

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN

ESCUELA DE POSTGRADO

UNIDAD DE POST-GRADO DE LA FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS.



ANÁLISIS DE CUERPOS NO RÍGIDOS USANDO ALGORITMOS DE DIFUSIÓN PARA LA DETECCIÓN DE SIMETRÍA EN MODELOS 3D

Tesis presentada por el Magister
Cristian José López Del Alamo

Para optar el grado de Doctor en ciencias
de la computación.

Asesor:
Dr. César Beltrán Castañón.

Arequipa – Perú

2014

Análisis de cuerpos no rígidos usando algoritmos de difusión para la detección de simetría en modelos 3D

TESIS PARA OPTAR EL GRADO DE DOCTOR EN CIENCIAS DE
LA COMPUTACIÓN

Mcs. Cristian José López Del Alamo

Escuela Profesional de Ingeniería de Sistemas e Informática

Orientador:

Cesar Beltrán Castañon

Departamento de Ingeniería de Sistemas e Informática
Facultad de Producción y Servicios
Universidad Nacional de San Agustín

Presentado: 22/07/2014

Comité Evaluador:

Profesor: Dr. Ernesto Cuadros Vargas.

Profesor: Dr. Iván Túpac Valdivia.

Profesor: Dr. César Beltrán Castañón.

Copyright © 2014 por Cristian José López Del Alamo.

Resumen:

The aim of this thesis is the development of a new algorithm for detection of symmetry in non-rigid bodies. This involves using 3D models, which are clouds of points, which are interconnected to form a closed mesh of triangles. Said mesh approximates the surface representing the model. Methods were employed for the detection of salient points on the surface of the structure. These points, known as key points, are then used for finding key components within the structure. Finally, these components are used for detection of symmetry in the non-rigid bodies using of the diffusion distance. As part of the thesis, two significant contributions were developed, the first is a new algorithm of linear complexity to the number of triangles, to detect key points or interest points and; the second, is a new algorithm to detect key components or key components based to diffusion processes. Our results showed that both algorithms are invariant to isometric changes, topological, noise and other non-rigid transformations. Furthermore, experiments on detection of symmetry through diffusion processes, no were invariantes in isometric transformations. However, when isometric transformations are not so abrupt, then symmetry detection works correctly.

Resumen:

El objetivo de esta tesis es el desarrollo de un nuevo algoritmo para la detección de simetría en cuerpos no rígidos. Esto implica el uso de modelos 3D, los cuales son nubes de puntos interconectados, formando una malla de triángulos. Dicha malla se aproxima a la superficie que representa el modelo. Se emplean métodos para la detección de puntos salientes en la superficie de la estructura. Estos puntos, conocidos como *key points*, son posteriormente utilizados para encontrar componentes clave (*key components*) dentro de la estructura. Finalmente, estos componentes son utilizados para la detección de simetría en los cuerpos no rígidos mediante el uso de la distancia de difusión. Como parte de la tesis, dos aportes significativos fueron desarrollados, el primero es un nuevo algoritmo de complejidad lineal respecto al número de triángulos, para detectar *key points* o puntos de interés y, el segundo es un nuevo algoritmo para detectar *key components* o componentes clave basado en procesos de difusión. Nuestros resultados mostraron que ambos algoritmos son invariantes a cambios isométricos, topológicos, ruido y otras transformaciones no rígidas. Por otro lado, los experimentos en la detección de simetría por procesos de difusión mostró no ser invariante a cambios isométricos abruptos. Sin embargo, cuando las transformaciones isométricas no son tan pronunciadas, la detección de simetría funciona correctamente.

ÍNDICE

Lista de Figuras	v
Lista de Tablas	vii
1 Introducción	1
1.1 Motivación y contexto	1
1.2 Planteamiento del problema	2
1.3 Objetivo	5
1.4 Aportes	5
2 Conceptos Previos	7
2.1 Análisis de modelos 3D	7
2.1.1 Operador Laplace-Beltrami	7
2.1.2 Heat Kernel Signature	10
2.1.3 Wave Kernel Signature	12
2.2 Funciones de Distancia en modelos 3D	13
2.2.1 Distancia Euclidia	13
2.2.2 Distancia Geodésica	13
2.2.3 Distancia de Difusión	14
2.3 Simetría	15
2.3.1 Simetría Extrínseca	16
2.3.2 Simetría Intrínseca	17
3 Key Points: Minimal area interest point	19
3.1 Trabajos Relacionados	20
3.1.1 Detección de <i>Key Points</i> en Imágenes	20
3.1.2 Detección de <i>Key Points</i> en Objetos 3D	22
3.1.3 Detección de puntos de interés basados en la escala (HKS)	25
3.2 Propuesta y Metodología	27
3.2.1 Cálculo de áreas	27
3.2.2 Ordenamiento y selección de triángulos con áreas pequeñas	28
3.2.3 Detección de puntos de interés	29
3.3 Experimentos	29
3.3.1 El conjunto de datos	29
3.3.2 Evaluación Metodológica	30
3.3.3 Resultado y comparación con otros métodos	30
3.4 Conclusión	34
4 Detección de <i>Key Components</i> mediante procesos de Difusión	37
4.1 Aspectos Preliminares	37

4.2	Trabajos relacionados	38
4.2.1	Detección de <i>Key Components</i>	39
4.3	Propuesta y Metodología	42
4.3.1	Modelo	42
4.3.2	Mapa HKS/WKS	42
4.3.3	Detección de Componentes	43
4.3.4	Extracción de Componentes	44
4.4	Experimentos	44
4.4.1	El conjunto de datos	45
4.4.2	Criterio de Evaluación	45
4.4.3	Resultados	46
4.5	Conclusiones	51
5	Detección de Simetría en cuerpos no rígidos	53
5.1	Aspectos Preliminares	53
5.2	Trabajos relacionados	54
5.3	Propuesta y Metodología	56
5.3.1	Modelo	57
5.3.2	Detección de <i>Key Components</i>	57
5.3.3	Firmas WKS	58
5.3.4	Correspondencia de Key Components	59
5.3.5	Detección de Simetría	61
5.4	Experimentos	62
5.4.1	El conjunto de datos	62
5.4.2	Experimentos y Resultados	63
6	Conclusiones finales y trabajos futuros	73
	Bibliografía	75

LISTA DE FIGURAS

1.1	Modelos 3D de un cuerpo Humano en diferentes niveles de escala, ruido, y posturas.	2
1.2	Modelos 3D de Cuerpos no Rígidos	2
1.3	Key Componentes	4
2.1	(a) Peso de la arista W_{ab} (b) Peso de Arista en Grafo Regular (c) Peso de Arista en Grafo Irregular	8
2.2	(a) Superficie bidimensional (b) Campo vectorial gradiente (c) Divergente (fuente p y sumidero q)	9
2.3	Función e^{-t}	11
2.4	Disipación del calor en una superficie [ZRHAr]	11
2.5	Distancia euclida en un objeto tridimensional con cambio isométrico	14
2.6	Distancia geodésica en un objeto tridimensional con cambio isométrico	14
2.7	Distancia de difusión en un objeto tridimensional con cambio isométrico	15
2.8	Simetría Extrínseca	16
2.9	Simetría en cuerpo Rígido	17
2.10	Simetría Intrínseca	18
3.1	Mapa de Bits de la imagen de Lena	21
3.2	k ring [SB11]	23
3.3	Malla de triángulos	23
3.4	Rotación de la malla mediante PCA	24
3.5	Discretización de una superficie mediante una malla de triángulos	24
3.6	(a) Forma NULL (b) Cambios isométrico (c) Cambio Topológico	26
3.7	Metodología	27
3.8	Triangulación (tamaño de los triángulos)	28
3.9	Modelos con <i>Key points</i> para diferentes valores de t	28
3.10	Transformación del modelo caballo con un nivel de intensidad de 5 y $t = 20\%$	31
4.1	(a) <i>key points</i> , (b) <i>Key componentes</i> obtenidos a partir de (a) [SB12]	39
4.2	(a) <i>key points</i> , (b) <i>Key componentes</i> obtenidos a partir de (a) [SB12]	40
4.3	(a) <i>key points</i> , (b) <i>Key componentes</i> obtenidos a partir de (a) [SB12]	41
4.4	(a) <i>key points</i> , (b) <i>Key componentes</i> obtenidos a partir de (a) [SB12]	41
4.5	Metodología	42
4.6	<i>N-Eigen</i> vs Número de Eigen Vectores	47
4.7	<i>N-Eigen</i> vs Número de Eigen Vectores	48
4.8	<i>Max-Points</i> vs Key components en modelo 0001.null.0	49
4.9	<i>Max-Points</i> vs Key components en modelo 0002.null.0	49
4.10	<i>Max-Points</i> vs Key components en modelo 0003.null.0	50
4.11	Resultados de correspondencias entre modelo transformado y modelo null	50

5.1	Metodología	56
5.2	Detección de <i>Key components</i>	57
5.3	Firmas de los Key components	58
5.4	Centroides Geodésicos	61
5.5	Correspondencia de Key Points	61
5.6	Firmas de <i>key components</i> detectados con nuestro algoritmo	64
5.7	Centroides Geodésicos	66
5.8	Detección de Simetría mediante distancias geodésica en modelos con simetría extrínseca	67
5.9	Detección de Simetría mediante distancias geodésica en modelos con simetría intrínseca	68
5.10	Detección de Simetría mediante distancias de difusión en modelos con simetría extrínseca	69
5.11	mapa de Voronoi formado por la distancia de difusión en modelo humano con transformaciones isométricas	70
5.12	mapa de Voronoi formado por la distancia de difusión en modelo humano con transformaciones no rígidas	71
5.13	Detección de Simetría en rostros 3D con cambios isométricos	72

LISTA DE TABLAS

3.1	Tipos de superficies que se presenten en una imagen 3D	22
3.2	Repetibilidad del algoritmo de detección de puntos de interés HKS2 [SB11].	31
3.3	Repetibilidad del algoritmo de detección de puntos de interés HKS3 [SB11].	32
3.4	Repetibilidad del algoritmo de detección de puntos de interés H3D [SB11].	33
3.5	Repetibilidad promedio de nuestro método. Número promedio de triángulos con menor área recuperada: 19670.	33
3.6	Método con la mejor performance para transformaciones de modelos e intensidades. HKS1 y HKS2 (<i>Heat Kernel Signature</i>). H3D (<i>H3D-Harris 3D</i>). y MAIP(<i>Minimum Area interest points</i>).	34
4.1	Valores de repetibilidad para Key Components [SB12]	51
4.2	Valores de repetibilidad con HKS	51
4.3	Valores de repetibilidad con <i>WKS</i>	51
5.1	Resultados de correspondencia entre centroides geodésicos	65

PREFACE

Esta tesis fue sometida al departamento de ingeniería de sistemas e informática de la facultad de producción y servicios de la Universidad San Agustín de Arequipa, como requisito para optar el grado de Doctor. El trabajo presentado fue llevado a cabo entre los años 2010-2014 y dirigido por el doctor Cesar Beltrán Castañón, con el apoyo del Doctor Iván Sipiran de la universidad de Chile y Alex Bronsteing de la universidad de Tel Aviv.

Dedicado a la mujer que me dio la vida dos veces, una cuando me permitió nacer y otra cuando me dio parte de su ser para que yo siga vivo. Mi madre.

Esta tesis es el resultado de varios años de aprendizaje, y es fruto de la discusión e intercambio de ideas con varias personas, como el Dr. Alex Bronsteing, Dr. Ivan Sipirán, Dr. Benjamin Bustos, Lic. Wilber Ramos Lovón y mi orientador, el Dr. Cesar Beltrán Castañón, a quienes agradezco profundamente.

Debo agradecer a mi familia, motor indiscutible en mi vida, por quienes he logrado cumplir varias metas propuestas, incluyendo, la culminación de esta tesis. Agradezco también, a mis mejores maestros, mis alumnos, quienes con sus preguntas, inquietudes y ganas de aprender han permitido que yo me supere a mi mismo. Ellos son la razón principal por la cual amo aprender.

Finalmente, agradezco a mis amigos, Dr. Álex Cuadros Vargas, Dr Juan Carlos Gutiérrez Cáceres, Dr. Yvan Tupac y Dr. Ernesto Cuadros Vargas, quienes siempre confiaron en mí e incentivaron mis deseos de superación por más problemas que uno pueda tener.

1

INTRODUCCIÓN

El primer capítulo hace referencia al planteamiento del problema, objetivo, motivación y trabajos previos de la tesis

1.1 Motivación y contexto

En la actualidad, con el avance de la tecnología, se han desarrollado nuevos dispositivos a bajo costo que facilitan la digitalización de objetos 3D, como por ejemplo; láser 3D, escáneres 3D, o herramientas como el *KINET* de Xbox de *Sony*, entre otros. Estos dispositivos permiten obtener, generalmente, una malla de triángulos, la cuales representan la superficie del objeto.

Esta superficie, se puede ver como una variedad de Riemman en un espacio tridimensional; en otras palabras, una superficie bidimensional embebida en un espacio tridimensional y, desde el punto de vista, de una estructura de datos, en un grafo poco denso. En tal sentido, es posible diseñar, implementar y aplicar diferentes algoritmos para trabajar sobre estas estructuras.

Las aplicaciones van desde encontrar características para búsquedas por similitud en objetos 3D [EKG10, DGD⁺11, JZ07, CTSO03], segmentación de objetos 3D [Sha08, ASC11a], detección de simetría [Sch78, MBP13, RBBK10a, XZJ⁺12, XZT⁺09], por mencionar algunas.

La simetría es un fenómeno universal en la naturaleza, la cual está presente en las ciencias y el arte [MPWC13]. En la actualidad, el análisis y procesamiento de simetría ha sido ampliamente estudiado para el caso de imágenes 2D. [AMWW88, Ata85]. Sin embargo, en el caso de objetos 3D, encontrar la simetría es más complejo.

Actualmente, se han desarrollado varios estudios para encontrar simetría en de cuerpos rígidos 3D, es decir, aquellos objetos a los cuales se les puede aplicar funciones de transformación como rotación, desplazamiento y escala. Este tipo de cuerpos presentan una simetría extrínseca. Sin embargo, cuando el objeto 3D sufre transformaciones isométricas, presenta ruido, cambios topológicos, presencia de hoyos en la estructura,

la simetría extrínseca se pierde, lo cual hace más complejo el hecho de encontrar la simetría.

En la figura 1.1, se puede observar los modelos de un cuerpo humano, con diferentes niveles de escala, ruido y posturas.

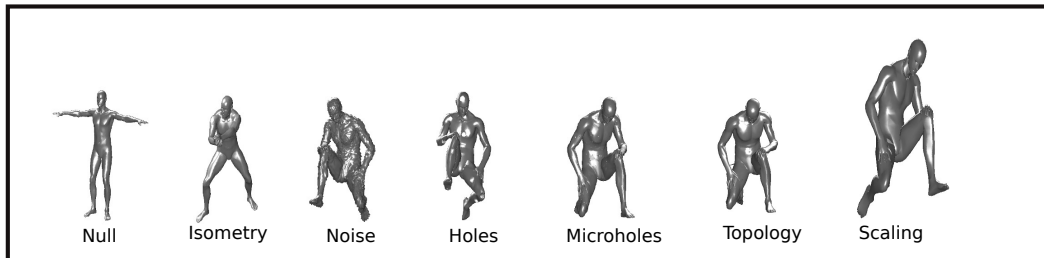


Figura 1.1: Modelos 3D de un cuerpo Humano en diferentes niveles de escala, ruido, y posturas.

Aunque el objetivo de esta investigación es proponer un nuevo algoritmo para detectar simetría en cuerpos no rígidos, las aplicaciones que se pueden obtener son variadas, desde reconstrucción de modelos 3D, análisis de modelos 3D, búsquedas de partes etc.

1.2 Planteamiento del problema

Encontrar la simetría en un cuerpo no rígido implica tener en cuenta varios aspectos. Por ejemplo, la cantidad de puntos que forman la malla del objeto, la cantidad de triángulos y las técnicas utilizadas para encontrar simetría en el objeto. De hecho, este no es un problema trivial, iniciando por la forma de la estructura, hasta las diferentes posturas, cambios topológicos, isométricos o ruido que puede tener los modelos 3D.

En la figura 1.2, se puede observar 6 modelos 3D de un mismo objeto. Sin embargo, la simetría debería mantenerse independiente de la postura. Por otro lado, está el problema de la escala y el ruido que pueda presentar la estructura. Por todos estos factores, este es un problema no trivial y, en esta tesis, se propone un algoritmo para determinar patrones de simetría en cuerpos no rígidos, independiente de la escala, la postura o el ruido en la estructura.

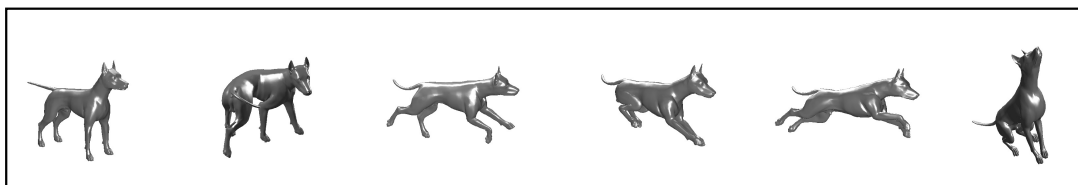


Figura 1.2: Modelos 3D de Cuerpos no Rígidos

El modelo 3D de un cuerpo, como en la figura 1.2, podría llegar a tener, por ejemplo, 10000 puntos, y la malla que representa el modelo hasta 30000 triángulos. Por lo tanto, utilizar un algoritmo que evalúe punto a punto cada vértice tratando de determinar si un vértice pertenece o no al eje de simetría, sería contraproducente en términos de costo computacional. En tal sentido, es necesario encontrar un conjunto de puntos en la malla, que sea invariante a la postura, al ruido y a la escala. En otras palabras, se requiere encontrar aquellos puntos, en la estructura, que se mantienen independientes de la transformación que se realice en el objeto. Dichos puntos se denominan *key points* o puntos de interés.

Los *key points*, son aquellos puntos que presentan una mayor variación en la superficie de la estructura respecto a sus vecinos. En el caso de las imágenes 2D, los *key points* son aquellos *pixels* con mayor variación de la intensidad con respecto a sus vecinos [SS88]. Varios algoritmos han sido desarrollados para encontrar los *key points* en imágenes 2D; sin embargo, cuando hablamos de objetos 3D, los *key points* se encuentran en los máximos o mínimos de la superficie de la estructura tridimensional. Para poder encontrar estos puntos, uno de los últimos algoritmos fue propuesto por [Glo09] y desarrollado por [SB12]. Este algoritmo es una extensión del algoritmo de [SS88] y será descrito en la sección 3.1.2 del capítulo 3.

Luego de determinar los *key points*, se presenta otro problema. El número de *key points*, puede ser aún muy grande como para poder determinar características simétricas y, por otro lado, un mismo cuerpo, en posturas distintas, no necesariamente presenta los mismos *key points*. Por lo tanto, los patrones de simetría podrían no ser los correctos. Para atacar el problema, se agrupan aquellos *key points*, de modo tal, que se obtenga componentes o submallas y, en lugar de trabajar con los *key points* de la estructura, trabajamos con los componentes.[SB12]

En la figura 1.3 podemos observar los *key components* de un cuerpo en posturas distintas, presencia de ruido, y hoyos. En esa imagen, se observa que los componentes, de color rojo, aparecen invariantes en todos los modelos 3D. Una breve descripción de la técnica propuesta por [SB12], se encontrará en la sección 4.2.1 del capítulo 2.

Como se puede ver, los *key components* son los mismos en todos los cuerpos y, por lo tanto, trabajar con *key components* implica un menor costo computacional, debido a que la cantidad de componentes es inferior a la cantidad de *key points*. Aunque, Sipiran y colegas [SB12] propusieron, recientemente, una técnica para hallar los componentes basada en *key point*, en esta tesis, se propone, además, un nuevo método basado en difusión de calor y ondas con el mismo propósito. Esta propuesta será descrita en la sección 4.2.1 del capítulo 4

El tercer problema con el cual lidiar, es el cómo encontrar los patrones de simetría dado un conjunto de *key components*. Y teniendo en cuenta que, los patrones de simetría, separan el objeto en dos estructuras similares, el primer paso es determinar que componente es isométrico a que otro componente. En el caso de la figura 1.3, se

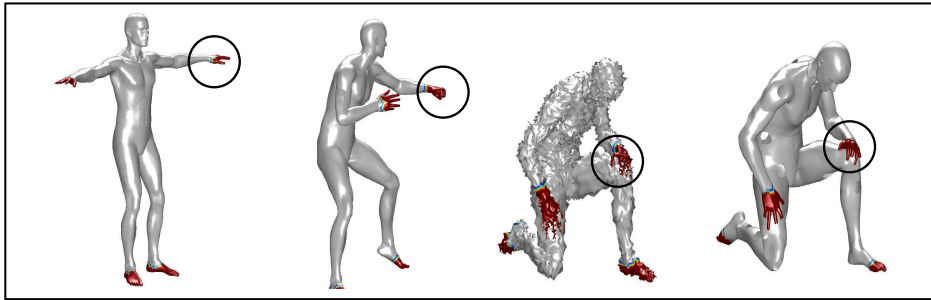


Figura 1.3: Key Componentes

observan 4 componentes, (pierna derecha, pierna izquierda, mano derecha y mano izquierda). En esta figura podemos agrupar los componentes mediante alguna técnica que nos permita medir el grado de similitud de estos componentes. De modo que, amabas manos formen un grupo y ambos pies otro.

Al tener los componentes agrupados, nos resta encontrar el conjunto de puntos s cuyas distancias geodésicas, al centroide geodésico de un componente a es igual, o muy similar, al centroide geodésico de otro componente b , de modo que ambos componentes a y b , pertenecen al mismo grupo, y por lo tanto, s deberá estar formado por los puntos del eje de simetría. Las definiciones de distancia geodésica se encuentra en la sección 2.2.2 del capítulo 2.

El cuarto problema es, cómo saber si dos componentes son isométricos. Para lograr esto, proponemos el uso de *Head kernel Signature* [SOG09] y *Wave Kernel Signature* [ASC11b]. El *Head kernel Signature* (*HKS*) conocido como firma de calor, está basado en la ecuación del calor, una ecuación diferencial, que modela el hecho de aplicar una fuente de calor a un punto p de una superficie y obtener, el grado en el cual, el calor se disipa en ese punto p en el tiempo t . Por otro lado, el *Wave Kernel Signature* (*WKS*), se basa en la idea de como las ondas se mueven por medio de una superficie. Por ejemplo, si damos un golpe a una superficie en un punto p , este golpe generará ondas, los cuales se extenderán por la superficie; de modo que la cantidad de energía en el punto p irá disminuyendo en el tiempo. Está técnica es similar al *HKS*, sin embargo ha demostrado mejores resultados en varias aplicaciones [VBCG10, ZGLG12, PS13, OMMG10, BK10, RBBK10b]. Los conceptos y definiciones de *HKS* y *WKS* se describen en la sección 2.1.2 y 2.1.3 del capítulo 2.

Dado que la superficie no es continua, si no que, es una malla de triángulos, es necesario aplicar conceptos del operador *Laplace Beltrami*, *Eigen vectores* y *Eigen valores*, los cuales serán explicados en la sección 2.1 del capítulo 2.

1.3 Objetivo

El objetivo de esta investigación es el desarrollo de un nuevo algoritmo para encontrar *key components* y detectar simetría en cuerpos no rígidos.

1.4 Aportes

La presente investigación tiene 3 aportes al estado del arte:

1. Proponemos un nuevo método para la detección de *Key points* basado en el análisis de las mallas triangulares. Este método será descrito en la sección 3.2 del capítulo 3.
2. Dado que se requiere identificar componentes isométricos para hallar la simetría de un objeto o modelo, nosotros desarrollamos un nuevo algoritmo para la detección de componentes basados en proceso de difusión de calor y detección de componentes fuertemente conexos. Este método será descrito en la sección 4.3 del capítulo 4.
3. Finalmente, desarrollamos un nuevo algoritmo para la detección de simetría en cuerpos no rígidos, mediante distancia geodésica, difusión de calor, y ecuación de onda. Este método será descrito en la sección 5.3 del capítulo 5.

2

CONCEPTOS PREVIOS

Este capítulo describe la teoría necesaria para poder entender la propuesta de la investigación

2.1 Análisis de modelos 3D

En la siguiente sección se explica conceptos del operador de *Laplace-Beltrami*, *Head Kernel signature*, *Wave Kernel Signature*, simetría extrínseca, simetría intrínseca, entre otros. Estos conceptos y la teoría matemática que los respalda, son piezas clave para el desarrollo y entendimiento de la propuesta.

2.1.1 Operador Laplace-Beltrami

El operador de Laplace-Beltrami es una generalización del laplaciano para espacios planos o superficies. Este operador juega un rol importante en muchas áreas, como procesamiento de imágenes, procesamiento de señales, procesamiento de superficies y, el estudio de ecuaciones diferenciales parciales. [Xu04].

Grafo Laplaceano Uno de los problemas cuando se trabaja con estructuras 3D representadas mediante una malla de polígonos, es el cómo manipular la estructura manteniendo información geométrica y morfológica. Es posible representar cualquier topología mediante una jerarquía de funciones base, las cuales están definidas mediante auto funciones del operador Laplace-Beltrami. Este operador define funciones base, las cuales, se adaptan bien a la geometría y topología del objeto 3D [Rus07a]

En otras palabras, podemos representar la geometría y la topología de una superficie mediante un conjunto de funciones y trabajar con una discretización de la superficie mediante una malla de polígonos. La representación de esa superficie discreta se realiza por medio de un grafo Laplaceano.[Xu04]

El grafo Laplaciano $L = (a_{i,j})$ no es más que una matriz, la cual esta definida como:

$$a_{i,j} = \begin{cases} w_{i,j} > 0 & \text{si } (i, j) \text{ es una arista} \\ -\sum w_{i,j} & \text{si } i = j \\ 0 & \text{En otro caso} \end{cases} \quad (2.1)$$

Donde $w_{i,j}$ son los pesos de las aristas asociados al grafo y A representa el área de *voronoi* de uno de los vértice de la arista W_{ab} . La ecuación 2.1, representa el grafo *laplaceano*. La matriz almacena información de la estructura del objeto mediante los pesos. Los pesos del Laplaciano pueden ser calculados de diferentes maneras, como por ejemplo mediante la siguiente ecuación 2.2:

$$w_{i,j} = \frac{\cot(\alpha_{ij}) + \cot(\beta_{ij})}{2A} \quad (2.2)$$

En la figura 2.1 (a), el peso W_{ab} de la arista azul se calcula sumando los ángulos opuestos a esta arista. Es decir $W_{ab} = (\cot(a) + \cot(b))/2A$. Este método, permite medir la regularidad de una triangulación, por ejemplo en (b) tenemos un grafo regular y, en tal sentido, los pesos de las aristas comunes al punto central p , serán similares. Sin embargo, si hay un cambio abrupto en la triangulación, los pesos cambian y, por lo tanto, la geometría del objeto. Esto se puede observar en figura (c), donde los ángulos de los triángulos son distintos. De esta forma, los pesos del laplaceano permiten almacenar información geométrica y topológica del objeto.

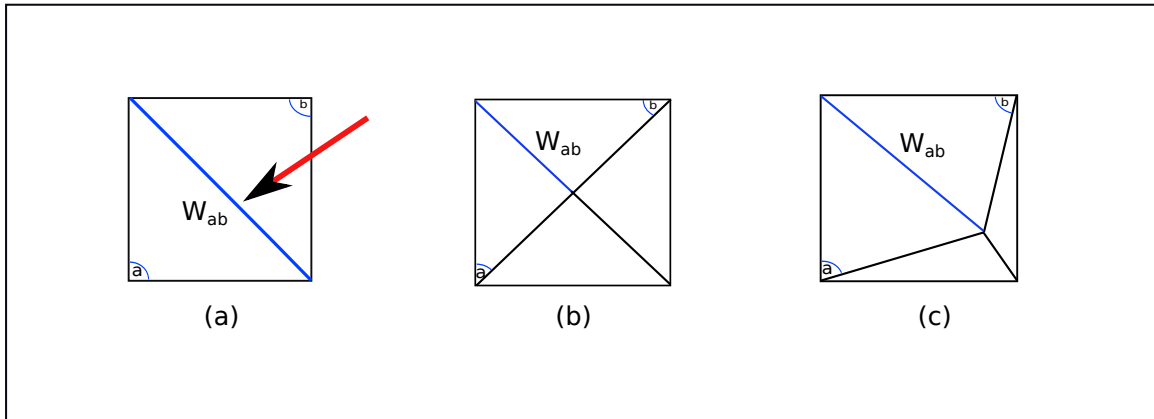


Figura 2.1: (a) Peso de la arista W_{ab} (b) Peso de Arista en Grafo Regular (c) Peso de Arista en Grafo Irregular

Si una estructura se representa mediante un grafo de 1000 vértices, entonces el grafo *laplaceano* sería una matriz poco densa de complejidad espacial $O(10^6)$. Por lo tanto, en lugar de trabajar con la matriz, es factible trabajar con los auto vectores de la matriz y, dado que los auto vectores y un auto valores, son combinaciones lineales de la matriz L , que representa el grafo *laplaciano*, entonces, un solo auto vector mantiene información topológica y geometría pero con una complejidad espacial de $O(10^3)$. Por otro lado, el conjunto de auto vectores de la matriz L es una base vectorial que reconstruye la superficie de un objeto 3D.[Rus07a]

Operador Laplace-Beltrami El operador Laplace-Beltrami se define de la siguiente manera:

$$\Delta = \text{div}(\text{grad}(M)) \quad (2.3)$$

Donde Δ operador lapla es grad representa el campo vectorial gradiente en una superficie y div , es la divergente.

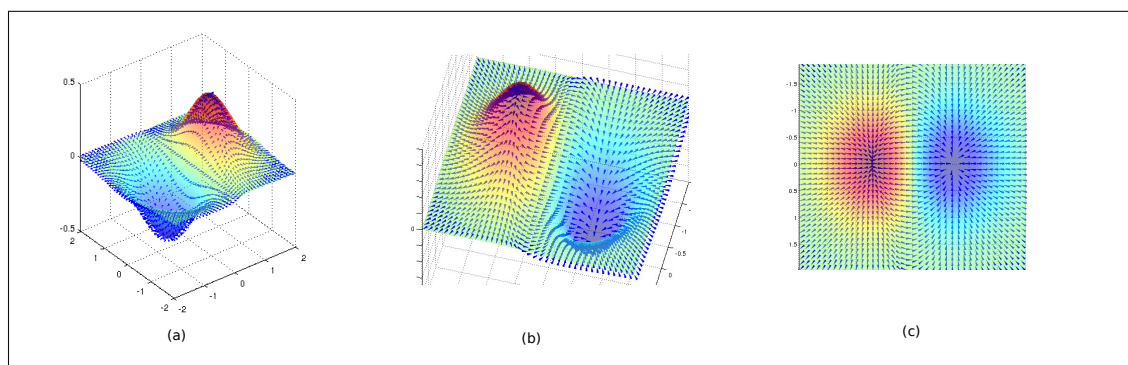


Figura 2.2: (a) Superficie bidimensional (b) Campo vectorial gradiente (c) Divergente (fuente p y sumidero q)

En la figura 2.2 se observa en (a) una superficie bidimensional. Al aplicar la gradiente en cada punto de la superficie, se obtiene un campo vectorial como en (b); este campo vectorial revela información respecto a la razón de cambio y la magnitud, de cada punto, respecto a la superficie. Finalmente en (c) se observa los vértices fuente, es decir aquellos vértices para las cuales la divergencia es positiva y, los sumideros, es decir los vértices para los cuales la divergente es negativa. La información que revela la divergente de la gradiente es importante para los procesos de difusión, de aquí la importancia del operador Laplace-Beltrami en el análisis de modelos 3D.

