

Towards 3D Pipe Reconstruction Employing Affine Transformations from Video Information

. Reyes-Acosta, I. Lopez-Juarez, R. Osorio-Comparan, G. Lefranc *Life Senior Member, IEEE*

Abstract— The development of a virtual scenario is desirable in a number of hostile environments such as in mining, or sewer pipelines to inspect its current condition and to identify possible failures. Obtaining a virtual image through the reconstruction of the mineshaft or inner pipe is a valuable tool for assessment and fault detection. In this paper, we present results for our 3D reconstruction approach, which proposes to employ regions with affine transformations that are used as dense correspondences. These features are calculated through local correspondences, such as the SIFT (Scale-invariant feature transform) and SURF (Speeded Up Robust Features). The correspondences are required to obtain the orientation and position of the camera and to get the structure of the scene leading to the creation of the 3D environment. The algorithm is presented as well as its results, which highlights the potential of the method.

Keywords— Dense correspondence, Local correspondence, Affine transformation.

I. INTRODUCCION

LA inspección de tuberías es una actividad importante para garantizar la calidad de las instalaciones. Esta actividad se lleva a cabo en diversos escenarios como la inspección de minas, instalaciones de petróleo, tuberías eléctricas y de agua o de drenaje con el fin de identificar posibles defectos, para inspeccionar su estado actual y poder programar alguna posible rehabilitación.

El servicio de inspección de tubería usando robots móviles ha recibido recientemente mucha atención, ya que simplifica algunas tareas de operadores. Una inspección de este tipo es una tarea muy demandante para el operador debido a que las longitudes de inspección pueden ser desde 100 metros hasta varios kilómetros y lo cual puede generar descuidos en la inspección pasando por alto defectos. En términos de procesamiento de imagen, existen otras complicaciones a considerar, tales como condiciones pobres de iluminación, cantidad excesiva de información de video, fusión de imágenes, detección de defectos, tiempo de procesamiento, entre otros aspectos.

Una solución viable es utilizar robots móviles para la evaluación de la calidad del tubo interior mediante el uso de técnicas de reconstrucción 3D a partir de la información de video.

A. Reyes-Acosta e I. Lopez-Juarez están en CINVESTAV - Saltillo, México. e-mail: alfredoreyesacosta@hotmail.com, ismael.lopez@cinvestav.edu.mx

Roman Osorio-Comparan está en IIMAS-UNAM, Circuito Escolar s/n, Ciudad Universitaria Ciudad de México, México. e-mail: roman@unam.mx

Gaston Lefranc está en la Escuela de Ingeniería Eléctrica en la Pontificia Universidad Católica de Valparaiso, Chile. e-mail: gaston.lefranc@pucv.clA.

Corresponding author: Ismael Lopez

De este modo, la detección de defectos y clasificación se pueden realizar fuera de línea y llevar a cabo acciones preventivas y/o correctivas.

El artículo está organizado como sigue. Después de la introducción en esta sección, se presentan las etapas necesarias para la reconstrucción 3D usando transformaciones afines en la sección II. La sección III presenta formalmente el método propuesto. En la sección IV se proporcionan resultados, finalmente las conclusiones y trabajo futuro se presenta en la sección V.

II. RECONSTRUCCIÓN 3D A PARTIR DE VIDEO

La típica secuencia para una reconstrucción en 3D de una escena a partir de un video [1], [2] es detectar y realizar correspondencias de puntos característicos, recuperar movimiento de la cámara y la estructura de la escena, generar un mapa denso y para obtener un modelo 3D como se muestra en la Fig. 1.



Fig. 1. Típica secuencia para la reconstrucción en 3D.

Para capturar el video, el robot se desplaza dentro de la tubería, el video es procesado para generar el ambiente virtual el cual considera ciertas limitaciones tales como un escenario estático y que se pueden obtener correspondencias entre las imágenes. En cuanto al problema de las correspondencias, hay dos métodos principales para atacarlo. Se puede utilizar métodos locales o globales dependiendo del conocimiento a priori de las imágenes. Si se encuentran imágenes consecutivas con un pequeño movimiento entre imágenes, es común usar un método global como flujo óptico. Por otro lado, si hay cambios de vista importantes entre imágenes, entonces el uso de las correspondencias locales es una mejor opción. En los párrafos siguientes, se describen las operaciones requeridas y una revisión del trabajo relacionado en el tema para entender el modelo propuesto.

A. Correspondencias locales.

Para obtener correspondencias de puntos característicos normalmente se emplean detectores y descriptores. Un buen detector se utiliza para detectar características localizadas en un punto o región, debe ser repetible y confiable. Según Mikolajczyk et al. [3] no hay un detector que funcione mejor para la mayoría de los casos. Un descriptor genera, normalmente, un vector el cual describe un punto característico. Mikolajczyk y Schmid [4] realizaron la

comparación de rendimiento de diversos descriptores del estado del arte, encontrando que el SIFT y la extensión del SIFT tuvieron el mejor rendimiento y muy cercano entre ellos. Lowe [5] menciona, sobre el descriptor SIFT, que las superficies de textura plana se pueden identificar de forma fiable con una rotación de 50 grados en cualquier dirección y bajo casi cualquier condición de iluminación visible.

