

DETECCIÓN DE DEFORMACIONES 3D CALCULANDO ESQUELETOS DE CURVATURAS

Carlos M. Mateo¹, Pablo Gil², Fernando Torres²

¹Instituto Universitario de Investigación Informática.

²Dpto. Física, Ingeniería de Sistemas y Teoría de la Señal.

Universidad de Alicante

{cm.mateo,pablo.gil,fernando.torres}@ua.es

Resumen

El artículo presenta un método de análisis 3D para medir la variación en superficies planas. El método está orientado a la detección de deformaciones elásticas en objetos formados por superficies planas. Estas deformaciones son causadas habitualmente cuando dos cuerpos, uno sólido y otro elástico, intersectan, generando una presión de contacto entre sus caras. El método describe una estrategia para modelar la forma de las deformaciones, usando una aproximación matemática basada principalmente en dos conceptos: Histograma y Mapa de Curvatura. En concreto se describe el algoritmo encargado de la detección de deformaciones, con el objetivo de realizar tareas de control visual e inspecciones para el proceso de manipulado con manos robóticas. Los resultados de varios experimentos se muestran de tal forma que se logra evaluar la validez y robustez del método para detectar y medir deformaciones en tareas de agarre. Para realizar estos experimentos, varias escenas virtuales fueron creadas con el fin de simular el contacto entre los dedos de una mano robot y un objeto elástico.

Palabras clave: Formas 3D, Superficie, Curvatura, Normales a la superficie, Modelado geométrico, Percepción visual de deformaciones, esqueletizado.

1. INTRODUCCIÓN

Por lo general, la investigación sobre el manipulado robotizado ha estado centrada en el reconocimiento de objetos rígidos como es el caso de [15]. Pero en los últimos años, el proceso de manipulado ha cambiado sus objetivos para abarcar tareas más complejas, de tal forma que se requiere reconocer objetos articulados [5], objetos deformables [4], objetos blandos [1], objetos semi-sólidos [3] o tejidos [14]. Por lo general, tres tipos básicos de objetos pueden ser considerados: sólidos, elásticos y deformables.

El coeficiente de flexibilidad del objeto puede ser medido matemáticamente a partir de tres concep-

tos características: la rigidez, la cual depende de la fuerza y el tamaño del área donde es aplicada; la dureza, la cual define la fuerza requerida para penetrar el material; y la resistencia, la cual es el total de energía que un material puede tolerar antes de que este se fracture.

Para realizar el control de procesos de agarre se suele usar un control táctil/fuerza [7]. Por lo que, una mano robot capaz de realizar tareas de manipulado de objetos, debe de realizar el control de las fuerzas que se aplican, estas fuerzas suelen ser lo suficientemente pequeñas como para no romper la estructura de la superficie o para perforar la misma. Como regla, las fuerzas nunca exceden un valor que puede causar una ruptura o penetración del objeto manipulado. Información adicional es necesaria para controlar procesos de manipulación si se desea controlar y medir las deformaciones en objetos no rígidos [6]. En estas situaciones, la información táctil/fuerza no es suficiente o es ambigua para detectar y analizar deformaciones en un objeto el cual está siendo manipulado [13]. A diferencia de [2], el objetivo de este artículo es modelar e identificar las deformaciones en objetos semi-sólidos con propiedades elásticas [8], desde sensores sin información de contacto. Por lo tanto, los sensores visuales pueden ayudar a otros sensores [17, 9]. En estos trabajos primero construyen un modelado geométrico de los objetos, para después identificar las deformaciones de ellos comparando entre los puntos de la superficie con y sin deformación.

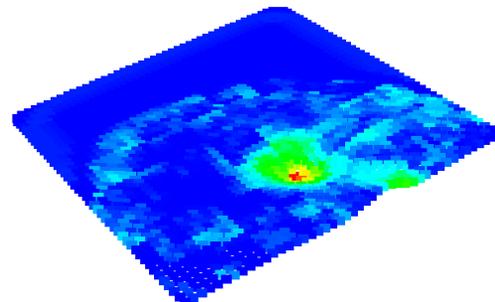


Figura 1: Conjuntos de curvas de nivel calculados desde una superficie con deformaciones

