

CÁLCULO DE MEDIAS GEOMÉTRICAS EN EL CONO DE LAS MATRICES  
SIMÉTRICAS SEMIDEFINIDAS POSITIVAS

JUAN ANDRÉS OLMOS ROJAS

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE CIENCIAS  
ESCUELA DE MATEMÁTICAS  
BUCARAMANGA  
2020

CÁLCULO DE MEDIAS GEOMÉTRICAS EN EL CONO DE LAS MATRICES  
SIMÉTRICAS SEMIDEFINIDAS POSITIVAS

JUAN ANDRÉS OLMOS ROJAS

Trabajo de investigación para optar al título de  
Pregrado en Matemáticas

Director:

Fabio Martínez Carrillo

Ph.D en Ingeniería de Sistemas y Computación

Codirector:

Juan Carlos Galvis

Ph.D en Matemáticas

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE CIENCIAS

ESCUELA DE MATEMÁTICAS

BUCARAMANGA

2020

## **AGRADECIMIENTOS**

El autor expresa su agradecimiento a la escuela de matemáticas por su formación y al grupo de investigación Bivl2ab de la escuela de ingeniería de sistemas por permitir extender los conocimientos a nuevas áreas de la ciencia. Este trabajo no habría sido posible sin la orientación y apoyo de los profesores Fabio Martínez en el área en la cuál se aplicó el trabajo y Juan Carlos Galvis en la formación matemática en nuevas áreas. Muchas gracias por su dirección, su paciencia, su excelente formación y dedicación.

Por último gracias a mi familia, seres cercanos y a las matemáticas.

## CONTENIDO

	pág.
<b>1. INTRODUCCIÓN Y PLANTEAMIENTO DEL PROBLEMA . . . . .</b>	<b>13</b>
<b>2. PRELIMINARES DE GEOMETRIA RIEMANNIANA . . . . .</b>	<b>20</b>
<b>3. FORMULACIÓN VARIACIONAL DE LA MEDIA GEOMÉTRICA . . . . .</b>	<b>28</b>
3.1. PROBLEMA DE OPTIMIZACIÓN . . . . .	29
<b>4. ALGORITMOS ESTUDIADOS . . . . .</b>	<b>40</b>
4.1. ALGORITMO PRINCIPAL DE GRADIENTE DESCENDIENTE: SOLUCIÓN DE PENNEC . . . . .	40
4.1.1. EJEMPLO MATRICES DE 2X2 . . . . .	41
4.1.2. CASO PARTICULAR: MEDIA GEOMÉTRICA . . . . .	45
4.2. CONTROL SOBRE EL GRADIENTE: ALGORITMO DE FLETCHER . . .	47
4.3. OTRA MÉTRICA PARA EL CÁLCULO DE LA MEDIA: LOG-EUCLIDEAN	48
4.4. ALGORITMO DE UNA LIBRERÍA DE GEOMETRÍA RIEMANNIANA APLI- CADA: PYRIEMANN . . . . .	49
4.5. MODIFICACIONES DEL ALGORITMO MEDIANTE AUTOVALORES GE- NERALIZADOS . . . . .	51
4.6. MODIFICACIONES DEL ALGORITMO MEDIANTE EL USO DE LA DES- COMPOSICIÓN DE CHOLESKY . . . . .	53
<b>5. RECONOCIMIENTO DE ACCIONES USANDO MEDIAS RIEMANNIANAS</b>	<b>56</b>
5.1. APLICACIÓN DEL FLUJO ÓPTICO A ACCIONES . . . . .	56
5.2. REGULARIZACIÓN DE MATRICES . . . . .	61

5.3. CLASIFICACIÓN DE ACCIONES: CLASIFICADOR BAYESIANO INGENUO . . . . .	62
5.4. PROCESO DE VALIDACIÓN CRUZADA . . . . .	65
<b>6. DISEÑO EXPERIMENTAL . . . . .</b>	<b>66</b>
6.1. SIMULACIONES CON MATRICES ALEATORIAS . . . . .	66
6.1.1. EJEMPLO DE CLASIFICACIÓN CON MATRICES DE BAJA DIMENSIONALIDAD . . . . .	66
6.2. RECONOCIMIENTO DE ACCIONES . . . . .	69
<b>7. EVALUACIÓN Y RESULTADOS . . . . .</b>	<b>71</b>
7.1. RESULTADOS EN SIMULACIONES CON MATRICES ALEATORIAS . . .	71
7.2. RESULTADOS EN RECONOCIMIENTO DE ACCIONES . . . . .	82
<b>8. CONCLUSIONES Y PERSPECTIVAS . . . . .</b>	<b>85</b>
<b>BIBLIOGRAFÍA . . . . .</b>	<b>87</b>
<b>ANEXOS . . . . .</b>	<b>91</b>