B. Correspondencias globales.

Tomando en cuenta los métodos globales, el flujo óptico es comúnmente utilizado, el cual genera correspondencias densas. Dentro de estos métodos densos, hay algoritmos que intentan fusionar las dos estrategias principales, locales y globales, como SIFT Flow [6], [7]. También hay métodos globales sin el uso de flujo óptico, como [8], [9]. Por otro lado, existen métodos que realizan correspondencias empleando regiones con transformaciones afines [10], [11], [12]. Ferrari [13] realiza estas correspondencias a través de regiones, tomando en cuenta la ubicación geométrica entre ellas para eliminar y aumentar el número de regiones.

C. Regiones Affine.

Las regiones tipo *affine*, utilizadas en varios métodos de correspondencia, se pueden calcular por medio de 3 pares de puntos o más. Un ejemplo se muestra en la Fig. 2, los cuales generan una matriz de transformación de 3 x 3, la cual se puede simplificar a 2 x 3; dicha matriz tiene 6 grados de libertad, los cuales son los valores de la matriz. La transformación dada por esta matriz mantiene el paralelismo de las geometrías, o sea, que, si hay líneas paralelas, seguirán paralelas después de la transformación [14].

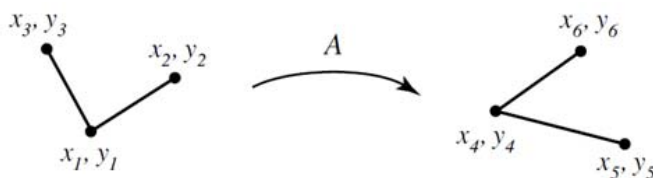


Figura 2. Ejemplo de una transformación tipo *affine* con 3 pares de puntos.

Dicha matriz se puede escribir como:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = A \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (1)$$

que se puede simplificar como:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = A \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2)$$

Genéricamente hablando, se puede obtener dicha transformación con más de 3 puntos. Para calcular la matriz supongamos el conjunto de puntos (\$CP_1\$) "n" en una de las imágenes como en la ecuación (3).

$$P_1 = \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \rightarrow CP_1 = \begin{pmatrix} x_1 & \dots & x_n \\ y_1 & \dots & y_n \\ 1 & \dots & 1 \end{pmatrix} \quad (3)$$

Por otro lado, se puede escribir las correspondencias de la otra imagen en el conjunto 2 de puntos de forma similar:

$$P_2 = \begin{pmatrix} x'_1 \\ y'_1 \\ 1 \end{pmatrix} \rightarrow CP_2 = \begin{pmatrix} x'_1 & \dots & x'_n \\ y'_1 & \dots & y'_n \\ 1 & \dots & 1 \end{pmatrix} \quad (4)$$

Entonces se puede obtener A por medio de la división de estos conjuntos de correspondencias.

$$A = CP_2 / CP_1 \quad (5)$$

Entre más correspondencias locales se tengan para el cálculo de la transformación, se reduce el ruido que puedan causar la baja resolución, cambios de iluminación, de escala entre otros, lo cual mejora la correspondencia densa.

D. Recobrar movimiento y reconstrucción 3D.

Para recobrar el movimiento y la estructura, se toman las correspondencias entre imágenes, de las cuales se pueden agregar restricciones geométricas y así generar las matrices de proyección las cuales representan información de movimiento de la cámara, ya que suponemos un escenario estático, o una cámara fija y que se mueva toda la escena sin deformaciones [15], se pueden tener otros sensores en lugar de la cámara para la reconstrucción [16].

Una cámara tiene tanto parámetros intrínsecos como extrínsecos (internos y externos). Los parámetros intrínsecos son aquellos que describen el funcionamiento de una cámara, como la distancia focal. Los parámetros externos de la cámara son aquellos que definen la posición y la orientación del cuadro de referencia de la cámara con respecto al mundo real.

La reconstrucción con solo el conocimiento de las correspondencias es solamente posible con una reconstrucción proyectiva. Existen varias maneras de obtener esas matrices proyectivas, como la matriz fundamental o el tensor focal, como lo menciona Trung [1], sin restricciones adicionales no es posible generar una reconstrucción métrica.

III. MODELO PROPUESTO

El escenario en el que se desea trabajar carece de líneas rectas y esquinas, utilizadas para obtener detectores rápidos en cómputo, el color y la textura tienden a ser muy similar en las imágenes y la iluminación es variante a lo largo del trayecto, un ejemplo de las características mencionadas se muestra en la Fig. 3.