Como consecuencia, los objetos elásticos pueden ser interpretados simplemente como un objeto rígido, puesto que no hay penetración de la superficie o la ruptura de esta. El módulo elástico a menudo se ha utilizado para medir las propiedades de rigidez en el estudio de los materiales cuando se conoce a priori. El módulo elástico mide la fuerza aplicada por unidad de área para deformar la superficie de un objeto. Sin embargo, este no puede ser usado cuando los objetos están compuestos de un material desconocido. En este artículo, la rigidez puede ser medida a través de la curvatura descrita por los puntos de la superficie subyacente a la geometría del objeto. Las características de la curvatura serán calculadas a partir de los puntos de la malla, los cuales modelan la superficie del objeto.

El artículo está estructurado de la siguiente manera: En la sección 2, se introduce el concepto de curvatura a partir de geometría diferencial. En la sección 3 se presenta el método, basado en la esqueletización de variaciones de superficie para modelar, detectar y medir deformaciones en ellas. Los resultados experimentales de las deformaciones causadas por la simulación de tareas de manipulado robotizado se muestran en la sección 4. Finalmente, la sección 5 contiene las conclusiones del trabajo.

2. CÁLCULO DE CURVATURAS

En este artículo, la geometría diferencial de curvas se ha usado como una herramienta que nos permite analizar la superficie 3D (Fig 1). Normalmente, en visión por computador, la superficie de un objeto 3D consiste en una nube de puntos no-estructurada representada como $\mathbf{P} = \{\mathbf{p}_i \in \mathbb{R}^3\}$. Si un objeto plano en el espacio Euclídeo es deformado por la aplicación de fuerza en su superficie exterior, las propiedades geométricas cambian, apareciendo ciertas curvatura en ella. De ahora en adelante, la curvatura se entiende como la variación de la superficie, y ella puede ser calculada midiendo los cambios de orientación de los vectores normales a los puntos de la superficie.

Por lo general, cada punto \mathbf{p}_i satisface un conjunto de axiomas en relación con su entorno de vecindad. Entonces, el cálculo de las propiedades geométricas de las curvas depende de los k puntos vecinos más cercanos a cada \mathbf{p}_{ij} . Por lo tanto, \mathbf{P} puede ser agrupado como un conjunto de regiones \mathbf{N}_j y cada uno es un subconjunto de puntos centrados en \mathbf{p}_{ij} . Tanto el tamaño (radio) como el número de puntos de cada región (densidad) influyen en la precisión para calcular las propiedades geométricas de la curva. Esto hace que grupos con pocos puntos causen imprecisiones en el cálculo, y mu-

chos puntos distorsionen los valores, dificultando la detección de cambios de orientación en la superficie, o lo que es lo mismo, suavicen la detección. Esto hace importante encontrar un equilibrio en la densidad de puntos dentro de las regiones \mathbf{N}_j .

Para estimar las propiedades geométricas locales de las regiones de la superficie se utiliza un análisis de los valores propios extraídos desde la matriz de covarianza de los puntos dentro del entorno de vecindad [16]. Para definir la matriz de covarianza se aplica un análisis de las componentes principales (PCA) como sigue:

$$\mathbf{C}_P = \mathbf{P}\mathbf{P}^T = \begin{bmatrix} \mathbf{p}_{i1} - \bar{\mathbf{p}} \\ \dots \\ \mathbf{p}_{ik} - \bar{\mathbf{p}} \end{bmatrix} \begin{bmatrix} \mathbf{p}_{i1} - \bar{\mathbf{p}} \\ \dots \\ \mathbf{p}_{ik} - \bar{\mathbf{p}} \end{bmatrix}^T \quad (1)$$

donde \mathbf{p}_{ij} es un punto dentro del entorno de vecindad \mathbf{N}_j y $\bar{\mathbf{p}}$ es el punto central de la región. Además, k define el numero de puntos dentro de \mathbf{N}_j .