## LISTA DE FIGURAS

	pág.
Figura 1. Descripción de video por medio de características . . . . .	14
Figura 2. Descripción de video como secuencia de matrices de covarianza . .	16
Figura 3. Ilustración de baja dimensión del algoritmo de la media . . . . .	42
Figura 4. Ilustración de baja dimensionalidad del calculo de la media de matrices múltiples de otra. . . . .	46
Figura 5. Ejemplo imagen flujo óptico . . . . .	59
Figura 6. Características extraídas del flujo óptico . . . . .	61
Figura 7. Ejemplo de clasificación de matrices de baja dimensionalidad . . . .	67
Figura 8. Ejemplo imágenes del conjunto de datos . . . . .	69
Figura 9. Comparación de tiempo y tamaño con tolerancia $10^{-2}$ . . . . .	72
Figura 10.Comparación de tiempo y tamaño con tolerancia $10^{-6}$ . . . . .	73
Figura 11.Comparación de tiempo y tamaño con tolerancia $10^{-9}$ . . . . .	74
Figura 12.Comparación de tiempo y cantidad de matrices con tolerancia $10^{-2}$ .	75
Figura 13.Comparación de tiempo y cantidad de matrices con tolerancia $10^{-6}$ .	76
Figura 14.Comparación de tiempo y cantidad de matrices con tolerancia $10^{-9}$ .	78
Figura 15.Comparación del número de iteraciones y la cantidad de matrices . .	80
Figura 16Exactitud de los algoritmos en la clasificación de videos, secuencia 1	83
Figura 17Exactitud de los algoritmos en la clasificación de videos, secuencia 2	83
Figura 18.Tiempos de los algoritmos en el cálculo de la media de un video, secuencia 1 . . . . .	83

Figura 19.Tiempos de los algoritmos en el cálculo de la media de un video,	
secuencia 2 . . . . .	84

## LISTA DE ANEXOS

	<b>pág.</b>
Anexo A. Funciones de matrices . . . . .	91
Anexo B. Problema de autovalor generalizado . . . . .	93
Anexo C. Definiciones y resultados variacionales en matrices . . . . .	95
Anexo D. Cálculo de geodésicas . . . . .	98



## RESUMEN

**TÍTULO:** CÁLCULO DE MEDIAS GEOMÉTRICAS EN EL CONO DE LAS MATRICES SIMÉTRICAS SEMIDEFINIDAS POSITIVAS \*

**AUTOR:** JUAN ANDRÉS OLMOS ROJAS \*\*

**PALABRAS CLAVE:** RECONOCIMIENTO DE ACCIONES, MATRICES DE COVARIANZA, MATRICES SIMÉTRICAS DEFINIDAS POSITIVAS, VARIEDAD DE RIEMANN, MEDIA GEOMÉTRICA.

**DESCRIPCIÓN:** En el área de la visión por computador, la descripción de videos a partir de matrices de covarianza ha sido usada como un descriptor compacto para el reconocimiento de acciones, esto pues un video puede ser descrito por la media de las matrices de covarianza que describen a cada una de las imágenes que componen el video. Las matrices de covarianza resultan ser matrices simétricas semi definidas positivas y regularizando pueden ser vistas como matrices simétricas definidas positivas (SPD), las cuales forman una variedad riemanniana. El problema geométrico es entonces encontrar una media en este espacio, este ha sido estudiado a partir de definir diferentes métricas y con ellas plantear diferentes algoritmos para el cálculo de la media. En este proyecto se desarrolla una descripción matemática de este espacio mediante definiciones, construcciones y ejemplos de baja dimensión para tener una noción más clara del problema y posteriormente llevarlo a la práctica. Además, se plantean dos algoritmos: usando autovalores generalizados y otro haciendo una descomposición de Cholesky, esto con el fin de poder comparar teórica y computacionalmente los diferentes algoritmos existentes y los nuestros en problemas reales. Utilizando un conjunto de datos públicos de reconocimiento de acciones, se compararon 4 algoritmos para el cálculo de la media junto a los dos propuestos. El algoritmo propuesto, con autovalores generalizados, logró una exactitud del 69.95 %. Además, en experimentos complementarios con generadores aleatorios de matrices SPD, se evidenció una convergencia más rápida para el algoritmo Log-Euclidean.