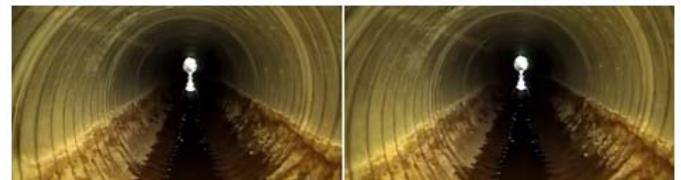


Figura 3. Ejemplos de imágenes en el interior de un tubo de drenaje

El modelo propuesto intenta superar las condiciones antes mencionadas trabajando bajo condiciones de iluminación deficientes usando correspondencia densa. El entorno de trabajo no tiene líneas rectas o esquinas que se utilizan normalmente para la computación rápida. El color y la textura

tienden a ser muy similares en las imágenes procesadas. Un ejemplo de dos imágenes consecutivas tomadas por nuestro robot móvil diseñado para fines de inspección se muestra en la Fig. 4.

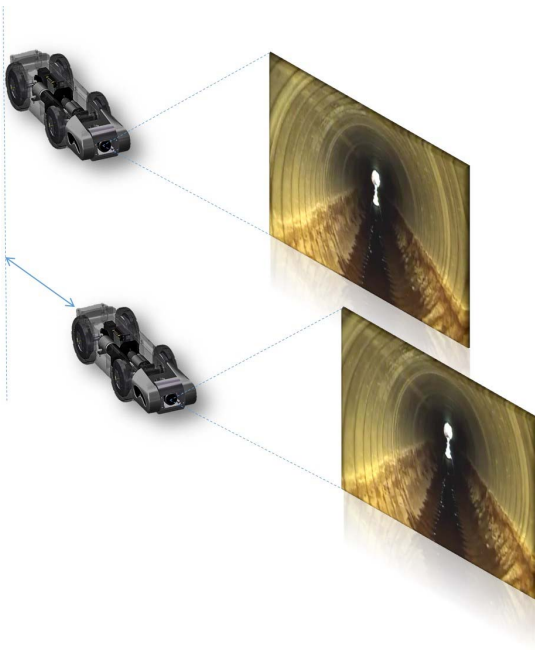


Figura 4. Imágenes en el interior de un tubo de drenaje tomadas por robot

Los pasos convencionales para la reconstrucción son: a) Detectar y realizar correspondencias, b) Recuperar movimiento y estructura, c) Generar un mapeo denso, y finalmente, d) Reconstrucción del modelo 3D. Una vez que el algoritmo propuesto determina buenas correspondencias, entonces se determina la pose de la cámara. El método busca correspondencias (utilizando detectores y descriptores) entre dos imágenes realizando la eliminación por medio de regiones con transformaciones afín, para posteriormente obtener la pose de la cámara.

El método comienza llenando la imagen con regiones circulares referidas como candidatos que se traslapan como se muestra en la Fig. 5 (Izquierda). El tamaño del círculo está vinculado a la velocidad y precisión del método. El empleo de regiones pequeñas tiende a obtener mejores resultados; Sin embargo, el proceso se vuelve lento. Si el escenario es plano, se recomienda utilizar regiones grandes para acelerar el proceso.

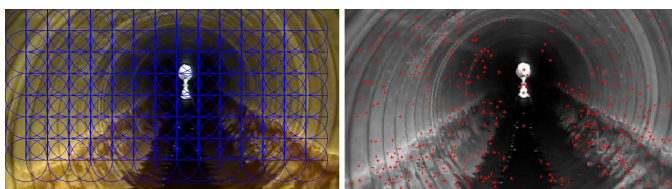


Figura 5. Regiones candidatas

La correspondencia se realiza mediante detectores y descriptores comúnmente utilizados en el campo como SIFT

[5] y SURF [17]. El método emplea regiones circulares fijas, para eliminar las coincidencias incorrectas. Más de tres puntos, reducen ruido en la correspondencia densa, pero también se reduce el número de regiones candidatas. Los puntos de correspondencia entre regiones se realizan mediante transformaciones afín con el fin de crear información densa para la reconstrucción y las malas correspondencias se eliminan localmente empleando el algoritmo 1.

