtrado, transformación de color, modificaciones del histograma, entre otras. *Segmentación*, elimina el fondo de la escena y realiza un modelo de este. Finalmente, un componente de *Reconocimiento y estimación de alturas*, que realiza la medición de las dieciséis variables respecto a una escala métrica, Fig. 2.

Fig. 2. DIAGRAMA EN BLOQUES DEL ALGORITMO DE PROCESAMIENTO DE IMÁGENES PARA MEDICIÓN DE ALTURA PIEZOMÉTRICAS



Fuente: autores.

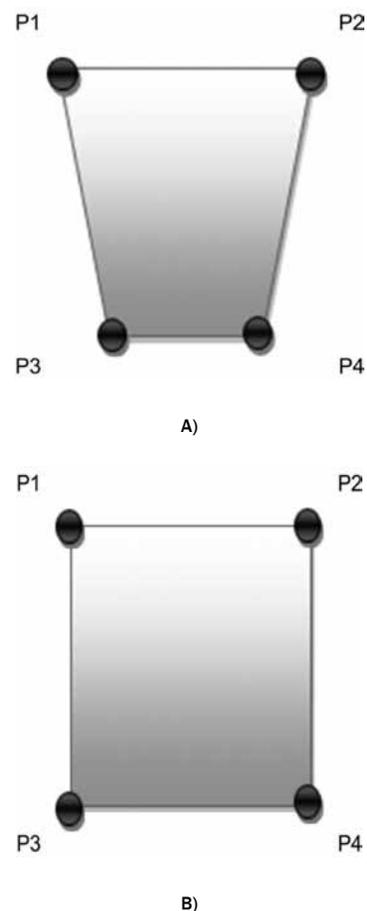
2.1.1 Captura de la imagen

Esta consta de una cámara digital que adquiere cuadro a cuadro la secuencia de imágenes tomadas de la escena del tablero piezométrico, con una resolución 800 x 600 píxeles y tasa de captura de 15 cuadros por segundo (fps). Se usa el módulo de configuración y lectura de cámaras digitales de OpenCV para leer y modificar las propiedades de la captura: cuadros por segundo, resolución en píxeles, brillo, contraste, entre otras [18].

2.1.2 Mejora de la imagen

En la *mejora de la imagen* las características de la secuencia de imágenes capturadas son modificadas, resaltando zonas especiales que aporten mayor información de las alturas piezométricas. Dichas modificaciones se realizan en dos etapas: a) corrección de perspectiva y b) filtrado espacial. En la Fig. 3 se observa la corrección de perspectiva en un rectángulo, donde el foco óptico está situado de manera que se visualiza una distorsión, requiriéndose de visualización sin distorsiones de perspectiva, ver Fig. 3.a; según lo anterior y la teoría de corrección de perspectiva, aplicada al rectángulo, se debe realizar una transformación geométrica proyectiva (TGP), tal que los puntos P1, P2, P3 y P4, ver Fig. 3.a, se mapeen en los nuevos puntos P1, P2, P3 y P4, corrigiéndose la distorsión, ver Fig. 3.b.

Fig. 3. CORRECCIÓN DE LA TGP EN A) IMAGEN DISTORSIONADA Y EN B) IMAGEN CON CORRECCIÓN



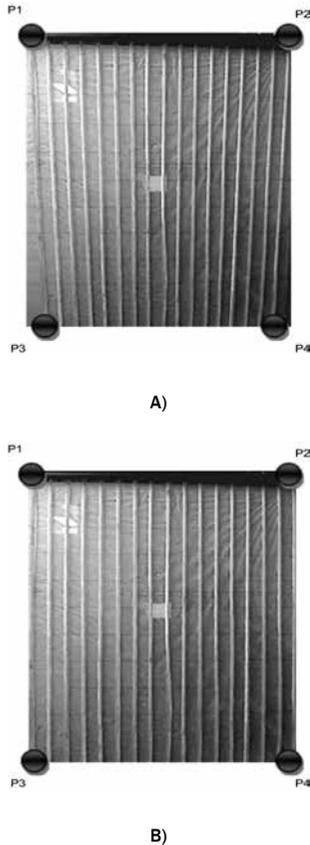
Fuente: autores.