La ecuación (1) se resuelve usando el método Turk and Pentland [18] aplicando descomposición de valores singulares (SVD). Así, se obtienen los valores propios λ_j , y los vectores propios \mathbf{v}_j de \mathbf{C}_P tal que:

$$\mathbf{P}^T \mathbf{P} \cdot \mathbf{v}_j = \lambda_j \cdot \mathbf{P} \mathbf{v}_j \quad (2)$$

La suma de los valores propios provee información sobre la variación de la superficie entre cada punto de los puntos del entorno \mathbf{N}_j y su punto central. Además, el menor valor entre los valores propios proporciona una medida de la variación entre el vector normal y la superficie. Por lo tanto, los valores propios pueden ayudar a clasificar la concavidad de la superficie de cada punto, y el conjunto de todos los parámetros que definen la curvatura. Además, la curvatura máxima local [12] de \mathbf{p}_{ij} dentro de \mathbf{N}_j pueden ser calculadas por:

$$c_{ij} = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \quad (3)$$

donde $\lambda_0 \leq \lambda_1 \leq \lambda_2$ son los valores propios de \mathbf{C}_P . El autovector \mathbf{v}_0 asociado a λ_0 es el vector normal al plano tangente del entorno de la superficie \mathbf{P} que define \mathbf{v}_1 y \mathbf{v}_2 . El conjunto de las curvas de nivel está definida por la función $\mathbf{S}_P : \mathbb{R}^3 \rightarrow \mathbb{R}$ como sigue:

$$\mathbf{S}_P(\Phi) = \{(x, y, z) \in \mathbb{R}^3 \mid \Phi(x, y, z) = l\} \quad (4)$$

donde l es un valor constante y representa una curva de nivel en la superficie. Con el objetivo de representar las curvas de nivel, todas ellas están agrupadas y representadas con el mismo color, el cual representan los puntos de \mathbf{P} con valores similares de curvatura (Fig. 2b).

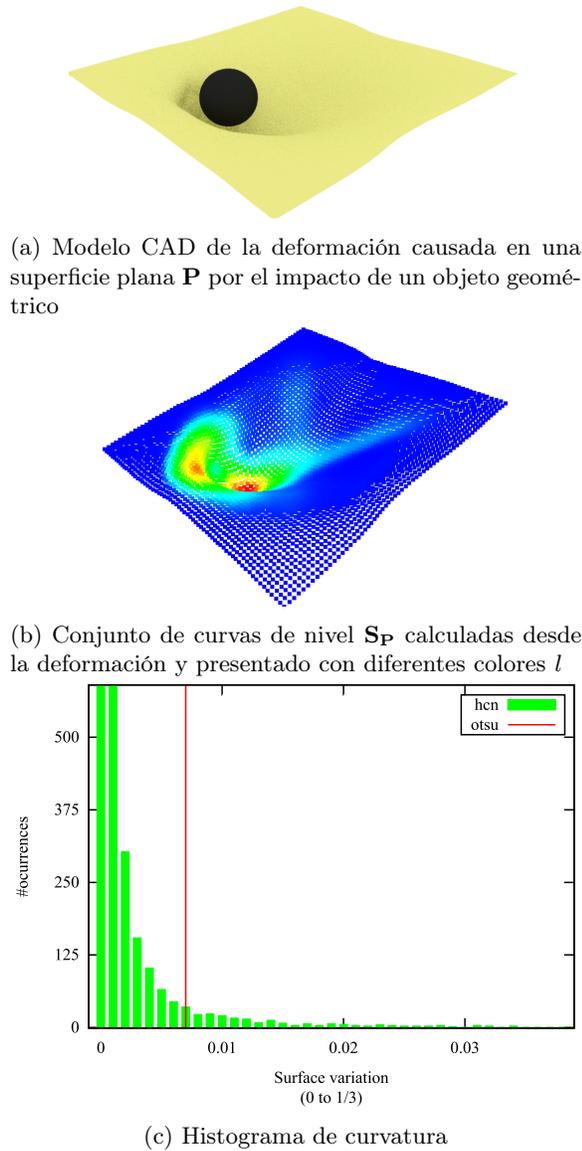


Figura 2: Representación de la información a partir del modelado de una deformación