Dado que nuestros modelos son superficies bidimensionales, embebidas en espacios tridimensionales y, están representadas mediante una malla de triángulos, es factible definir, en ellas, el operador Laplace-Beltrami.

El operador de Laplace-Beltrami Δ puede ser descompuesto en un conjunto de auto vectores y auto valores, de modo que, $\phi_i : M \rightarrow R$ son los autovalores y $\lambda_i \in R$ los auto valores, tal que:

$$\Delta \phi_i = \lambda_i \phi_i \quad (2.4)$$

Esta descomposición en auto vectores y auto valores presenta las siguientes propiedades [Ovs11]

1. Los auto vectores ϕ_i son invariantes bajo deformaciones isométricas.

2. El conjunto de auto vectores forman una base ortogonal completa, para el espacio de funciones L_2 en la superficie.
3. El operador Laplaceano únicamente determina la métrica de la superficie y, por otro lado, la distancia geodésica entre los puntos.

De la propiedad 1 y propiedad 3 se desprende que dos superficies son isométricamente deformables una en la otra, si y solo si, sus operadores de Laplace tiene los mismos auto vectores y auto valores.

De manera formal decimos, sea M y N dos superficies compactas con operadores de Laplace-Beltrami Δ_M y Δ_N , entonces, un *homeomorfismo* $T : N \rightarrow N$ preserva las distancias geodésicas, si y solo si, para cada auto vector ϕ_i de Δ_N asociada con λ_i , $\phi_i \circ T$ es una auto vector de Δ_M .

Esta propiedad nos permite, directamente, crear algoritmos para encontrar *key components* y simetría, sin embargo, el número de comparaciones posibles para encontrar puntos de simétricas es exponencial. Por esta razón, en la siguiente sección, se analizan conceptos de *Heat Kernel Signature* y *Wave Kernel Signature*, los cuales serán indispensables para el entendimiento del método propuesto para encontrar *key components*, descrito en el capítulo 4 y el método propuesto para encontrar simetría, descrito en el capítulo 5.

2.1.2 Heat Kernel Signature

Imaginemos que tenemos una superficie M , a la cual se le aplica una fuente de calor en un punto p de la superficie M . El problema de encontrar la cantidad de disipación de calor en el punto p luego de un tiempo t , se resuelve por medio de la ecuación del calor, cuya solución se denomina como *Heat Kernel*. De manera formal decimos que, si $u(p, t)$ es el promedio de calor en el punto p luego de un tiempo t , entonces:

$$u(p, t) = \frac{\partial u}{\partial t} = \Delta u \quad (2.5)$$

Donde Δu es el operador de Laplace-Beltrami.

Por otro lado, es lógico pensar que el calor se disipa en un punto p , a medida que se incrementa el tiempo t . Un posible modelo es mediante la fórmula e^{-t} .

En la figura 2.3, se observa que en un tiempo $t = 0$, la cantidad de calor es máxima y, a medida que el tiempo se incrementa, la intensidad de calor disminuye. Sin embargo, este modelo no representa correctamente lo que ocurre, ya que, la intensidad de calor disminuye teniendo en cuenta la topología y la geometría de la superficie

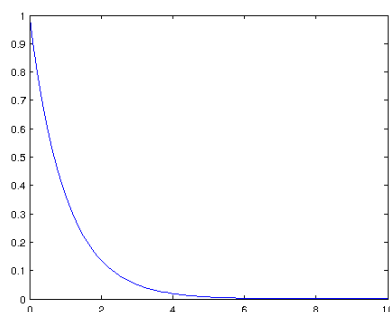


Figura 2.3: Función e^{-t}

M . Entonces, para obtener la disipación de calor de manera correcta, se resuelve la ecuación diferencial 3.13 y se obtiene la ecuación 2.6

$$u(p, t) = e^{-t\Delta} f \quad (2.6)$$

Donde Δ es el operador de Laplaz Beltrami.

La ecuación 2.6, muestra que la disipación de calor, no solo depende del tiempo; si no también, de la geometría y topología, esto debido a que el operador Δ mantiene información geométrica y topológica de una superficie M . Por otro lado, Sun et al. [SOG09] propusieron el *Heat Kernel Signature* como una restricción del dominio temporal del *Heat Kernel*, de modo que se puede obtener una firma para cada vértice del grafo que representa el modelo 3D.

$$HKS(x, t) = \sum_{i=0}^{\infty} e^{-\lambda_i t} \phi_i^2(x) \quad (2.7)$$

En la ecuación 2.7, λ_i y ϕ_i son los auto valores y auto vectores del operador de Laplace-Beltrami, respectivamente y t es el tiempo de exposición de la fuente de calor.

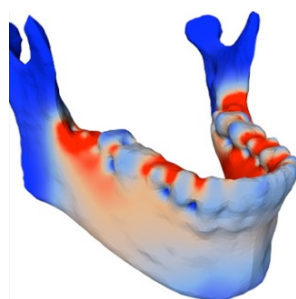


Figura 2.4: Disipación del calor en una superficie [ZRHAr]

En la figura 2.4 se aprecia como el calor se disipa en la estructura y, la manera como se disipa, varía en relación a características geométricas y topológicas.

2.1.3 Wave Kernel Signature

El Wave Kernel Signature es un método para análisis de superficies tridimensional representadas en forma de mallas triangulares recientemente propuesto por [ASC11b]. La idea del método se basa en representar un punto x , de una superficie M , mediante la probabilidad promedio de una partícula cuántica de diferentes niveles de energía medidos en ese punto x , la cual esta gobernada por la ecuación de Schrödinger.

La ecuación de Schrödinger mide la evolución de una partícula cuántica, en un superficie y , se define de la siguiente manera:

$$\frac{\partial \psi(x, t)}{\partial t} = i\lambda_i \psi(x, t) \quad (2.8)$$

Como se observa, la ecuación 2.8 es una ecuación diferencial parcial, la cual describe el cambio o la evolución de una función de honda en el tiempo y , de manera similar a la propagación del calor, es posible evaluar cómo se comportan las ondas, en una superficie embebida en un espacio tridimensional. La ecuación de Schrödinger puede ser calculada mediante la auto descomposición del grafo que representa una superficie y , por lo tanto, también esta relacionada a la auto descomposición del operador *Laplace-Beltrami*.

El *Wave Kernel signature* se represente mediante la siguiente ecuación:

$$WKS(E, x) = \sum_{k=0}^{\infty} \phi_k(x)^2 f_E(E_k)^2 \quad (2.9)$$

Donde E representa la cantidad de energía promedio en un punto x de una superficie M , x es un punto en la superficie de la malla M donde se mide la cantidad de energía. $\phi_k(x)$ y $f_E(E_k)$ representan la auto descomposición del grafo laplaciano que representa la superficie M .

Como se puede observar, la ecuación 2.9 es similar al *Heat kernel signature*, descrita en la ecuación 3.14. Sin embargo, a diferencia de la propagación de calor descrita por el *HKS*, el *WKS* caracteriza la propagación de las ondas en una superficie.

Aunque la ecuación 2.9 es similar a la ecuación de calor, la dinámica es drásticamente diferente. Mientras la ecuación de calor se concentra en el proceso de disipación, la ecuación de Schrödinger esta relacionada a la oscilación de las partículas. Lo cual, crea una comportamiento drásticamente diferente.

2.2 Funciones de Distancia en modelos 3D

Podemos pensar en una estructura tridimensional discreta como una nube de puntos en un espacio euclídeo, o en una superficie bidimensional embebida en un espacio tridimensional representada como un grafo y, aplicar, a cada punto o vértice de la estructura, una función que transforme ese punto en un vector k dimensional. Por lo tanto, podemos medir o comparar los vectores k -dimensionales que representan los puntos de los objetos, partes de los objetos o los objetos mismos, mediante alguna métrica.

A continuación se describen las métricas que posteriormente podrán definir conceptos de simetría extrínseca y simetría intrínseca.

2.2.1 Distancia Euclidia

La distancia euclídea, es un función conocida que nos permite medir la distancia de 2 puntos k - *dimensionales* en un espacio euclideo.

$$d(v_1, v_2) = \sqrt{\sum_{i=0}^k (v_1(i)^2 - v_2(i)^2)} \quad (2.10)$$

La ecuación 2.10, representa la recta más pequeña que une dos puntos en un espacio euclídeo. Uno de los problemas con esta métrica, es que es susceptible a cambios isométricos y topológicos. Como se puede ver en la figura 2.5, la imagen de la izquierda, muestra dos puntos de color rojo (uno en el dedo índice y, el otro, en el dedo gordo de la misma mano). La imagen de la derecha, es la misma mano, con un cambio isométrico y, como se puede notar, el dedo índice y el dedo gordo de la mano se han acercado, lo cual implica un cambio en el valor de la distancia entre ambos puntos. Esto hace que la distancia euclídea no funcione en objetos con transformaciones no rígidas .

2.2.2 Distancia Geodésica

La distancia geodésica se define como el camino más corto entre dos puntos en la superficie de la tierra y, por lo tanto, no se trata, necesariamente, de una recta, si no, de una curva. Dato que las superficies con las que trabajamos, se representa de

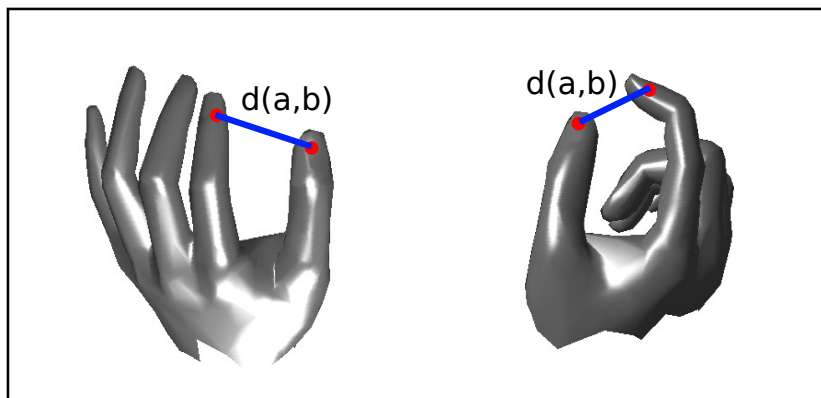


Figura 2.5: Distancia euclida en un objeto tridimensional con cambio isométrico

manera discreta, con grafos; entonces, la distancia geodésica se obtiene encontrando la ruta más corta entre dos puntos en el grafo. Y por lo tanto, solo es necesario de aplicar el algoritmo de Dijkstra.

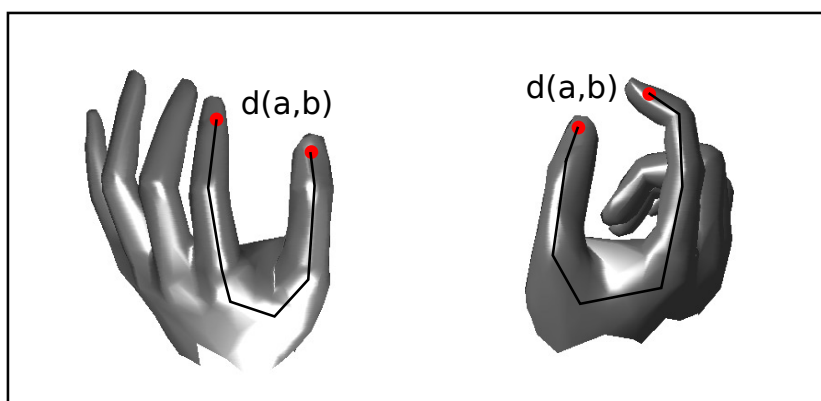


Figura 2.6: Distancia geodésica en un objeto tridimensional con cambio isométrico

En la figura 2.6 se observa la distancia geodésica entre dos puntos. A diferencia de la distancia euclídea, ver figura 2.5, la cual varía cuando existen transformaciones no rígidas, como isometrías o estiramientos, la distancia geodésica permanece similar. Sin embargo, con la presencia de cambios topológicos, la distancia geodésica deja de funcionar.

2.2.3 Distancia de Difusión

La distancia de difusión [CL06], es una mediada de distancia no lineal, la cual computa la distancia entre dos puntos, simulando la difusión de calor y recordando el tiempo de ir de un punto a otro mediante una difusión aleatoria. [LDR09]. La distancia de

difusión esta relacionada con el *Heat Kernel Signature*, como se pueden ver en las ecuaciones 2.11, 2.12 y 2.13.

$$d_{X,t}(v_x, v_y) = \|h_t(v_x, \cdot) - h_t(v_y, \cdot)\|_{L_2(X)}^2 \quad (2.11)$$

$$d_{X,t}(v_x, v_y) = \|h_t(v_1, v_k) - h_t(v_k, v_y)\|^2 dz \quad (2.12)$$

$$d_{X,t}(v_x, v_y) = \sum_{i=1}^{\infty} e^{-2\phi_i t} (\lambda_i(v_1) - \lambda_i(y))^{v_2} \quad (2.13)$$

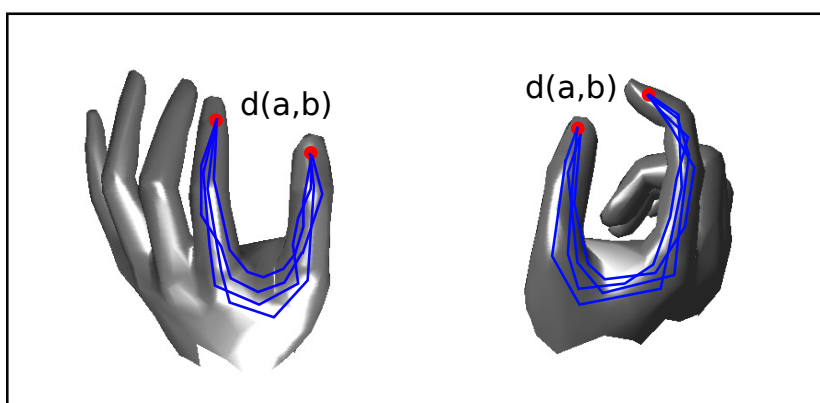


Figura 2.7: Distancia de difusión en un objeto tridimensional con cambio isométrico

En la figura 2.7, se observa la distancia entre 2 puntos en un mano y, a diferencia de la distancia geodésica, que calcula el camino más corto en el grafo que representa el objeto, la distancia de difusión, calcula el promedio de los caminos de ir desde un punto a a un punto b ; lo cual, la hace menos susceptible a cambios isométricos, topológicos y ruido en la estructura.

2.3 Simetría

Distintos algoritmos han sido desarrollados para encontrar simetría en objetos 2D; incluso si estos objetos son sometidos a transformaciones rígidas como rotación, traslación o escala. Sin embargo, en el mundo real, los objetos son tridimensional y las transformaciones que sufren no necesariamente son rígidas. Es decir, pueden sufrir transformaciones, como estiramientos o deformaciones, en partes del objeto; lo cual, puede generar cambios en la curvatura del objeto y, por lo tanto, de su topología. A estos cambios se les denomina transformaciones no rígidas y estas transformaciones afectan

las distancia euclídea entre dos puntos de la malla que representan el modelo. Por otro lado, las transformaciones rígidas mantienen la distancia euclídea entre pares de puntos. En tal sentido, es posible determinar dos tipos de simetría, extrínseca e intrínseca.

2.3.1 Simetría Extrínseca

La simetría extrínseca puede ser definida como invariante bajo transformaciones rígidas. Una transformación rígida se refiere a aquella que preserva la distancia euclídea. En la figura 2.8, se observa un objeto con su eje de simetría representado por el color rojo. Luego, dados dos puntos x, y en el lado izquierdo del modelo, y dados dos puntos x', y' en el lado derecho del modelo y, suponiendo que, x' es el resultado de aplicar una transformación rígida de x y y' es el resultado de aplicar la misma transformación a y , entonces decimos que la distancia euclídea entre x y y es igual a la distancia euclídea entre x' y y' .

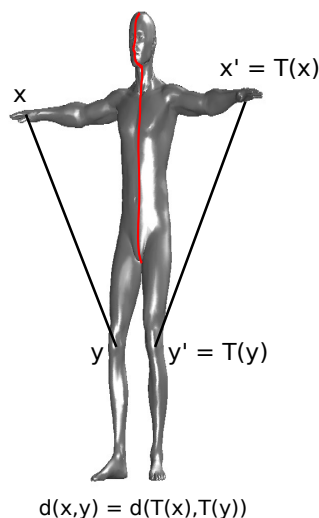


Figura 2.8: Simetría Extrínseca

De manera más formal definimos la simetría extrínseca de la siguiente manera: Dado un modelo M representando mediante una variedad de Riemann, específicamente una superficie bidimensional compacta, suave y conectada embebida en un espacio tridimensional y, dos puntos $x \in M$ y $y \in M$, tal que, $x' = T(x)$, y $y' = T(y)$, entonces se cumple la ecuación 2.14, donde d es la distancia euclídea.

$$d(x, y) = d(T(x), T(y)) \quad (2.14)$$

En la figura 2.9 se aprecia como la distancia euclídea d_1 en la imagen (a) es diferente a la distancia euclídea d_2 en la figura (b). Esto se da, debido a que la figura (a) es una transformación no rígida de (b) y, por lo tanto, las distancia entre pares de puntos de figura (a) no preservan las distancias euclídeas entre los pares de puntos en la figura (b).

Por otro lado, una transformación $T : M \rightarrow \mathbb{R}^3$ que preserva la geometría extrínseca de un modelo M es llamada congruencia y, M y $T(M)$ son llamados congruentes. Luego si $T(M) = M$, entonces a T se le denomina auto-congruencia o una simetría extrínseca de M . En general, la simetría extrínseca forma un grupo de funciones de composición de la siguiente manera [RBBK07].

$$S_e(M) = \{T | M \rightarrow M : d_{euclídea}(x, y) = d_{euclídea}(x', y') \wedge x' = T(x) \wedge y' = T(y), \forall x', x, y, y' \in M\} \quad (2.15)$$

Un subconjunto de S_e son la reflexión y la rotación.

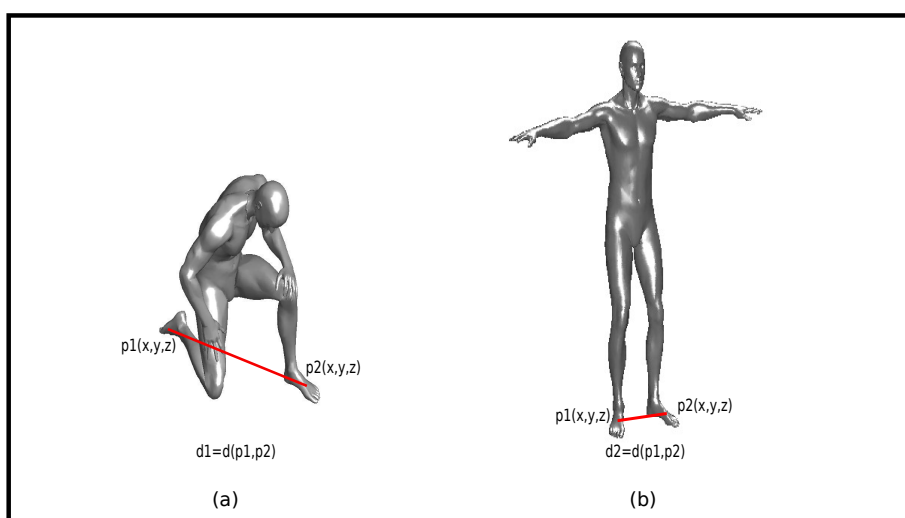


Figura 2.9: Simetría en cuerpo Rígido

2.3.2 Simetría Intrínseca

Supongamos que tenemos una variedad de Riemann M , compacta y sin bordes. Definimos una simetría intrínseca si existe un homeomorfismo $T : M \rightarrow M$ en una variedad de Riemann, de modo tal, que se preservan las distancias geodésicas. Esto es:

$$d(x, y) = d(T(x), T(y)) \quad (2.16)$$

Donde x, y son dos vértices de la malla que representa la variedad de Riemann M , d es una función de distancia geodésica y T representa alguna transformación no rígida, como una transformación isométrica, cambios topológicos, ruidos entre otros.

Como se puede ver en la imagen 2.10, la distancia d_1 entre el punto p_1 y p_2 en la figura (a) debe ser la misma que la distancia d_1 entre el punto p_1 y p_2 de la figura (b). Esto se da, solo si se trata de la distancia geodésica, la cual fue definida en la subsección 2.2.2.

Por otro lado, una transformación $T : M \rightarrow \mathbb{R}^3$ que preserva la geometría intrínseca de un modelo M , es llamada una isometría y, el modelo M y $T(M)$ se refieren como isométricos. Una función de transformación isométrica aplicada así misma, es llamada auto-isometría o simetría intrínseca del modelo M . Al igual que la simetría extrínseca, la simetría intrínseca forma un grupo isométrico representado por la ecuación:

$$S_i(M) = \{T | M \rightarrow M : d_{geodesica}(x, y) = d_{geodesica}(x', y') \wedge x' = T(x) \wedge y' = T(y), \forall x', x, y, y' \in M\} \quad (2.17)$$

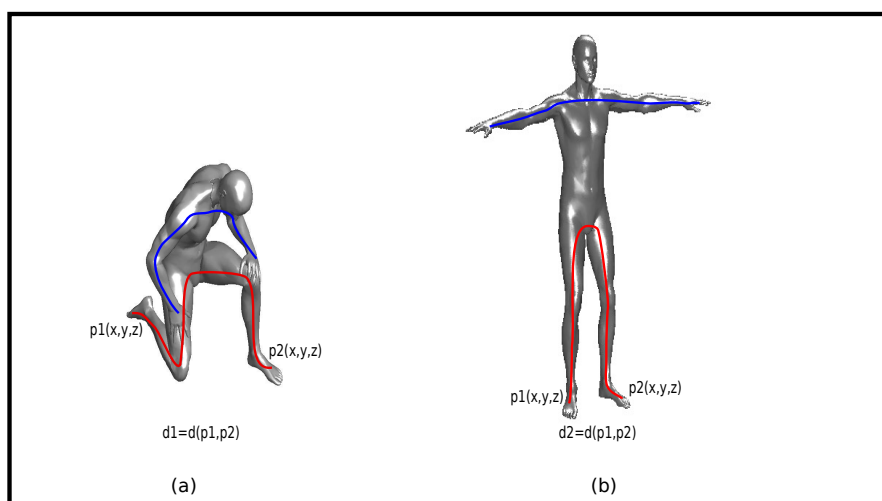


Figura 2.10: Simetría Intrínseca

3

KEY POINTS: MINIMAL AREA INTEREST POINT

Este capítulo describe un nuevo algoritmo eficiente para la detección de key points

Debido a la creciente cantidad de información multimedia y la reducción de los costos en los dispositivos de adquisición de datos 3D, es que se ha incrementado el interés en el desarrollo de algoritmos de extracción de características para modelos tridimensionales, de tal manera, que estos sean eficientes, robustos e invariantes a cambios isométricos, topológicos y presencia de ruido.

Una de las tareas clave, para la extracción de características en modelos 3D, es la detección de puntos de interés o *key points*. Estos puntos son aquellos que están presentes en las estructuras sobresalientes o protusiones del modelo y pueden ser utilizadas en lugar de todo el objeto.

La detección de puntos de interés o *key points*, es el proceso inicial que sirve para desarrollar tareas subsecuentes, como detección de componentes clave [SB12], registro parcial de modelos en bases de datos 3D, detección de simetría extrínseca, recuperación y reconocimiento de modelos 3D [OOFB08], coincidencia parcial de mallas [LVJ05], segmentación y simplificación de mallas [DCG12], entre otros.

Por otro lado, definir el concepto de punto de interés es ambiguo, debido a la necesidad de incorporar la percepción humana. En este trabajo, los puntos de interés o *key points* se definen mediante el nivel de la protuberancia de las estructuras locales en los modelos 3D [SB11].

La detección de puntos de interés ha sido bastante estudiado en el caso de imágenes 2D. Sin embargo, en modelos 3D, se presentan nuevos problemas más complejos, como la deformación, la escala, presencia de ruido, cambios topológicos e isométricos, entre otros. Algunos trabajos se han desarrollado últimamente para lidiar con este problema; tales como, mallas salientes [LVJ05], puntos salientes [CCFM08], 3D Harris [SB11], 3D sift [OOFB08], esquinas dependientes de la escala [NN07], detección de puntos de interés basados en la escala (HKS) [SOG09]. Todas estas técnicas fueron comparadas en eficiencia y robustez en [DCG12].

En esta investigación, se presenta un nuevo algoritmo eficiente para detectar puntos de interés en modelos 3D, mediante el análisis de los triángulos que componen la malla

del modelo tridimensional. Este algoritmo, es distinto a otras propuestas recientes, las cuales son más complejas y menos eficientes, tales como 3D Harris [SB11] o detección de puntos de interés basados en la escala (HKS) [SOG09].

3.1 Trabajos Relacionados

Los puntos de interés, son aquellos puntos con una alta variación en la superficie de la estructura con respecto a sus vecinos. Para imágenes 2D, los puntos de interés son aquellos pixels con alta variación de intensidad. Mientras que, en modelos 3D, los puntos de interés son aquellos que se encuentran en zonas de alta protusión del modelo.

Como se mencionó en la sección anterior, varias propuestas han sido desarrolladas para detectar puntos de interés en modelos 3D. Muchas de ellas son extensiones de los detectores propuestos en imágenes. Sin embargo, dos métodos, planteados recientemente, para detección de *key points* en modelos 3D son *3D Harris*, el cual fue propuesto por [Glo09] e implementado por [SB11] y; el de detección de puntos de interés basados en la escala (HKS) expuesto por [SOG09].

3.1.1 Detección de *Key Points* en Imágenes

Harris and Stephens [Det88] propusieron un algoritmo para encontrar esquinas en imágenes bidimensionales. Este algoritmo analiza el cambio de intensidad de un píxel en relación a sus vecinos, de modo tal, que los píxeles con mayor promedio de cambio, son aquellos con mayor probabilidad de representar esquinas o bordes en las imágenes.

En figura 3.1 se observa la imagen en mapa de bits de Lena. Al analizar esta figura, podemos ver varios *pixels* que tienen colores muy similares, estos forman un grupo. Por otro lado, los *pixels* de grupos distintos tienen *pixels* con valores de intensidades diferentes. Entonces, es lógico pensar que si un píxel p_i que pertenece a un grupo g_j de la imagen, se encuentra en el centro del grupo, lo más probable es que el cambio de intensidad con respecto a sus vecinos sea mínimo. Por el contrario, si el *pixel* p_i del grupo g_i se encuentra muy cerca a varios *pixel* de otro grupo g_j , entonces el cambio de intensidad con respecto a sus vecinos deberá ser mayor.

Por otro lado, una imagen 2D a color, puede ser vista como una superficie bidimensional F , de modo que $I = F(x, y)$, donde I es el valor de intensidad en la coordenada x, y de la superficie. Luego, podemos analizar cada *pixel* de la imagen, como si fuera un

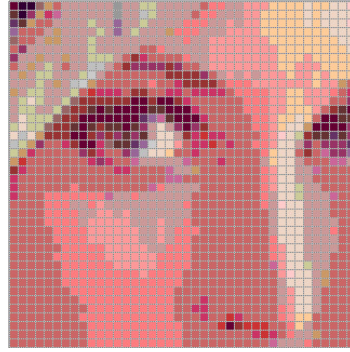


Figura 3.1: Mapa de Bits de la imagen de Lena

punto de una superficie continua y, evaluar el promedio de las derivadas con respecto a sus vecinos. Si este promedio tiende a ser cero, entonces, indica que el píxel, no tiene un cambio abrupto de intensidad y, por lo tanto, no se trata de un esquina ni un contorno. Pero si el valor es mayor o menor a cero, entonces, existe un cambio abrupto de intensidad y, por lo tanto, es probable que se trate de una esquina o contorno.

Basado en esta idea, Harris y Stephens, propusieron un algoritmo, conocido como *descriptor de Harris* para la detección de puntos de interés en imágenes 2D.

Sea I una imagen bidimensional, y sea $I(x_i, y_j)$ el valor de la intensidad del píxel en la posición (x_i, y_j) ; entonces, para encontrar la variación de la intensidad de un *píxel* (x, y) , hallamos el promedio de la variación del *píxel* (x, y) con respecto a una vecindad gaussiana W . La ecuación 3.1 representa este promedio de variación de la intensidad.

$$e(x, y) = W(x_i, y_j) \sum_{x_i, y_i} [I(x_i + \Delta x, y_i + \Delta y) - I(y_i, y_i)] \quad (3.1)$$

Un problema con la ecuación 3.1, es que trabaja con valores discretos. Por lo tanto; es necesario trabajar con valores continuos para poder aplicar las derivadas a cada puntos de la superficie que representa la función de intensidad. Por esta razón, Harris propone utilizar la expansión de Taylor truncada, para aproximar una superficie continua, a la función discreta $F(x, y)$. Es así que se obtiene la ecuación 3.12

$$E(x, y) = S \begin{bmatrix} \sum_{x_i, y_i} \cdot W I_x^2 & \sum_{x_i, y_i} \cdot W I_y * I_x \\ \sum_{x_i, y_i} \cdot W I_x * I_y & \sum_{x_i, y_i} \cdot W I_y^2 \end{bmatrix} S^T \quad (3.2)$$

Donde $S = [\Delta x \Delta y]$

$$e(x, y) = SE(x, y)S^T \quad (3.3)$$

$$h(x, y) = \det(E) - k(\text{Tr}(E))^2 \quad (3.4)$$

En la ecuación 3.4, $h(x, y)$, se conoce como el operador de harris; $E(x, y)$ es una matriz que representa las variaciones promedio de las intensidades de un pixel con respecto a sus vecinos, det es la determinante de una matriz y Tr es la traza de la matriz.

Ahora, el algoritmo se reduce a aplicar el *operador de Harris* a cada píxel de la imagen con respecto a una vecindad Gaussiana. Los valores $h(x, y)$ iguales a cero, indican que no hay cambios con respecto a sus vecinos y, por lo tanto, no se trata de un punto importante. Sin embargo, si el valor de $h(x, y)$ es mayor a cero y menor a cero, entonces, estamos en un máximo local o un mínimo local en la superficie formada por los valores de las intensidades de las imágenes y, por lo tanto, se trata de un *pixel* que posiblemente pertenezca a un borde o contorno. Dado que una imagen puede tener un número muy grande de puntos de interés, solo es necesario tomar los k mayores valores absolutos de $h(x, y)$ como puntos de interés.

3.1.2 Detección de *Key Points* en Objetos 3D

Spirán y Bustos [SB11], propusieron una extensión del algoritmo de *Harris*, para que en lugar de encontrar puntos de interés en imágenes 2D, sea posible encontrar puntos de interés pero en mallas de objetos 3D. La idea de Spirán y Bustos, es muy similar a la de Harris, solo que en lugar de analizar *pixels*, ahora se analizan vértices. Por otro lado, dado que los objetos 3D están modelados como una malla de polígonos, la estructura de datos deja de ser una matriz, para convertirse en un grafo.