---

\* Trabajo de investigación

\*\* Facultad de Ciencias. Escuela de Matemáticas. Director: Fabio Martínez Carrillo, Ph.D. Codirector: Juan Carlos Galvis, Ph.D.

## ABSTRACT

**TITLE:** GEOMETRIC MEANS ON THE CONE OF SYMMETRIC POSITIVE SEMI-DEFINITE MATRICES \*

**AUTHOR:** JUAN ANDRÉS OLMOS ROJAS \*\*

**KEYWORDS:** ACTION RECOGNITION, COVARIANCE MATRICES, POSITIVE DEFINITE MATRICES, RIEMANNIAN MANIFOLD, GEOMETRIC MEAN.

**DESCRIPTION:** In the area of computer vision, the description of videos from covariance matrices has been used as a compact descriptor for the recognition of actions, this because a video can be described by the mean of the covariance matrices that describe each one of the frames of the video. The covariance matrices are symmetric positive semi-defined (SPD) matrices and regularizing can be seen as symmetric positive defined matrices (SPD), which form a Riemannian manifold:  $S_{++}^n$ . The problem then is a geometric problem of finding a mean in this space. This problem has been studied from the definition of different metrics and with them propose different algorithms for calculate the mean. In this project, the aim is then to develop a clear mathematical description of this space by definitions, constructions and low-dimensional examples, in order to have a clearer notion of the geometric problem in order to later put it into practice. In addition, algorithms were also proposed: using generalized eigenvalues and doing a Cholesky decomposition, this in order to be able to compare theoretically and computationally different existing algorithms and ours in real problems. Using a public data set of action recognition, with 60 videos and 6 actions, 4 algorithms for the calculation of the mean were compared with two algorithms proposed in this work. The proposed algorithm, with generalized eigenvalues, achieved an accuracy of 69.95%. Furthermore, in complementary experiments with random covariance matrix generators, a faster convergence was evidenced for the *Log-Euclidean* algorithm. In general, all algorithms converge in a number of iterations ( $\sim 25$  iterations) for the calculation of the mean, regardless the number of matrices.

---

\* Bachelor work

\*\* Faculty of Science. School of Mathematics. Advisor: Fabio Martínez Carrillo, Ph.D. Co-advisor: Juan Carlos Galvis, Ph.D.

## 1. INTRODUCCIÓN Y PLANTEAMIENTO DEL PROBLEMA

El análisis de imágenes y la visión por computador juegan un rol fundamental en áreas como la biomedicina, los deportes, la vigilancia automática y el estudio de materiales, entre muchas otras. En particular, la descripción y clasificación de alto nivel de secuencias de vídeos es uno de los problemas mas importantes en visión por computador debido a sus desafíos inherentes a la alta dimensionalidad de los videos, y la alta variabilidad de representación que tienen las acciones en estas secuencias <sup>1,2</sup>. El principal objetivo en este campo de acción es la descripción y clasificación (automática) de alto nivel de secuencias de vídeos a partir de patrones de movimiento y de apariencia <sup>3</sup>. Esta tarea, sin embargo, resulta compleja por la alta variabilidad de movimientos y apariencia para representar las acciones y el fondo. También variaciones de la cámara, cambios de iluminación y subjetividad en la descripción de acciones pueden hacer mas complejo este problema. Además, los enfoques de la literatura son computacionalmente costosos y la exactitud en su clasificación depende de la dimensionalidad de los datos con los que se trabajen <sup>3</sup>. Particularmente en una secuencia de video, cada imagen  $I$  que lo compone es codificada en arreglos de dimensión  $(W \times H)$ , correspondiente a su ancho ( $W$  *width* en ingles) y alto ( $H$  *high* en ingles). Esta imagen  $I$  es representada por un conjunto

---

<sup>1</sup> D. Geronimo y col. "Survey of Pedestrian Detection for Advanced Driver Assistance Systems". En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.7 (2010), págs. 1239-1258. DOI: 10.1109/TPAMI.2009.122.