La TGP se hace utilizando (1) [20]:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} m00 & m01 & m02 \\ m10 & m11 & m12 \\ m20 & m21 & m22 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

Donde x', y' representan las coordenadas 2D de la imagen con la corrección aplicada, w' es la tercera dimensión no utilizada, x, y representan las coordenadas 2D de la imagen transformada y M es la matriz de la TGP. En la Fig. 4 se observa el resultado de aplicar la TGP al tablero piezométrico, la cámara se encuentra en el extremo superior del tablero y por tanto se presenta un problema de perspectiva óptica; en la Fig. 4.a se muestra la imagen sin corrección y en la imagen de la Fig. 4.b el resultado de la corrección [21].

Fig. 4. CORRECCIÓN DE LA TGP DEL TABLERO PIEZOMÉTRICO EN A) IMAGEN ORIGINAL Y EN B) IMAGEN CON CORRECCIÓN



Fuente: autores.

En el procesamiento siguiente la secuencia de imágenes se transforma del espacio de color RGB a escala de grises, con la finalidad de mejorar contraste y brillo de la imagen y resaltar caracte-

rísticas pertinentes de las alturas piezométricas. Esto se hace con una función tipo LUT (*look up table*) de OpenCV [21].

2.1.3 Segmentación

Esta logra la separación de las alturas piezométricas del resto o fondo de la escena. Se organiza en una primera etapa de extracción de fondo y primer plano, donde se modela el fondo y se extrae el primer plano y una segunda etapa de filtrado morfológico donde se separan los objetos de interés.

• Extracción de fondo y primer plano

Un algoritmo de extracción de fondo o *background* y uno de primer plano o *foreground* se usan en la primera etapa. Esto se hace mediante un algoritmo de *codebook* [18] que permite realizar la estimación de primer plano y fondo. Un *codebook* es una estructura compuesta por pequeñas unidades de información llamadas *codewords*, que almacenan datos como distorsión de color, cambios de luminosidad, tiempo que ha pasado entre accesos al *codeword* y frecuencia con la que el *codeword* ha sido accedido para actualizarlo [15]. El *codeword* consiste en un vector RGB $v_i = (\overline{R_i}, \overline{G_i}, \overline{B_i})$ y una 6-tupla $(I_{\min}, I_{\max}, f, MNRL, fa, la)$ donde se almacena:

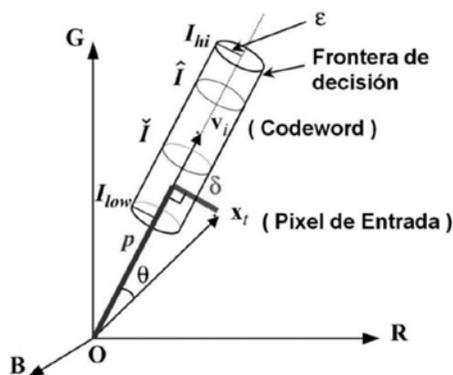
- $I_{\min}, I_{\max}, (\check{I}, \hat{I})$ Luminosidad máxima y mínima de todos los pixeles asignados a ese *codeword*.
- f Frecuencia con la que ese *codeword* ha ocurrido.
- $MNRL$ El mayor intervalo durante la creación de fondo en la cual este *codeword* no fue accedido.
- fa, la Primer y último tiempo de acceso al *codeword*.

Al contrario de los típicos modelos de color usados, donde se emplean colores normalizados, en el modelo *codeword* la distorsión de color se basa en una ponderación de la luminosidad en el modelo normalizado de color [15]. Este efecto se representa geoméricamente como un re-escalamiento o normalización a un vector *codeword* de un píxel según su luminosidad, ver Fig. 5.

Si se toma un píxel de entrada $x_t = (R, G, B)$ y un *codeword* C_i que posee un vector de color $v_i = (\overline{R_i}, \overline{G_i}, \overline{B_i})$, ver (2):

$$\begin{aligned}
 \|x_t\|^2 &= (R^2 + G^2 + B^2) \\
 \|v_i\|^2 &= (\overline{R_i^2} + \overline{G_i^2} + \overline{B_i^2}) \\
 \langle x_t, v_i \rangle^2 &= (R\overline{R_i} + G\overline{G_i} + B\overline{B_i})^2 \\
 \langle x_t, v_i \rangle &= (R\overline{R_i} + G\overline{G_i} + B\overline{B_i})
 \end{aligned} \quad (2)$$

Fig. 5. MODELO DE COLOR Y REPRESENTACIÓN GEOMÉTRICA DE CODEWORD



Fuente: autores.