3. MÉTODO DE ESQUELETIZADO DE VARIACIONES DE SUPERFICIES

En este trabajo, las superficies están representadas como nubes de puntos. Vale la pena señalar que los datos de \mathbf{P} pueden ser obtenidos desde una nube de puntos estructurada o no-estructurada. \mathbf{P} es una nube estructurada si esta se encuentra almacenada en un matriz, tal que $\mathbf{P}_{X \times Y}$ donde X e Y son el número de filas y columnas, respectivamente. Como norma, esto ocurre si \mathbf{P} es adquirida desde un sensor, como podría ser un sensor RGBD o ToF, donde cada punto corresponde con una posición del sensor. Por otro lado, \mathbf{P} se considera no-estructurada si esta se encuentra alma-

cenada como un vector matriz, tal que la nube de puntos se encuentra como $\mathbf{P}_{X \times 1}$. En este último caso, \mathbf{P} no tiene una relación directa con los datos de la imagen de rango. \mathbf{P} se presenta normalmente como no-estructurada cuando está construida (adquirida) desde un modelo CAD (Fig. 2a). Por lo tanto, con el fin de buscar los puntos vecinos a cada punto de \mathbf{P} se suelen utilizar estructuras de ordenado de datos en forma de árbol (árbol-kd o árbol-octal).

El algoritmo propuesto consta de dos fases: Inicialización y extracción de curvaturas. La primera fase solo se calcula cuando \mathbf{P} es una nube no-estructurada, puesto que necesita estar estructurada para poder realizar la segunda parte del algoritmo. Esto es, los puntos más cercanos en el espacio Euclídeo deben ser almacenados como vecinos dentro de \mathbf{P} . De modo que, es posible buscar puntos con las mismas propiedades geométricas y localizados en el mismo entorno de vecindad. Esto es esencial para calcular las curvas de nivel desde los parámetros de curvatura y para detectar las variaciones de superficie dadas por el camino transversal entre las curvas de nivel.

El método propuesto detecta deformaciones de cualquier superficie representada como \mathbf{P} . Está basado en la búsqueda de puntos críticos de \mathbf{P} en los cuales se encuentran variaciones de superficie. En este trabajo, los puntos críticos son puntos pertenecientes a las diferentes curvas de nivel Φ de la superficie, pero además yacen en el camino transversal que cruza las curvas de nivel por medio del ajuste de puntos singulares. Los puntos singulares están definidos como los puntos con máximo valor de curvatura, y cuyo valor de curvatura ha sido estimado a partir de un umbral de curvatura extraído desde el histograma de curvatura.

3.1. HISTOGRAMA DE CURVATURAS

Una vez los parámetros de curvatura han sido calculados, de acuerdo con (3), para todos los puntos dentro de \mathbf{P} (Fig. 2b), se construye un histograma de curvatura (Fig. 2c). Esto representa la distribución de la variación de la superficie y puede ser calculada como:

$$H_{\mathbf{P}} = \frac{\text{number}(c_{ij})}{\text{size}(\mathbf{P})} \quad (5)$$

donde c_{ij} es el valor de curvatura, y $\text{size}(\mathbf{P})$ representa la densidad o número de puntos usados para muestrear la superficie. $H_{\mathbf{P}}$ cambia dependiendo del tamaño del entorno de vecindad \mathbf{N}_j usado para calcular cada c_{ij} . El usuario debe escoger el radio de acuerdo a la precisión para la correcta detección de cambios abruptos en la superficie.