Algoritmo 1

Eliminar las correspondencias dentro de las regiones locales, donde n es el total de Correspondencias dentro de una región, CanReg son las regiones candidatas.

```

1: for Reg = 1 to CanReg do
2:   if n < Umbral (5 para el ejemplo) then
3:     Borrar Reg de CanReg y regresa a 1.
4:   else
5:     for i = 1 to n do
6:       for j = 1 to n do
7:         D1[i][j] = Distancia n(i) a n(j) en imagen 1.
8:         D2[i][j] = Distancia n(i) a n(j) en imagen 2.
9:         D[i][j] = D1[i][j]/D2[i][j] (para normalizar).
10:        D[j][i] = D[i][j]. (Para rellenar la matriz).
11:      end for
12:      Obtener la median de n(i) con la ecuación (5)
13:      for j = 0 to n do
14:        if Si no cumple ecuación (6) then
15:          Agrega un PuntoMalo a n(i) y n(j).
16:        end if
17:      end for
18:    end for
19:    Obtener las correspondencias con max PuntoMalo.
20:    if P untoMalo > Umbra (4 para el ejemplo) then
21:      Borrar la crrespondencia.
22:      Ir a 2 (Para verificar la region de nuevo)
23:    end if
24:  end if
25: end for
    
```

Para el algoritmo, el promedio de la comparación de las distancias euclidianas se determina con:

$$Prom_{ij} = \frac{\sum_{k=1}^n \frac{D1_k^2}{D2_k^2}}{n} \tag{6}$$

donde:

- $\frac{D1_i^2}{D2_i^2}$ distancia de un punto “i” a “j” en la imagen 1.
- $\frac{D2_i^2}{D2_j^2}$ distancia del punto “i” a “j” en la imagen 2.
- n Número de correspondencias dentro de la región.

De tal forma que:

$$0.9Prom_{ij} \leq \frac{D1_i^2}{D2_i^2} \leq 1.1Prom_{ij} \tag{7}$$

Para la eliminación, se toma un punto cualquiera y se toma

la distancia hacia las demás correspondencias para ambas imágenes, como se muestra en la Fig. 6.

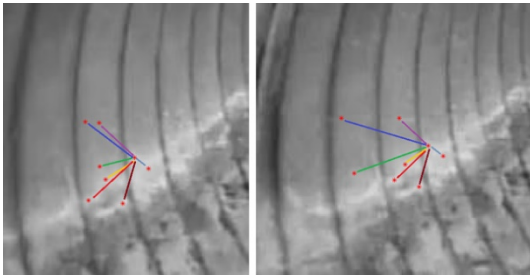


Figura 6. Distancia de una correspondencia a las demás.

En la Fig. 7, se muestran en diferentes colores las distancias de una correspondencia ejemplo hacia las demás. Se pueden apreciar 2 líneas visiblemente más grandes en comparación con las demás.

Al realizar este proceso con el resto de las correspondencias, se tiene un efecto similar, por lo que se puede eliminar las correspondencias “malas”. En la Fig. 9 se muestra la transformación de la región candidata al eliminar las correspondencias.

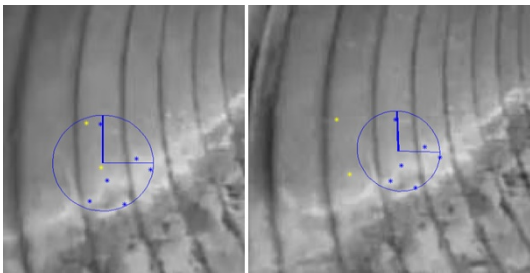


Figura 7. Ejemplo de transformación tipo *affine*.

Este proceso se realiza para la imagen completa, obteniendo regiones en casi toda la imagen. El resultado de dichas regiones se muestra en la Fig. 8, los puntos de color azul son las correspondencias “buenas”, los puntos de color amarillo, son las correspondencias eliminadas por no cumplir con las dimensiones, y en rojo las correspondencias que no se tiene suficientes datos para tomarlas como “buenas” o “malas”.

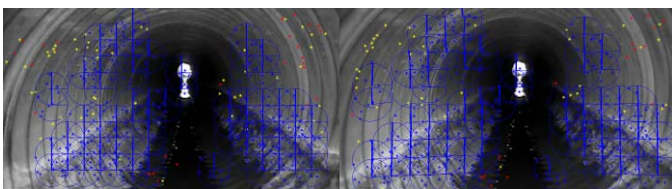


Figura 8. Regiones sobre toda la imagen.

Debido al movimiento de las imágenes, se puede saber que las regiones no pueden tener una excentricidad (parámetro que determina el grado de desviación de una sección cónica con respecto a una circunferencia) muy diferente a las demás, lo mismo aplica para el tamaño y el ángulo al que apunta la región, por lo cual se puede realizar un filtro tomando en cuenta estas características en las regiones, como lo muestra la

Fig. 9.

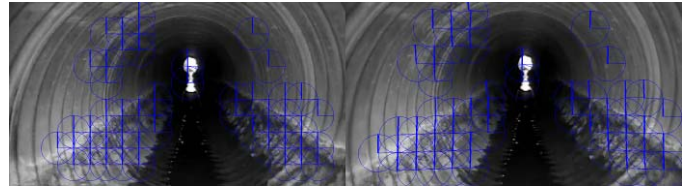


Figura 9. Regiones fijas después de filtrarlas tomando en cuenta la excentricidad y el ángulo.