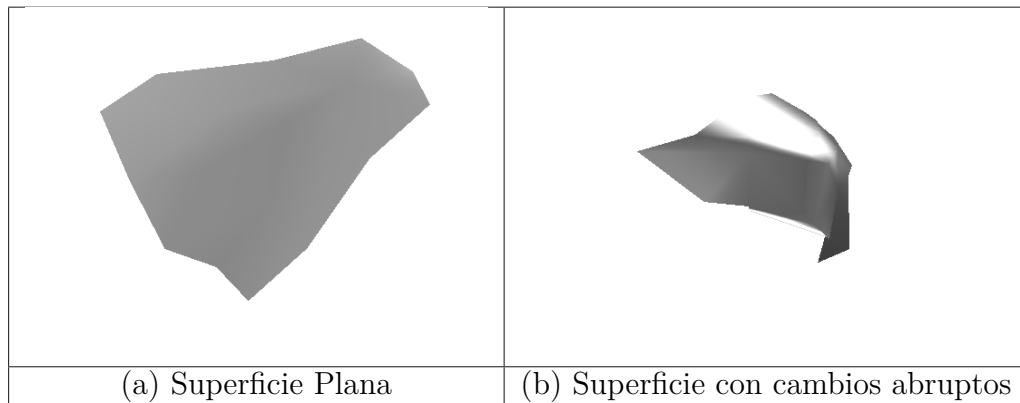


Table 3.1: Tipos de superficies que se presenten en una imagen 3D

En la tabla 3.1, se puede apreciar dos tipos de superficies. En (a), se presenta una superficie plana y, por lo tanto, el cambio de la estructura entre un vértice y otro, es nulo. En (b), por otro lado, se aprecia una superficie irregular; entonces, algunos vértices causarán una variación fuerte en la estructura, y son esos vértices los que se desea encontrar. Luego, haciendo uso de esta característica, Spirán y Bustos desarrollaron un método para encontrar aquellos vértices en el grafo que mayor variación aportan a

la estructura tridimensional, resultando en un algoritmo simple y muy similar al de Harris.

Uno de los problemas a resolver en este tema, es como definir la vecindad; ya que no contamos con una matriz, sino con una nube de puntos en un espacio tridimensional, interconectados por aristas. Sipirán y Bustos, proponen utilizar k -anillos al rededor de un vértice.

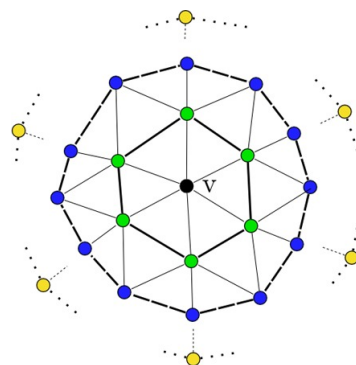


Figura 3.2: k ring [SB11]

En la figura 3.2, el vértice v , es el nodo desde el cual se tomarán k -anillos, los nodos de color verde representan los vecinos de 1-anillo, los de color azul son vecinos 2-anillos, y los de color amarillo son los 3-anillos. Los vértices que pertenecen a estos anillos forma el grupo de vecinos de v , y son estos vecinos y el mismo v , los que serán analizados para evaluar si el vértice v proporciona una mayor deformación en la superficie, en relación a sus vecinos. De ser este el caso, v se considera un vértice con alta probabilidad de ser un punto de interés.

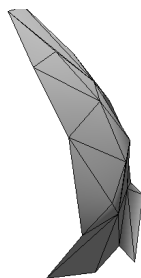


Figura 3.3: Malla de triángulos

Sin embargo, como se puede notar en la figura 3.3, el conjunto de vértices es una discretización de una superficie; y para poder calcular el valor de *Harris*, como en el caso de las imágenes 2D, se requiere una superficie continua y, de este modo, encontrar las derivadas. Por este motivo, Iván Sipirán y Bustos, encuentran el centroide, $v(k)$, del conjunto de vértices y, trasladan al conjunto de puntos, de modo que, el centroide se encuentre en el origen del sistema de coordenadas 3D. Para esto, se aplica Análisis de componente Principales (PCA) y se escoge al auto vector con el menor auto valor

asociado como la normal del plano de ajuste. La figura 3.4 es el resultado de aplicar la rotación mediante PCA a la figura 3.4. Es importante notar que el conjunto de puntos es rotado, de modo tal, que la normal esté en dirección del plano z .

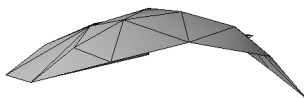


Figura 3.4: Rotación de la malla mediante PCA

Una vez que el conjunto de puntos ha sido rotado y, dado que para poder aplicar *Harris* es necesario tener una superficie continua; entonces, se ajusta una superficie cuadrada al conjunto de vértices, utilizando el método de mínimos cuadrados. La idea es buscar un paraboloides cuya forma está representada mediante la ecuación 3.5

$$z = f(x, y) = \frac{p_1}{2}x^2 + p_2xy + \frac{p_3}{2}y^2 + p_4x + p_5y + p_6 \quad (3.5)$$

La figura 3.5, muestra la superficie de ajuste aplicada a la imagen de la figura 3.4

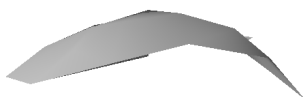


Figura 3.5: Discretización de una superficie mediante una malla de triángulos

Como se requiere evaluar las derivadas en un punto v , entonces solo es necesario evaluar las derivadas de $f(x, y)$ en ese punto. Sipirán propone integrar las derivadas en una vecindad Gaussiana, para obtener el grado en que varían el punto v en relación a su vecindad. Las ecuaciones 3.6, 3.7 y 3.8 representan las derivadas con respecto a x , con respecto a y y con respecto a x y a y a la vez.

$$A = \frac{1}{\sqrt{2\pi\sigma}} \int_R e^{-\frac{(x^2+y^2)}{2\sigma^2}} \cdot f_x(x, y)^2 dx dy \quad (3.6)$$

$$B = \frac{1}{\sqrt{2\pi\sigma}} \int_R e^{-\frac{(x^2+y^2)}{2\sigma^2}} \cdot f_y(x, y)^2 dx dy \quad (3.7)$$

$$C = \frac{1}{\sqrt{2\pi\sigma}} \int_R e^{-\frac{(x^2+y^2)}{2\sigma^2}} \cdot f_x(x, y) \cdot f_y(x, y)^2 dx dy \quad (3.8)$$

Resolviendo las integrales y derivadas, se obtiene los siguientes valores:

$$A = p_4^2 + 2p_1^2 + 2p_2^2 \quad (3.9)$$

$$B = p_5^2 + 2p_2^2 + 2p_3^2 \quad (3.10)$$

$$C = p_4p_5 + p_1p_2 + p_2p_3 \quad (3.11)$$

Finalmente, al igual que *Harris 2D*, se llega a la siguiente ecuación:

$$E = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \quad (3.12)$$

Donde A representa el cambio promedio que genera el vértice v con respecto a sus vecinos en dirección x , B es el cambio promedio que genera el vértice v con respecto a sus vecinos en dirección y y C es el cambio promedio que genera el vértice v con respecto a sus vecinos en dirección x y y .

Luego, el valor del operador de Harris es calculado como en 3.4. Finalmente, se toman aquellos k vértices, cuyos valores absolutos de Harris son los mayores. Estos k vértices resultantes son los puntos de interés o *key points*.

En la figura 3.6 se observa un ejemplo de aplicar el operador de Harris 3D y obtener los 100 mejores *key points*. En la figura, (a) es el modelo 3D, cuya posición representa la mayor simetría extrínseca, (b) es el mismo modelo con cambios isométricos y, (c) representa cambios topológicos. Como se puede observar, los *key points* se mantienen robustos, independiente de los cambios isométricos, topológicos, e incluso en cambios de escala, presencia de hoyos, o ruido, como lo demuestran en [SB11]

3.1.3 Detección de puntos de interés basados en la escala (HKS)

Supongamos que tenemos una superficie M , a la cual se le aplica una fuente de calor en un punto p de la superficie M . El problema de encontrar el promedio de calor

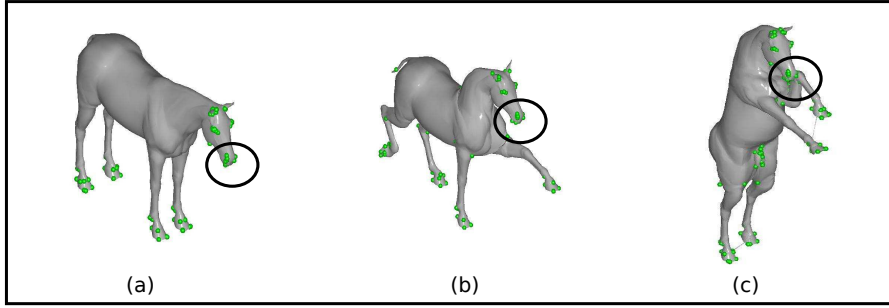


Figura 3.6: (a) Forma NULL (b) Cambios isométrico (c) Cambio Topológico

que se disipa en un punto p luego de un tiempo t se representa mediante la ecuación diferencial del calor, cuya solución se denomina *Heat Kernel*. Formalmente, se dice que si $u(p, t)$ es el promedio de calor en un punto p luego de un tiempo t , entonces el *heat kernel* esta definido como en la ecuación 3.13.

$$u(p, t) = \frac{\partial u}{\partial t} = \Delta u \quad (3.13)$$

Donde Δu es el operador de Laplace-Beltrami [DK10].

En [SOG09] se propone el *Heat Kernel Signature* con una restricción en el dominio del tiempo del *Heat Kernel*, de modo que se pueda obtener una firma para cada vértice del grafo que representa el modelo.

La siguiente ecuación representa el *Heat kernel Signature*

$$HKS(x, t) = \sum_{i=0}^{\infty} e^{-\lambda_i t} \phi_i^2(x) \quad (3.14)$$

En la ecuación 3.14, λ_i y $\phi_i(x)$ son los auto vectores y auto valores del operador Laplace-Beltrami.

Finalmente, un vértices es seleccionado como punto de interés, cuando, para valores altos de tiempo, su firma es máxima con respecto a sus vértices vecinos. [OSG08].

Se debe tener en cuenta que el costo computacional, para encontrar los puntos de interés, mediante este método es alto. Esto debido a la eigen descomposición del operador de Laplace-Beltrami cuya complejidad computacional tiene a ser $O(kn^2)$, donde k es el número de auto vectores y auto valores a computar y n es el número de vértices la malla que representa el modelo. Por otro lado, dado que el número de vértices en las mallas originales son demasiado grandes, éstas deben ser reducidas para poder realizar el cómputo en un tiempo razonable.

Para propósitos de evaluación, dos variantes de *Heat Kernel Signature* fueron utilizados, las cuales, se describen a continuación.

- *HKS2*: El software *The Geomagic* fue utilizado para remover aristas *non-manifold*, y los triángulos se consideraron con orientación. Los resultados de esta variante fueron tomados de ⁱ.
- *HKS3*: Un proceso de filtrado fue utilizado para eliminar los puntos de interés poco estables. Para esto, se mantuvo aquellos puntos de interés con homólogos persistentes. Los resultados fueron tomados de [BBB⁺10a].

3.2 Propuesta y Metodología

En esta investigación, se propone un nuevo método para detección de puntos de interés eficiente. A diferencia de otros métodos más complejos propuestos recientemente, nosotros solo requerimos calcular el área de los triángulos que forman la malla del modelo 3D y, luego se ordenan de manera creciente. Nuestro algoritmo, al que denominamos, *Minimal area interest point*, como los otros, es robusto a cambios isométricos, topológicos, presencia de ruido y hoyos, entre otros. Con la ventaja adicional, de ser simple y lineal respecto al número de triángulos.

La metodología seguida se resumen en la figura 3.7.

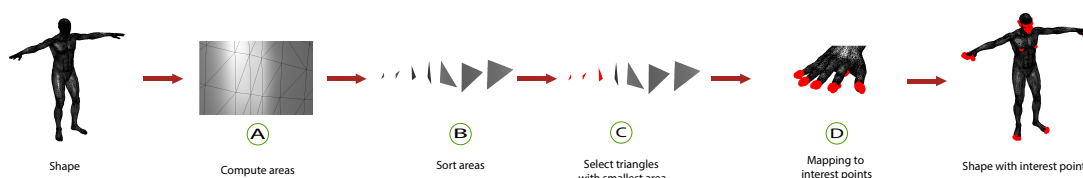


Figura 3.7: Metodología

3.2.1 Cálculo de áreas

Sea M una malla triangular, que representa un modelo, donde $v_1, v_2, v_3 \in R^3$ conforman un triángulo en M . El área se define como:

$$area = \frac{\det(A)}{2} \quad (3.15)$$

ⁱ<http://www.geomtop.org/sunjian/software/hks.html>

where

$$A = \begin{vmatrix} v_1(x) & v_1(y) & v_1(z) \\ v_2(x) & v_2(y) & v_2(z) \\ v_3(x) & v_3(y) & v_3(z) \end{vmatrix}$$

Para cada triángulo, se computa el área de acuerdo a la ecuación 3.15

3.2.2 Ordenamiento y selección de triángulos con áreas pequeñas

Dado que los triángulos con pequeñas áreas, están localizadas en zona de altas protusiones, entonces, los vértices que componen esos triángulos deben ser puntos de interés. Nuestros experimentos, evidencian que esta presunción es correcta, como se podrá ver la sección 3.3.

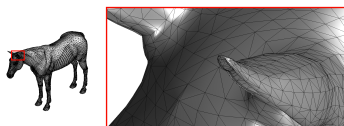


Figura 3.8: Triangulación (tamaño de los triángulos)

La figura 3.8, muestra que los triángulos que están en zonas de alta protusión (orejas del caballo), tienen áreas pequeñas, respecto a los triángulos que están en zonas planas, como la cabeza del caballo.

Nuestro algoritmo, ordena todos los triángulos de la malla, en orden creciente y seleccionamos los k primeros triángulos, posteriormente se utiliza un umbral t , para determinar en número de triángulos que se tomarán para formar los *key points*.

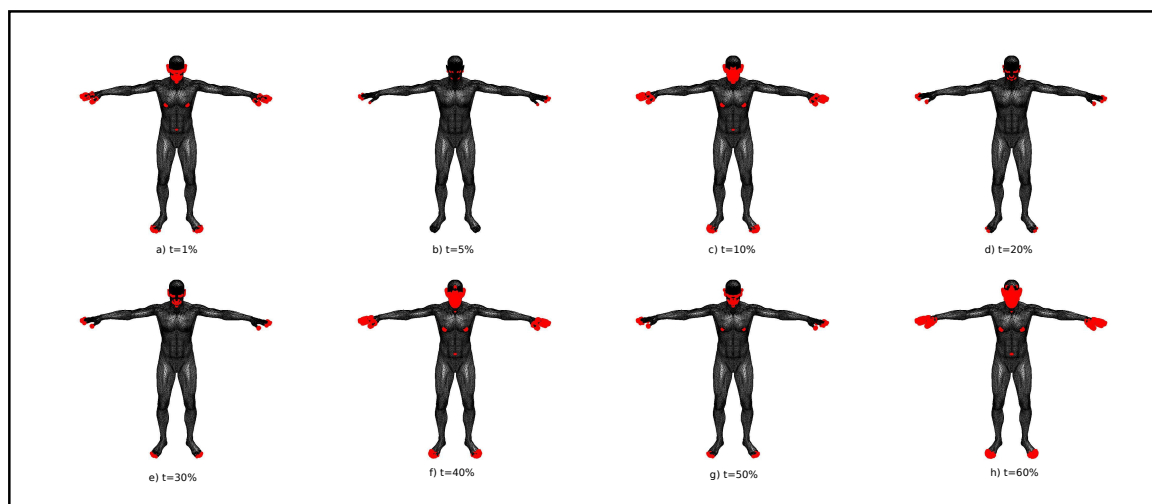


Figura 3.9: Modelos con *Key points* para diferentes valores de t

La figura 3.9 muestra distintos conjuntos de puntos de interés que dependen de un umbral t , para el modelo humano. Los puntos de interés están representados por puntos rojos.

3.2.3 Detección de puntos de interés

Para el cálculo de los puntos de interés, se computan todos los vértices que conforman los triángulos, de modo que, los vértices sean únicos; estos vértices resultantes son los puntos de interés.

3.3 Experimentos

En esta sección, se describen los experimentos realizados con el método propuesto y se realizan comparaciones con otros métodos del estado del arte.

3.3.1 El conjunto de datos

Se utilizó como base de datos el *benchmark* de detección y descripción de características robustas **SHREC** [BBB⁺10a]. Esta base de datos esta compuesta por 3 modelos (Humano, perro y caballo) representados como una malla de triángulos con aproximadamente 10,000 – 50,000 vértices. Para cada modelo, 9 transformaciones fueron aplicadas. Las transformaciones son: isometría (deformaciones no rígidas y elásticas), topología (unión de vértices de modo que se obtiene una triangulación diferente), micro hoyos, hoyos grandes, escala local y global, ruido gaussiano aditivo, ruido de disparo y disminución de resolución.

Los niveles de intensidad de la transformación fueron de cinco, y el número más alto representan transformación con mayor intensidad, a excepción de la clase isometría, la cual presente 5 cambios isométricos distintos. Finalmente, el número total de modelos es de 138 y para cada modelo existen, un total de 45 transformaciones.

3.3.2 Evaluación Metodológica

Sea $F(Y) = \{y_k\}_k$, un conjunto de puntos, los cuales son detectados por un método de detección de puntos de interés. Entonces, para medir la calidad de los puntos de interés se utiliza el criterio de repetibilidad [SB11].

Sea un modelo X conocido como *null Shape*, el cual es utilizado para evaluar la tasa de aciertos del método de detección de puntos de interés, y sea Y un modelo transformado de X , entonces, se espera que los puntos de interés encontrados en Y sean similares a los puntos de interés en X . Luego, para cada par de puntos $C_0(X, Y) = \{(y'_k, x_k)\}_{k=1}^{|Y|}$, un punto de interés en $y_k \in F(Y)$ se dice que es repetible si una bola geodésica de radio R al rededor del punto correspondiente $x'_k : (x'_k, y_k) \in C_0(X, Y)$, contiene un punto característico $x_j \in F(X)$. El subconjunto $Fr(Y) \subset F(Y)$ de puntos de interés repetibles esta dado por $F_{R,X}(Y) = \{y_k \in F(Y) : F(X) \cap B_R(x'_k) \neq \emptyset, (x'_k, y_k) \in C_0(X, Y)\}$, donde $B_R(x'_k) = \{x \in X : d_X(x, x'_k) \leq R\}$ y d_X denota una función de distancias geodésica en X .

La repetibilidad $rep(Y, X)$ de $F(Y)$ en X se definida como el porcentaje de puntos de interés de $F(Y)$ que son repetibles,

$$rep(X, Y) = \frac{F_{R,X}(Y)}{F(Y)}.$$

en un modelo transformado Y y en su correspondiente modelo null X . La calidad de la detección de características fue medida como $(rep(Y, X) + rep(X, Y))/2$. El valor de $R = 5$ fué utilizado en el benchmark. Este radio constituye aproximadamente el 1% del diámetro del modelo. Las características sin correspondencia con el modelo de evaluación (modelo null) son ignorados.

3.3.3 Resultado y comparación con otros métodos

Los experimentos fueron realizados en una máquina intel core-i7, con 12G de memoria RAM y, a diferencia del método *HKS*, los modelos no fueron sometidos a un proceso de reducción, es decir, se utilizaron las malla originales. Las mallas cuentan con entre 10000 y 50000 vértices.

En la figura 3.10, (a) representa el modelo Null, es decir, el modelo sin transformación, (b) el modelo con grandes hoyos, (c) el modelo con un cambio isométrico, (d) el modelo con una reducción de vértices, (e) modelo con ruido, (f) Modelo con cambios topológicos, (g) modelos con micro hoyos, (h) modelos con cambio de escala e (i)

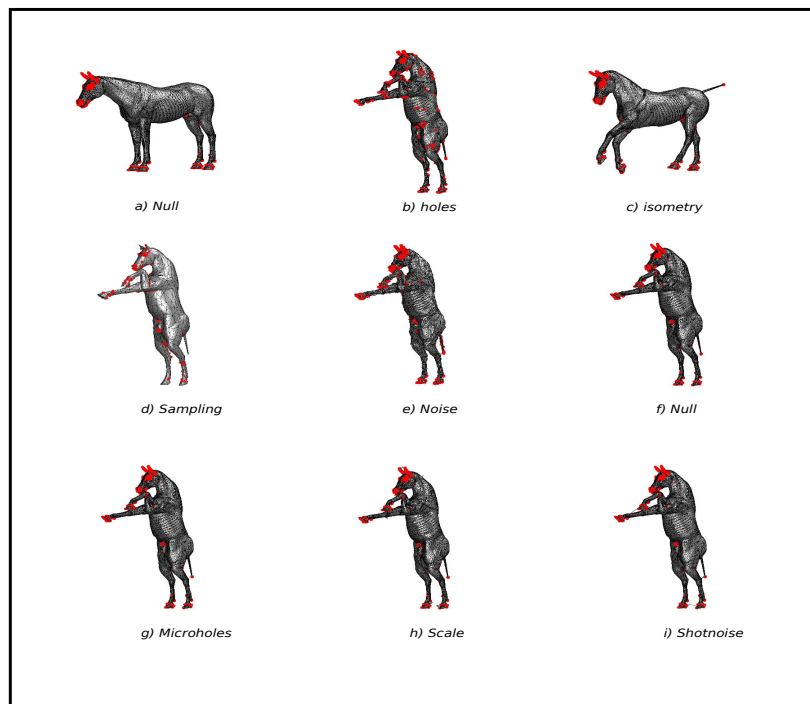


Figura 3.10: Transformación del modelo caballo con un nivel de intensidad de 5 y $t = 20\%$

modelo con ruidos simples. En cada una de las imágenes se muestran los puntos de interés, de color rojo, detectados con nuestro método.

En la figura 3.10, también se puede observar que los puntos de interés detectados en cada uno de los modelos transformados son similares al modelo null; salvo en el caso de la imagen (b) (hoyos grandes), donde los puntos de interés se agrupan cerca a los hoyos del modelo.

Las tablas 3.2,3.3 y 3.4, son los resultados de repetibilidad para los algoritmos Heat Kernel Signature 2, algoritmos Heat Kernel Signature 3 y 3D Harris respectivamente.

Transformación	Intensidad					Promedio
	1	≤ 2	≤ 3	≤ 4	≤ 5	
Holes	91.48	90.60	86.78	83.73	81.86	86.89
Isometry	98.08	98.72	98.01	97.88	98.04	98.15
Local Scale	98.08	94.83	90.09	83.05	78.31	88.87
Micro Holes	98.08	96.69	96.00	95.52	94.87	96.23
Noise	95.30	92.78	91.67	89.24	87.62	91.32
Sampling	97.05	97.88	97.39	96.27	92.35	96.19
Scale	99.36	99.36	98.50	97.90	97.68	98.56
Shot Noise	98.08	96.22	93.39	90.45	87.32	93.09
Topology	97.44	96.10	92.26	91.22	88.64	93.13
Promedio	96.99	95.91	93.79	91.70	89.63	93.60

Table 3.2: Repetibilidad del algoritmo de detección de puntos de interés HKS2 [SB11].

En tabla 3.2 se observa los resultados, de repetibilidad promedio, correspondientes

al algoritmo *Heat Kernel Signature 2*. Aquí se puede observa que *HKS2* presenta buenos resultados, superando en promedios el 90% de repetibilidad en la mayoría de los casos.

Transformación	Intensidad					Promedio
	1	≤ 2	≤ 3	≤ 4	≤ 5	
Holes	80.54	79.00	75.25	72.10	69.99	75.38
Isometry	100.00	100.00	100.00	100.00	100.00	100.00
Local Scale	97.44	96.79	93.02	87.25	82.90	91.48
Micro Holes	100.00	100.00	98.15	96.58	95.64	98.07
Noise	100.00	95.19	93.16	89.37	85.77	92.70
Sampling	100.00	100.00	100.00	100.00	96.20	99.24
Scale	100.00	100.00	100.00	98.61	97.78	99.28
Shot Noise	100.00	95.30	90.03	82.10	74.38	88.36
Topology	94.44	90.38	87.45	88.70	85.76	89.35
Promedio	96.94	95.18	93.01	90.52	87.60	92.65

Table 3.3: Repetibilidad del algoritmo de detección de puntos de interés HKS3 [SB11].

En tabla 3.3 se muestra los resultados de repetibilidad promedio, correspondientes al algoritmo *Heat Kernel Signature 3*. A diferencia de *HKS2*, el *HKS3*, llegan al 100% de repetibilidad en varios de los experimentos, sin embargo, el comportamiento promedio es similar al *HKS2*.

Una de las desventajas de ambos algoritmos, es la necesidad de realizar la eigén descomposición del operador Laplace-Beltrami, cuya complejidad computacional, es de $O(kn^2)$, donde k representa el número de eigen vectores a utilizar y, n es el número de vértices de la malla que representa el modelo. Por otro lado, en la práctica solo se requiere un número pequeño de eigen vectores, entre 15 a 20 y, por lo tanto, el costo computacional de estos algoritmos tienden a n^2 . Sin embargo, dado que el número de vértices de las mallas llegan hasta 500000, se hace necesario realizar una reducción del tamaño de las mallas. Este proceso previo, supone una de las principales desventajas de estos dos algoritmos.

En la tabla 3.4 se observan los resultados de repetibilidad promedio, correspondientes al algoritmo *3D Harris*. Los resultados muestran que en algunos casos, como en isometría, scale y sampling, los resultados no son superiores a HKS3, sin embargo, en promedio, *Harris 3D*, tiene resultados ligeramente superiores. Lo importante en relación a las anteriores propuestas, es que Harris 3D, tiene un costo computacional de orden lineal respecto a los vértices del modelo y, en tal sentido, no requiere la simplificación de las mallas. Sin embargo, el algoritmo, es ligeramente más complejo que nuestra propuesta, y requiere el cálculo del análisis de componentes principales para cada matriz formada por un vértices y sus vecinos; además, de generar la superficie de ajuste a cada *r-ring*. Por lo tanto, el costo computacional de este algoritmo, aunque lineal, es mayor que nuestra propuesta, la cual es más simple y eficiente.

Transformación	Intensidad					Promedio
	1	≤ 2	≤ 3	≤ 4	≤ 5	
Transform	1	≤ 2	≤ 3	≤ 4	≤ 5	AVG
Holes	94.62	94.43	94.10	94.01	93.81	94.19
Isometry	96.01	96.73	96.26	96.60	96.62	96.44
Local Scale	96.24	94.96	93.40	91.26	88.84	92.94
Micro Holes	96.01	96.01	95.98	95.96	95.95	95.98
Noise	93.09	92.58	91.59	90.33	88.79	91.28
Sampling	95.31	93.62	92.08	89.13	80.42	90.11
Scale	97.06	96.89	96.28	95.62	94.94	96.16
Shot Noise	96.03	95.66	95.00	93.83	92.79	94.66
Topology	96.01	95.97	95.82	95.73	95.71	95.85
Promedio	95.60	95.21	94.50	93.61	91.99	94.18

Table 3.4: Repetibilidad del algoritmo de detección de puntos de interés H3D [SB11].

Transformación	Intensidad					Promedio
	1	≤ 2	≤ 3	≤ 4	≤ 5	
Holes	94.12	93.10	92.67	88.27	78.21	89.27
Isometry	97.53	96.90	94.66	96.43	94.24	95.95
Local Scale	88.87	82.00	69.26	50.89	45.77	67.36
Micro Holes	97.53	97.52	97.51	97.51	97.47	97.51
Noise	88.63	75.17	67.89	63.61	60.33	71.13
Sampling	40.46	19.90	9.08	3.76	0.76	14.79
Scale	97.55	97.53	97.52	97.52	97.54	97.53
Shot Noise	97.51	97.52	97.52	97.49	97.45	97.50
Topology	97.52	97.53	97.53	97.52	97.53	97.53
Promedio	88.86	84.13	80.40	77.00	74.37	80.95

Table 3.5: Repetibilidad promedio de nuestro método. Número promedio de triángulos con menor área recuperada: 19670.

En tabla 3.5 se muestra los resultados de repetibilidad de nuestro método. En la tabla se puede apreciar que, los resultados son cercanos al 90% en la mayoría de casos y, aunque en promedio, son menores a los métodos de *3D Harris*, *HKS2* y *HKS3*, la simplicidad de la propuesta sumada a la complejidad computacional bastante menor que las propuestas anteriores, hacen de nuestro algoritmo, eficiente y con resultados razonables para ser aplicado en tareas subsecuentes.

Finalmente, en la tabla 3.6, se observa una comparación entre los 4 métodos. Se puede notar que los mejores resultados, en la mayoría de los casos, son proporcionados por *HKS3*. Sin embargo, para todos los modelos como *holes* (grandes ojos), los mejores resultados son obtenidos de aplicar el algoritmo *H3D*. Por otro lado, nuestro algoritmo, tuvo un mejor desempeño en todos los modelos con cambios topológicos. Estos resultados indican que los tres métodos dan buenos resultados en generar, pero sus mejores resultados dependen del tipo de transformación aplicada al modelo.

Un análisis distinto, tiene que ver con los modelos *sampling* y *holes*; que tuvieron los

Transformación	Intensidad					Promedio
	1	≤ 2	≤ 3	≤ 4	≤ 5	
Holes	H3D	H3D	H3D	H3D	H3D	H3D
Isometry	HKS3	HKS3	HKS3	HKS3	HKS3	HKS3
Local Scale	HKS2	HKS3	H3D	H3D	H3D	H3D
Micro Holes	HKS3	HKS3	HKS3	MAIP	MAIP	HKS3
Noise	HKS3	HKS3	HKS3	H3D	H3D	HKS3
Sampling	HKS3	HKS3	HKS3	HKS3	HKS3	HKS3
Scale	HKS3	HKS3	HKS3	HKS3	HKS3	HKS3
Shot Noise	HKS3	MAIP	MAIP	MAIP	MAIP	MAIP
Topology	MAIP	MAIP	MAIP	MAIP	MAIP	MAIP

Table 3.6: Método con la mejor performance para transformaciones de modelos e intensidades. HKS1 y HKS2 (*Heat Kernel Signature*). H3D (*H3D-Harris 3D*). y MAIP (*Minimun Area interest points*).

peores resultados con nuestro algoritmo. En el caso de *sampling*, con intensidades mayores 3, la cantidad de vértices son disminuidos hasta en un 70%. Esto afecta la triangulación y, por lo tanto, el área de los triángulos cambia drásticamente, decrementando la eficiencia de nuestro método.

Por otro lado, en el caso de *holes* se observa, por ejemplo; que los puntos de interés se agrupan cerca a los hoyos. Esto se debe a que los triángulos, cerca a los hoyos, son bastante pequeños, y ,en tal sentido, son seleccionados para tomar sus vértices como puntos de interés.

Sin embargo, para las otras transformaciones ha demostrado tener resultados muy cercanos a los otros algoritmos, e incluso superiores en algunas transformaciones como en los casos de *topology* y *Shot Noise*.

3.4 Conclusión

Se presentó un método para detección de puntos de interés en modelos 3D. Nuestros experimentos confirman que, efectivamente, los triángulos con valores de áreas pequeñas, corresponden a zonas de alta protusión, y los vértices que conforman esos triángulos son considerados puntos de interés. Por otro lado, nuestro algoritmo, es muy simple y eficiente. Una ventaja importante es que no es necesario realizar cálculos complejos y el costo computacional es lineal respecto al número de triángulos en el modelo.

En las tablas de comparación de la sección anterior, se observa que nuestro método es robusto a cambios topológicos e isompetricos, así como, a la presencia de ruido, escala, entre otro. Más aún, nuestros resultados son similares a los de *3D Harris* y

HKS. Por otro lado, podemos ver, según la tabla 3.6, que tenemos mejores resultados en las transformaciones de *micro holes*, *shot noise* y *topology*.