<sup>2</sup> Y. Wang, K. Huang y T. Tan. "Human Activity Recognition Based on R Transform". En: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. 2007, págs. 1-8. DOI: 10.1109/CVPR.2007.383505.

<sup>3</sup> David A. Forsyth y Jean Ponce. *Computer Vision - A Modern Approach, Second Edition*. Pitman, 2012, págs. 1-791.

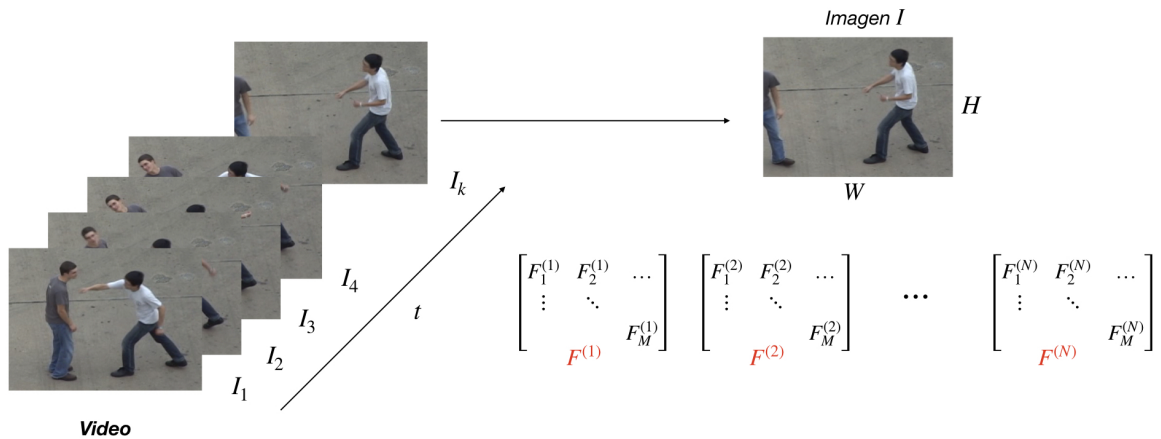
de  $N$  características, denotadas por  $F^{(1)}, F^{(2)}, \dots, F^{(N)}$  (donde podemos asumir que todas son de dimensión  $W \times H$ ). Por lo tanto la dimensión de representación de cada imagen  $I$  es  $W \times H \times N$ . Ver Figura 1 para una ilustración de lo anterior.

Cada característica  $F^{(i)}$  puede ser organizada en un vector de dimensión  $M = WH$ , es decir:  $F^{(i)} = [F_{\ell,k}^{(i)}]_{\ell=1,k=1}^{W,H} = [F_1^{(i)}, \dots, F_M^{(i)}]$ . Un enfoque alternativo para el análisis de secuencias de video y el manejo de la alta dimensionalidad es el uso de matrices de covarianza por su representación vectorial <sup>5</sup>. En este sentido, la covarianza entre dos características puede ser definida como:

---

<sup>5</sup> Xavier Pennec. "Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements". En: *Journal of Mathematical Imaging and Vision* 25.1 (2006). A preliminary appeared as INRIA RR-5093, January 2004., págs. 127-154. DOI: 10.1007/s10851-006-6228-4.

**Figura 1.** Un video es un conjunto de imágenes de dimensión  $W \times H$ , donde cada imagen será descompuesta en  $N$  características  $F^{(1)}, F^{(2)}, \dots, F^{(N)}$  de dimensión  $W \times H$ . Imágenes tomadas del conjunto de datos *UT-Interaction* <sup>4</sup>.