El histograma $H_{\mathbf{P}}$ resulta de gran utilidad para

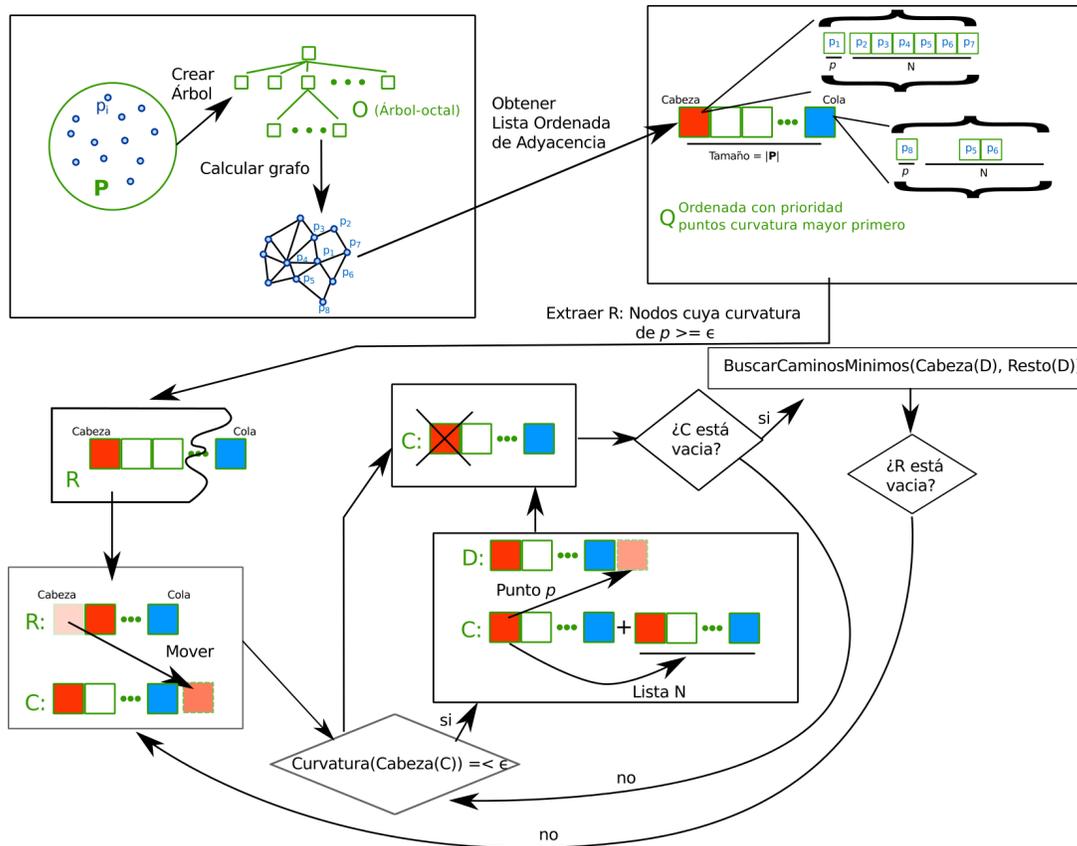


Figura 3: Algoritmo para la caracterización de las deformaciones mediante esqueleto de curvaturas

realizar la búsqueda de los puntos singulares. Tanto los puntos con $\max(c_{ij})$ como los puntos límites que define el borde donde no hay variación de curvatura en la superficie. Los puntos límites separan la región de curvatura de la región sin curvatura. Ellos están calculados desde H_P usando la técnica presentada en [10]. El histograma nos permite buscar el umbral de curvatura ϵ_{otsu} por medio de la minimización de la desviación estándar de la distribución Gaussiana que representan las dos zonas (Fig 2c).

3.2. ESQUELETIZADO DE VARIACIÓN DE SUPERFICIE

Asumimos que la superficie es siempre identificable, entonces los gradientes de la superficie en cada punto es o bien cero, o bien perpendicular al nivel de curva el cual representa la variación de la superficie a cada punto. El método inspecciona los puntos críticos de la función de la superficie a lo largo de sus curvas de nivel. Los puntos críticos son aquellos de máxima curvatura los cuales se encuentran en el centro de la deformación.

El algoritmo propuesto se usa para buscar tanto puntos críticos como los puntos límite, y el camino transversal el cual define la deformación en

la superficie. El algoritmo usa un árbol-octal O como [11] con el objetivo de almacenar todos los puntos \mathbf{p} de \mathbf{P} como una estructura ordenada, donde cada punto \mathbf{p} se encuentra almacenado en un nodo hoja (los nodos que se encuentran en el nivel más profundo del árbol) del árbol. Después, creamos una cola de prioridad Q con el par $\{p, N\}$, donde p es cada punto de O y N es su lista de adyacencia. N representa los puntos de vecindad con una conexión de 26-conectado, lo que es lo mismo, sus 26 directos puntos vecinos (si los tuviera). En la cabeza de la cola de prioridad Q se encuentran los puntos p con el mayor valor de curvatura.