Es importante, tener en cuenta que, la presencia de *holes*(grandes hoyos), cambian drásticamente el tamaños en las triángulos donde se presenta los hoyos y; por lo tanto, decrementa, la posibilidad de detectar correctamente los puntos de interés.

4

DETECCIÓN DE *Key Components* MEDIANTE PROCESOS DE DIFUSIÓN

En este capítulo se describe el método propuesto para encontrar los *key components* en modelos 3D.

El objetivo de la propuesta es descomponer una malla 3D en un conjunto de k regiones, donde cada región k_i sea representativa del modelo. Estos componentes o regiones tienen más ventajas que los *key points* debido a que son menos sensibles a deformaciones como el ruido. Por otro lado, el número de componentes es inferior comparado a representaciones locales como es el caso de los *key points*, lo cual permite realizar otras tareas subsecuentes a menor costo. [SB12]

4.1 Aspectos Preliminares

La descomposición de mallas beneficia a muchas aplicaciones. En metamorfosis [GSL⁺99, ZSH00, STK02], la descomposición de mallas es utilizada para establecer correspondencias. La compresión [KG00] y simplificación [GH97] hace uso de la descomposición para mejorar el ratio de la compresión. En recuperación de modelos 3D, la descomposición de grafos sirve como una firma invariante a cambios no rígidos [ZTS02]. En la detección de colisiones, la descomposición facilita el cálculo de los límites del volumen para determinar eficientemente el lugar de colisión y la deformación subsecuente [LWTH01], otras aplicaciones incluyen modificación y modelado de partes y, en esta tesis, en particular, es utilizada para detectar simetría intrínseca en modelos 3D.

En este capítulo se describe un nuevo método para encontrar *key components* basados en procesos de difusión. A cada vértice del grafo que representa el modelo, se le aplica una función de difusión obteniendo un vector *k-dimensional* que representa, de manera única, cada vértice. A estos vectores se les conocen como firma del vértice. A continuación, se recupera solo aquellos vértices cuya valor absoluto, del módulo de sus firmas, sea mayor a un umbral. Finalmente, los vértices que se obtienen son los que conforman los *key components*.

En los siguientes secciones se describen algunos trabajos relacionados, el método propuesto, y finalmente las evaluaciones de nuestro método respecto al estado del arte.

4.2 Trabajos relacionados

Una de las primeras técnicas para la descomposición de mallas, basada en el manejo de sus características, fue propuesta por Mortara et al. [MPS⁺04]. El método, estima la curvatura de un vértice con respecto a una vecindad variable y además, toma en cuenta la topología de la vecindad. Para esto, utiliza una esfera con crecimiento variable en cada vértice y estudia la forma en que se da la intersección entre la esfera y la superficie que envuelve. Por otro lado, Huang et al [HWAG09] propusieron la descomposición de modelos 3D basado en análisis modal tomando la eigen descomposición de la matriz Hessiana de una función energía definida en la malla, donde la matriz Hessiana es una matriz cuadrada formada por las derivadas parciales de segundo orden de una función y esta describe la curvatura local de una función de varias variables.

Otro método de descomposición, fue propuesta por [KT03] y se denomina "*Mesh decomposition using fuzzy clustering and cuts*". La idea de este algoritmo es encontrar componentes significativos utilizando algoritmos de agrupamiento como el *kmeans*, mientras se mantiene los límites entre los componentes de manera difusa. Este algoritmo se centra en encontrar pequeñas área difusas y busca los límites exactos mientras se expande a lo largo del modelo. Por otro lado, el método de [KLT05] denominado "*Mesh segmentation using feature point and core extraction*", propone utilizar el algoritmo *Multi-dimensional scaling* (MSD) para transformar los vértices de la malla, de modo tal que, el modelo se encuentre en una postura invariante a la representación; luego, los puntos de alta protusión o puntos de interés son extraídos. Finalmente, el componente principal de la malla es encontrado; es decir, el componente donde los puntos de interés proveen suficiente información para obtener otros componentes significativos.

Estos métodos, y muchos otros, segmentan el modelo completo en un subconjunto de partes, de modo que la unión de los subconjuntos genera el objeto global. Sin embargo, nosotros estamos interesados en encontrar una subconjunto de componentes significativos, y no segmentar el objeto completo en todos sus componentes. Sipirán y Bustos propusieron el método denominado "*Key-Component Detection On 3D Meshes Using Local Features*" [SB12] y fueron los primeros en acuñar el término *key components* o componentes clave, para referirse a este subconjunto de componentes que preservan la mayoría de las características significativas del modelo.

En la sección 4.2.1 se explica el método propuesto por [SB12], en la sección 4.3 se describe nuestra propuesta, en la sección 4.5 se detallan los experimentos y resultados

y finalmente, en la sección 4.5 se presentan las conclusiones de este capítulo.

4.2.1 Detección de *Key Components*

Los *key points* o puntos de interés nos permiten obtener descriptores de imágenes invariantes al ruido, rotación y escala. Sin embargo, el número de *key points* puede ser demasiado alto como para poder, por ejemplo, encontrar correspondencia entre objetos 3D. Por otro lado, como se aprecia en la figura 4.1, existen algunos *key points* que están muy cercanos entre sí, mientras que otros están aislados. Esta idea permite generar grupos de *key points* y obtener sub mallas de la estructura, de modo que en lugar de trabajar con *key points*, ahora trabajamos con sub mallas, a las cuales *Sipirán y Bustos* las denominaron *key components* o componentes clave [SB12] y, estos componentes son aquellos que presentan mayores cambios dentro de la estructura.

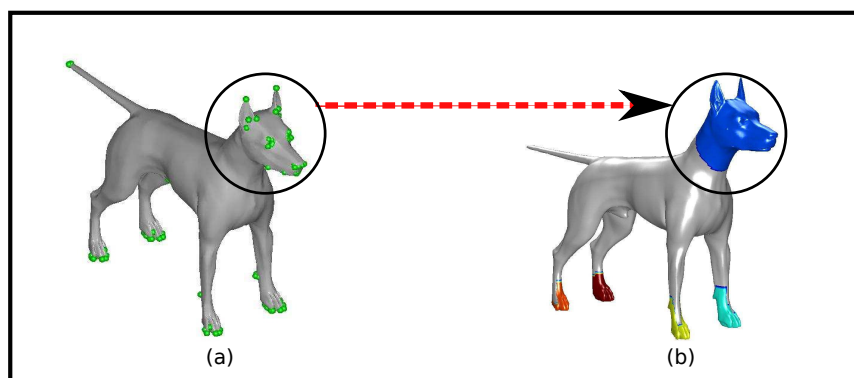


Figura 4.1: (a) *key points*, (b) *Key componentes* obtenidos a partir de (a) [SB12]

En la figura 4.1 (a) se aprecia los *key points*, de color verde, localizados en la superficie del modelo 3D. Como se puede ver en la figura, estos puntos aparecen en lugar de mayor cambio en la estructura. Por otro lado, también se puede observar que varios puntos se encuentran geodésicamente cercanos entre sí, estos son agrupados en componentes como en la figura (b).

El método propuesto por [SB12], en primer lugar, encuentra *key points* o puntos de interés, haciendo uso del algoritmo, propuesto por el mismo autor, y descrito en la sección 3.1.2 del capítulo 3. Una vez, seleccionados los puntos de interés, estos son agrupados de acuerdo a su concentración y, teniendo en cuenta, altas características locales. Una forma para medir las características locales es por medio de la distancia geodésica entre los puntos de interés y, agruparlos de acuerdo a su cercanía geodésica.

Sea $S = s_1, s_2, \dots, s_n$ el conjunto de *key points* previamente encontrados, el objetivo es hallar una partición tal que $S_i \in S, i = 1 \dots m$ de acuerdo a las siguientes propiedades.

1. $d_{geod}(x, y) \leq T_0, \forall x, y \in S_i$
2. $d_{geod}(x, y) \geq T_p, \forall x \in S_i, \forall y \in S_j, i \neq j$
3. $\cup_{i=1}^m S_i \subseteq S$.
4. $S_i \cap S_j = \emptyset, i \neq j$

La propiedad 1, indica que la distancia geodésica entre dos puntos x, y pertenecientes a un componente debe ser menor a un valor T_0 , esto permite delimitar el tamaño de los *key components*. La propiedad 2, indica que la distancia geodésica entre todo punto x de un componente, a todo punto y de otro componente, debe ser lo suficientemente grande para considerar que ambos componentes son distintos. Para lograr este propósito, se utiliza como umbral el valor T_p . La propiedad 3, indica que la unión de todos los *key components* deben dar como resultado un subconjunto del modelo completo. Finalmente, la propiedad 4 nos indica que no debe haber intersección entre componentes.

Dado que el método se basa en analizar las distancias geodésicas, los autores proponen una técnica de agrupamiento que tenga en cuenta la distancia geodésica entre los puntos de interés. Para lograr este objetivo, el método calcula un conjunto $P \in \mathbb{R}^2$, de modo que, las distancias euclídeas entre los puntos pertenecientes a P preserven las distancias geodésicas entre los elementos en S . Entonces, se requiere encontrar el conjunto P , tal que:

$$P = \arg \min_{p_1, \dots, p_n} \sum_{i < j} (||p_i - p_j|| - d_{geod}(s_i, s_j)) \quad (4.1)$$

donde cada $p_i \in \mathbb{R}$ corresponde al *key points* S_i .

Para computar estos puntos se utiliza el método "*Multidimensional Scaling*" propuesto por [BG05], el cual embebe un conjunto de puntos de una dimensión en otra. Para mantener las distancia geodésicas se utiliza algoritmo de Dijkstra considerando la malla como un grafo.

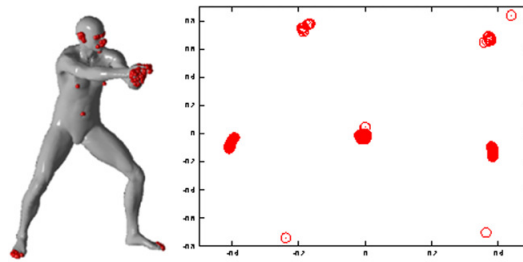


Figura 4.2: (a) *key points*, (b) *Key components* obtenidos a partir de (a) [SB12]

En la figura 4.2, se observa el resultado de calcular el conjunto de puntos en un espacio bidimensional, manteniendo las distancias geodésicas entre los *keypoints* del modelo.

Para encontrar los *key components*, los autores utilizan el algoritmo propuesto por [LL04]. El algoritmo, de partición, tiene en cuenta el número de elementos que hay en cada componente; de modo que, componentes con un número pequeño de puntos son eliminados, con el objetivo de evitar la formación de componentes que no aporten información importante sobre la estructura de la malla.

En la figura 4.3, se puede apreciar el resultado de obtener los *key componentes* luego de aplicar el algoritmo de agrupamiento..

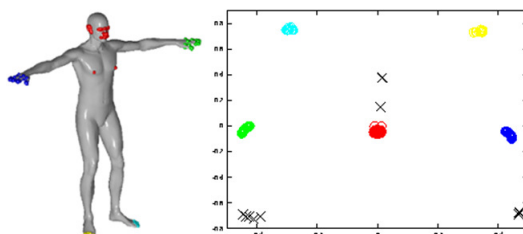


Figura 4.3: (a) *key points*, (b) *Key componentes* obtenidos a partir de (a) [SB12]

Finalmente, en la figura 4.4 presentamos los resultados de aplicar el algoritmo de Sipirán, al modelo Humano con transformaciones isométricas, topológicas, presencia de ruido, escala, hoyos, entre otros.

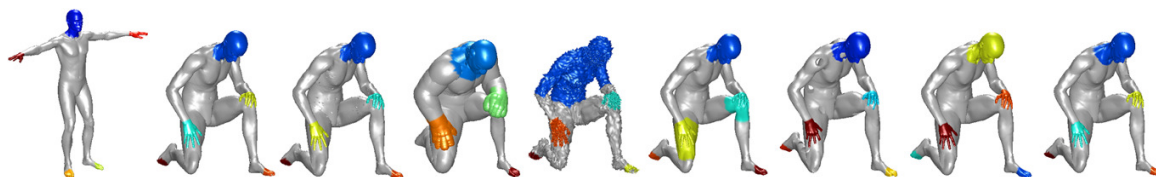


Figura 4.4: (a) *key points*, (b) *Key componentes* obtenidos a partir de (a) [SB12]

Una de las desventajas de este método es que requiere, en primer lugar, detectar los *key points* o puntos de interés, y como se describió en el capítulo anterior, para encontrar buenos *key points*, se requiere varios parámetros de entrada. Por lo tanto, una buena detección de *key Components*, requiere una etapa de preprocesamiento previa, y si esta estaba no es lo suficientemente buena, la detección de *key components* no será la adecuada.

En la siguiente sección, se presenta nuestra propuesta, la cual no requiere de etapas de preprocesamiento previa.

4.3 Propuesta y Metodología

En esta sección se describe la metodología de nuestro método, que a diferencia de la propuesta de Sipirán y Bustos, no requiere el cálculo previo de *key points*. Nuestra propuesta computa, directamente, los *key components* mediante técnicas de difusión y algoritmos para la detección de componentes fuertemente conexos. Esta es una ventaja, debido a que no se requiere configurar muchos parámetros para encontrar componentes estables. En la figura 4.5 se muestran cada una de las etapas para el cálculo de los componentes.

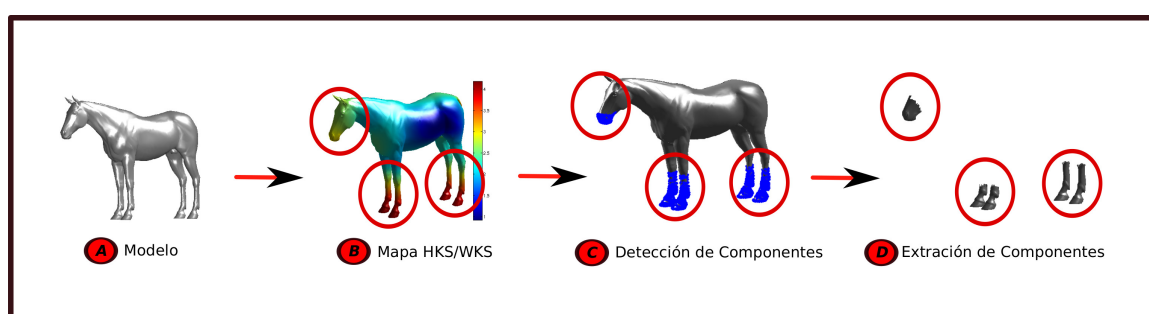


Figura 4.5: Metodología

4.3.1 Modelo

El modelo de entrada es una malla de triángulos que representa una estructura tridimensional. La malla de triángulos será representada mediante un grafo como estructura de datos, al cual le denominaremos grafo $G(V, F)$, donde V es el conjunto de todos los vértices en el grafo y F es el conjunto de todas las caras de los triángulos de la malla. El grafo G , es la entrada de la siguiente etapa, cálculo de *Heat Kernel Signature* o el *Wave Kernel Signature*, dependiendo de la técnica (HKS/WKS) para hallar los *key components*.

4.3.2 Mapa HKS/WKS

En esta segunda etapa, se utiliza el operador de *Laplace-Beltrami* y su correspondiente eigen descomposición a fin de obtener el espectro de grafo $G(V, F)$. La eigen descomposición del grafo G , permite obtener los k primeros auto valores y auto vectores.

Estos auto vectores y auto valores son utilizados para el computo del *Heat Kernel Signature* o el *Wave Kernel Signature* mediante las fórmulas 2.7 y 2.9 del capítulo 2.

Al aplicar HKS/WKS en cada vértice v del grafo G , se obtiene un vector T dimensional; donde T representa el tiempo de exposición a la fuente de calor o de onda, dependiendo si se utiliza *HKS* o *WKS*. A este vector se le conoce como firma, dado que dos vértices en posiciones distintas del grafo, tienen grados de difusión de calor, o de onda, distintas, entonces; sus firmas también son diferentes. Para poder ver este comportamiento, optamos por hallar el valor absoluto del módulo de cada vector $T - dimensional$, de modo que ahora tenemos un único valor que representa a cada vértices. Estos valores son mapeados en la imagen (b) de la figura 4.5. En la figura, los colores más cercanos al rojo son las zonas más calientes, mientras, que los colores más cercanos al azul son las zonas más frías. Aquellas partes, en la imagen (b), que están encerradas en un círculo, representan las zonas más calientes. Es decir, vértices cuyo valor absoluto del módulo de sus firmas es mayor y, como se puede apreciar en la imagen, estas zonas corresponden a los componentes clave que representan el modelo.

4.3.3 Detección de Componentes

Dada la observación anterior, en esta etapa utilizamos un umbral para determinar el conjunto de vértices que formarán los componentes. El algoritmo 4.1 describe los pasos para hallar los vértices que conforman los componentes.

Algorithm 4.1: Cálculo de las p firmas más relevantes

input : $G(V, F)$, Umbral U , Número de Eigen Vectores n , op Opción HKS/WKS

output : C vértices del componente, R módulos con mayor valor absoluto

```

1 if  $op == 1$  then
2    $Firmas_{n \times k} \leftarrow \text{computeHKS}(v, f, n)$ ;
3   else ;
4    $Firmas_{n \times k} \leftarrow \text{computeWKS}(v, f, n)$ ;
5  $S \leftarrow \emptyset$ ;
6  $C \leftarrow \emptyset$ ;
7 for each  $v \in G(V, G)$  do
8    $N \leftarrow \text{abs}(\text{norm}(Firma_{n \times k} \in v))$ ;
9   if  $N \geq U$  then
10     $R \leftarrow R \cup N$  ;
11     $C \leftarrow C \cup v$ 
12 return  $(R, C)$ ;
```

En el algoritmo 4.1, la entrada es un grafo $G(V, F)$, donde V son los vértices y F son las caras de los triángulos en la malla, U es el umbral, a partir del cual, los vértices

serán tomados para formar los *key components*; n es el número de auto valores y auto vectores que se utilizarán para el cálculo del HKS/WKS y, finalmente; op es una opción que indica, al programa, si el cálculo de los componentes se hará mediante *Heat Kernel Signature* o *Wave Kernel Signature*. El algoritmo retorna en C el conjunto de vértices que forman parte de los componentes y en R el conjunto de valores para cada vértice. En la imagen (c) de la figura 4.5 se muestra en color azul, todos los vértices con mayor umbral.

4.3.4 Extracción de Componentes

Luego de aplicar el algoritmo 4.1 se sabe cuales vértices formarán parte de los componentes, pero los componentes aun no han sido extraídos de la malla. En esta última etapa, se aplica un algoritmo para calcular componentes fuertemente conexos en el grafo [Len79]. Es decir, dado que cada conjunto de vértices de color azul en la figura 4.5, forman sub grafos del grafo $G(V, F)$, entonces, para poder extraerlos, aplicamos un algoritmo para determinar componentes fuertemente conexos y obtenemos como resultado, la imagen (d) de la figura 4.5.

Algorithm 4.2: Algoritmo de Tarjan. [Len79]

input : Grafo $G(V, F)$ de modelo

output : Conjunto S de componentes fuertemente conexos

```

1 index  $\leftarrow$  0 ;
2 S  $\leftarrow$  empty ;
3 for  $\forall v \in V$  do
4   | if visitado( $v$ )==FALSE then
5   | | strongconnect( $v$ );

```

El algoritmo 4.3 y 4.2 computa los componentes fuertemente conexos en un grafo. Este algoritmo fue propuesto por *Robert Tarjan* [Len79] y es utilizado para separar los *key components* de un modelo.

4.4 Experimentos

En esta sección se explica y evalúa los resultados experimentales de la implementación de nuestro método utilizando un *benchmark* estándar. La presentación de los resultados son divididos en tres etapas. La primera, evalúa la mejor combinación de parámetros, de modo que, podamos analizar el efecto de los parámetros en la repetibilidad de los

Algorithm 4.3: Algoritmo Fuertemente conexos

input : $v \in G(V, F)$
output : Conjunto S de componentes fuertemente conexos

```

1 v.index  $\leftarrow$  index;
2 v.lowlink  $\leftarrow$  index;
3 index  $\leftarrow$  index + 1;
4 S.push(v);
5 for  $\forall(v, w) \in F$  do
6   | if visitado(w) = FALSE then
7     |   strongconnect(w);
8     |   v.lowlink  $\leftarrow$  min(v.lowlink, w.lowlink);
9   | else if  $w \in S$  then
10  |   | v.lowlink  $\leftarrow$  min(v.lowlink, w.index) ;
11 if  $v.\text{lowlink} = v.\text{index}$  then
12 |   repeat  $w \leftarrow S.\text{pop}()$ ;
13 |   until ( $w == v$ )

```

key components. En la segunda etapa, se evalúa dos implementaciones distintas de nuestro método. Una con *Heat Kernel Signature* y la otra, con *Wave Kernel Signature*, con el objetivo de determinar cual de ambos presenta mejores resultados. Finalmente, en la última etapa, se realizan comparaciones de nuestro método, con el trabajo de Iván Sipiran y Benjamín Bustos [SB12].

4.4.1 El conjunto de datos

En nuestros experimentos y, con el fin de evaluar nuestro método, se utilizaron los modelos del *benchmark* SHREC 2010 [BBB⁺10b]. Esta base de datos está compuesta por 3 modelos, los cuales presentan simetría extrínseca (NULL Shapes). Cada modelo, a su vez, tiene un conjunto de modelos obtenidos mediante la aplicación de transformaciones sobre los modelos NULL. Las transformaciones aplicadas a los modelos son isometría, micro agujeros, agujeros grandes, cambios topológicos, ruido plano, escala global y local, y disminución de la resolución.

4.4.2 Criterio de Evaluación

El objetivo, al igual que [SB12], es determinar si los *key components* de los modelos con transformaciones, son consistentes con las *key components* de los modelos sin trans-

formaciones. De manera más formal, si tenemos un modelo X cuyos *key components* son X_1, \dots, X_n , y un modelo Y , resultado de aplicar una transformación a X , con *key components* Y_1, \dots, Y_m , entonces, para cada Y_j se calcula su componente correspondiente X' en X . Luego, se define la repetibilidad entre X y Y de la siguiente manera:

$$R(X, Y) = \sum_{i=1}^m \max_{1 < i \leq n} O(X_i, X'_i) \quad (4.2)$$

Donde la superposición O entre dos componentes esta definida en la fórmula 4.3

$$O(X_i, Y_j) = \frac{A(X_i \cap X'_j)}{A(X_i \cup X'_j)} \quad (4.3)$$

La ecuación 4.3 permite representar de manera formal, el grado o porcentaje de superposición entre un *key components* del modelo sin transformación y un *key component* del modelo transformado. La ecuación 4.2 calcula, la máxima superposición entre los *key components* del modelo Y con el modelo X . Valores cercanos a 1, indican que la superposición entre los *key components* es alta, y por lo tanto, el método sería robusto.

4.4.3 Resultados

En esta sección se describen los experimentos y resultados obtenidos. En cuanto a los experimentos fueron divididos en 3 etapas

Evaluación de Parámetros En esta etapa, se realizaron varios experimentos a fin de determinar la mejor combinación de parámetros. Los parámetros utilizados son *N-Eigen* y *Max-Points*. *N-Eigen*, se refiere al número de autovalores y autovectores que son utilizados para el cálculo del espectro del grafo. *Max-Points*, es el número máximo de vértices que serán tomados para obtener los *key components*, una vez que han sido ordenados de forma decreciente y de acuerdo al valor absoluto del módulo de sus firmas.

Para el parámetro *N-Eigen* se utilizaron los siguiente valores 10,15,20,...,50, y para *Max-Points* se tomaron los valores 1000,1100, 1200, ..., 3000. Se realizaron todas las combinaciones con estos valores y para cada modelo, tanto con *WKS* como con *HKS*. El objetivo es determinar la mejor combinación de parámetros de modo que se obtenga los mejores resultados.

En las figuras 4.6 y 4.7, las imágenes (a), (b) y (c) muestran las comparaciones entre *N-Eigen* y el número de *key components* generados por nuestro método, utilizando *Heat*

Kernel Signature y *Wave Kernel Signature*, respectivamente. En el eje de las abscisas se encuentra las variaciones del parámetro *N-Eigen*; este inicia en 10 auto vectores y se van incrementando de 5 en 5 hasta los 50 autovectores. Números mayores a 50 no mejoran los resultados y, al contrario, decrementan la velocidad de computo. Por otro lado, el eje de las ordenadas, muestra la cantidad de *key components* encontrados por nuestro método.

La imagen (a) son los resultados con el modelo NULL del humano, la imagen (b) son los resultados del modelo NULL del perro y finalmente, la imagen (c) muestra los resultados con el modelo NULL del caballo. Se debe especificar que el modelo NULL, se refiere al modelo original sin ninguna transformación isométrica, topológica o presencia de ruidos.

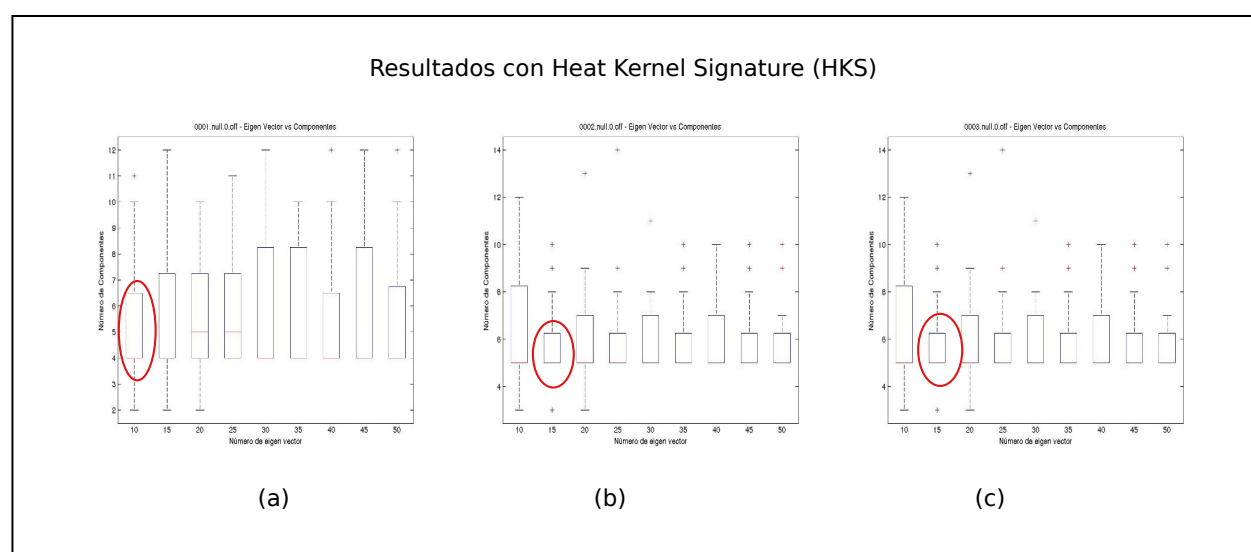


Figura 4.6: *N-Eigen* vs Número de Eigen Vectores

En la figura 4.6, resultados con *HKS* se pueden notar que, en todos los casos (a),(b) y (c), existen una relación entre número de autovectores y *key components*. Es decir, cuando se incrementa en número de autovectores, entonces se incrementa el número de componentes generados por el método. Dado que lo deseable es tener pocos componentes y, que estos sean lo suficientemente significativos y representativos del modelo; entonces, solo es necesario utilizar valores entre 10 a 15 autovectores para obtener una media de 5 y 6 *key components*. Los círculos de color rojo en la figura muestran los mejores resultados, los cuales se dan al seleccionar 15 eigen vectores.

Es importante, también, tener en cuenta que cuanto más autovectores se seleccionen para la generación de los componentes, mayor será el costo computacional y; por otro lado, se generaran muchos componentes, los cuales, no serán útiles para el proceso de identificación de la simetría.

En la figura 4.7, resultados con *WKS*, se observa que los mejores resultados se dan con 15 autovectores, del mismo modo que en el caso del *HKS*.

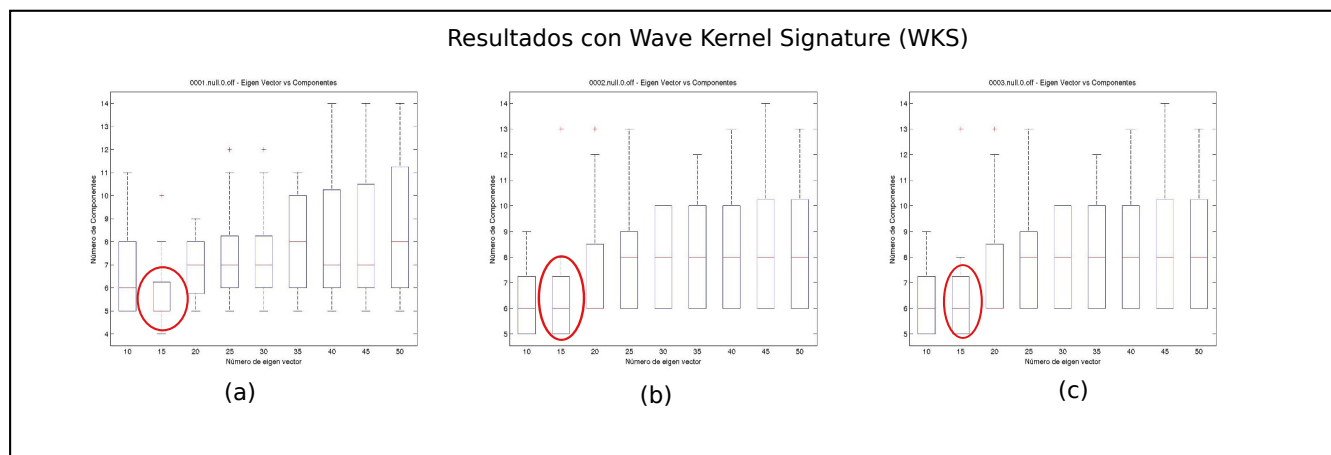


Figura 4.7: *N-Eigen* vs Número de Eigen Vectores

En las figuras 4.8, 4.9 y 4.10, se muestran las comparaciones entre el parámetro *Max-Points* y el número de *key components* generados. En este caso, en el eje de las abscisas se encuentran los diferentes valores que toma *Max-Points*. Estos valores van desde 1000 y van incrementando de 100 en 100 hasta 3000; note que el gráfico las abscisas están etiquetadas con valores de 10, 11, ... 30; sin embargo, se debe aclarar que, estos valores son en realidad 1000, 1100, ... 3000. Se escribió de esta manera por cuestiones de notoriedad y espacio. El eje de las ordenadas representan los componentes generados por nuestro método, con distintos valores de autovectores.

Nuevamente, es posible determinar una relación entre el parámetro *Max-Point* y los *key components*. Valores altos para *Max-Points* generan varios componentes y, debido a que se espera tener pocos componentes; entonces, se opta por utilizar entre 1700 a 1900. Vea los círculos de color rojo en las figuras 4.8, 4.9 y 4.10.

Es importante, también, evaluar cómo es que la cantidad de componentes se dispersan de la media. Vea la línea de color rojo en cada uno de los *box* de la gráfica, la cual representa la media. Por ejemplo, en el modelo de la imagen 4.9 con 16(x100) vértices se puede apreciar una media de 6 componentes; sin embargo, su desviación respecto a la media es alta y, en tal sentido, no conviene considerarla.

Luego de realizar y evaluar los experimentos, concluimos que se requieren entre 10 a 15 *N-Eigen* y entre 1700 a 1900 *Max-Points*.

La figura 4.11 muestra las correspondencias de modelos transformados con su respectivo modelo null. Estas correspondencias permiten evaluar la repetibilidad del algoritmo para encontrar *key components*

Comparación con otros métodos En esta sección, se presenta el análisis de repetibilidad de nuestro método. El análisis de repetibilidad fue descrito en la sección 4.4.2. Por otra parte, se realizan las comparaciones con el método propuesto por [SB12].