$$C_{ij} = C_{ij}(F) = \frac{1}{M-1} \sum_{\ell=1}^M \left( F_{\ell}^{(i)} - m_i \right) \left( F_{\ell}^{(j)} - m_j \right), \quad 1 \leq i, j \leq N,$$

donde  $m_i = \frac{1}{M} \sum_{\ell=1}^M F_{\ell}^{(i)}$  es la media de la característica  $F^{(i)}$ . De esta forma, a una imagen  $F$  se le puede asociar una matriz de covarianza

$$C_F = [C_{ij}]_{i=1, j=1}^{N, N},$$

de dimensión  $N \times N$ . Observe que si definimos la matriz

$$\check{F} = [F^{(1)} - m_1, F^{(2)} - m_2, \dots, F^{(N)} - m_N],$$

entonces

$$C_F = \frac{1}{M-1} \check{F}^T \check{F},$$

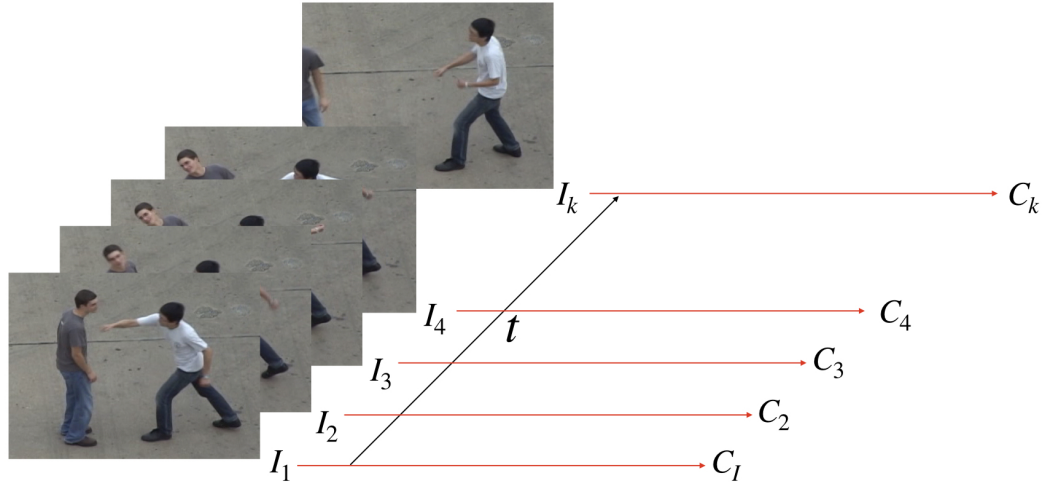
de donde se puede ver que  $C_F$  es simétrica claramente ( $C_{ij} = C_{ji}$ ) y semi-definida positiva. Más aun, para un  $v \neq 0$ ,  $v \check{F}^T \check{F} v^T = (\check{F} v^T)^T (\check{F} v^T) > 0$  siempre que  $\check{F}$  tenga columnas linealmente independientes. Al ser simétrica, vemos que para describirla solo se necesita su parte triangular superior que se podría ver como un vector de dimensión  $\frac{N(N+1)}{2}$  y mediante un proceso de regularización (vea Sección 5), estas matrices pueden asumirse simétricas definidas positivas. Ahora, si se toma  $N \ll \min \{W, H\}$  vemos que  $\frac{N(N+1)}{2} \ll W \times H \times N$ , y por tanto puede resultar atractivo (en términos de memoria) calcular estas matrices para describir propiedades de una imagen o un video. En la practica las matrices de covarianza tienen la ventaja de poder compactar y reducir la dimensión de una posible gran cantidad de datos, y además, permite analizar de manera mas eficiente la relación existente entre las características estudiadas.

El reconocimiento de acciones requiere de video, así que se entenderá a este como un conjunto ordenado de imágenes denotadas por  $I_{(1)}, I_{(2)}, \dots, I_{(K)}$ , descritas por

$N$  características. Así, cada una de las  $K$  imágenes puede ser representada por una matriz de covarianza. Denotando  $C_{(1)}, C_{(2)}, \dots, C_{(K)}$ , donde  $C_{(i)}$  es la matriz de covarianza que le corresponde a la imagen  $I_{(i)}$  para  $i \in \{1, 2, \dots, K\}$ , ver Figura 2. Entonces, una forma compacta de representar estas  $K$  matrices de covarianza es mediante una descripción estadística del vídeo, como por ejemplo, una *media*. Esta media de las matrices  $C_{(1)}, C_{(2)}, \dots, C_{(K)}$  resulta ser un arreglo de  $N \times N$  que representa la información del video de dimensión  $W \times H \times N \times K$  (pues tiene  $K$  imágenes de  $W \times H$  con  $N$  características).