La distorsión del color está dada por (3),

$$\delta = \sqrt{(\|x_t\|^2 - p^2)}$$

Siendo,

$$p^2 = \|x_t\|^2 \cos^2 \theta = \frac{\langle x_t, v_i \rangle^2}{\|v_i\|^2} \quad (3)$$

$$\delta \leq \epsilon$$

Esta distorsión se compara con un ϵ con el fin de fijar una frontera de decisión o validez del *codeword*. Respecto a la luminosidad, como en la 6-tupla *codeword* se almacenan tanto la máxima como la mínima, esto proporciona un rango en el cual la luminosidad varía. Ver (4).

$$\begin{aligned}
 I_{low} &= \alpha I_{max} \\
 I_{hi} &= \min\left(\beta I_{max}, \frac{I_{min}}{\alpha}\right)
 \end{aligned} \quad (4)$$

Donde $\alpha < 1$ y $\beta > 1$. Típicamente α está entre 0,4 y 0,7 y β entre 1,1 y 1,5

Finalmente, la función para detectar cambios en la luminosidad. Ver (5).

$$brightness = \begin{cases} true, & si \ I_{low} \leq \|x_t\| \leq I_{hi} \\ false, & en \ otro \ caso \end{cases} \quad (5)$$

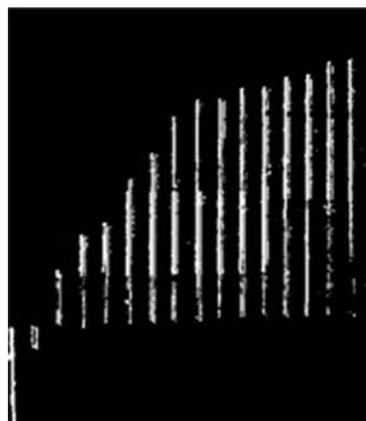
Según lo anterior, se construye un algoritmo de extracción en dos etapas: 1) se genera el modelo de fondo a partir de una secuencia de imágenes que entregan información del fondo, en ello se usan (3) y (5), y 2) se utiliza el modelo de fondo

creado y se calcula el primer plano a partir de este, a fin de realizar la detección de movimiento. Al final de esta etapa el modelo generado contiene para cada uno de los píxeles un *codeword*. El resultado de la extracción de fondo del panel piezométrico entrega los objetos en blanco correspondiente al primer plano y en negro el fondo, Fig. 6.a y Fig. 6.b.

Fig. 6. EXTRACCIÓN DE FONDO TABLERO PIEZÓMETRO EN A) IMAGEN ORIGINAL Y EN B) IMAGEN EXTRAÍDA



A)



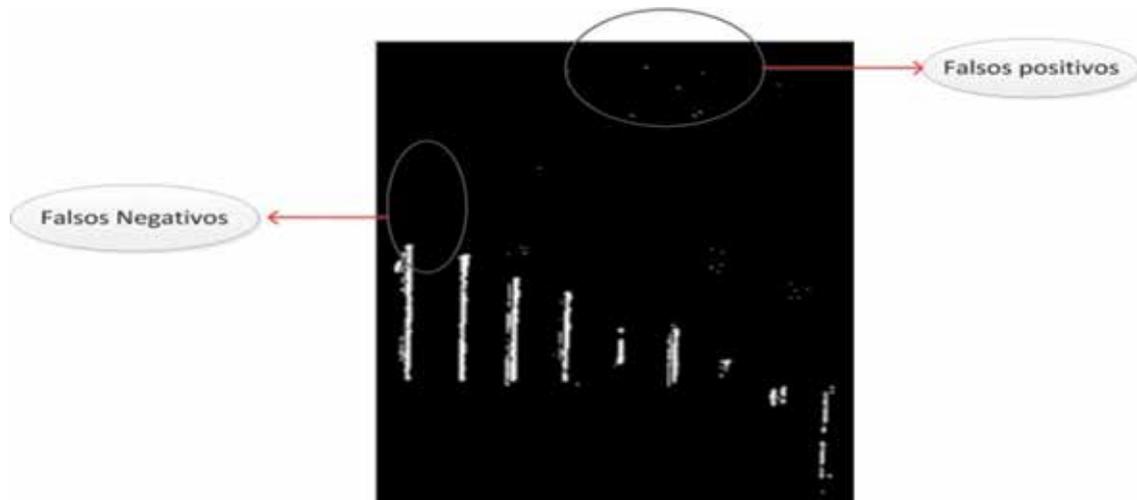
B)

Fuente: autores.