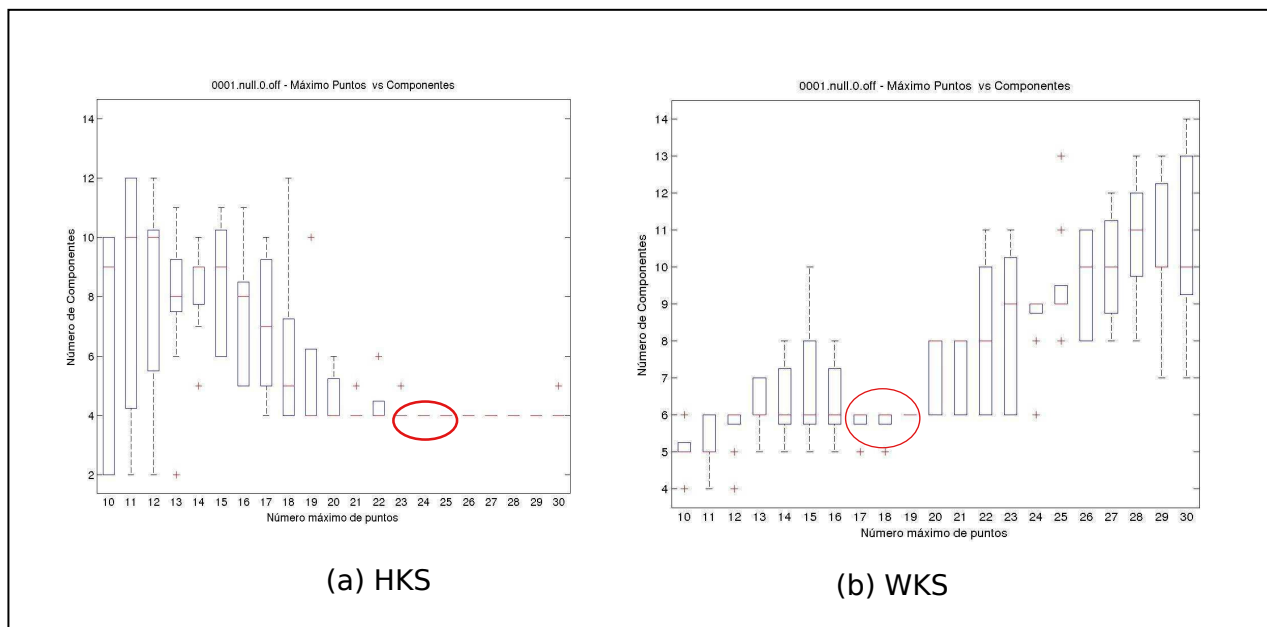


Figura 4.8: *Max-Points* vs Key components en modelo 0001.null.0

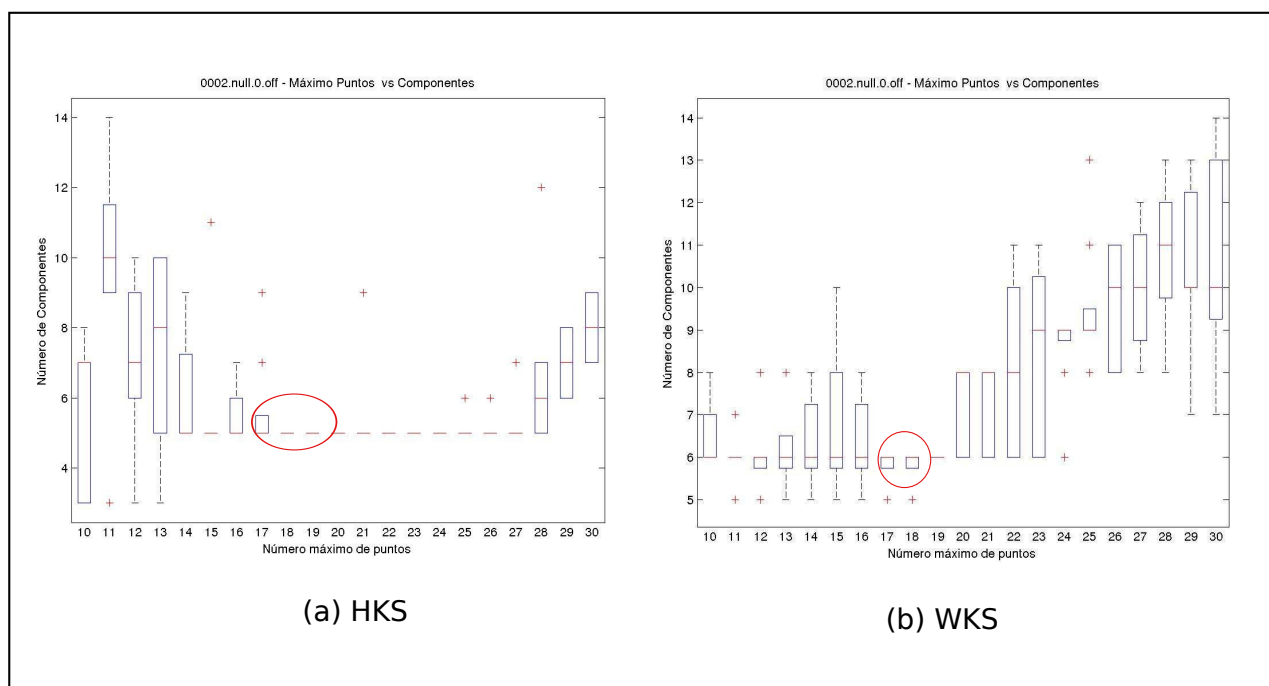


Figura 4.9: *Max-Points* vs Key components en modelo 0002.null.0

Es importante mencionar que el método de Sipirán utiliza *key points*, el cual fue descrito en la sección 3.1.2 del capítulo 3 y posteriormente realiza un proceso de agrupamiento de *key points* geodésicamente cercanos para encontrar los componentes. En contraste, nuestro método utiliza directamente los promedios de las firmas de cara vértices a las cuales se les ha aplicado un proceso de difusión.

En la tabla 4.1 se muestran los resultados de repetibilidad obtenidos por el método

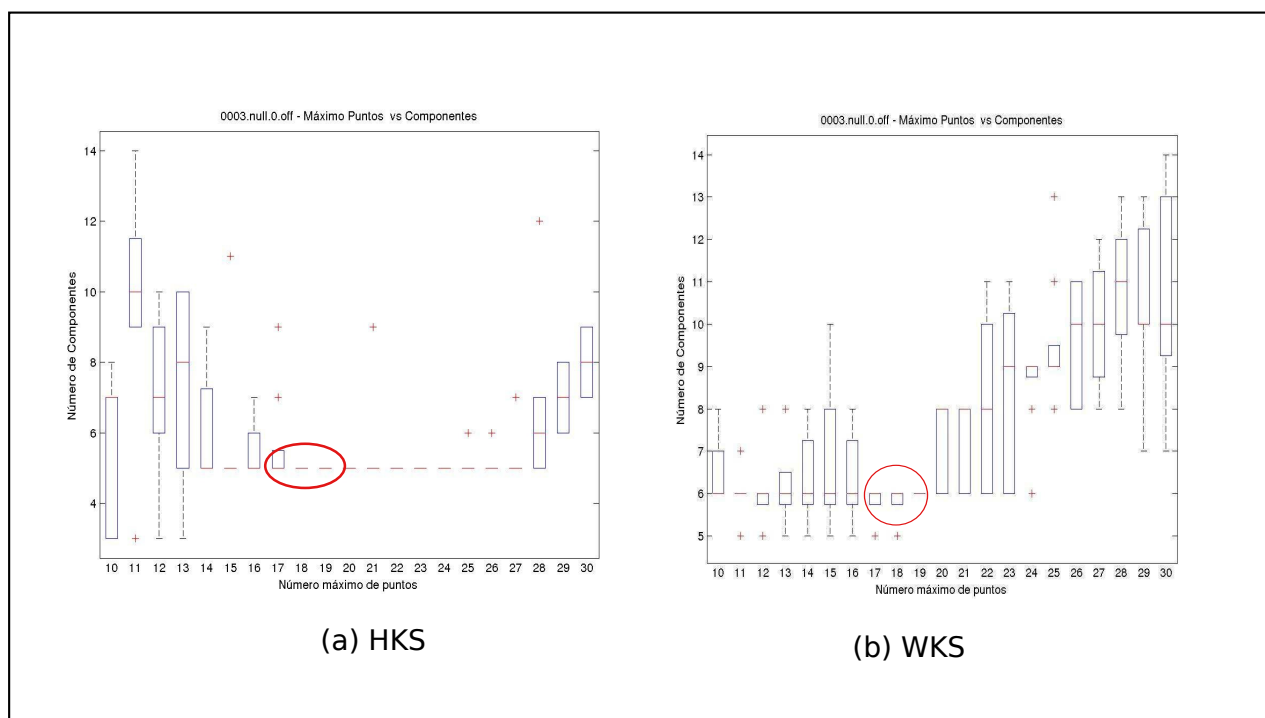


Figura 4.10: *Max-Points* vs Key components en modelo 0003.null.0

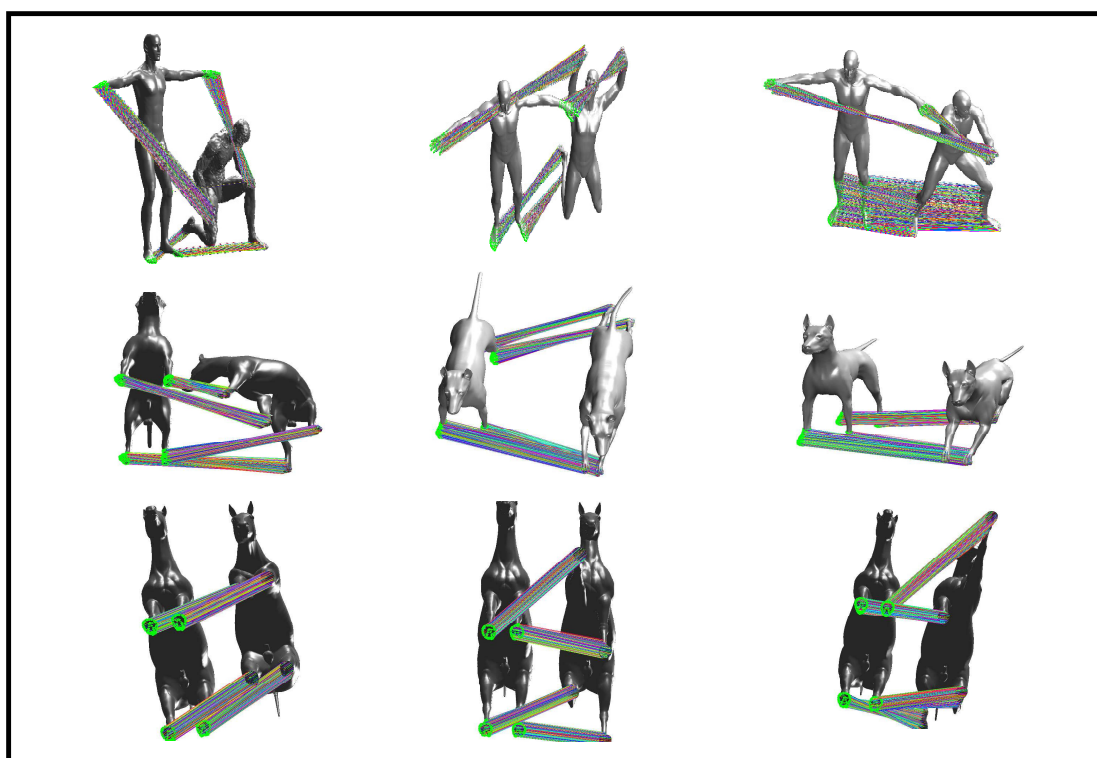


Figura 4.11: Resultados de correspondencias entre modelo transformado y modelo null

de [SB12]. El método muestra los mejores resultados en transformaciones de escala y micro holes, llegando a un promedio de repetibilidad de 90%. Sin embargo, en los casos

como transformaciones *holes* y *sampling*, el porcentaje de repetibilidad disminuye hasta un 50%. Observe que a medida que se incrementa la intensidad de la transformación, el porcentaje repetibilidad disminuye, lo cual indica, que cuanto más drásticas son las transformaciones, más complejo encontrar los componentes correctos. En las tablas 4.2 y 4.3 se muestran los resultados de repetibilidad de nuestro método.

Transformación	Intensidad				
	1	≤ 2	≤ 3	≤ 4	≤ 5
Isometry	0.58	0.83	0.86	0.95	0.85
Topology	0.74	0.76	0.75	0.79	0.86
Micro holes	0.89	0.85	0.90	0.89	0.90
Scale	0.91	0.92	0.93	0.81	0.83
Local Scale	0.84	0.87	0.82	0.93	0.75
Sampling	0.53	0.31	0.21	0.03	0.00
Noise	0.87	0.83	0.71	0.79	0.72
Shot Noise	0.84	0.83	0.85	0.89	0.73
Holes	0.54	0.58	0.48	0.44	0.41
Promedio	0.75	0.75	0.72	0.72	0.67

Table 4.1: Valores de repetibilidad para Key Components [SB12]

Transformación	Intensidad				
	1	≤ 2	≤ 3	≤ 4	≤ 5
Isometry	0.62	0.92	0.92	0.92	0.92
Topology	0.70	0.77	0.77	0.95	0.95
Microholes	0.62	0.62	0.62	0.62	0.62
Scale	0.62	0.63	0.62	0.61	0.61
Local Scale	0.59	0.88	0.97	0.70	0.42
Sampling	0.62	0.62	0.62	0.00	0.00
Noise	0.51	0.52	0.99	0.63	0.64
Shot noise	0.61	0.61	0.62	0.63	0.63
Holes	0.62	0.62	0.62	0.62	0.62
Promedio	0.61	0.69	0.75	0.63	0.6

Transformación	Intensidad				
	1	≤ 2	≤ 3	≤ 4	≤ 5
Isometry	0.90	0.84	0.80	0.88	0.90
Topology	0.80	0.75	0.68	0.68	0.68
Microholes	1.00	1.00	1.00	0.92	0.92
Scale	0.90	0.90	0.90	0.90	0.90
LocalScale	1.00	0.92	0.92	0.83	0.75
Sampling	1.00	1.00	0.92	0.00	0.00
Noise	1.00	1.00	1.00	1.00	1.00
Shot Noise	0.91	0.91	0.89	0.89	0.83
Holes	1.00	1.00	1.00	0.92	1.00
Promedio	0.95	0.92	0.9	0.78	0.78

Table 4.2: Valores de repetibilidad con HKS **Table 4.3:** Valores de repetibilidad con WKS

En la tabla 4.2 se observan los resultados de nuestro método, utilizando *Heat kernel signature*. Como se puede apreciar en la tabla, los resultados de nuestro método, utilizando como proceso de difusión el *HKS*, es inferior, en promedio, al propuesto por [SB12]. Sin embargo, en los casos de *holes* y *sampling*, nuestro método tiene mejores resultados.

En la tabla 4.3 se observan los resultados de nuestro método, utilizando *Wave kernel signature*. En este caso, los resultados son superiores al propuesto por [SB12] y a nuestro método con *HKS*. Como se puede apreciar en tabla, los resultados llegan a un 100% de repetibilidad en la mayoría de los casos. *Local Scape* y *Sampling* presentan valores de al menos al 100%, sin embargo, logran promedios de repetibilidad mayores que el propuesto por [SB12].

4.5 Conclusiones

Luego del análisis de los experimentos podemos concluir que nuestro método, utilizando *Wave Kernel Signature*, es superior al propuesto por [SB12], dando como resultado 95% de repetibilidad para cada una de las transformaciones y en sus diferentes intensidades. Por otro lado, al comparar nuestro método utilizando *HKS* y *WKS*, los resultados

indican que el *WKS* es más robusto para encontrar *key components*, independientemente de la transformación y la intensidad de esta.

5

DETECCIÓN DE SIMETRÍA EN CUERPOS NO RÍGIDOS

En el siguiente capítulo se presenta un nuevo algoritmo para detectar simetría en cuerpos no rígidos.

5.1 Aspectos Preliminares

Con el avance de la tecnología y con la disminución de costos de nuevos dispositivos para la adquisición de información de objetos, en forma de nube de puntos o de mallas de triángulos, ahora es posible desarrollar algoritmos para descubrir patrones, segmentación de mallas, extracción de información de simetría, mejoramiento de modelos, recuperación de modelos 3D, entre otros.

La simetría, referida en algún contexto, como auto-similitud o auto-invarianza es la piedra angular de la naturaleza, exhibiéndose, a si misma, por medio de modelos de la creación de la naturaleza que nosotros vemos cada día, así como también, por medio de las leyes de la física [BBK]

El interés de la simetría en modelos o formas, data desde inicios de la civilización humana. Las primeras evidencias, de que nuestros predecesores atribuyeron importancia a la simetría, pueden ser encontrados en muchas culturas, desde monumentos arquitectónicos de las pirámides de Egipto hasta las decoraciones tradicionales de los griegos.

Johannes Kepler fue el primero quien dio una formulación geométrica a la simetría en su tratado *Ön the six-cornered snowflake* " a inicios de 1611. Unas pocas centurias mas tarde, el estudio de la simetría de formas llega a ser la piedra angular de la cristalografía. Finalmente, la simetría de objetos mas complejos, puede ser encontrada en la física teórica moderna, en temas de la materia el espacio y el tiempo [BBK].

Aunque muchos objetos naturales son simétricos, la ruptura de la simetría puede ser, a menudo, una indicación de anormalidad o comportamiento anormal [RBB⁺10]. Por otro lado, la detección de asimetría se plantea en numerosos problemas prácticos. Por

ejemplo, la detección de tumores en imágenes médicas pueden basarse en la desviación de la simetría [MGM⁺05]. Por otra parte, la detección de simetría y/o asimetría facial es importante en cirugía cráneo facial [HFSvW⁺04], ya que las características de simetría facial son a menudo asociadas con belleza y estética [MBT99].

La asimetría facial puede ser, también, una indicación de varios síndromes y desordenes como la acromegalia, cretinismo, síndrome de donw, entre otros. Entonces la detección de simetría y la asimetría pueden ser utilizadas como un conocimiento a priori en muchos problemas.

La detección de simetría puede también, por ejemplo, facilitar la reconstrucción de superficies [SML00], detección de rostros, reconocimiento y extracción de características [DNGM97, RY92].

Tradicionalmente, la simetría es considerada como propiedades geométricas extrínsecas del modelo, por ejemplo, relacionado al modo en que el modelo u objeto es representado en el espacio euclidiano [RBBK07]. Esta descripción es buena para modelos rígidos; sin embargo, es inapropiada para formas no rígidas o deformables y esto, debido a que, las deformaciones, que los modelos pueden sufrir, hacen que la simetría extrínseca puede perderse, mientras intrínsecamente la forma permanece aún simétrica.

En otras palabras, la simetría bilateral extrínseca, de un cuerpo, puede romperse cuando el cuerpo asume posturas distintas. Aún, desde el punto de vista de geometría intrínseca, la nueva forma permanece, al menos, idéntica [RBBK07].

5.2 Trabajos relacionados

Estudios existentes para la detección de simetría reflexiva se han centrado en la búsqueda de simetría perfecta [Ata85, WWV85]. Los primeros trabajos en esta área están basados en algoritmos de *matching* de subcadenas [KMP77], tomando la ventaja de la circularidad de una cadena, la cual es simétricamente reflexiva. Por otro lado, aunque, el *matching* de subcadenas es inherentemente una pregunta binaria, esos algoritmos solo pueden detectar simetría perfecta y son inestables con la presencia de ruido e imprecisión. En consecuencia, estos algoritmos no son adecuados para muchas formas de registro y aplicaciones de *matching* y, mucho menos, para la detección de simetría en cuerpos no rígidos.

Por otro lado, muchos otros trabajos para la detección de simetría, se han concentrado en simetría euclídea en dos o tres dimensiones. Y, debido a que el espacio de búsqueda está limitado a transformaciones rígidas y escala, entonces los métodos desarrollados sobre este tipo de transformaciones son eficientes. Pero, en el caso, de las deformaciones no rígidas, como cambios isométricos, topológicos y otros, la detección de simetría

sigue siendo un problema complejo. Podolak y sus colegas [PSG⁺06] desarrollaron un método que mide la simetría reflexiva de un modelo, con respecto a todos los planos en un volumen, extendiendo el trabajo de Kazhan y otros [KFR04].

Rustamov y colegas [Rus07b], en su trabajo denominado, *Laplace-Beltrami Eigenfunctions for Deformation Invariant*, propusieron un método denominado *Global Point Signature* o simplemente, **GPS**. En su método, los autores, utilizan las eigen funciones de la eigen descomposición del operador *Laplace - Beltrami*, con el objeto de representar cada punto en la superficie del modelo, en un vector k -dimensional, de modo que, los puntos no dependan de su localidad en las coordenadas cartesianas. El GPS se define de la siguiente manera:

Dado un punto p en una superficie, el *GPS* en p se define como:

$$GPS(p) = \left(\frac{1}{\sqrt{\lambda_1}}\phi_1, \frac{1}{\sqrt{\lambda_2}}\phi_2, \frac{1}{\sqrt{\lambda_3}}\phi_3, \dots \right) \quad (5.1)$$

Donde λ_i y ϕ_i , son los auto valores y auto vectores de la eigen descomposición del operador *Laplace-Beltrami*. Estos vectores, constituyen una firma única para cada vértice del modelo y, es utilizado, en conjunto, con técnicas de agrupamiento, como el *k-means* para la segmentación de modelos tridimensional invariantes a cambios isométricos y topológicos.

Mitra y sus colegas [MSHS06] propusieron un método para buscar simetría en objetos 3D. La simetría se encuentra examinando los valores extremos de los coeficientes armónicos esféricos. Zhiyuan Zhang y otros [ZYF13], hallaron los campos armónicos de un par de puntos característicos en el modelo y, sus campos vectoriales gradientes. Entonces, calcularon la intersección de esos campos gradientes para derivar un campo angular con signos, el cual introduce un sentido de orientación y, de este modo, es factible determinar la autosimetría.

Maksims Ovsjanikovs [Ovs11], propuso un método para la detección de simetría intrínseca global, mediante la eigen descomposición del operador *Laplace-Beltrami*. Basándose en el trabajo de [Rus07b], Ovsjanikovs utilizó el **GPS** para transformar todos los vértices del modelo, en un conjunto de vectores únicos para cada vértices, y en este espacio, se analizó cada eigen función evaluando si es positiva o negativa; de forma tal que, cada simetría intrínseca T , puede ser identificada con una secuencia de signos (+ o -). En otras palabras, para un modelo M con una simetría intrínseca M , el espacio de firmas restringido induce una secuencia de signos.

Todos estos métodos fueron aplicados a modelos tridimensionales y calcularon la simetría, bien como patrones simétricos o como un plano que divide al objeto en dos modelos simétricamente reflexivos. Sin embargo, ninguno de los métodos anteriores, ha detectado el conjunto de vértices, en el modelo, que forman parte de la autosimetría.

Esta es una tarea compleja, debido a que cambios en la posición del objeto, deformaciones, cambios isométricos, afectan la localidad de los vértices cercanos a eje de la autosimetría.

En esta parte de la tesis, nosotros proponemos un nuevo método, para encontrar el conjunto de vértices que se encuentran en la zona de autosimetría; para lo cual, hacemos uso del *Heat Kernel Signature*, *Wave Kernel Signature*, distancias de difusión, distancia geodésica y la detección de *key componets*. Para la detección de *key componets* utilizamos nuestro propio algoritmo, el cual fue descrito en el capítulo anterior.

En la siguiente sección se describe la metodología. Posteriormente, se detallarán los experimentos y resultados y; finamente, en la última sección, se describen las conclusiones y trabajos futuros de este capítulo.

5.3 Propuesta y Metodología

En esta sección se describe la metodología para hallar simetría en modelos 3D. En la figura 5.1 se muestran cada una de las etapas para encontrar la simetría.

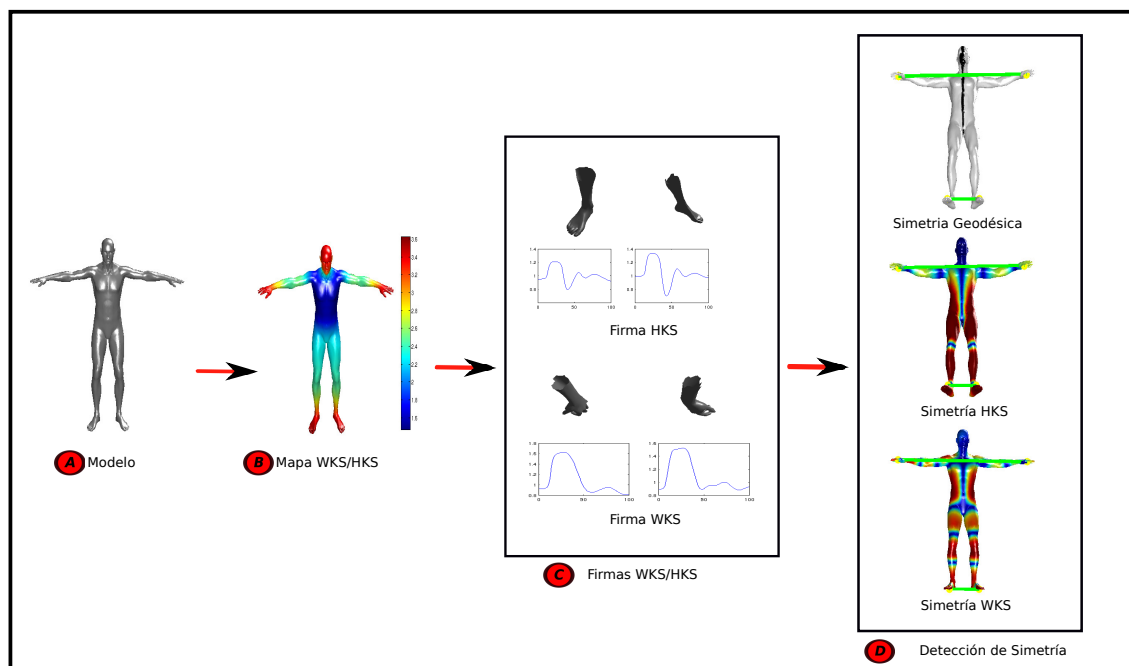


Figura 5.1: Metodología

A continuación describiremos cada uno de estas etapas.

5.3.1 Modelo

El modelo es una malla de triángulos que representa una estructura tridimensional. La malla de triángulos se modela mediante un grafo al cual le denominaremos grafo $G(V, F)$, donde V es el conjunto de todos los vértices en el grafo y F es el conjunto de todas las caras de los triángulos de la malla. El grafo G , es la entrada de la siguiente etapa, el cual calcula el *Wave Kernel Signature* para hallar los *key components* y posteriormente la firma de cada componente. En la figura 5.1 la imagen (a) representa al modelo de entrada

5.3.2 Detección de *Key Components*

En el capítulo 4 se demostró que el *Wave Kernel Signature* tiene un mejor desempeño en cuanto a repetibilidad de *key components* que el *Heat Kernel Signature* y, por lo tanto, el cálculo y detección de los *key components* se realiza mediante el algoritmo propuesto por el autor y descrito en el capítulo 4. Se utilizaron los parámetros que mejores resultados proporcionaron en los experimentos y resultados, los cuales se encuentran en la sub sección 4.4.3 del capítulo 4.

En la figura 5.2 se muestra los componentes detectados con el algoritmo, *Key Components mediante procesos de Difusión*, descrito en el capítulo anterior. En la imagen, se aprecia un modelo con 5 transformaciones isométricas. Los *key components* en cada imagen se muestran en color rojo. Como se puede observar, los 5 objetos presentan *key components* similares, esto debido a que se trata de un mismo objeto, al cual se le han aplicado transformaciones no rígidas.

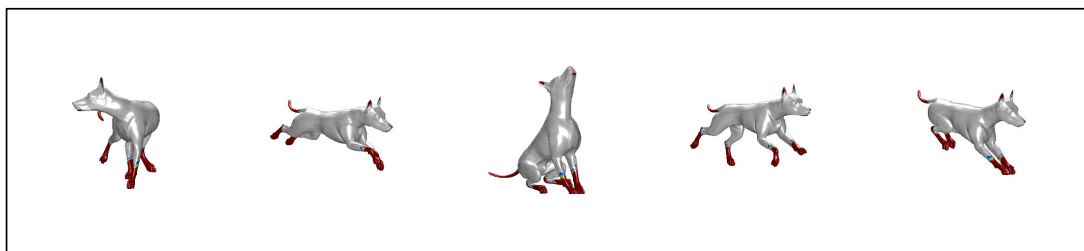


Figura 5.2: Detección de *Key components*

5.3.3 Firmas WKS

La firma se refiere a un valor o conjunto de valores que representan de manera única a un objeto. En el caso de los *key components*, la firma es un vector *k-dimensional* y cada *key component* cuenta con una firma única; sin embargo, se espera que componentes similares, tengan firmas similares.

En la figura 5.3 se observa 4 *key components* (KC1 Mano Derecha, KC2 Mano Izquierda, KC3 Pie Derecho y KC4 Pie Izquierdo). Para un humano es posible determinar la similitud entre el KC1 y KC2, es decir mano derecha y mano izquierda, incluso cuando existen cambios isométricos, topológicos o ruido; lo mismo ocurre en el caso de KC3 y KC4. Con el objetivo que el algoritmo pueda identificar la correspondencia entre dos componentes, en la investigación aplicamos *Wave Kernel Signature* a cada componente, de modo que podemos transformar cada vértices en un vector *k-dimensional*.

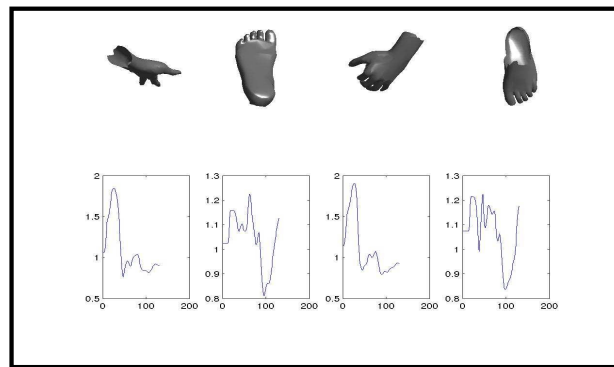


Figura 5.3: Firmas de los Key components

El algoritmo 5.1 computa las firmas para cada uno de los vértices de un *key component*. La entrada es un grafo $G'(V', F')$ que representa un *key component* extraído de un modelo, donde V' son los vértices y F' son las caras de los triángulos en la malla, k es el número de auto valores y auto vectores que se utilizarán para el cálculo de WKS. El algoritmo retorna $Firmas_{nk}$, el cual es una matriz nk dimensional, la cual está conformada por n vectores k – dimensionales que representan las firmas de cada uno de los n vértices.

Luego de obtener las firmas de cada vértice del *key component*, se presenta el problema de hallar *key components* correspondientes. Es decir, que *key components* tienen una estructura geométrica similares a otro. Para lograr esto, se propone utilizar el centroíde geodésico, el cual se describe en el algoritmo 5.2.