El algoritmo propuesto utiliza correspondencias locales para crear regiones con transformaciones *affine*. Estas regiones son utilizadas para eliminar malas correspondencias (y obtener correspondencias densas). Las correspondencias restantes se utilizan para obtener la matriz fundamental por medio del algoritmo Normalizado de 8 puntos, posteriormente se ejecuta la descomposición de valores singulares (SVD) de la matriz esencial (La matriz esencial es una generalización de la matriz fundamental) para determinar la pose de la cámara [1].

La diferencia entre el algoritmo propuesto y otros métodos comúnmente empleados como RANSAC, el cual es un método iterativo para calcular los parámetros de un modelo matemático de un conjunto de datos que contiene valores atípicos, es no determinista en el sentido de que produce un resultado razonable sólo con una cierta probabilidad, mayor a medida se realicen más iteraciones. Para el ejemplo, se utilizó el método RANSAC con las mismas imágenes que se muestran en la Fig. 10.

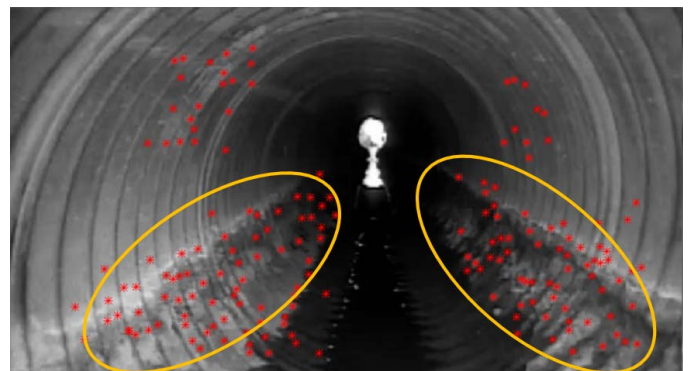


Figura 10. Correspondencias restantes en el método propuesto.

Como se puede observar, RANSAC no puede encajar todas las correspondencias, normalmente se ajusta a las correspondencias encerradas en el círculo porque allí se encuentra la mayor concentración de puntos. Existe la opción de usar RANSAC en las regiones locales en lugar del método propuesto, pero en el caso donde no hay una correspondencia local real RANSAC siempre da una solución y el usuario debe tomar una decisión. La idea del método propuesto es eliminar algunas regiones localmente o eliminar todas las correspondencias dentro de una región. Si es así. El algoritmo propuesto tiene la ventaja de considerar pequeñas regiones, por lo que es mejor para las superficies continuas como las

empleadas en esta investigación.

V. RESULTADOS.

Para probar el método, se consideraron dos casos. La primera prueba con un caso genérico donde la escena es rica en textura y el segundo caso fue probado dentro de la tubería bajo condiciones reales. Las imágenes de la escena rica en textura fueron tomadas en dos posiciones diferentes como se muestra en la Fig. 11. Se eliminaron las regiones que no contenían correspondencias.

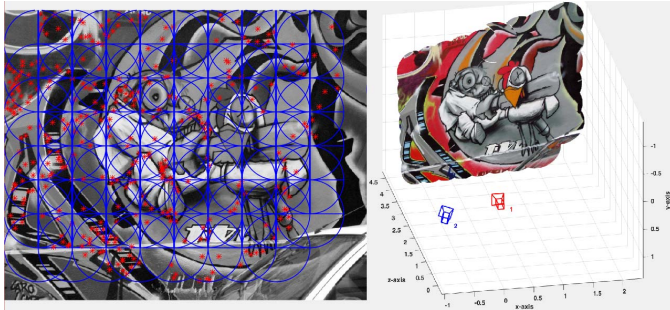


Fig. 11. Regiones del método y su reconstrucción 3D

La posición estimada de la cámara se muestra en color rojo y azul. Como se puede observar la reconstrucción fue completa con la excepción de la esquina inferior derecha donde la superficie no contiene textura.

En la figura 12 se muestran imágenes de condiciones reales dentro del tubo. Se muestra en particular otra región candidata del tubo, similar a las regiones anteriores, se muestra las correspondencias seleccionadas, eliminadas y el plano por el cual se realizó la selección, así como su transformación afín.

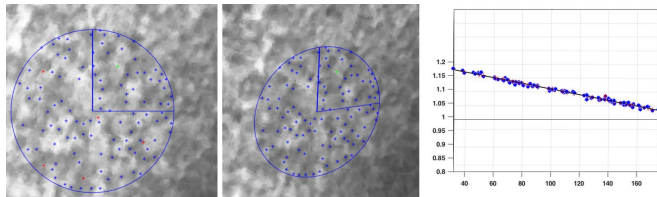


Figura 12. Otro ejemplo de transformación afín con correspondencias restantes.