Una media que ha sido usada para representar estadísticamente el conjunto de

**Figura 2.** Un video es una secuencia de imágenes  $I_{(1)}, I_{(2)}, \dots, I_{(K)}$  que se puede describir con las matrices de covarianza  $C_{(1)}, C_{(2)}, \dots, C_{(K)}$ , respectivamente. Imágenes tomadas del conjunto de datos *UT-Interaction* <sup>6</sup>.



matrices de covarianza  $C_1, C_2, \dots, C_K$ , es la media de Fréchet mencionada en <sup>7,8,9</sup>. Este conjunto de matrices de covarianza ha sido estudiado precisamente porque calcular una media teniendo en cuenta el espacio que forman no es sencillo debido a su geometría. La idea entonces es calcular una media *geométrica* que tenga en cuenta la geometría del cono convexo formado por las SPD (como subconjunto del espacio vectorial de las matrices simétricas). Luego, se debe considerar una métrica en este cono, y una vez definida, surge la pregunta de cómo poder calcular una matriz que con respecto a esta métrica esté lo mas cerca de todas las matrices  $C_1, C_2, \dots, C_K$ . Pennec y Fletcher en <sup>7,8</sup> proponen algoritmos que se basan en el gradiente descendiente de Newton para solucionar

$$\min_C \sum_{i=1}^k \text{dist}(C, C_i)^2.$$

En <sup>7,8</sup> se verifica que, para una distancia apropiada, la matriz definida positiva  $C$  que minimiza esta suma es única. El algoritmo iterativo planteado en <sup>7</sup> para calcular esta matriz consiste en la iteración

$$C_{t+1} = \exp_{C_t} \left( \frac{1}{k} \sum_{i=1}^k \log_{C_t}(C_i) \right),$$

donde  $C_t$  es la aproximación anterior y  $C_0$  es una aproximación inicial. Las funciones

---

<sup>7</sup> Xavier Pennec, Pierre Fillard y Nicholas Ayache. "A Riemannian Framework for Tensor Computing". En: *International Journal of Computer Vision* 66.1 (2006), págs. 41-66. DOI: 10.1007/s11263-005-3222-z.

<sup>8</sup> P Thomas Fletcher y Sarang Joshi. "Riemannian Geometry for the statistical analysis of diffusion tensor Data". En: *Signal Processing* 87 (feb. de 2007), págs. 250-262. DOI: 10.1016/j.sigpro.2005.12.018.

<sup>9</sup> Xavier Pennec. "Probabilities and Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements". En: ene. de 1999, págs. 194-198.

de matrices  $\exp_{C_t}$  y  $\log_{C_t}$  son definidas a partir de la definición de funciones de matrices (ver <sup>10</sup> y 1).

Por todo lo anterior es de notar que las matrices de covarianza resultan ser una alternativa de representación compacta para la representación de videos en problemas de visión por computador. Algunos autores han propuesto explicaciones, descripciones y motivaciones matemáticas relacionadas al cálculo estadístico en estos conjuntos de matrices de covarianza; ver <sup>11,1,2,7,8</sup>. Para la definición y calculo de una media de un conjunto de matrices de covarianza, es importante saber que estas matrices forman una variedad de Riemann. Por tanto todos estos aspectos matemáticos son necesarios para entender, implementar y reconocer los alcances, deficiencias, ventajas y desventajas de los algoritmos para el cálculo de medias de matrices de covarianza. Además, y como es ya conocido en el campo de métodos numéricos y matemática aplicada, el entendimiento de estos aspectos genera oportunidades para la correcta aplicación de estas técnicas en tareas como el reconocimiento, clasificación y representación de acciones.

Este trabajo desarrolló un estudio del cálculo numérico de medias geométricas aplicado al problema de descripción compacta de secuencias de vídeo para el reconocimiento de acciones. Para ello, se hizo un estudio de medias geométricas basados en los ya existentes como el propuesto por P. Thomas Fletcher en el año 2007 en su artículo *Riemannian Geometry for the Statistical Analysis of Diffusion Tensor Data* <sup>8</sup> o el mencionado por Pennec en su artículo *A Riemannian Framework for Tensor Computing* <sup>7</sup>. El cálculo de la media geométrica se resume en un proceso iterativo, en el

---

<sup>10</sup> Nicholas J. Higham. *Functions of Matrices: Theory and Computation (Other Titles in Applied Mathematics)*. Philadelphia, PA, USA: Society for Industrial y Applied Mathematics, 2008.