Más tarde, Q es filtrada y almacenada en R para obtener solo los valores de los elementos p de Q que son más grandes que ϵ_{otsu} . Siguiendo el diagrama de flujo mostrado en la Fig 3 se observa que el algoritmo genera una lista D que almacena los puntos críticos en la cabeza siguiendo con los puntos que se encuentran dentro de las curvas de nivel. Para finalizar el algoritmo calcula todos los caminos mínimos (utilizando el algoritmo Dijkstra) desde la cabeza de la lista D al resto de puntos en dicha lista, siempre y cuando estos estén dentro de un rango de curvatura, tal que, $\epsilon - \alpha \leq curvatura \leq \epsilon + \alpha$. Donde α es un error permitido. Se observa que el punto en la cabeza

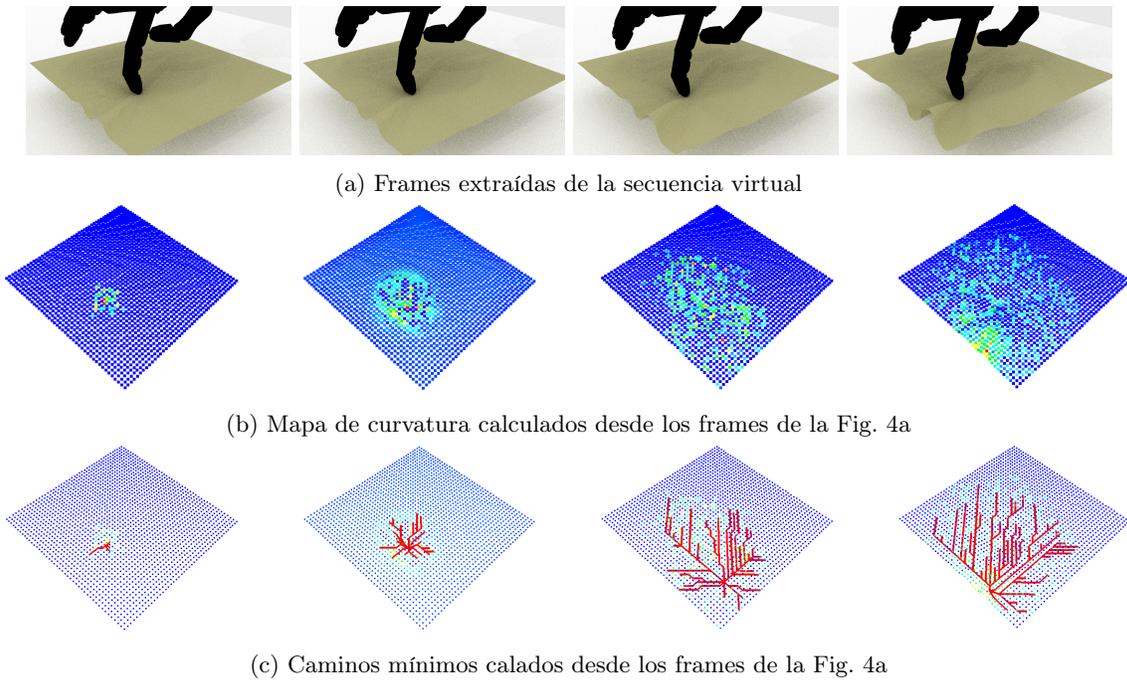


Figura 4: Evolución temporal de la deformación de una superficie plana causada por el contacto de un dedo robot desde dos secuencias de movimientos distintas de la mano robot

de la lista D es el punto con curvatura máxima, y el resto de puntos en D , que cumple la condición, son los puntos del conjunto de los puntos límites. El algoritmo se ejecuta iterativamente hasta encontrar todos las deformaciones en el plano P .

4. EXPERIMENTACIÓN

Diversos experimentos y pruebas han sido realizadas. Para ello hemos modelado los movimientos de una mano robot real considerando tanto la cinemática como el modelo virtual (forma y estructura) de sus dedos y palma sin las restricciones físicas ni considerando las singularidades, por medio del uso del software libre Blender. En particular el modelo de la mano robot corresponde a la mano robot de la empresa Shadow Robot disponible en nuestro laboratorio de investigación.

Nuestros experimentos simulan varias deformaciones causadas por el contacto entre los dedos de la mano virtual de la mano robot y un objeto plano. Las deformaciones dependen de la orientación de los dedos, la trayectoria usada en el proceso de contacto, la forma de los dedos y la presión producida por estos en el momento del contacto (Fig. 4).