• Filtrado morfológico

En este se realiza una binarización [19] para separar regiones u objetos de interés en la imagen, las áreas en blanco representan las alturas piezométricas en el tablero o valor verdadero y en negro las áreas sin interés o valor falso Fig. 7. Las detecciones falsas y omitidas se descartan en la etapa de reconocimiento de alturas.

Fig. 7. RESULTADO BINARIO DEL FILTRADO MORFOLÓGICO CON DETECCIONES FALSAS Y OMITIDAS



Fuente: autores.

2.1.4 Reconocimiento y estimación de alturas

En este se encuentra y calcula la medida real en milímetros (mm) de cada altura binarizada. Se usa unión de blobs (regiones conectadas de píxeles con igual intensidad) [22] y se finaliza con cálculo de las coordenadas (x,y) de cada blob detectado para estimar la altura piezométrica.

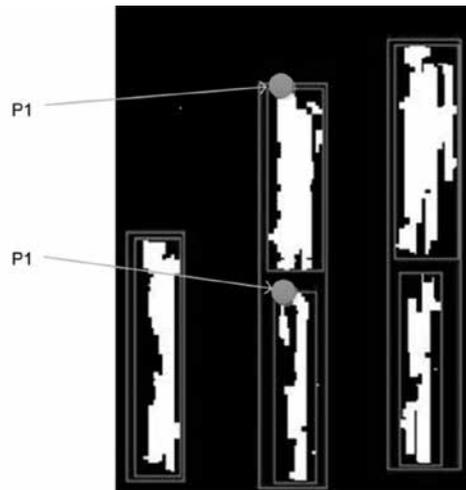
- **Unión de blobs (merging)**

Aquí se agrupan múltiples blobs cercanos en un único blob [22], [23]; se realiza a partir de un criterio de distancia entre blobs: distancia euclidiana, distancia vertical u horizontal. Estas últimas se definen como la distancia en píxeles de puntos verticales u horizontales. Sean dos blobs B_1 y B_2 asociados a una *caja de rectángulo* o *Bounding-Box* (rectángulo más pequeño que contiene un blob) dados por los puntos $P_1(x,y), P_2(x,y)$, la distancia vertical D_v se define como la diferencia de la coordenada y de los puntos, $D_v = y_{P_1} - y_{P_2}$, y la distancia horizontal D_h como la diferencia de las coordenadas x de los puntos $D_h = x_{P_1} - x_{P_2}$. Si se unen o no dos blobs se tiene en cuenta la distancia horizontal y vertical entre las alturas piezométricas, según (6),

$$\text{unir si} = \begin{cases} D_v < thr_vertical \\ D_h < thr_horizontal \end{cases} \quad (6)$$

La aplicación de la técnica une los blobs que se encuentran cercanos de manera vertical Fig. 8.

Fig. 8. UNIÓN DE BLOBS PERTENECIENTES A UNA MISMA ALTURA PIEZOMÉTRICA POR DISTANCIA VERTICAL



Fuente: autores.

- **Estimación de alturas**

Esta consta de dos etapas: 1) se realiza la estimación de altura en píxeles, y en 2) se realiza conversión a unidades de longitud, según una relación milímetros por píxel. Dichas etapas se detallan a continuación:

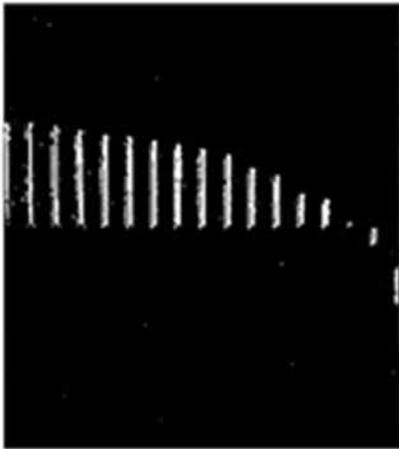
- 1) Estimación de altura en píxeles. Con cada blob en su *bounding box* se toma el valor superior medio del mismo en un punto del plano (x, y). Para conocer cada punto en el plano del blob o altura piezométrica se realiza un rastro de la imagen de arriba hacia abajo y se

marcan los puntos extremos de la imagen binaria. Todo punto de las alturas piezométricas se guarda en una estructura para su posterior análisis.