<sup>11</sup> Wilson Moreno y Fabio Martinez. "Frame-Level Covariance Descriptor for Action Recognition Riemannian Geometry for the statistical analysis of diffusion tensor Data". En: *Colombian Conference on Computing* (sep. de 2018), págs. 276-290.



que cada iteración, consiste en mapear a la variedad usando la función exponencial el promedio de matrices de covarianza proyectadas al plano tangente (logaritmos de las covarianzas). En este trabajo se propusieron el uso de autovalores generalizados y la descomposición de Cholesky para disminuir los cálculos algebraicos involucrados en el cálculo de las funciones logaritmo y exponencial. Esta media de matrices de covarianza, brinda un gran panorama de las ventajas, desventajas y alcances en cuanto a la descripción compacta de vídeos para la clasificación y reconocimiento de acciones. También permite hacer un estudio para entender y explicar matemáticamente el cálculo de esta media junto con propiedades algebraicas y geométricas. Esto conllevará a reconocer criterios para su uso eficiente en diferentes aplicaciones de las ciencias de la computación, matemáticas aplicadas y áreas relacionadas. Este trabajo está organizado en un conjunto de capítulos que describen tanto el problema de reconocimiento de acciones en visión por computador, como los fundamentos de la geometría riemanniana y álgebra lineal para el análisis de matrices de covarianzas. En la sección 2 se describe como los vídeos pueden ser descritos como un conjunto de matrices SPD <sup>7</sup>, entendiendo las características de entrada y su representación a nivel de cuadro. Además en la Sección 3 se mostrará el problema de optimización a resolver para el cálculo de la media. Posteriormente, en la Sección 4 se mostrarán diferentes algoritmos con los que se ha resuelto este problema de optimización y algunos ejemplos de baja dimensionalidad. De manera particular, para el dominio de aplicación, en la Sección 5 se explicarán conceptos de clasificación y flujo óptico para posteriormente llevar a cabo aplicaciones con simulaciones y datos reales. En anexo se encuentran definiciones sobre funciones de matrices (1), resultados variacionales en matrices (3), el cálculo de geodésicas (4) y el cálculo de funciones de matrices con autovalores generalizados (2) que se utiliza en la Sección 4 para implementar dos algoritmos: mediante autovalores generalizados y mediante el uso de una descomposición de Cholesky.

## 2. PRELIMINARES DE GEOMETRIA RIEMANNIANA

Para trabajar sobre el espacio de las matrices SPD (denotado por  $S_{++}^n$ ), es importante primero entender la geometría de esta variedad. En primer lugar se definirá el concepto de variedad diferenciable, plano tangente y métrica riemanniana. Luego se mostrará como medir distancias en estas variedades, se presentarán las geodésicas y algunos teoremas como el de Hopf-Rinow y de Cartan-Hadamard para poder definir las funciones exponencial y logaritmo, muy importantes para este trabajo. Los resultados que se mostrarán en esta sección son basados en su mayoría del libro <sup>3</sup>, sin embargo hay otros libros de geometría riemanniana como <sup>12,13</sup> y más introductorios sobre geometría diferencial y variedades diferenciables como <sup>14,15</sup>.

Se puede comenzar entonces definiendo a un espacio *localmente euclidiano de dimensión  $n$*  como un espacio topológico  $\mathcal{M}$  tal que para cada  $p \in \mathcal{M}$  existe un abierto  $U \in \mathcal{M}$ , un abierto  $\tilde{U} \subset \mathbb{R}^n$  y un homeomorfismo  $\varphi : U \rightarrow \tilde{U}$  tal que  $p \in U \subset \mathcal{M}$ . Si  $\mathcal{M}$  es localmente euclidiano de dimensión  $n$ , Hausdorff y verifica el segundo axioma de numerabilidad se dice entonces que  $\mathcal{M}$  es una  *$n$ -variedad topológica*. Esta  $n$ -variedad topológica se puede empezar a cubrir con *cartas* que son parejas  $(U, \varphi)$  donde  $U$  es un abierto de  $\mathcal{M}$  que se conoce como el dominio de la carta y  $\varphi : U \rightarrow \tilde{U}$  es un homeomorfismo que se conoce como parametrización. Si se tienen dos cartas  $(U, \varphi)$  y  $(V, \phi)$  con  $U \cap V \neq \emptyset$ , entonces la composición

---

<sup>12</sup> Manfredo do Carmo. *Riemannian Geometry*. Birkhäuser Base, 1992.