En el algoritmo 5.2, $G(V, F)$ es el grafo que representa un *key components*, $Firma_{n \times k}$ son las firmas de cada vértice. El algoritmo calcula el promedio de todas las firmas en C . Pero, el promedio C puede no corresponder a ningún vértice del grafo del

Algorithm 5.1: Firmas de cada vértice de un *Key Components*

input : *Key Component*: $G'(v', f')$; Número de *Eigen Vectores*: k

output : $Firma_{nk}$

- 1 $L = \text{Laplaciano}(G)$;
 - 2 $W = \text{Pesos}(G)$;
 - 3 $[\lambda, \phi] = \text{eiges}(L, W)$;
 - 4 $Firmas_{nk} \leftarrow \sum_{i=0}^{\infty} \phi_i(v_i)^2 f_E(E_i)^2 \forall v_i \in$;
 - 5 return $Firmas_{nk}$
-

Algorithm 5.2: Firma de *Key Components*

input : $G(V, F)$: *Key component*, $Firma_{nk}$: Firmas de los vértices

output : $\text{Centroide}G$

- 1 **for** $j \leftarrow 1 : k$ **do**
 - 2 $C[j] \leftarrow 0$;
 - 3 **for** $i \leftarrow 1 : n$ **do**
 - 4 **for** $j \leftarrow 1 : k$ **do**
 - 5 $C[j] \leftarrow C[j] + Firma_{ij}$;
 - 6 **for** $j \leftarrow 1 : k$ **do**
 - 7 $C[j] \leftarrow \frac{C[j]}{n}$;
 - 8 $\text{Centroide}G \leftarrow \min \text{dis}(v, C) \forall v \in V$;
 - 9 return $\text{Centroide}G$
-

componente y, en tal sentido, no se podría aplicar un algoritmo de difusión de calor o de onda. Para resolver el problema, proponemos encontrar el vértice v del grafo G cuya distancia euclídea al centroide C sea la mínima.

En resumen, la firma de un *key component* se calcula hallando el vértice v del grafo $G(V, F)$ cuya distancia euclídea sea la mínima al promedio de las firmas de cada vértice del grafo G .

Si analizamos nuevamente la figura 5.3, las firmas de cada *key components* están representadas en las imágenes (a),(b),(c) y (d). Observe que las firmas de KC1 y KC2, es decir las firmas (a) y (b) son similares, incluso cuando sus modelos tiene cambios isométricos. Lo mismo ocurre en los *key components* KC3 y KC4 cuyas firmas son (c) y (d) respectivamente.

5.3.4 Correspondencia de Key Components

En la sección 5.3.3 se describió el proceso para encontrar las firmas de los *key components* y, dado que un modelo puede contar con m *key components* es necesario encontrar

pares de *key components* cuyas firmas sean las más similares. Esto con el objeto de calcular la simetría del modelo.

De manera formal, sea M un modelo representado por el grafo $G(V, F)$ y C_i un *key component* de M representado por el grafo $G_i(V_i, F_i)$, donde $G_i(V_i, F_i) \subset G(V, F)$. El problema se reduce a encontrar todos los pares de grafos $G_i(V_i, F_i)$ y $G_j(V_j, F_j)$ cuya diferencias de firmas promedio sea mínima.

Algorithm 5.3: Correspondencia de *Key Componentes*

input : $Firmas_{mk}$: Matriz de firmas de *key components*

output : $Pares_{n2}$: Matriz de Componentes Pares

```

1 pares = [];
2 dist = [];
3 visitados = zeros(1,Npaths);
4 for i ← 1 : m do
5     if visitados(i) < 1 then
6         [p q] = knnsearch(Firmasmk,Firmasik,2);
7         pares = [pares ; p];
8         dist = [dist ; q];
9         visitados(p(1)) = 1;
10        visitados(p(2)) = 1;
11 return Pares

```

En algoritmo 5.3 recibe como entrada una matriz de m filas y k columnas, donde m es el número de *key components* y k es la dimensión de la firma de cada componente. En otras palabras, la fila i de la matriz $Firmas_{mk}$ representa la firma k – dimensional del i – ésimo *key component*.

La salida $Pares_{n2}$, es una matriz de $n \times 2$, donde n es el número de *key components* pareados. Esta matriz almacena, en la fila i – ésima, dos índices; los cuales corresponden a los *key components* cuyas firmas son las más similares.

El algoritmo, en la línea 6, básicamente busca, para cada $Firmas_{ik}$, en la matriz de $Firmas_{mk}$, la firma más cercana. Las firmas similares y sus distancias, son almacenadas en la línea 7 y 8 respectivamente. Finalmente, los *key components* que ya han sido emparejados, no deben ser evaluados nuevamente, y por lo tanto se les marca como visitados, de manera que en la siguiente iteración no son evaluados. Es importante mencionar, que dado que este proceso se realiza varias veces, se opta por una estructura de datos que minimice la búsqueda de los componentes pares. En este caso se ha utilizado un KDTree.

La figura 5.7 muestra un modelo con 5 transformaciones isométricas. Sin embargo, y como se puede observar; todos los modelos presentan los centroides geodésicos en zonas similares. Por otro lado, en la figura 5.5, se aprecia el resultado de aplicar el algoritmo 5.3 a cada uno de los modelos. En la figura se puede observar que los

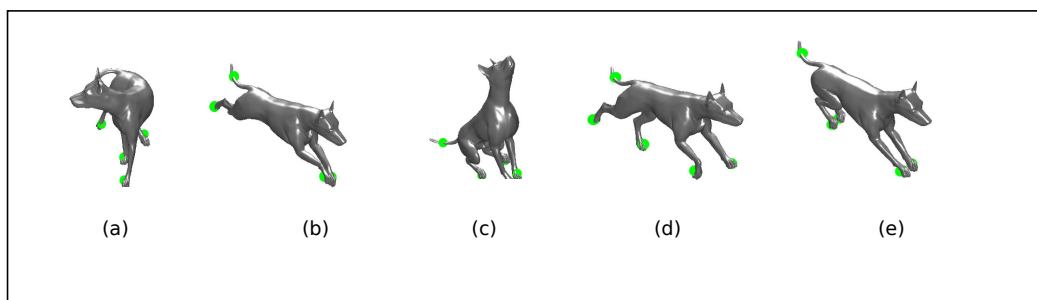


Figura 5.4: Centroides Geodésicos

componentes cuya geometría y topología son similares han sido pareados. Esto debido a que sus firmas son similares. En caso que el modelo presente un número impar de *key Components*, uno de los componentes es descartado.

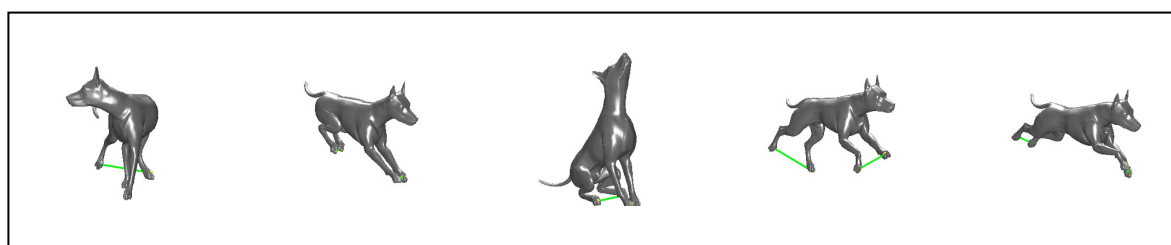


Figura 5.5: Correspondencia de Key Points

5.3.5 Detección de Simetría

En las secciones anteriores se describió el proceso para encontrar los *key components*, los centroides geodésicos y; finalmente, las correspondencias de cada *key component*. En esta sección detallaremos el método propuesto para encontrar simetría en modelos 3D.

Teniendo en cuenta que cada *key component* está representado por un único vértice, el cual es el centroide geodésico del componente y, dado que sabemos cual es el par correspondiente de cada centroide, ahora podemos encontrar la simetría en modelos 3D de la siguiente manera.

Sea Kc_i y Kc_j dos *key components* que pertenecen a un modelo M , y sean C_i y C_j los centroides geodésicos de Kc_i y Kc_j respectivamente; y supongamos que Kc_i y Kc_j forman un par correcto. Es decir, sus firmas son similares y, por lo tanto, su geometría y topología es similar. Entonces, los vértices v_k que corresponden a posiciones aproximadamente simétricas del modelo M , deben ser aquellos que cumplan con la ecuación 5.2.

$$S = \{v_k | abs(d(v_k, C_i) - d(v_k, C_j)) \simeq 0\}, \forall v_k \in G(V, F) \quad (5.2)$$

Donde, S es el conjunto de todos los vértices v_k del grafo $G(V, F)$, que están a una distancia aproximadamente equidistante de los centroides C_i, C_j ; d es una función de distancia invariante a cambios isométricos, como la distancia geodésica o la distancia de difusión.

5.4 Experimentos

En esta sección se explica y evalúa los resultados experimentales de nuestro método utilizando un *benchmark* estándar.

Los experimentos fueron realizados en tres etapas:

- Se determinaron los mejores key componentes, de modo que la repetibilidad de estos sea máxima. Estos experimentos fueron discutidos en la sección 4.4.3 del capítulo 4.
- Se evalúa la correspondencia entre centroides geodésicos.
- Se analiza los resultados de simetría y se compara los resultados respecto a distancias geodésicas y distancias de difusión.

5.4.1 El conjunto de datos

En nuestros experimentos y, con el fin de evaluar nuestro método, se utilizaron dos bases de datos distintas. La primera, es el *benchmark* SHREC 2010 [BBB⁺10b]. Esta base de datos esta compuesta por 3 modelos, los cuales presentan simetría extrínseca (NULL Shapes). Cada modelo, a su vez, tiene un conjunto de modelos obtenidos mediante la aplicación de transformaciones sobre los modelos NULL. Las transformaciones aplicadas a los modelos son isometría, micro agujeros, agujeros grandes, cambios topológicos, ruido plano, escala global y local, y disminución de la resolución. La segunda base de datos, esta compuesta por modelos de rostros tridimensionales mostrando distintas expresiones faciales [BBM⁺07]. La base de datos contiene 15 expresiones del mismo rostro. Cada objeto contiene aproximadamente 2000 vértices. Es importante mencionar que, los cambios en las expresiones faciales, generan cambios en algunas zonas de la malla que representa el modelo y, en tal sentido, es ideal para verificar si el algoritmo puede encontrar la simetría incluso con esas variaciones o transformaciones isométricas.

5.4.2 Experimentos y Resultados

Los experimentos se llevaron a cabo en una máquina corei7, con 12G de memoria RAM.

Experimento 1: Base de datos SHREC 2010 [BBB+10b]

1. *Determinación de los mejores Key Component*: se utilizó nuestro algoritmo para detección de *key components* basado en difusión, propuesto en la sección 4.3 del capítulo 4. En ese capítulo, se determinó la mejor combinación de parámetros para encontrar los *key components*, los cuales fueron 15 auto vectores para el parámetro *N-Eigen* y entre 1700 a 1900 vértices para el parámetro *Max-Point*. Por otro lado, el método para la creación de los componentes fue el *wave kernel signature*, debido a que mostró un mejor desempeño en relación al *Heat Kernel Signature*. Para verificar este hecho ver los resultados de los experimentos en la sección 4.4.3 del capítulo 4.
2. *Evaluación de correspondencia entre centroides geodésicos*:

En la figura 5.6, se observa los componentes y sus respectivas firmas para diferentes transformaciones del modelo NULL. La firma de un componente i , fue calculada aplicando el algoritmo 5.3.3, el cual retorna el promedio de las firmas *WKS* de cada vertice que pertenece al componente i .

En la figura 5.6, (a) es el modelo humano sin transformación; (b),(c) y (d) son los *key components* de 3 transformaciones isométricas del modelo NULL. En (e), (f), (g) y (h) se observan los *key components* obtenidos de modelos con hoyos, micro hoyos, ruido y escala, respectivamente. También, se puede apreciar, que ambas manos tienen firmas similares, y ambos pies también. Por ejemplo, observe las firmas de los pies en la figura (a). En esa figura, se puede notar que ambas firmas son similares, lo propio ocurre con las firmas de ambas manos y este patrón se repite para todos los modelos. Bajo esta observación, se aplicó el algoritmo de correspondencia, utilizando la similitud entre las firmas de los componentes, para cada uno de los 135 modelos, y los resultados se describen en la tabla 5.6.

En la tabla 5.1 se muestran los resultados, en porcentaje, de aplicar la correspondencia entre los centroides geodésicos. En la tabla, la primera columna representa los diferentes tipos de transformación aplicadas al modelo NULL, la columna % Correcto, se refiere al porcentaje de modelos cuyos centroides geodésicos tiene correspondencias correctas. Por ejemplo, mano derecha con mano izquierda, y pie derecho con pie izquierdo y, la tercera columna, % Incorrecto, es el porcentaje de los centroides con correspondencias incorrectas. Se puede observar que, en

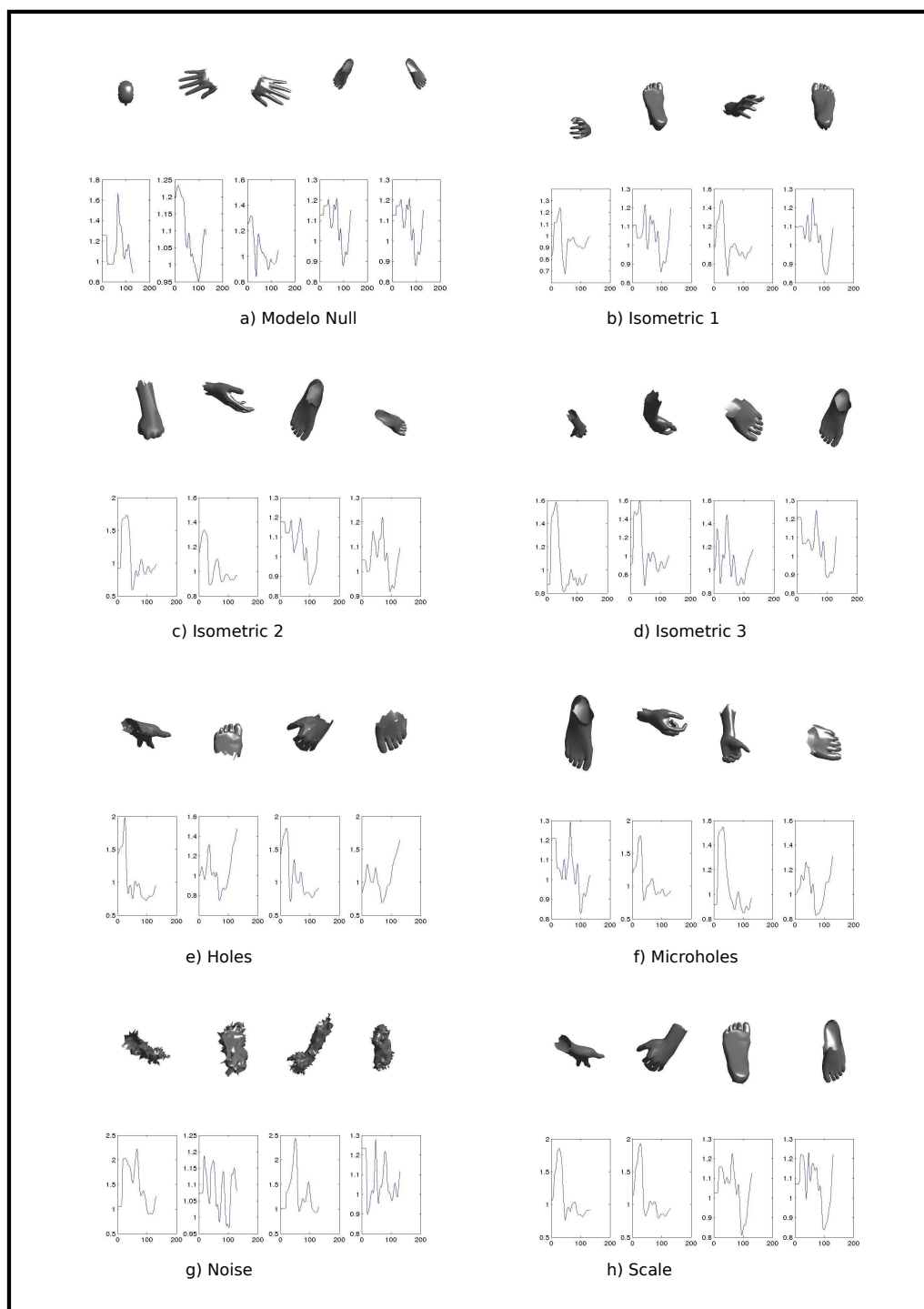


Figura 5.6: Firmas de *key components* detectados con nuestro algoritmo

la mayoría de transformaciones, los componentes presentan correspondencias correctas, superando el 90%.

Sin embargo, las transformaciones *Local Scale* y *topology*, redactadas en color rojo en la tabla 5.1, muestran los resultados de correspondencia correcta con niveles de 55% y 60%, respectivamente. La pregunta es ¿porqué ocurren estos

Transformación	Correspondencia	
	% Correcto	% Incorrecto
Holes	93.33	6.66
Isometry	93.33	6.66
Local Scale	53.33	46.66
Micro Holes	86.66	13.33
Noise	100	0
Sampling	100	0
Scale	93.33	6.66
Shot Noise	100	0
Topology	60	40
Promedio	78	12

Table 5.1: Resultados de correspondencia entre centroides geodésicos

resultados tan bajos con este tipo de transformaciones?. Analizamos los efectos de estas transformaciones en el computo de los *key componentes* y centroides geodésicos y esto se explica en el la figura 5.7. Aquí, (a) representa un modelo de caballo con trasformación topológica. En esta imagen, podemos ver, encerrado entre círculos rojos y apuntado por flechas, que existen deformaciones en la malla, de modo que, ciertas partes, que deberían estar separadas, se encuentran unidas. Por ejemplo, la pata derecha superior y la pata izquierda superior del caballo. Esto afecta al proceso de selección de *key components*, de modo que en lugar de encontrar dos componentes distintos, uno para cada pata superior, ahora encuentra uno solo. Por lo tanto, el número de centroides geodésicos y la propia correspondencia falla. Por otro lado, en la imagen (b), se presenta el modelo de un perro con transformaciones *Local Scape*, o escala local. Aquí se puede observar, nuevamente encerrados con círculos rojos y apuntados por flechas, que solo algunas partes del modelo han sido escaladas, mientras otras se mantienen invariables. Por ejemplo, observe la pata derecha inferior y la pata izquierda inferior del modelo en la imagen (b) de la figura 5.7. La pata izquierda ha sido escalada, mientras la derecha permanece igual. Dada la escala desigual en estos componentes, las firmas de sus vértices, producto de aplicar el *WKS*, tendrán variaciones distintas, y en tal sentido, las firmas de los centroides geodésicos de ambos componentes no serán similares como se espera, y por lo tanto, la correspondencia falla. Por otro lado, incluso con esta desventaja en este tipo de transformaciones, nuestro método ha demostrado muy buenos resultados para todas las otras transformaciones.

3. Análisis de simetría

Una vez calculados los centroides geodésicos y sus respectivas correspondencias, se realizaron los experimentos para encontrar la simetría en cada uno de los modelos. Esta etapa fue dividida en dos partes. Detección de simetría con distancias geodésicas y detección de simetría con procesos de difusión.

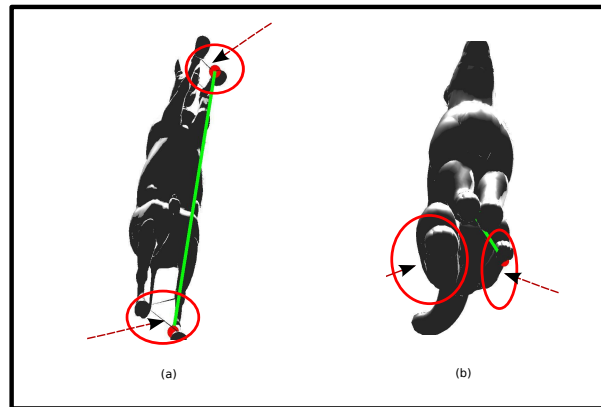


Figura 5.7: Centroides Geodésicos

- Simetría con distancia geodésica:

En la figura 5.8 se muestra los resultados de aplicar el algoritmo de detección de simetría en los modelos NULL; es decir, aquellos que presentan simetría extrínseca. Las pruebas fueron realizadas utilizando la distancia geodésica. En primero lugar se encontraron los *key components*, el cual fue descrito en la sección 4.3.3 del capítulo 4. Posteriormente, se aplicó el algoritmo para detectar el centroide geodésico y las correspondencia entre estos; descrito en el algoritmo 5.2 de la sección 5.3.3 y finalmente, se computó los vértices, que se encuentran en la zona de simetría de los modelos, para esto aplicamos la ecuación 5.2. En esta figura, se puede apreciar que la simetría en los 3 modelos han sido encontradas de manera correcta. Sin embargo, ¿que ocurre con los modelos con transformaciones isométricas?

En la figura 5.9 podemos observar los resultados de aplicar el algoritmo de simetría mediante distancia geodésica, y tal como lo muestra la teoría, los resultados no son detectados de manera correcta. Esto se debe a que pequeños cambios en la estructura de la malla, genera variaciones en las distancias geodésicas y por lo tanto, en el calculo de simetría. Sin embargo, como se menciona en la sección 2.2.3 del capítulo 2, la distancias de difusión debería ser invariante a cambios isométricos. En tal sentido, experimentamos lo que ocurre al aplicar nuestro algoritmo para encontrar simetría con distancia de difusión.

- Simetría con distancia de difusión:

Se realizaron experimentos mediante la distancia de difusión. Para estos experimentos se siguieron los mismo pasos que en el caso anterior. Es decir, se encontraron los *key components*, el cual fue descrito en la sección 4.3.3 del capítulo 4. Posteriormente, se aplicó el algoritmo para detectar el centroide geodésico y las correspondencia entre estos; descrito en el algoritmo 5.2 de la sección 5.3.3 y finalmente, se computó los vértices, que se encuentran

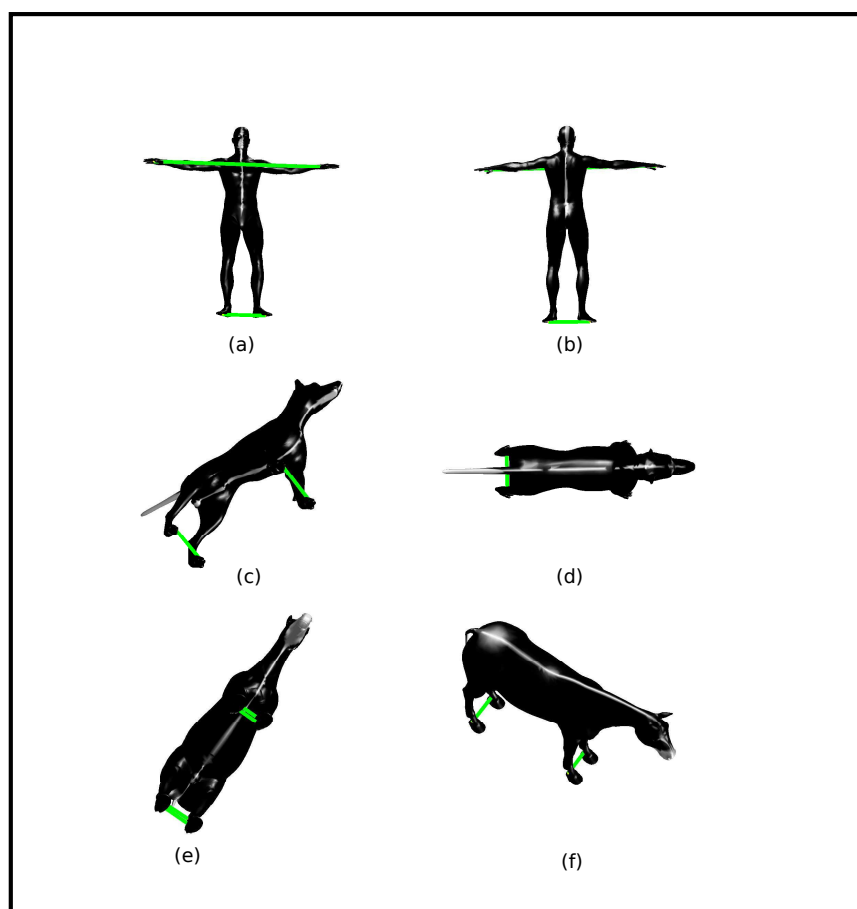


Figura 5.8: Detección de Simetría mediante distancias geodésica en modelos con simetría extrínseca

en la zona de simetría de los modelos, para esto aplicamos la ecuación 5.2. La diferencia con la detección de simetría mediante distancia geodésica estriba en el uso de la distancia de difusión en la ecuación 5.2 en lugar de la distancia geodésica.

En la figura 5.10, se observa los resultados de nuestro algoritmo para detectar simetría mediante la distancia de difusión en modelos con simetría extrínseca. Aquí, podemos observar, que a diferencia del método con distancia geodésica, los patrones de simétrica no solo se encuentran en el eje central del modelo; si no que, se encuentran en distintas zonas de los modelos. Por ejemplo, vea imagen del humano de la figura 5.10 y observe que no solo se encuentra el eje central de simetría, si no que los patrones de simetría se encuentra en los pies, manos, y la parte central del cuerpo.

En la figura 5.11 se observa la detección de simetría en cuerpos con transformaciones isométricas. Calculamos las regiones más cercanas a los centroides geodésicos, detectando la separación entre las regiones de voronoi. Para

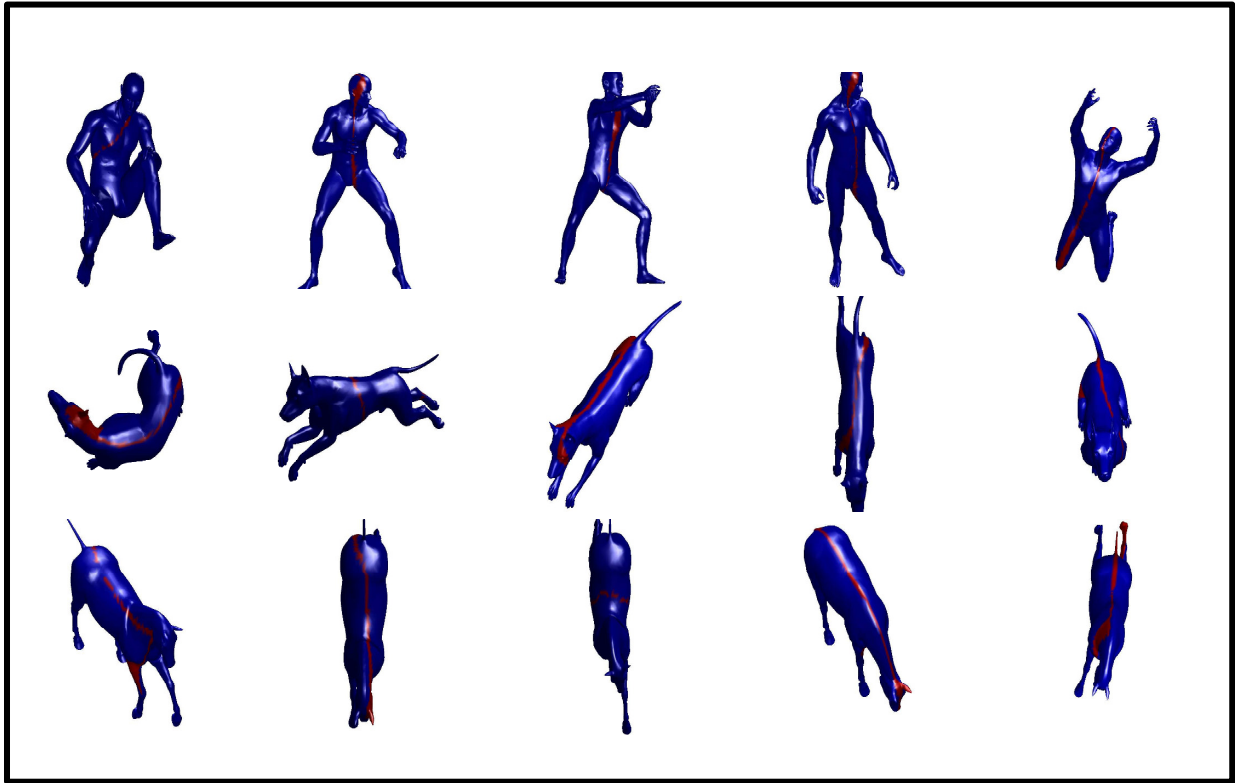


Figura 5.9: Detección de Simetría mediante distancias geodésica en modelos con simetría intrínseca

esto aplicamos el siguiente criterio

$$R_i = \{x \in M \mid d_{difusion}(x, C_i) \leq d_{difusion}(x, C_i)\} \quad (5.3)$$

Donde R_i es la región i con centroide geodésico C_i , M es la malla que representa el modelo y x es un vértice que pertenece a la malla. Es otras palabras, encontramos las regiones de voronio, y detectamos los puntos frontera. Como se puede observar en la figura 5.11 la detección de simetría es mejor, en contraste con la detección de simetría mediante distancia geodésica, Ver figura 5.9. Sin embargo, los resultados, como se puede observar en la figura (c),(d) y (f), las líneas de simetría se alejan del centro real. La pregunta es por que ocurre esto, si la teoría indica que la distancia de difusión es robusta a cambios isométricos. Esto se explica debido que cambios abruptos en la posición del modelo, generan en cambios de la triangulación, y en tal sentido, el cálculo del operador *Laplace-Beltrami* y, dato que la eigen descomposición del operador *Laplace-Beltrami* se utiliza para el computo de la distancia de difusión, entonces la detección de la simetría se ve afectada. Esto se puede apreciar, con mayor efecto, la figura 5.12 donde los cambios en el modelo y por ende en la triangulación es más abrupto.