- 2) Conversión a unidades de longitud. Finalmente se calculan las alturas piezométricas de la cuantificación realizada en el plano (x, y) . Para ello se usa una constante de relación λ , ver (7), para determinar la medida en milímetros de cada altura piezométrica. Si la imagen es de 800x600 píxeles corresponde a una medida de 1000x800 milímetros (área del tablero piezométrico), Fig. 9, por lo que:

$$\lambda = \frac{\text{Altura (mm)}}{\text{Alturapi (píxeles)}} = 1,25\text{mm/píxel} \quad (7)$$

Fig. 9. DETECCIÓN DE ALTURAS PIEZOMÉTRICAS



Fuente: autores.

Antes de proceder con la aplicación experimental del algoritmo se realizó un proceso de calibración, en él se delimita el área efectiva del tablero

piezométrico que cubrirá la cámara. Para ello se suministran los datos correspondientes al alcance y cero del transmisor multinivel de alturas. Para el caso particular del tablero piezométrico usado experimentalmente el alcance es de 1000 mm y el cero de 0 mm, que tras la aplicación del TGP permite relacionarlo con el alto en píxeles del tablero piezométrico. Otros dos parámetros de calibración configurados son el brillo y el contraste de la cámara, con el fin de lograr valores en los que las alturas piezométricas presenten buena separación del fondo del tablero, para ello se debe garantizar una buena y constante iluminación convencional de luces fluorescentes sobre el tablero.

3. RESULTADOS

En la validación se hicieron cálculos del desempeño temporal de cada componente del mismo, mientras se variaban los niveles en un tablero de 16 alturas piezométricas. Se tomaron medidas por observación directa, y se contrastaron con las medidas estimadas utilizando el algoritmo implementado, para diez caudales de entrada del circuito hidráulico, con el propósito de hacer el cálculo y estimación del error producido en la medida calculada por el algoritmo, además de verificar el seguimiento de las alturas piezométricas. A partir de los datos obtenidos, se obtuvo el desempeño temporal, el error relativo y la precisión del algoritmo.

3.1 Desempeño temporal

El algoritmo ejecutado en el computador de escritorio logró un tiempo promedio de procesamiento de 250ms permitiendo procesar 5 marcos por segundo, Tabla I.

TABLA I
TIEMPOS DE PROCESAMIENTO PROMEDIO DE CADA COMPONENTE DEL ALGORITMO (MS)

Captura de imagen	Mejora de imagen	Segmentación		Reconocimiento y estimación de alturas	
		Extracción de fondo	Filtrado	Unión blobs	Estimación
4	41	45	50	56	60

Fuente: autores.

3.2 Cálculo del error de medición

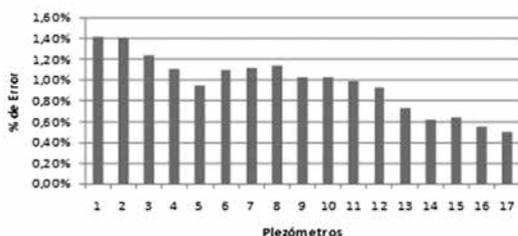
Se tomó como patrón de referencia las medidas realizadas manualmente, utilizando cinta métrica de precisión Clase II. El error relativo porcentual por piezómetro, según una serie de diez medidas tomadas para cada uno, fue calculado según (8).

$$Er = ((X_{patrón} - X_{algoritmo}) / X_{patrón}) * 100\% \quad (8)$$

Donde $X_{patrón}$ corresponde al dato manualmente medido, y $X_{algoritmo}$ corresponde al dato medido por el algoritmo.

Aplicada (8) para medir el promedio del error relativo en la medición de cada piezómetro, Fig. 10, el porcentaje de error mínimo es de 0,4% para el piezómetro 17, mientras que el porcentaje de error más alto, cerca al 1,4%, lo presentan los piezómetros 1 y 2.