Las correspondencias dadas por el método propuesto se utilizan para obtener la matriz fundamental, posteriormente se ejecuta la descomposición de valores singulares (SVD) de la matriz esencial (la matriz esencial es una generalización de la matriz fundamental) para determinar la pose de la cámara [1] y con las matrices de la cámara se puede realizar la triangulación de las correspondencias para generar la reconstrucción 3D del escenario.

Para la obtención de la pose de la cámara, se emplea el algoritmo LTS, el cual nos permite eliminar un porcentaje de las correspondencias obtenidas por el método propuesto, con lo que se mejora el resultado. Experimentalmente se determinó que un valor de 10% fue suficiente para la eliminación de

correspondencias mal ubicadas dentro de la región de interés, en la figura 13 se muestra la posición de la cámara en las diferentes imágenes, así como las correspondencias en 3D.

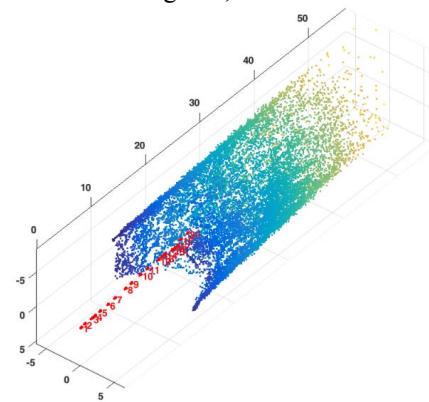


Figura 13. Pose de la cámara y correspondencias en 3D.

Se obtienen las correspondencias densas por medio de las transformaciones afín y se toma la textura de las imágenes para obtener una reconstrucción 3D como se muestra en la fig. 14 la cual muestra un tramo del tubo.

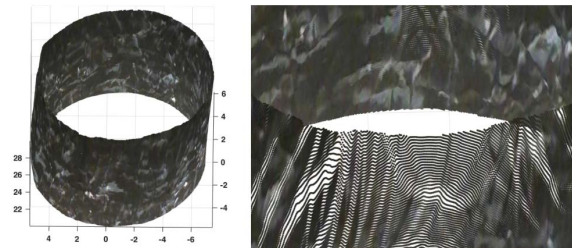


Figura 14. Reconstrucción 3D de un tramo del tubo.

Como paso siguiente se evaluó el algoritmo en ambientes distintos. Los escenarios se muestran en la fig. 15, empleando diferentes casos que incluye imágenes de tuberías (4-6). La evaluación se llevó a cabo utilizando la biblioteca VLFeat [18] y OpenCV [19] por medio de Matlab con el fin de encontrar coincidencias en las imágenes.

En [4] Realizaron la comparación de rendimiento de diversos descriptores del estado del arte, encontrando que el SIFT y el GLOH tuvieron el mejor rendimiento.



Figura 15. Imágenes ejemplo para la evaluación de detectores.

En los experimentos realizados en este trabajo se utilizaron

los descriptores SIFT y los propios de los detectores binarios empleados cuyos resultados se muestran en la fig. 16 acomodados de menor a mayor tiempo de procesamiento. Es importante mencionar que, aunque los detectores binarios resultaron en tiempos cortos de procesamiento, el número de correspondencias no resultan ser suficientes para una adecuada reconstrucción como se muestra más adelante.

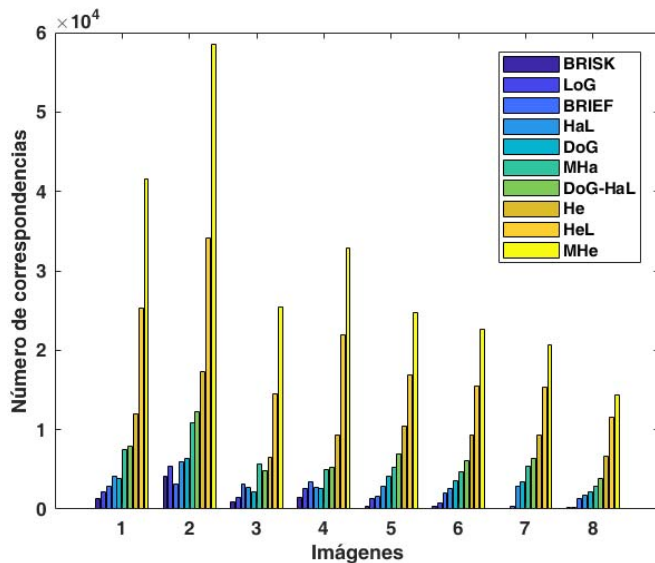


Figura 16. Numero de correspondencias empleando diferentes detectores.