Cada experimento consiste en un movimiento reproducido en una secuencia de imágenes. Cada secuencia esta compuesta por 60 imágenes. El algoritmo calcula los conjuntos de curvas de nivel a partir de los parámetros de curvatura y los histogramas de curvatura. Esto nos permiten obtener

la evolución de las curvaturas durante un tiempo secuencia a través del proceso de formación de las deformaciones (Fig. 4 y 5). Después, estimamos los caminos mínimos que recorren las deformaciones desde los puntos críticos hasta los puntos límites para medir la variación de la superficie y el perfil topográfico de la deformación (Fig. 4 y 6).

El algoritmo está implementado en el lenguaje de programación C/C++ y usando las librerías de código abierto PCL, Boost y Eigen. Ha sido ejecutado sobre un ordenador equipado con un procesador Intel Core i7-4770k, con 8GB de memoria RAM y una tarjeta gráfica Nvidia GeForce 760GTX.

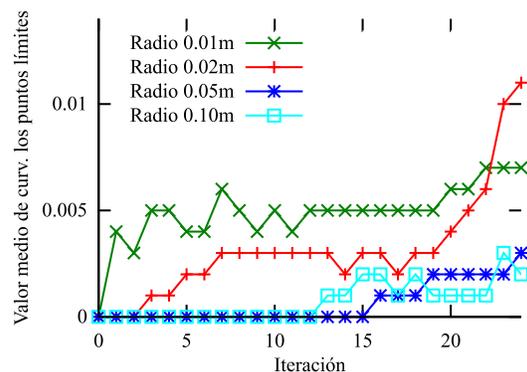


Figura 5: Evolución del valor de la curvatura a través de la secuencia de los movimientos mostrada en la Fig 4

La Fig. 5 muestra la dependencia de la medida de la curvatura con respecto al radio de vecindad seleccionado para calcular los vectores normales a los puntos de la superficie. Si el radio es lo suficientemente grande los pequeños concavidades en la superficie no serán detectados por el algoritmo. La Fig. 5 muestra la evolución de los puntos límites y como crecen rápidamente en las primeras iteraciones. Este hecho indica que el contacto entre la mano robot y la superficie elástica está ocurriendo en ese momento. Después, el crecimiento se suaviza y se hace más progresivo. La pendiente de la curva de las líneas del esqueleto indica el nivel de deformación generado durante el momento del agarre. La Fig. 6 muestra como el mínimo camino transversal para los esqueletos aporta información para determinar la deformación en un instante de tiempo. En la prueba ilustrada en la Fig 4, se ha seleccionado el frame numero 49 para estudiar. De los 200 caminos generados por el algoritmo, solo se representa 1 de cada 20 por claridad. Podemos observar que a excepción de los caminos 40, 80 y 160, son caminos sin una gran pendiente, pero con una gran longitud, esto indica que se trata de una deformación suave y muy progresiva. Por otro lado vemos que los pocos caminos que si que tiene una gran pendiente pero un corto recorrido, se corresponden a la parte de la deformación que está sufriendo más estrés, la parte de intersección entre el dedo y el plano.

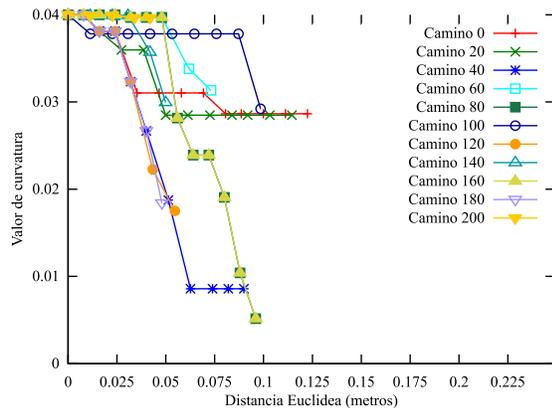


Figura 6: Caracterización de la variación del camino transversal calculado desde los puntos críticos desde el frame 49 de la prueba (en la Fig 4 corresponde a la tercera columna)

5. CONCLUSIONES

Este artículo ha descrito una aproximación novel al análisis de la deformación de superficies planas. El método propuesto construye mapas de variación de curvaturas y esqueletos para medir el desnivel local en la superficie por medio de la comparación entre los puntos que yacen en la misma

superficie. Además, el método usa un algoritmo novel para caracterizar y describir la variación de curvas de nivel causada por las deformaciones. Para realizar esto, nuestro algoritmo identifica las curvas de nivel y después, busca los puntos críticos los cuales junto con los puntos límites definen el camino transversal entre las curvas de nivel. Lo que significa que, generamos un perfil topográfico en la dirección de los máximos gradientes locales.