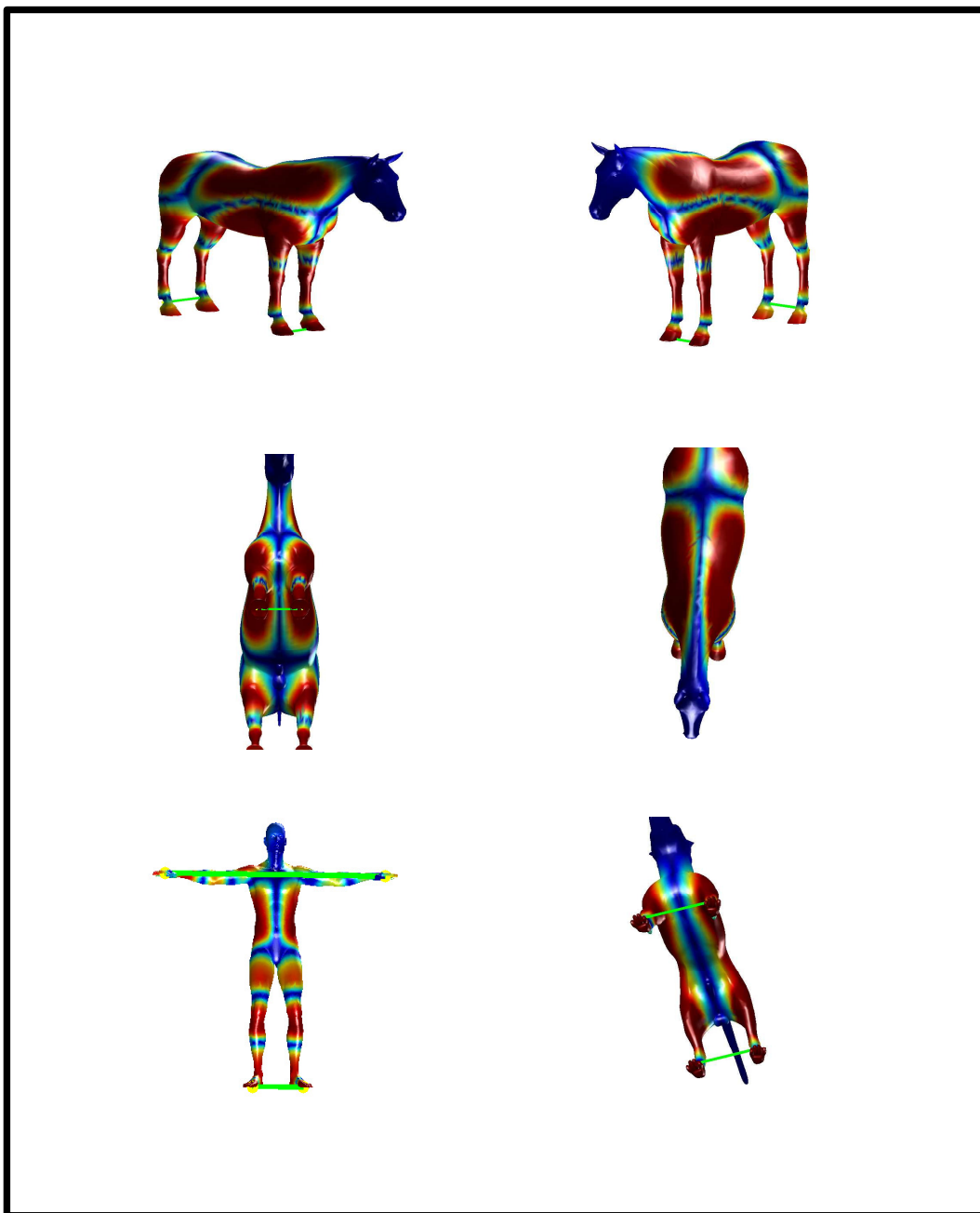


Figura 5.10: Detección de Simetría mediante distancias de difusión en modelos con simetría extrínseca

En la figura 5.12 se observa que, los peores resultados, se dan con cambios topológicos (c), *shotnoise* (d) y escala local (e). Y es justamente, en estos modelos donde los cambios de la triangulación es más abrupta. Observe las flechas en las imágenes (c), (d) y (e). Aquí se observa los efectos de cambios abruptos en los modelos y la consecuencia en nuestro algoritmo.

Experimento 1: Base de datos de expresiones faciales [BBM⁺07] Dado que cambios abruptos generan problemas al encontrar simetría en cuerpo no rígidos,

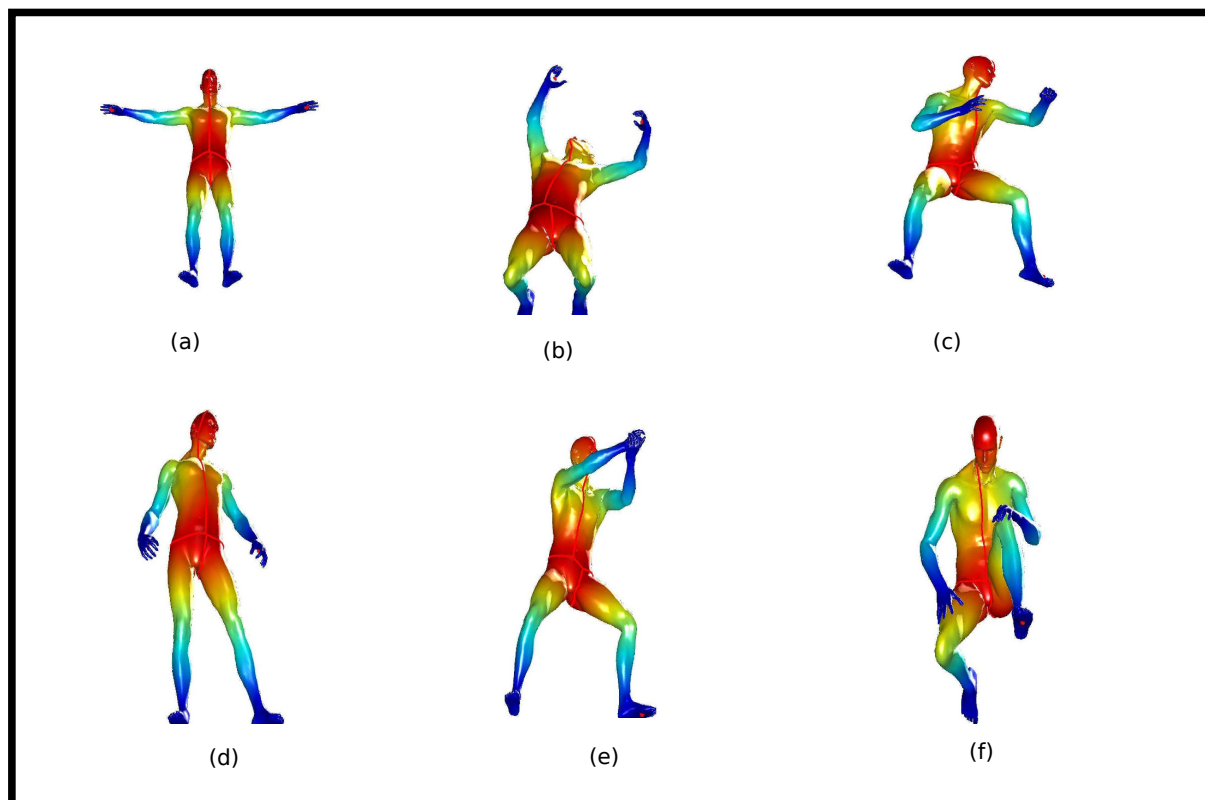


Figura 5.11: mapa de Voronoi formado por la distancia de difusión en modelo humano con transformaciones isométricas

optamos por analizar el comportamiento, de nuestro algoritmo, en modelos con cambios isométricos no pronunciados, como en el caso de los cambios de un rostro producto de expresiones faciales. Los pasos que se siguieron fueron los mismos que los anteriores y los resultados se pueden observar en la figura 5.13. Para cada rostro, se detectaron entre 4 y 6 *key componentes*. Cada componente tiene un máximo de 200 vértices y un mínimo de 50 vértices. Se tomaron 15 eigen vectores para el computo del *WKS*. En los resultados, se observa una detección de simetría en el 100% de los modelos. Esto refuerza el hecho de que, la distancia por difusión, no es invariante a cambios isométricos pronunciados, si no que, es invariante a cambios isométricos suaves.

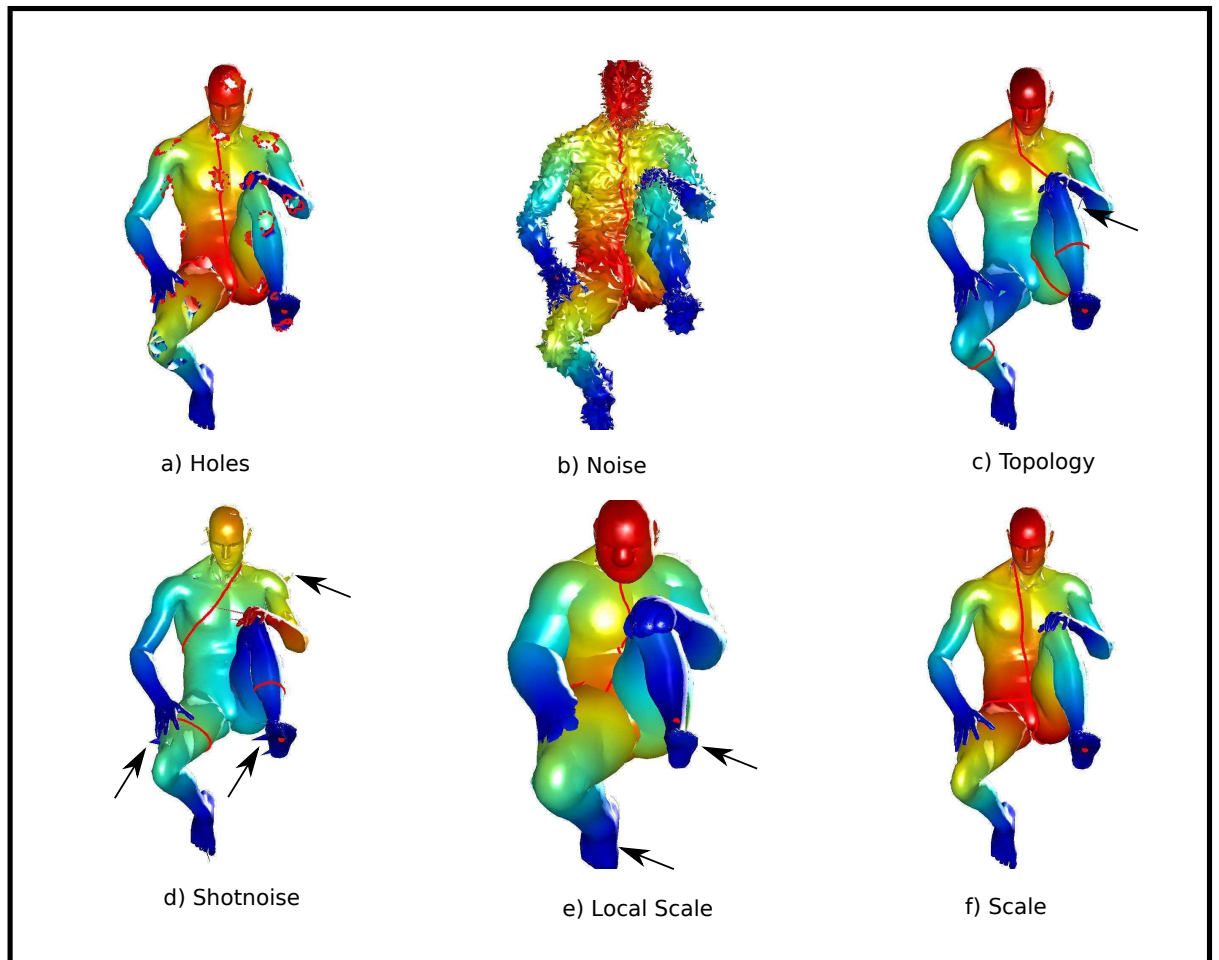


Figura 5.12: mapa de Voronoi formado por la distancia de difusión en modelo humano con transformaciones no rígidas

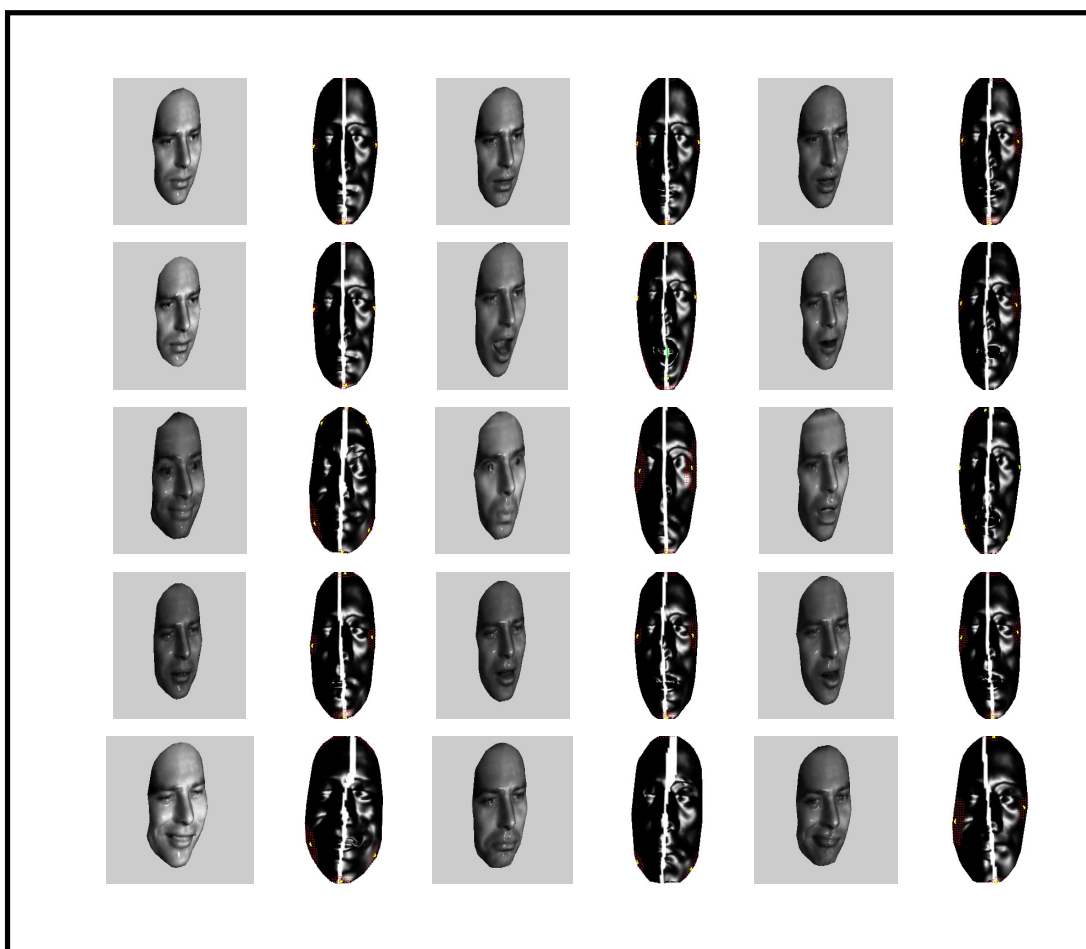


Figura 5.13: Detección de Simetría en rostros 3D con cambios isométricos

6

CONCLUSIONES FINALES Y TRABAJOS FUTUROS

En esta tesis, se presentaron 3 nuevos algoritmos. El primero, al cual denominamos, *Minimal area interest point*, permite detectar *key points* o puntos de interés mediante un algoritmo eficiente, es decir de complejidad computacional lineal respecto al número de *faces* de la triangulación del modelo. Por otra parte, nuestros resultados fueron contrastados con 2 técnicas recientes del estado del arte, *Harris 3D* y detección de *key points* mediante *HKS*. Nuestros resultados demostraron ser robustos a los cambios isométricos y topológicos así como, a la presencia de ruido, escala, entre otros. También, identificamos que la presencia de *holes*(grandes hoyos), cambian drásticamente el tamaños en las triángulos donde se presenta los hoyos y; por lo tanto, decrementa, la posibilidad de detectar, correctamente, los puntos de interés. Sin embargo, en transformaciones con cambios topológicos y *shotnoise*, tuvimos mejores resultados que *Harris 3D* y *HKS*.

El segundo algoritmo que propusimos, permite detectar *key components* basado en procesos de difusión y fue comparado con el método propuesto por *Ivan Sipiran*, quien acuño el término de *key components* para referirse a regiones del modelo que representan altas protusiones en la estructura. Demostramos que nuestro método es superior al propuesto por [SB12], y contrastamos los resultados utilizado dos procesos de difusión diferentes; el *Heat Kernel Signature* y el *Wave Kernel Signature*. Evaluamos nuestros resultados mediante la repetibilidad de los *key component* y encontramos que el *WKS* es más robusto para encontrar *key components* que el *HKS*, independientemente de la transformación y la intensidad de esta.

Finalmente, el tercer algoritmo propuesto, permite detectar la simetría mediante procesos de difusión. En nuestros experimentos, encontramos que la detección de simetría funciona correctamente para el caso de modelos con simetría extrínseca, tanto para la distancia geodésica, como con la distancia por difusión. Sin embargo, nuestro algoritmo, utilizando distancia geodésica falla al encontrar la simetría en cuerpos no rígidos, es decir con simetría intrínseca. Esto era de esperar, debido a que la distancia geodésica no es invariante a cambios topológicos e isométricos. Por otro lado, se realizaron experimentos para encontrar simetría, en cuerpos no rígidos, con distancia de difusión. Encontramos, que aunque los resultados son mejores con respecto a la distancia geodésica, aún así falla cuando los modelos presentan cambios topológicos,

shotnoise y escala local. También, descubrimos que este comportamiento se debe a que cuando la posición del modelo cambia abruptamente, entonces la triangulación cambia para adaptarse a la nueva forma del modelo, y esto genera cambios abruptos en el operador *Laplace-Beltrami* y, por lo tanto en la función de distancia de difusión. Por otro lado, realizamos experimentos, con otra base de datos, con pequeños cambios isométricos, y determinamos que la distancia de difusión no es invariante a cambios isométricos abruptos; pero, si los cambios isométricos no son muy abruptos, nuestro método proporciona buenos resultados.

En cuanto a los trabajos futuros, actualmente ya se está trabajando en los siguientes problemas.

1. Nuevo algoritmo para detectar *key Components*: la idea es evitar llevar los *key points* a un espacio bidimensional para aplicar un algoritmo de agrupamiento y luego encontrar los componentes. En lugar de esto, nuestra propuesta va en el sentido de computar *k-rings*, en cada *key points*, de modo que todas las *rings* que se interceptan formarán un *key component*. Esto no solo reducirá el costo computacional, si no que, es susceptible a ser paralelizado debido a que es posible determinar el *ring* en cada *key points* por separado.
2. Nuevo algoritmo para completitud de mallas 3D: La idea es aprovechar los resultados de nuestro algoritmo para detectar *key componets* en los modelos con hoyos, debido a que este crea *key points*, solo al rededor de los hoyos, luego es posible generar parches que mantengan la curvatura en los hoyos del modelo con el objetivo de repararlos.
3. Búsqueda de similitud parcial en modelos 3D: Dado que nuestro algoritmo para determinar *centroides geodésicos*, calcula firmas únicas para cada *key component* y dado que cada *key component* es una pequeña región del modelo que representa cambios abruptos en el modelo, es posible, crear un descriptor formado solo con las firmas de las centroides geodésicos y posteriormente, indexar estos descriptores en una estructura de datos métrica para hacer búsquedas por similitud total o incluso parcial.

BIBLIOGRAFÍA

- [AMWW88] Helmut Alt, Kurt Mehlhorn, Hubert Wagners, and Emo Welzl. Congruence, similarity, and symmetries of geometric objects. *Discrete & Computational Geometry*, 3(1):237–256, 1988.
- [ASC11a] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. Pose-consistent 3d shape segmentation based on a quantum mechanical feature descriptor. In *Pattern Recognition*, pages 122–131. Springer, 2011.
- [ASC11b] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference On*, pages 1626–1633. IEEE, 2011.
- [Ata85] Mikhail J Atallah. On symmetry detection. *Computers, IEEE Transactions On*, 100(7):663–666, 1985.
- [BBB⁺10a] Alexander Bronstein, Michael Bronstein, Benjamin Bustos, Umberto Castellani, Marco Crisani, Bianca Falcidieno, Leonidas Guibas, Iasonas Kokkinos, Vittorio Murino, Ivan Sipiran, Maks Ovsjanikovy, Giuseppe Patanè, Michela Spagnuolo, and Jian Sun. Shrec 2010: robust feature detection and description benchmark. In *Eurographics 2010 Workshop on 3D Object Retrieval (3DOR'10)*, pages 79–86. Eurographics Association, 2010.
- [BBB⁺10b] AM Bronstein, MM Bronstein, B Bustos, U Castellani, M Crisani, B Falcidieno, LJ Guibas, I Kokkinos, V Murino, M Ovsjanikov, and Others. Shrec 2010: Robust feature detection and description benchmark. *Proc. 3DOR*, 2(5):6, 2010.
- [BBK] Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. *Numerical Geometry Of Non-Rigid Shapes*. Springer.
- [BBM⁺07] Er M. Bronstein, Michael M. Bronstein, Student Member, Student Member, Ron Kimmel, and Senior Member. Calculus of non-rigid surfaces for geometry and texture manipulation. *IEEE Trans. Visualization and Comp. Graphics*, 13:902–913, 2007.
- [BG05] Ingwer Borg and Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer, 2005.
- [BK10] Michael M Bronstein and Iasonas Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Computer Vision And Pattern Recognition (CVPR), 2010 IEEE Conference On*, pages 1704–1711. IEEE, 2010.
- [CCFM08] Umberto Castellani, Marco Cristani, Simone Fantoni, and Vittorio Murino. Sparse points matching by combining 3d mesh saliency with statistical descriptors. *Comput. Graph. Forum*, 27(2):643–652, 2008.
- [CL06] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied And Computational Harmonic Analysis*, 21(1):5–30, 2006.
- [CTSO03] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3d model retrieval. In *Computer Graphics Forum*, volume 22, pages 223–232. Wiley Online Library, 2003.

- [DCG12] Helin Dutagaci, Chun Pan Cheung, and Afzal Godil. Evaluation of 3d interest point detection techniques via human-generated ground truth. *Vis. Comput.*, 28(9):901–917, September 2012.
- [Det88] Edge Detector. Chris harris and mike stephens plessey research hoke manor. *United Kingdom*, 1988.
- [DGD⁺11] Helin Dutagaci, Afzal Godil, Petros Daras, Apostolos Axenopoulos, G Litos, Stavroula Manolopoulou, Keita Goto, Tomohiro Yanagimachi, Yukinori Kurita, Shun Kawamura, and Others. Shrec'11 track: Generic shape retrieval. In *Proceedings Of The 4th Eurographics Conference On 3D Object Retrieval*, pages 65–69. Eurographics Association, 2011.
- [DK10] Anastasia Dubrovina and Ron Kimmel. Matching shapes by eigendecomposition of the laplace-beltrami operator. In *Proc. 3DPVT*, volume 2, 2010.
- [DNGM97] Francesco GB De Natale, Daniele D Giusto, and Fabrizio Maccioni. A symmetry-based approach to facial features extraction. In *Digital Signal Processing Proceedings, 1997. DSP 97., 1997 13th International Conference on*, volume 2, pages 521–525. IEEE, 1997.
- [EKG10] Amir Egozi, Yosi Keller, and Hugo Guterman. Improving shape retrieval by spectral matching and meta similarity. *Image Processing, IEEE Transactions On*, 19(5):1319–1327, 2010.
- [GH97] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings Of The 24th Annual Conference On Computer Graphics And Interactive Techniques*, pages 209–216. ACM Press/Addison-Wesley Publishing Co., 1997.
- [Glo09] Przemyslaw Glomb. Detection of interest points on 3d data: Extending the harris operator. In *Computer Recognition Systems 3*, pages 103–111. Springer, 2009.
- [GSL⁺99] Arthur Gregory, Andrei State, Ming C Lin, Dinesh Manocha, and Mark A Livingston. Interactive surface decomposition for polyhedral morphing. *The Visual Computer*, 15(9):453–470, 1999.
- [HFSvW⁺04] Clara E Huisinga-Fischer, Jim PHJA Souren, Frederik vd Werken, Birte Prahlandersen, Floris van Ginkel, et al. Perception of symmetry in the face. *Journal of Craniofacial Surgery*, 15(1):128–134, 2004.
- [HWAG09] Qi-Xing Huang, Martin Wicke, Bart Adams, and Leonidas Guibas. Shape decomposition using modal analysis. In *Computer Graphics Forum*, volume 28, pages 407–416. Wiley Online Library, 2009.
- [JZ07] Varun Jain and Hao Zhang. A spectral approach to shape-based retrieval of articulated 3d models. *Computer-Aided Design*, 39(5):398–407, 2007.
- [KFR04] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Symmetry descriptors and 3d shape matching. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 115–123. ACM, 2004.
- [KG00] Zachy Karni and Craig Gotsman. Spectral compression of mesh geometry. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 279–286. ACM Press/Addison-Wesley Publishing Co., 2000.
- [KLT05] Sagi Katz, George Leifman, and Ayellet Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21(8-10):649–658, 2005.

- [KMP77] Donald E Knuth, James H Morris, Jr, and Vaughan R Pratt. Fast pattern matching in strings. *SIAM journal on computing*, 6(2):323–350, 1977.
- [KT03] Sagi Katz and Ayellet Tal. *Hierarchical mesh decomposition using fuzzy clustering and cuts*, volume 22. ACM, 2003.
- [LDR09] Jianye Lu, Julie Dorsey, and Holly Rushmeier. Dominant texture and diffusion distance manifolds. In *Computer Graphics Forum*, volume 28, pages 667–676. Wiley Online Library, 2009.
- [Len79] Thomas Lengauer. Robert endre tarjan. In *Conference Record Of The Eleventh Annual ACM Symposium On Theory Of Computing: Papers Presented At The Symposium, Atlanta, Georgia, April 30-May 2, 1979*, page 262. Association For Computing Machinery, 1979.
- [LL04] Wee Kheng Leow and Rui Li. The analysis and applications of adaptive-binning color histograms. *Computer Vision and Image Understanding*, 94(1):67–91, 2004.
- [LVJ05] Chang Ha Lee, Amitabh Varshney, and David W. Jacobs. Mesh saliency. *ACM Trans. Graph.*, 24(3):659–666, July 2005.
- [LWTH01] Xuetao Li, Tong Wing Woon, Tiow Seng Tan, and Zhiyong Huang. Decomposing polygon meshes for interactive applications. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 35–42. ACM, 2001.
- [MBP13] Anirban Mukhopadhyay, Suchendra M Bhandarkar, and Fatih Porikli. Detection and characterization of intrinsic symmetry. *ArXiv Preprint ArXiv:1309.7472*, 2013.
- [MBT99] Linda Mealey, Ruth Bridgstock, and Grant C Townsend. Symmetry and perceived facial attractiveness: a monozygotic co-twin comparison. *Journal of personality and social psychology*, 76(1):151, 1999.
- [MGM⁺05] Matei Mancas, Bernard Gosselin, Benoît Macq, et al. Fast and automatic tumoral area localisation using symmetry. In *ICASSP (2)*, pages 725–728, 2005.
- [MPS⁺04] Michela Mortara, Giuseppe Patané, Michela Spagnuolo, Bianca Falcidieno, and Jarek Rossignac. Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. *Algorithmica*, 38(1):227–248, 2004.
- [MPWC13] Niloy J Mitra, Mark Pauly, Michael Wand, and Duygu Ceylan. Symmetry in 3d geometry: Extraction and applications. In *Computer Graphics Forum*, volume 32, pages 1–23. Wiley Online Library, 2013.
- [MSHS06] Aurélien Martinet, Cyril Soler, Nicolas Holzschuch, and François X Sillion. Accurate detection of symmetries in 3d shapes. *ACM Transactions on Graphics (TOG)*, 25(2):439–464, 2006.
- [NN07] J. Novatnack and K. Nishino. Scale-dependent 3d geometric features. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007.
- [OMMG10] Maks Ovsjanikov, Quentin Mérigot, Facundo Mémoli, and Leonidas Guibas. One point isometric matching with the heat kernel. In *Computer Graphics Forum*, volume 29, pages 1555–1564. Wiley Online Library, 2010.
- [OOFB08] R. Ohbuchi, K. Osada, T. Furuya, and T. Banno. Salient local visual features for shape-based 3d model retrieval. In *Shape Modeling and Applications, 2008. SMI 2008. IEEE International Conference on*, pages 93–102, June 2008.

- [OSG08] Maks Ovsjanikov, Jian Sun, and Leonidas Guibas. Global intrinsic symmetries of shapes. In *Computer Graphics Forum*, volume 27, pages 1341–1348. Wiley Online Library, 2008.
- [Ovs11] Maksims Ovsjanikovs. *Spectral methods for isometric shape matching and symmetry detection*. Stanford University, 2011.
- [PS13] Giuseppe Patané and Michela Spagnuolo. Heat diffusion kernel and distance on surface meshes and point sets. *Computers & Graphics*, 37(6):676–686, 2013.
- [PSG⁺06] Joshua Podolak, Philip Shilane, Aleksey Golovinskiy, Szymon Rusinkiewicz, and Thomas Funkhouser. A planar-reflective symmetry transform for 3d shapes. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 549–559. ACM, 2006.
- [RBB⁺10] Dan Raviv, Alexander M Bronstein, Michael M Bronstein, Ron Kimmel, and Guillermo Sapiro. Diffusion symmetries of non-rigid shapes. In *Proc. International Symposium On 3D Data Processing, Visualization And Transmission (3DPVT)*. Citeseer, 2010.
- [RBBK07] Dan Raviv, Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Symmetries of non-rigid shapes. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference On*, pages 1–7. IEEE, 2007.
- [RBBK10a] Dan Raviv, Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Full and partial symmetries of non-rigid shapes. *International Journal Of Computer Vision*, 89(1):18–39, 2010.
- [RBBK10b] Dan Raviv, Michael M Bronstein, Alexander M Bronstein, and Ron Kimmel. Volumetric heat kernel signatures. In *Proceedings Of The ACM Workshop On 3D Object Retrieval*, pages 39–44. ACM, 2010.
- [Rus07a] Raif M. Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing, SGP '07*, pages 225–233, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.
- [Rus07b] Raif M Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 225–233. Eurographics Association, 2007.
- [RY92] Daniel Reissfeld and Yehezkel Yeshurun. Robust detection of facial features by generalized symmetry. In *International Conference on Pattern Recognition*, pages 117–117. IEEE COMPUTER SOCIETY PRESS, 1992.
- [SB11] Ivan Sipiran and Benjamin Bustos. Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, 27(11):963–976, 2011.
- [SB12] Ivan Sipiran and Benjamin Bustos. Key-component detection on 3d meshes using local features. In *Proceedings Of The 5th Eurographics Conference On 3D Object Retrieval*, pages 25–32. Eurographics Association, 2012.
- [Sch78] Erwin Schmid. An algorithm to compute the eigenvectors of a symmetric matrix. 1978.
- [Sha08] Ariel Shamir. A survey on mesh segmentation techniques. In *Computer Graphics Forum*, volume 27, pages 1539–1556. Wiley Online Library, 2008.
- [SML00] Ilan Shimshoni, Yael Moses, and Michael Lindenbaum. Shape reconstruction of 3d bilaterally symmetric surfaces. *International Journal of Computer Vision*, 39(2):97–110, 2000.

- [SOG09] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer Graphics Forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.
- [SS88] C Harris SI and M Stephens. A combined corner and edge detection. In *Proc. 4th Alvey Vision Conference*, pages 147–151, 1988.
- [STK02] Shymon Shlafman, Ayellet Tal, and Sagi Katz. Metamorphosis of polyhedral surfaces using decomposition. In *Computer Graphics Forum*, volume 21, pages 219–228. Wiley Online Library, 2002.
- [VBCG10] Amir Vaxman, Mirela Ben-Chen, and Craig Gotsman. A multi-resolution approach to heat kernels on discrete surfaces. *ACM Transactions On Graphics (TOG)*, 29(4):121, 2010.
- [WWV85] Jan D Wolter, Tony C Woo, and Richard A Volz. Optimal algorithms for symmetry detection in two and three dimensions. *The Visual Computer*, 1(1):37–48, 1985.
- [Xu04] Guoliang Xu. Convergence of discrete laplace-beltrami operators over surfaces. *Computers & Mathematics With Applications*, 48(3):347–360, 2004.
- [XZJ⁺12] Kai Xu, Hao Zhang, Wei Jiang, Ramsay Dyer, Zhiquan Cheng, Ligang Liu, and Baoquan Chen. Multi-scale partial intrinsic symmetry detection. *ACM Transactions On Graphics (TOG)*, 31(6):181, 2012.
- [XZT⁺09] Kai Xu, Hao Zhang, Andrea Tagliasacchi, Ligang Liu, Guo Li, Min Meng, and Yueshan Xiong. Partial intrinsic reflectional symmetry of 3d shapes. In *ACM Transactions On Graphics (TOG)*, volume 28, page 138. ACM, 2009.
- [ZGLG12] Wei Zeng, Ren Guo, Feng Luo, and Xianfeng Gu. Discrete heat kernel determines discrete riemannian metric. *Graphical Models*, 74(4):121–129, 2012.
- [ZRH_{ar}] Valentin Zobel, Jan Reininghaus, and Ingrid Hotz. Generalized heat kernel signature. *Journal of WSCG, International Conference on Computer Graphics, Visualization and Computer Vision*, 2011 to appear.
- [ZSH00] Malte Zöckler, Detlev Stalling, and Hans-Christian Hege. Fast and intuitive generation of geometric shape transitions. *The Visual Computer*, 16(5):241–253, 2000.
- [ZTS02] Emanoil Zuckerberger, Ayellet Tal, and Shymon Shlafman. Polyhedral surface decomposition with applications. *Computers & Graphics*, 26(5):733–743, 2002.
- [ZYF13] Zhiyuan Zhang, KangKang Yin, and Kelvin WC Foong. Symmetry robust descriptor for non-rigid surface matching. In *Computer Graphics Forum*, volume 32, pages 355–362. Wiley Online Library, 2013.