Fig. 10. PORCENTAJE DE ERROR EN LA MEDICIÓN POR CADA PIEZÓMETRO



Fuente: autores.

3.3 Precisión del algoritmo

En la Tabla II se registra el cálculo de la variabilidad para cada piezómetro, se procesaron 5 medidas por piezómetro, cuando el circuito hidráulico opera con un caudal de 22 litros/minuto. La sexta columna de la Tabla II consigna los valores del coeficiente de variación, según cálculos dados por (9).

$$CV = \frac{desv.estándar}{media} * 100\% \quad (9)$$

En la Tabla II, los valores del coeficiente de variación inferiores a 0,10 indican una alta concentración, entre 0,1 y 0,5 una concentración media y valores superiores a 0,5 una alta dispersión y una media poco o nada representativa [24]. El coeficiente de variación en cada piezómetro es inferior al 0,075 % y la desviación estándar inferior a 0,52; lo que se traduce en precisión en cada una de las medidas entregadas por el algoritmo, logrando un alto grado de reproducibilidad del proceso de medición en condiciones normales de operación.

4. CONCLUSIONES

Se diseñó e implementó un algoritmo de procesamiento óptico digital de imágenes, concebido como un multitransmisor de nivel, para la medición en línea de dieciséis alturas piezométricas, en un banco de pérdidas de carga en conductos

TABLA II
MEDIDAS DE VARIABILIDAD EN LA MEDICIÓN DE LOS 16 PIEZÓMETROS

Piezómetro	Media		Mediana (cm)	Moda (cm)	Desviación estándar (mm)	Coeficiente de variación (CV)
	cm	pixeles				
1	28,77	230	29	28,61	0,17	0,060%
2	47,21	377	47	47,17	0,11	0,023%
3	66,67	533	67	66,67	0,00	0,000%
4	68,97	551	69	69,11	0,52	0,075%
5	81,64	653	82	81,71	0,32	0,039%
6	74,06	592	74	74,12	0,10	0,013%
7	78,96	631	79	79,00	0,11	0,014%
8	86,64	693	87	86,59	0,13	0,015%
9	89,37	714	89	89,16	0,49	0,055%
10	90,32	722	90	90,38	0,12	0,013%
11	92,78	742	93	92,82	0,13	0,014%
12	92,93	743	93	92,96	0,12	0,013%
13	94,83	758	95	94,72	0,14	0,015%
14	95,21	761	95	95,26	0,11	0,012%
15	97,29	778	97	97,29	0,11	0,011%
16	97,61	780	98	97,57	0,09	0,009%

Fuente: autores.

a presión. El algoritmo consta de cuatro componentes: captura de la imagen, realiza la adquisición de la secuencia de imágenes en 4ms, mejora de la imagen donde se corrige la deformación de perspectiva, entre otros, en 41 ms, segmentación para modelar y eliminar el fondo de la escena, en 95 ms, reconocimiento y estimación de alturas que en 110 ms estima la altura en mm.

La técnica de extracción de fondo basada en el algoritmo de *Codebook*, combinada con el filtrado morfológico y unión de blobs, son usados en los componentes de segmentación y reconocimiento y estimación de alturas, en general se considera que la metodología usada en el diseño e implementada en las cuatro fases del algoritmo del multitransmisor de nivel es adecuada para la estimación de alturas piezométricas. Sin embargo, el algoritmo presenta dificultades dependiendo del no cumplimiento de las condiciones establecidas en el proceso de calibración de la cámara, en particular en lo concerniente a la iluminación de la escena. A pesar de lo anterior el multitransmisor de nivel se constituye en una valiosa herramienta de apoyo para la realización de las prácticas de pérdidas de carga en conductos a presión en los laboratorios de hidráulica.

El algoritmo del multitransmisor de nivel obtiene simultáneamente las 16 alturas piezométricas de presión con un porcentaje de error menor a 1,4 % y con una precisión menor al 0,1 %, lo que hace al algoritmo confiable para la medición de pérdidas de presión en conductos. El algoritmo se implementó en lenguaje C usando Open CV 2.3 y se ejecuta en 200 ms, permitiendo realizar 5 medidas por segundo en cada uno de los 16 piezómetros.