Este análisis de superficie proporciona una investigación empírica sobre el efecto de las deformaciones sobre la superficie de objetos elásticos cuando se realizan varias tareas de agarre con una mano robot. Y lo más importante, se presenta la implementación de un algoritmo para medir estas deformaciones. Este algoritmo ha sido programado en el lenguaje de programación C/C++, y puede ser integrado en una plataforma robótica.

Agradecimientos

La investigación que nos ha llevado a estos resultados ha sido financiada por el Gobierno Español y los fondos europeos FEDER (DPI2012-32390) y el Gobierno Regional de Valencia (PROMETEO/2013/085).

Referencias

- [1] Berenson, D. (2013). Manipulation of deformable objects without modeling and simulating deformation. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 4525–4532. IEEE.
- [2] Boonvisut, P. and Çavuşoğlu, M. C. (2014). Identification and active exploration of deformable object boundary constraints through robotic manipulation. *The International journal of robotics research*, 33(11):1446–1461.
- [3] Gemici, M. C. and Saxena, A. (2014). Learning haptic representation for manipulating deformable food objects. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 638–645. IEEE.
- [4] Hirai, S., Tsuboi, T., and Wada, T. (2001). Robust grasping manipulation of deformable objects. In *Proc. IEEE Symposium on Assembly and Task Planning*, pages 411–416.
- [5] Katz, D. and Brock, O. (2008). Manipulating articulated objects with interactive perception. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 272–277. IEEE.
- [6] Khalil, F. F., Curtis, P., and Payeur, P. (2010). Visual monitoring of surface deformations on objects manipulated with a robotic hand. In *Robotic and Sensors Environments (ROSE), 2010*

- IEEE International Workshop on*, pages 1–6. IEEE.
- [7] Khalil, F. F. and Payeur, P. (2010). *Dexterous robotic manipulation of deformable objects with multi-sensory feedback-a review*. INTECH Open Access Publisher.
- [8] Mateo, C. M., Gil, P., et al. (2015). Analysis of shapes to measure surfaces: An approach for detection of deformations. In *Informatics in Control, Automation and Robotics (ICINCO), 2015 12th International Conference on*. IEEE.
- [9] Mkhitarian, A. and Burschka, D. (2012). Vision based haptic multisensor for manipulation of soft, fragile objects. In *Sensors, 2012 IEEE*, pages 1–4. IEEE.
- [10] Otsu, N. (1975). A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27.
- [11] Papon, J., Abramov, A., Schoeler, M., and Worgotter, F. (2013). Voxel cloud connectivity segmentation-supervoxels for point clouds. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2027–2034. IEEE.
- [12] Pauly, M., Gross, M., and Kobbelt, L. P. (2002). Efficient simplification of point-sampled surfaces. In *Proceedings of the conference on Visualization'02*, pages 163–170. IEEE Computer Society.
- [13] Pomares, J., Gil, P., García, G. J., Sebastián, J. M., and Torres, F. (2008). Improving detection of surface discontinuities in visual-force control systems. *Image and Vision Computing*, 26(10):1435–1447.
- [14] Ramisa, A., Alenya, G., Moreno-Noguer, F., and Torras, C. (2013). Finddd: A fast 3d descriptor to characterize textiles for robot manipulation. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 824–830. IEEE.
- [15] Saxena, A., Driemeyer, J., and Ng, A. Y. (2008). Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2):157–173.
- [16] Shaffer, E. and Garland, M. (2001). Efficient adaptive simplification of massive meshes. In *Proceedings of the conference on Visualization'01*, pages 127–134. IEEE Computer Society.
- [17] Smith, P. W., Nandhakumar, N., et al. (1996). Vision based manipulation of non-rigid objects. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 4, pages 3191–3196. IEEE.
- [18] Turk, M., Pentland, A. P., et al. (1991). Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 586–591. IEEE.