Paper I

*A new approach for fast interest points detection algorithm
in 3D shapes*

A new approach for fast interest points detection algorithm in 3D shapes

Cristian Lopez Del Alamo^{1,2} and Luciano Romero² Lizet Fuentes²

¹La Salle University & Arequipa, Peru

²National University of San Agustin & Arequipa, Peru

Abstract

Due to the increasing amount of data and the reduction of costs in 3D data acquisition devices, there has been a growing interest, in developing efficient and robust feature extraction algorithms for 3D shapes, invariants to isometric, topological and noise changes, among others. One of the key tasks for feature extraction in 3D shapes, is the interest points detection; where interest points are salient structures, which can be used, instead of the whole object. In this research, we present a new approach to detect interest points in 3D shapes by analyzing the triangles that compose the mesh which represent the shape, in different way to other algorithms more complex such as Harris 3D or HKS. Our results and experiments of repeatability, confirm that our algorithm is stable and robust, in addition to being of linear order in the number of faces of the mesh.

1. Introduction

In recent years, interest points detection have a growing trend; the aim is choose a subset of points, which represent distinctive characteristics, such as saliency or level of protrusion of outstanding local structure; instead a whole object. Interest point detection, is a initial process for other subsequent relevant process, like key component detection [SB13], registration partial shapes in 3D database, intrinsic simetry detection, 3D model retrieval [OOFB08] mesh recognition, partial matching, viewpoint selection [LVJ05], mesh segmentation, mesh simplification, registration [DCG12].

Define the concept of interest point is ambiguous, Because it is necessary to incorporate human perception; in this paper, interest point is defined as the level of protrusion of outstanding local structure [SB11a].

There are many works to detect relevant points in 2D images, but in 3D models new problems more complex appeared, such as deformation, scale and others.

On the other hand, compute interest points in 3D shapes efficiently with great amount of vertex and edges, is a complex task. Therefore, to reduce the information, many techniques has been developed, such as mesh saliency [LVJ05], salient points [CCFM08], 3D Harris [SB11a], 3D sift [OOFB08], scale dependt corners [NN07], HKS- based

interest point detection [SOG09a]; all this techniques, were compared in efficiency and robustness in [DCG12].

2. Related Works

Interest points are those points with a high variation in the surface of the structure with respect to its neighbors. For 2D images, the interest points are those pixels with higher intensity variation over its neighbors. Several algorithms have been developed to find the interest points in 2D images; however, in 3D shapes, the interest points are on the protrusions of the shapes.

Several approaches have been proposed for interest point detection in 3D shapes and, most of they, are extensions of the detectors, proposed for images. One recent method to detect interest points in 3D shapes, is Harris 3D, which was proposed for [Glo09] and implemented for [SB11b]. Other method was proposed for [SOG09b], which defines the Heat Kernel Signature as a temporal domain restriction of the Heat Kernel on a manifold, which is related to the Laplace Beltrami spectrum.

2.1. Harris 3D

Ivan and Bustos proposed a extension of Harris algorithm, In such a way that instead of find points of interest in 2D images, it is possible to find points of interest in 3D



Figure 1: Methodology

shapes [SB11b]. The idea is very similar to that of Harris in images [Det88], Only that instead of analyzing pixels, vertices are discussed now. Furthermore, since the 3D shapes are modeled as a polygon mesh, the data structure is no longer a matrix, to become a graph.

In Harris 3D, for each vertex v in the graph G , a r -ring is defined as the k nearest neighbors to the vertex v in the graph G . We use 2-ring as suggested Sipiran. Then, for the set of vertices of each 2-ring, a square surface is adjusted by the least squares method, see equation 1. Then, move the set of points, in a way that the centroid is at the origin of the 3D coordinate system, by principal component analysis. Finally, it is possible to calculate the derivatives at the vertex V .

$$z = f(x, y) = \frac{p_1}{2}x^2 + p_2xy + \frac{p_3}{2}y^2 + p_4x + p_5y + p_6 \quad (1)$$

The equations 2, 3 y 4, shows the integrals in a gaussian neighbors of the vertex v

$$A = \frac{1}{\sqrt{2\pi\sigma}} \int_R e^{-\frac{(x^2+y^2)}{2\sigma^2}} \cdot f_x(x, y)^2 dx dy \quad (2)$$

$$B = \frac{1}{\sqrt{2\pi\sigma}} \int_R e^{-\frac{(x^2+y^2)}{2\sigma^2}} \cdot f_y(x, y)^2 dx dy \quad (3)$$

$$C = \frac{1}{\sqrt{2\pi\sigma}} \int_R e^{-\frac{(x^2+y^2)}{2\sigma^2}} \cdot f_x(x, y) \cdot f_y(x, y)^2 dx dy \quad (4)$$

$$H = \begin{bmatrix} A & C \\ C & B \end{bmatrix}.$$

Where H is the Harris 3D value for one vertex.

Finally, we take those k vertices, whose absolute values from H are the greater. These resulting k vertices are the interest point.

2.2. HKS

Suppose we have a surface M , which is applied to a heat source at a point p of the surface M . The problem of finding

the amount of heat dissipation at point p after a time t is solved by the equation of the heat equation, whose solution is designated as Heat Kernel. Formally we say that if $u(p, t)$ is the average heat at point p after a time t , then Heat Kernel is defined as in the equation 5.

$$u(p, t) = \frac{\partial u}{\partial t} = \Delta u \quad (5)$$

Where Δu is the Laplace-Beltrami operator [DK10].

In [SOG09b] proposed the Heat Kernel Signature as a restriction on the time domain of the Heat Kernel, so that you can get a signature for each vertex of the graph representing the 3D model.

$$HKS(x, t) = \sum_{i=0}^{\infty} e^{-\lambda_i t} \phi_i^2(x) \quad (6)$$

In the equation 6, λ_i and $\phi_i(x)$ are the autovalues and autovectors of Laplace-Beltrami operator.

Finally, a vertex is selected as interest point, when for large time values, its signature has a maximum with respect to the neighbor vertices. [OSG08].

3. Methodology

In this research, we propose a new approach for fast interest point detection. In contrast to more complex algorithms such as Harris 3D [SB11b] and HKS [SOG09b], we just need to calculate the areas of the shape faces and place it in increasing order. Our *Minimal area interest points* (MAIP) algorithm, like the others, is robust to topological, isometric changes, and presence of noise and holes; with the advantage, that be of lineal order to number of faces.

The methodology followed is summarized in the figure 1.

3.1. Compute areas

Let M a triangular mesh, that represents a shape, where $v_1, v_2, v_3 \in \mathbb{R}^3$, conforms a face (triangle) in M . Area is

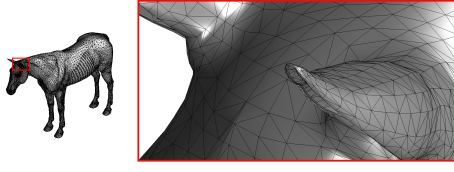


Figure 2: Shape areas

defined as:

$$area = \frac{det(A)}{2} \quad (7)$$

where

$$A = \begin{vmatrix} v_1(x) & v_1(y) & v_1(z) \\ v_2(x) & v_2(y) & v_2(z) \\ v_3(x) & v_3(y) & v_3(z) \end{vmatrix}$$

For each triangle, we compute areas using the equation 7

3.2. Sort areas

We have the assumption, that triangles with smaller areas, are located in high protusion zones, therefore, the vertices which composed the faces (triangles), are interest points. This will be demonstrated in our tests and experiments in the section 4.

Figure 2, shows smaller areas, are located in high protusion zones (horse ears), and big areas correspond to plane zones.

3.3. Select triangles with smallest areas

We use a threshold t , in order to select the faces with small areas, where t represent a percent of total number of areas.

Figure 3 shows different set of interest points which depends of threshold t , for null human shape. The interest points are represented by red spheres.

3.4. Detect interest points

We calculate all vertices that conforms these faces, so that, the vertices are unique; this resulting vertices are the interest points.

4. Experimental evaluation and discussion

In this section, we show experimental results of our implementation of the MAIP method; in order to evaluate our method, we used a standard benchmark and we present results obtained by repeatability test.

4.1. The data set

We used the SHREC robust feature detection and description benchmark [PST*]; this data set is composed by three null shapes (human, dog and horse), represented as triangular meshes with approximately 10,000 – 50,000 vertices. For each shape nine transformations were applied to them. The transformations are: isometry (non-rigid almost inelastic deformations), topology (welding of shape vertices resulting in different triangulation), micro holes, big holes, global and local scaling, additive Gaussian noise, shot noise and down-sampling.

The strength levels of transformation were five, and the higher number represent the stronger the transformation, except for the isometry class. The total dataset size is 138, and for each shape there are 45 transformations.

4.2. Evaluation methodology

We followed the evaluation methodology from [PST*]. Let $F(Y) = \{y_k\}_k$, a set of feature points, which are detected by an interest point detection method. To measure the quality, *repeatability* criterion was used.

For each transformed shape Y in the dataset the groundtruth dense correspondence to the null shape X to be given in the form of pairs of points $C_0(X, Y) = \{(y'_k, x_k)\}_{k=1}^{|Y|}$, a feature point $y_k \in F(Y)$ is said to be *repeatabile* if a geodesic ball of radius R around the corresponding point $x'_k : (x'_k, y_k) \in C_0(X, Y)$ contains a feature point $x_j \in F(X)$. The subset $Fr(Y) \subset F(Y)$ of repeatable features is given by

$$Fr_{R,X}(Y) = \{y_k \in F(Y) : F(X) \cap B_R(x'_k) \neq \emptyset, (x'_k, y_k) \in C_0(X, Y)\},$$

where $B_R(x'_k) = \{x \in X : d_X(x, x'_k) \leq R\}$ and d_X denotes the geodesic distance function in X .

The repeatability $rep(Y, X)$ of $F(Y)$ in X is defined as the percentage of features from $F(Y)$ that are repeatable,

$$rep(X, Y) = \frac{Fr_{R,X}(Y)}{F(Y)}.$$

For a transformed shape Y and the corresponding null shape X , the overall feature detection quality was measured as $(rep(Y, X) + rep(X, Y))/2$. The value of $R = 5$ was used in the benchmark. This radius constitute approximately 1% of the shapes diameter. Features without groundtruth correspondence were ignored.

4.3. Results and comparison with other methods

We used the sizes of shapes without mesh reduction. Figure 4, shows horse shapes with interest points detected by our method, for all transformations with strength level of 5.

Tables 4,6, shows repeatability values of our method using 20% and 40% of vertices of human shapes, where the average number of detected points is 9835 and 19670 respectively.

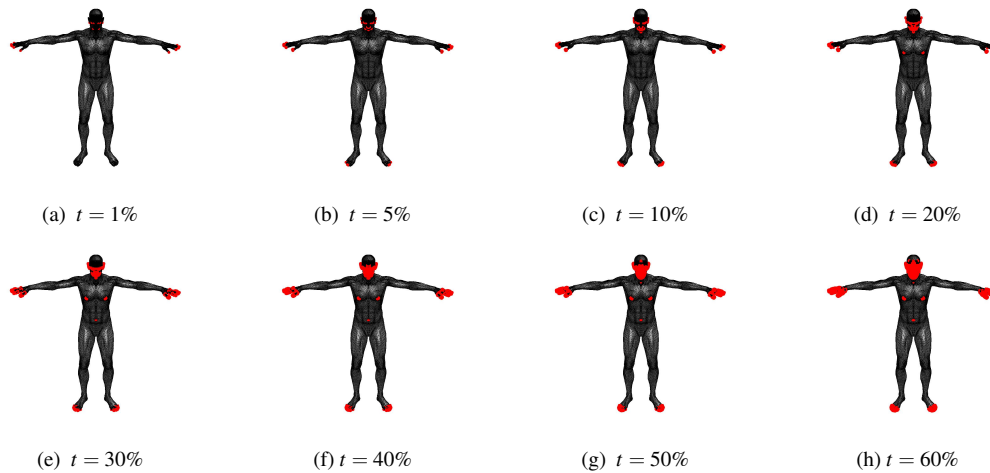


Figure 3: Shapes with interest points for different t values

Table 1: Repeatability of HKS2 feature detection algorithm [SB11a].

Transform	Strength					AVG
	1	≤ 2	≤ 3	≤ 4	≤ 5	
Holes	91.48	90.60	86.78	83.73	81.86	86.89
Isometry	98.08	98.72	98.01	97.88	98.04	98.15
Local Scale	98.08	94.83	90.09	83.05	78.31	88.87
Micro Holes	98.08	96.69	96.00	95.52	94.87	96.23
Noise	95.30	92.78	91.67	89.24	87.62	91.32
Sampling	97.05	97.88	97.39	96.27	92.35	96.19
Scale	99.36	99.36	98.50	97.90	97.68	98.56
Shot Noise	98.08	96.22	93.39	90.45	87.32	93.09
Topology	97.44	96.10	92.26	91.22	88.64	93.13
Average	96.99	95.91	93.79	91.70	89.63	93.60

Table 2: Repeatability of HKS3 feature detection algorithm [SB11a].

Transform	Strength					AVG
	1	≤ 2	≤ 3	≤ 4	≤ 5	
Holes	80.54	79.00	75.25	72.10	69.99	75.38
Isometry	100.00	100.00	100.00	100.00	100.00	100.00
Local Scale	97.44	96.79	93.02	87.25	82.90	91.48
Micro Holes	100.00	100.00	98.15	96.58	95.64	98.07
Noise	100.00	95.19	93.16	89.37	85.77	92.70
Sampling	100.00	100.00	100.00	100.00	96.20	99.24
Scale	100.00	100.00	100.00	98.61	97.78	99.28
Shot Noise	100.00	95.30	90.03	82.10	74.38	88.36
Topology	94.44	90.38	87.45	88.70	85.76	89.35
Average	96.94	95.18	93.01	90.52	87.60	92.65

Repeatability values of our method using the dog shapes are showed in the tables 8,9, using 20% and 40% of vertices, where the average number of detected points is 19670 and 4768 respectively.

Table 3: Repeatability of H3D feature detection algorithm [SB11a].

Transform	Strength					AVG
	1	≤ 2	≤ 3	≤ 4	≤ 5	
Holes	94.62	94.43	94.10	94.01	93.81	94.19
Isometry	96.01	96.73	96.26	96.60	96.62	96.44
Local Scale	96.24	94.96	93.40	91.26	88.84	92.94
Micro Holes	96.01	96.01	95.98	95.96	95.95	95.98
Noise	93.09	92.58	91.59	90.33	88.79	91.28
Sampling	95.31	93.62	92.08	89.13	80.42	90.11
Scale	97.06	96.89	96.28	95.62	94.94	96.16
Shot Noise	96.03	95.66	95.00	93.83	92.79	94.66
Topology	96.01	95.97	95.82	95.73	95.71	95.85
Average	95.60	95.21	94.50	93.61	91.99	94.18

Table 4: Repeatability of our method using 20% of vertices of human shapes. Average number of detected points: 9835.

Transform	Strength					AVG
	1	≤ 2	≤ 3	≤ 4	≤ 5	
Holes	93.91	92.57	92.54	85.93	75.52	88.10
Isometry	98.51	98.47	95.06	97.98	92.00	96.41
Local Scale	85.94	79.62	45.31	30.06	26.18	53.42
Micro Holes	98.51	98.51	98.53	98.53	98.53	98.52
Noise	74.97	55.74	47.06	42.50	39.13	51.88
Sampling	22.91	9.68	4.33	1.83	0.38	7.83
Scale	98.53	98.51	98.51	98.49	98.53	98.51
Shot Noise	98.51	98.51	98.51	98.51	98.51	98.51
Topology	98.51	98.51	98.51	98.53	98.51	98.51
Average	85.59	81.12	75.37	72.49	69.70	76.85

For the horse shapes, repeatability values of our method, using 20% and 40% of vertices, are showed in tables 10,11, where the average number of detected points is 3631 and 7264 respectively.

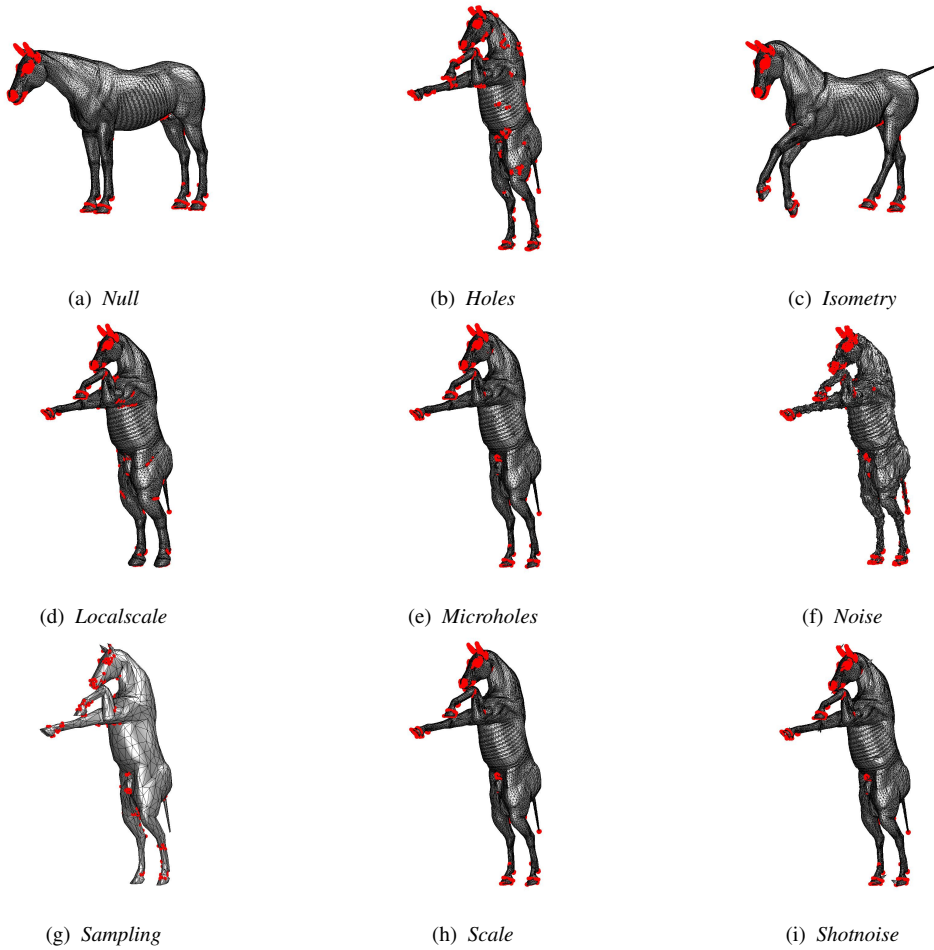


Figure 4: Shape transformations with strength level of 5 and $t = 20\%$

Table 5: Methods with the best performance by transformations and strengths of human shapes. HKS-Heat Kernel Signature. H3D-Harris 3D. MAIP using 20% of vertices.

Transform	Strength					
	1	≤ 2	≤ 3	≤ 4	≤ 5	AVG
Holes	H3D	H3D	H3D	H3D	H3D	H3D
Isometry	HKS3	HKS3	HKS3	HKS3	HKS3	HKS3
Local Scale	HKS2	HKS3	H3D	H3D	H3D	H3D
Micro Holes	HKS3	HKS3	MAIP	MAIP	MAIP	MAIP
Noise	HKS3	HKS3	HKS3	H3D	H3D	HKS3
Sampling	HKS3	HKS3	HKS3	HKS3	HKS3	HKS3
Scale	HKS3	HKS3	HKS3	HKS3	MAIP	HKS3
Shot Noise	HKS3	MAIP	MAIP	MAIP	MAIP	MAIP
Topology	MAIP	MAIP	MAIP	MAIP	MAIP	MAIP
Average	HKS2	HKS2	H3D	H3D	H3D	H3D

Table 6: Repeatability of our method using 40% of vertices of human shapes. Average number of detected points: 19670.

Transform	Strength					
	1	≤ 2	≤ 3	≤ 4	≤ 5	AVG
Holes	94.12	93.10	92.67	88.27	78.21	89.27
Isometry	97.53	96.90	94.66	96.43	94.24	95.95
Local Scale	88.87	82.00	69.26	50.89	45.77	67.36
Micro Holes	97.53	97.52	97.51	97.51	97.47	97.51
Noise	88.63	75.17	67.89	63.61	60.33	71.13
Sampling	40.46	19.90	9.08	3.76	0.76	14.79
Scale	97.55	97.53	97.52	97.52	97.54	97.53
Shot Noise	97.51	97.52	97.52	97.49	97.45	97.50
Topology	97.52	97.53	97.53	97.52	97.53	97.53
Average	88.86	84.13	80.40	77.00	74.37	80.95

Tables 1, 2 and 3 shows the repeatability values of methods evaluated in [SB11a], corresponding to algorithms

of HKS-Heat Kernel Signature (HKS2, HKS3) and H3D-Harris 3D. These results are used to compare with our results.

Table 7: Methods with the best performance by transformations and strengths of human shapes. HKS-Heat Kernel Signature. H3D-Harris 3D. MAIP using 40% of vertices.

Transform	Strength					
	1	≤ 2	≤ 3	≤ 4	≤ 5	AVG
Holes	H3D	H3D	H3D	H3D	H3D	H3D
Isometry	HKS3	HKS3	HKS3	HKS3	HKS3	HKS3
Local Scale	HKS2	HKS3	H3D	H3D	H3D	H3D
Micro Holes	HKS3	HKS3	HKS3	MAIP	MAIP	HKS3
Noise	HKS3	HKS3	HKS3	H3D	H3D	HKS3
Sampling	HKS3	HKS3	HKS3	HKS3	HKS3	HKS3
Scale	HKS3	HKS3	HKS3	HKS3	HKS3	HKS3
Shot Noise	HKS3	MAIP	MAIP	MAIP	MAIP	MAIP
Topology	MAIP	MAIP	MAIP	MAIP	MAIP	MAIP
Average	HKS2	HKS2	H3D	H3D	H3D	H3D

Table 8: Repeatability of our method using 20% of vertices of dog shapes. Average number of detected points: 4768.

Transform	Strength					
	1	≤ 2	≤ 3	≤ 4	≤ 5	AVG
Holes	70.23	64.71	58.04	51.89	44.85	57.94
Isometry	80.19	89.37	82.53	86.02	94.58	86.54
Local Scale	76.82	64.78	56.81	40.87	30.14	53.89
Micro Holes	80.22	80.22	80.22	80.26	80.26	80.24
Noise	77.97	65.19	57.96	51.05	47.44	59.92
Sampling	36.66	22.63	12.86	6.25	1.26	15.93
Scale	80.19	80.19	80.19	80.19	80.19	80.19
Shot Noise	80.19	80.19	80.22	80.35	80.26	80.24
Topology	80.19	80.19	80.19	80.19	80.19	80.19
Average	73.63	69.72	65.45	61.90	59.91	66.12

Tables 5, 7, show the best performance by transformation and strengths, between methods HKS1, HKS2, H3D evaluated in [SB11a] and our method, with $t=20$ and $t=40$ respectively, of human shapes. For each strength level, the average performance is calculated and also for each transformation.

From the table 5, we can note that our method outper-

Table 9: Repeatability of our method using 40% of vertices of dog shapes. Average number of detected points: 9537.

Transform	Strength					
	1	≤ 2	≤ 3	≤ 4	≤ 5	AVG
Holes	80.16	76.17	70.64	63.83	55.50	69.26
Isometry	85.17	93.64	86.56	91.81	94.99	90.43
Local Scale	82.96	74.05	67.86	54.04	45.40	64.86
Micro Holes	85.17	85.17	85.17	85.17	85.17	85.17
Noise	84.21	78.68	72.45	68.17	64.89	73.68
Sampling	51.01	34.03	20.46	10.24	2.32	23.61
Scale	85.17	85.17	85.17	85.17	85.17	85.17
Shot Noise	85.21	85.22	85.24	85.21	85.19	85.21
Topology	85.17	85.17	85.17	85.14	85.17	85.17
Average	80.47	77.48	73.19	69.86	67.09	73.62

Table 10: Repeatability of our method using 20% of vertices of horse shapes. Average number of detected points: 3631.

Transform	Strength					
	1	≤ 2	≤ 3	≤ 4	≤ 5	AVG
Holes	82.07	73.32	66.19	59.73	50.10	66.28
Isometry	92.64	97.28	93.08	98.35	96.38	95.55
Local Scale	92.31	90.45	86.89	81.69	77.29	85.72
Micro Holes	92.59	92.55	92.79	92.69	92.69	92.66
Noise	89.89	81.04	74.20	68.59	64.28	75.60
Sampling	32.10	18.21	11.31	6.55	0.90	13.81
Scale	92.59	92.64	92.59	92.59	92.64	92.61
Shot Noise	92.64	92.55	92.59	92.55	92.55	92.58
Topology	92.59	92.64	92.59	92.64	92.59	92.61
Average	84.38	81.19	78.03	76.15	73.27	78.60

Table 11: Repeatability of our method using 40% of vertices of horse shapes. Average number of detected points: 7264.

Transform	Strength					
	1	≤ 2	≤ 3	≤ 4	≤ 5	AVG
Holes	85.11	79.11	73.15	67.28	60.21	72.97
Isometry	89.98	95.78	91.04	92.76	92.00	92.31
Local Scale	89.61	88.06	85.54	79.09	76.36	83.73
Micro Holes	89.98	89.96	89.98	89.96	89.96	89.97
Noise	89.44	85.56	81.37	78.36	74.98	81.94
Sampling	46.33	28.26	16.99	8.96	1.69	20.45
Scale	89.93	89.93	89.93	89.93	89.93	89.93
Shot Noise	89.93	89.96	89.93	89.91	89.79	89.91
Topology	89.98	89.96	89.96	89.98	89.96	89.97
Average	84.48	81.84	78.66	76.25	73.87	79.02

forms in shot noise, topology and micro holes transformations.

From the table 7, our method outperforms in shot noise and topology transformations.

However, it is important to note that for both, sampling and noise, our method has low results. This is because the presence of noise modifies the triangulation mesh, so that in places where it occurs, noise areas of the triangles is less, then, it decreases the interest points detection.

5. Conclusions

We present a fast interest point detection method in 3D shapes. Our experiments confirm that, effectively, the faces (triangles) with smaller areas, correspond to high protrusion zones, and the vertices that conforms those triangles are considered interest points. Furthermore, the algorithm is very simple and efficient. An important advantage is that not necessary to do complex computations and has a complexity linear to number of faces.

In the comparison tables of the previous section, we can note that our method is robust to topological and isometric changes, as well, presence of noise, scaling, etc. Furthermore, our results obtained are similar to Harris 3D and HKS

methods. By the other hand, our results from the table 9 are superior in micro holes, shot noise and topology transformations.

Importantly, the presence of noise and the triangulation changes, decreases the ability to detect points of interest.

6. Future works

In future work, we develop an efficient method for find key components of the shapes. This algorithm clusters the key points geodesically closest, so that they form a cluster. On the other hand, it is possible to modify the algorithm to find the holes in the shapes, so that it is possible to perform a mesh repair process.

Acknowledgements

We thank the school of Computer Science of National University of San Agustin, La Salle University and the research group of Image Processing and Data Mining (IPRODAM), for supporting this research.

References

- [CCFM08] CASTELLANI U., CRISTANI M., FANTONI S., MURINO V.: Sparse points matching by combining 3d mesh saliency with statistical descriptors. *Comput. Graph. Forum* 27, 2 (2008), 643–652. 1
- [DCG12] DUTAGACI H., CHEUNG C. P., GODIL A.: Evaluation of 3d interest point detection techniques via human-generated ground truth. *Vis. Comput.* 28, 9 (Sept. 2012), 901–917. URL: <http://dx.doi.org/10.1007/s00371-012-0746-4>, doi:10.1007/s00371-012-0746-4. 1
- [Det88] DETECTOR E.: Chris harris and mike stephens plessey research hoke manor. *United Kingdom* (1988). 2
- [DK10] DUBROVINA A., KIMMEL R.: Matching shapes by eigendecomposition of the laplace-beltrami operator. In *Proc. 3DPVT* (2010), vol. 2. 2
- [Glo09] GLOMB P.: Detection of interest points on 3d data: Extending the harris operator. In *Computer Recognition Systems 3*. Springer, 2009, pp. 103–111. 1
- [LVJ05] LEE C. H., VARSHNEY A., JACOBS D. W.: Mesh saliency. *ACM Trans. Graph.* 24, 3 (July 2005), 659–666. URL: <http://doi.acm.org/10.1145/1073204.1073244>, doi:10.1145/1073204.1073244. 1
- [NN07] NOVATNACK J., NISHINO K.: Scale-dependent 3d geometric features. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (Oct 2007), pp. 1–8. doi:10.1109/ICCV.2007.4409084. 1
- [OOFB08] OHBUCHI R., OSADA K., FURUYA T., BANNO T.: Salient local visual features for shape-based 3d model retrieval. In *Shape Modeling and Applications, 2008. SMI 2008. IEEE International Conference on* (June 2008), pp. 93–102. doi:10.1109/SMI.2008.4547955. 1
- [OSG08] OVSJANIKOV M., SUN J., GUIBAS L.: Global intrinsic symmetries of shapes. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 1341–1348. 2
- [PST*] PRATIKAKIS I., SPAGNUOLO M., THEOHARIS T., (EDITORS R. V., BRONSTEIN A. M., BRONSTEIN M. M., CASTELLANI U., DUBROVINA A., GUIBAS L. J., HORAUD R. P., KIMMEL R., KNOSSOW D., LAVANTE E. V., MATEUS D., OVSJANIKOV M., SHARMA A.): Shrec 2010: robust correspondence benchmark. 3
- [SB11a] SIPIRAN I., BUSTOS B.: Harris 3d: A robust extension of the harris operator for interest point detection on 3d meshes. *Vis. Comput.* 27, 11 (Nov. 2011), 963–976. URL: <http://dx.doi.org/10.1007/s00371-011-0610-y>, doi:10.1007/s00371-011-0610-y. 1, 4, 5
- [SB11b] SIPIRAN I., BUSTOS B.: Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer* 27, 11 (2011), 963–976. 1, 2
- [SB13] SIPIRAN I., BUSTOS B.: Key-components: Detection of salient regions on 3d meshes. *Vis. Comput.* 29, 12 (Dec. 2013), 1319–1332. URL: <http://dx.doi.org/10.1007/s00371-013-0870-9>, doi:10.1007/s00371-013-0870-9. 1
- [SOG09a] SUN J., OVSJANIKOV M., GUIBAS L.: A concise and provably informative multi-scale signature based on heat diffusion. In *Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2009), SGP '09, Eurographics Association, pp. 1383–1392. URL: <http://dl.acm.org/citation.cfm?id=1735603.1735621>. 1
- [SOG09b] SUN J., OVSJANIKOV M., GUIBAS L.: A concise and provably informative multi-scale signature based on heat diffusion. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 1383–1392. 1, 2