Respecto a las correspondencias de los puntos característicos de las imágenes en el método, no sólo el tiempo de procesamiento se consideró, sino también el número de coincidencias y la ubicación de las correspondencias. Usando DoG en conjunto con Harris Laplace resultó la mejor combinación para los escenarios deseados. Como se puede observar en la fig. 17, el detector Multiescala Hessiano para las imágenes 1 al 4 es superior en el número de correspondencias obtenidas por el método, pero en la Tabla 1 se muestra los tiempos de procesamiento, los cuales son muy superiores al DoG-Harris Laplace. Por otro lado, están los binarios que tienen menor tiempo de procesamiento, de los cuales el BRISK para las imágenes 5 al 8 son menores a 100 correspondencias incluso sin correspondencias en la imagen 7. Para el detector DoG-Harris Laplace se obtuvo el mayor número de correspondencias sobre las imágenes, especialmente de la imagen 5 a la imagen 8 (imágenes 5 y 6 corresponden a tuberías).

La diferencia entre el método propuesto y otros enfoques similares como el RANSAC, es que RANSAC es un algoritmo para estimar los parámetros de un modelo mediante muestreo de datos aleatoriamente dado un conjunto de datos cuyos elementos contienen tanto inliers como outliers. RANSAC utiliza el esquema de votación para encontrar el resultado de ajuste óptimo, lo que significa que tratará de encajar la mayoría de los puntos coincidentes en el modelo, y eliminar el resto [Fischler y Bolles]. El método propuesto normaliza localmente las distancias de correspondencia por lo que no es necesario modificar el rango para aceptar correspondencias

como inliers.

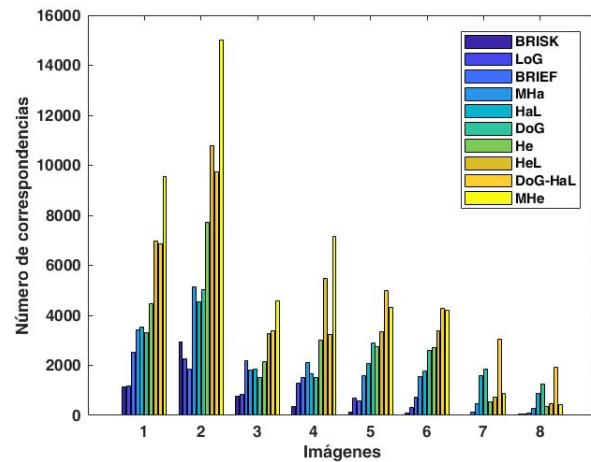


Figura 17. Numero de correspondencias empleando diferentes detectores.

Tabla 1. Tiempo promedio empleando diferentes detectores.

| Detector | Tiempo (minutos) |
|---------------------|------------------|
| BRISK | 0.6 |
| BRIEF | 1.1 |
| DoG | 1.9 |
| Harris Laplace | 2.9 |
| LoG | 4.5 |
| Harris Multiescala | 5.1 |
| DoG-Harris Laplace | 8.2 |
| Hessian | 14.2 |
| Hessian Laplace | 70.2 |
| Hessian Multiescala | 165.7 |

Teniendo en cuenta que el objetivo es reconstruir el tubo interior mientras el robot se mueve en línea recta, la única información útil es la que proviene del área más cercana a la cámara, ya que la información proveniente de la parte más profunda del tubo es innecesaria. Esta información se eliminó de modo que sólo se utilizan los parches de anillo (fig. 14). De esta manera, se pueden unir múltiples imágenes y reconstruir la pared interior del tubo en 3D, como en la fig. 18 la cual muestra un tramo de tubo de 3 metros.

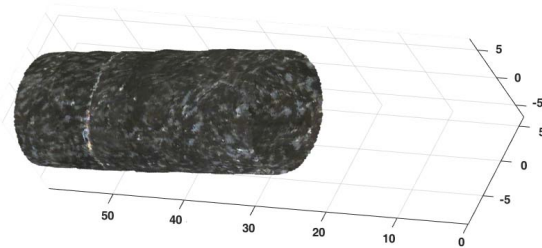


Figura 18 Reconstrucción de un tubo de 3 metros.

V. CONCLUSIONES.

El algoritmo propuesto es capaz como se muestra en este artículo, de obtener correspondencias densas a partir de correspondencias locales, dichas correspondencias son suficientes para generar un mayor número de regiones

alrededor de las ya existentes para rellenar los huecos que pueda llegar a generar. Estas regiones nuevas se pueden obtener a partir de las características geométricas de las regiones adyacentes.

La reconstrucción del escenario se puede realizar, como se demuestra en los resultados iniciales dados en este artículo, por medio de las correspondencias generadas ya que dichas correspondencias son suficientemente buenas para describir el ambiente.

AGRADECIMIENTOS.

Los autores agradecen el apoyo del Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca otorgada a A Reyes-Acosta para realizar sus estudios doctorales, así como del apoyo al proyecto INFR-2014-02 230761.

REFERENCIAS.