REFERENCIAS

- [1] L. Fortuna, S. Graziani, and M. G. Xibilia, *Soft sensors for monitoring and control of industrial processes*. 2007.
- [2] T.-H. Wang, M.-C. Lu, C.-C. Hsu, C.-C. Chen, and J.-D. Tan, "Liquid level measurement using a single digital camera," *Measurement*, vol. 42, pp. 604-610, 2009.
- [3] D. Yasuhiko and S. J. Ovaska, "Industrial applications of soft computing: A Review," *Proc. IEEE*, vol. 89, no. 9, 2001.
- [4] W. Healy, "Building america meeting on miagnostic measurement and performance feedback for residen-
tial space conditioning equipment," *Build. Sensors Energy Monit. Syst.*, no. July, 2010.
- [5] G. Heredia and A. Ollero, "Virtual sensor for failure detection, identification and recovery in the transition phase of a morphing aircraft," *Sensor*, vol. 10, pp. 2188-2201, 2010.
- [6] W. Wei, L. Chunfang, and X. Xinhe, "Simulation of robot localization based on virtual sensors," in *2008 Chinese Control and Decision Conference*, 2008, pp. 248-252.
- [7] J. P. Sebastiá, J. Alberola, and R. Lajara, "Physical signal conditioning for GMR magnetic sensors : Applied to traffic speed monitoring GMR sensors," *Sensors Actuators A Phys.*, vol. 137, pp. 230-235, 2007.
- [8] P. Ibarquengoytia, A. Reyes, M. Huerta, and J. Hermosillo, "Probabilistic virtual sensor for on-line viscosity estimation," in *Artificial Intelligence, 2008. MICAI '08. Seventh Mexican International Conference on*, 2008, pp. 3777-782.
- [9] N. Hardy and A. Ahmad, "De coupling for re use in design and implementation using virtual sensors," *Auton. Robot.*, vol. 6, pp. 265-280, 1999.
- [10] N. Hardy and A. Ahmad, "ViSIAR - A virtual sensor integration architecture," *Robotica*, vol. 17, pp. 637-647, 1999.
- [11] E. Velásquez, "Lectura automática de instrumentos de medida mediante técnicas de visión artificial," Universidad de Vigo, 2008.
- [12] E. Vázquez-Fernández, A. Dacal-Nieto, H. González-Jorge, F. Martín, A. Formella, and V. Álvarez-Valado, "A machine vision system for the calibration of digital thermometers," *Meas. Sci. Technol.*, vol. 20, no. 6, p. 065-106, Jun. 2009.
- [13] P. Jampana, "Computer vision based sensors for chemical processes outline introduction," University of Alberta, 2009.
- [14] E. Valverde and J. Valverde, "Diseño e implementación de un laboratorio remoto: caso de estudio planta de pérdidas de carga en conductos a presión del Laboratorio de Hidráulica," Universidad del Cauca, 2010.
- [15] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172-185, Jun. 2005.
- [16] R. Lagani, *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing Ltd, 2011.
- [17] I. Corporation, "Intel Corporation.pdf," 2012. [Online]. Available: <http://developer.intel.com>.

- [18] G. Bradski and A. Kaehler, *Learnig OpenCV*. O'Reilly Media, Inc., 2008.
- [19] R. Gonzalez and R. Woods, *Digital image processing*. Prentice Hall Upper Saddle River, 2002.
- [20] N. Raveendranathan, V. Loseu, E. Guenterberg, R. Giannantonio, R. Gravina, M. Sgroi, and R. Jafari, "Implementation of virtual sensors in body sensor networks with the SPINE framework," *2009 IEEE Int. Symp. Ind. Embed. Syst.*, pp. 124-127, Jul. 2009.
- [21] J. A. Cerquera and J. E. Córdoba, "Medición y pérdidas de presión usando visión por computador," Universidad del Cauca, 2012.
- [22] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, and R. Bolle, "Appearance models for occlusion handling," *Image Vis. Comput.*, vol. 24, no. 11-1, pp. 1233-1243, 2014.
- [23] F. Chang, C. Chen, and C. Lu, "A linear-time component-labeling algorithm using contour tracing technique," *Comput. Vis. image Underst.*, vol. 93, pp. 206-220, 2003.
- [24] I. Espejo, F. Fernández, M. López, M. Muñoz, A. Rodríguez, A. Sánchez, and C. Valero, "Estadística descriptiva y probabilidad: (Teoría y problemas)," Universidad de Cádiz, 2009.