- [1] D. Trung. Technical Report: A Review of 3D Reconstruction from Video Sequences. ISIS group, Universiteit van Amsterdam 2005.
- [2] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A Comparison of Affine Region Detectors. *International Journal on Computer Vision*, 65(1/2), pp.43-72, november 2005.
- [3] T. Tuytelaars, K. Mikolajczyk. Local Invariant Feature Detectors: A Survey. *Foundations and Trends in Computer Graphics and Vision*, Vol. 3, nb 3, pp 177-280, 2008
- [4] K. Mikolajczyk and C. Schmid. 2005. A Performance Evaluation of Local Descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 10 (October 2005), 1615-1630.
- [5] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, November 2004.
- [6] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. SIFT flow: dense correspondence across different scenes. *ECCV 2008*.
- [7] C. Liu, J. Yuen and A. Torralba. SIFT flow: dense correspondence across different scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, Vol 33, No.5, 2011.
- [8] M. Tau, T. Hassner. Dense Correspondences Across Scenes and Scales. *CoRR abs/1406.6323*, 2014.
- [9] S. Obdrzalek, M. Perdoch, J. Matas. Dense linear-time correspondences for tracking. in *Workshop on Visual Localization for Mobile Platforms. CVPR 2008*, Jun. 2008.
- [10] T. Tuytelaars and L. Van Gool. Wide Baseline Stereo based on Local, Affinely invariant Regions. *British Machine Vision Conference*, pp. 412-422, 2000.
- [11] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets, or 'How do I organize my holiday snaps?'," in *Computer Vision—ECCV 2002*.
- [12] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints. *International Journal of Computer Vision*, vol. 66, no. 3, March 2006, pp. 231-259.
- [13] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous Object Recognition and Segmentation from Single or Multiple Model Views. *International Journal of Computer Vision (IJCV)*, April 2006.
- [14] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*, Second edition, 2004, publisher Cambridge University Press, ISBN: 0521540518.
- [15] A Simple 3D Scanner Based On Passive Vision for Geometry Reconstruction. A. Narváez, E. Ramírez. *IEEE Latin America Transactions*, Vol. 10, No. 5, September 2012.
- [16] M. Zambrano, C. Medina and E. Galagarza. Target Localization and Reconstruction Using Compressive Sampling. *IEEE Latin America Transactions*, Vol. 13, No. 2, Febrero 2012.
- [17] Herbert Bay, Andreas Ess, Tinne Tuytelaars and Luc Van Gool. SURF:Speeded Up Robust Features. *Computer Vision and Image Understanding (CVIU)*. Vol. 110, No. 3, pp. 346–359, 2008
- [18] A Vedaldi, B Fulkerson. VLFeat: An open and portable library of computer vision algorithms. *Proceedings of the 18th ACM international conference on Multimedia*, 1469-1472
- [19] <https://github.com/opencv/opencv>. visitado 17 sept, 2018.



Alfredo Reyes-Acosta obtained a BEng in Mechatronics Engineering from Instituto Tecnológico de Saltillo in 2006. He received a Master degree in Robotics and Advanced Manufacturing from CINVESTAV-Saltillo in 2008. Currently, He is pursuing his PhD studies at Cinvestav. His research interests are in computer vision applications focused in the area of Robotics and Manufacturing.



Ismael López-Juárez obtained a BEng in Mechanical and Electrical Engineering from National Autonomous University of Mexico in 1991. He received a Master degree in Instrument Design and Application from UMIST in 1996, and the PhD in Intelligent Robotics from Nottingham Trent University in 2000. He was founder and leader of *Grupo de Investigación en Mecatrónica y Sistemas Inteligentes de Manufactura* in CIATEQ, A.C (2000-2006). Currently, he is principal researcher in the Research Group on Robotics and Advanced Manufacturing in CINVESTAV-Saltillo and SNI member Level II.



Roman Osorio obtain BEng at UNAM, México. He has been engineering consulting in enterprises. He has published more than 100 papers in several fields. His research lines are mMobile Robots, Automation, Artificial Vision, and software development. He is researcher at Departamento de Sistemas de Computación y Automatización del Instituto de Investigación de Matemáticas Aplicadas y Sistemas (IIMAS-UNAM) and Electronics Professor.



Gaston Lefranc obtained a BEng in Electrical Engineering at Universidad Técnica Federico Santa María en Chile, Ingeniería Civil Electricista at Universidad de Chile (1976), Mater of Sciece at Northwestern University at Evanston, Illinois, USA. He has been founder of ACC, Asociación Chilena de Control Automático, being President. He has been IEEE Cono Sur Council and four time IEEE Chile Section Chair. He is Full Professor in Escuela de Ingeniería Eléctrica at Pontificia Universidad Católica de Valparaíso, Chile. His research fields are Artificial Intelligence, Automatic Control, Robotics, Flexible Manufacturing Systems FMS and Biomedical Engineering. He has been co-author of 10 books, more than 200 papers in Conferences and Journals. He is IEEE Senior Member.