<sup>13</sup> Jürgen Jost. *Riemannian Geometry and Geometric Analysis*. Springer-Verlag, 2008.

<sup>14</sup> John M. Lee. *Introduction to Smooth Manifolds*. University of Washington, Department of Mathematics, 2000.

<sup>15</sup> Serge Lang. *Fundamentals of Differential Geometry*. Vol. 191. Springer, 1999.

$\Phi := \phi \circ \varphi^{-1} : \varphi(U \cap V) \rightarrow \phi(U \cap V)$  es llamada la *función de transición de  $\varphi$  a  $\phi$*  y resulta ser un homeomorfismo. Dos cartas  $(U, \varphi)$  y  $(V, \phi)$  son *diferenciabilmente compatibles* si  $U \cap V = \emptyset$  o si la función de transición  $\Phi := \phi \circ \varphi^{-1}$  es un difeomorfismo. Se puede continuar cubriendo la variedad topológica hasta tener una colección de cartas  $\{U_i\}_{i \in I}$ , esta colección recibe el nombre de *Atlas* (denotado como  $\mathcal{A}$ ) cuando los dominios de las cartas cubren a  $\mathcal{M}$ , si además cualesquiera dos cartas en  $\mathcal{M}$  son diferenciabilmente compatibles se dice que  $\mathcal{M}$  es un *atlas diferenciable*. Si toda carta que sea diferenciabilmente compatible con toda carta de  $\mathcal{A}$  está en  $\mathcal{A}$ , entonces se dice que este atlas es *máximal*, es decir, no está estrictamente contenido en un atlas diferenciable. Las definiciones anteriores tienen sentido en cuanto a que dada una  $n$ -variedad topológica en algunas ocasiones se quiere construir una *estructura diferenciable*. Esta estructura diferenciable es precisamente un Atlas diferenciable máximo sobre la variedad. A la pareja  $(\mathcal{M}, \mathcal{A})$  donde  $\mathcal{M}$  es una  $n$ -variedad topológica y  $\mathcal{A}$  es su estructura diferenciable se le conoce como *variedad diferenciable*. Para trabajar sobre estas variedades son necesarios los siguientes conceptos que pueden verse con mas detalle en <sup>12</sup>.

Dadas  $\mathcal{M}_1$  y  $\mathcal{M}_2$  dos variedades diferenciables de dimensión  $n$  y  $m$  respectivamente, una función  $\varphi : \mathcal{M}_1 \rightarrow \mathcal{M}_2$  es diferenciable en un punto  $P \in \mathcal{M}_1$  si dada una parametrización  $\varphi_2 : V \subset \mathbb{R}^m \rightarrow \mathcal{M}_2$  con  $V$  abierto y  $\varphi(P) \in \varphi_2(V)$  va a existir una parametrización  $\varphi_1 : U \subset \mathbb{R}^n \rightarrow \mathcal{M}_1$  con  $U$  abierto y  $P \in \varphi_1(U)$  tal que la función  $\varphi_2^{-1} \circ \varphi \circ \varphi_1 : U \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$  es diferenciable en  $\varphi_1^{-1}(P)$ . Por otra parte, dada una variedad diferenciable  $\mathcal{M}$  de dimensión  $m$ , una función  $\gamma : (-\epsilon, \epsilon) \rightarrow \mathcal{M}$  es llamada *curva diferenciable* en  $\mathcal{M}$ . Como se menciona en <sup>16</sup>, si  $\mathcal{M} \subset \mathbb{R}^n$  entonces dado  $P \in \mathcal{M}$  considere todas las curvas  $\gamma : (-\epsilon, \epsilon) \rightarrow \mathcal{M} \subset \mathbb{R}^n$  tales que  $\gamma(0) = P$ . Se define entonces el espacio vectorial  $T_P \mathcal{M}$  como el conjunto de todos los *vectores tangentes*

---

<sup>16</sup> Elon Lima. *Análise Real Volume 2*. Ene. de 2004.

$\dot{\gamma}(0) = \gamma'(0)$  de estas curvas. Este conjunto de todos los vectores tangentes a  $\mathcal{M}$  en  $P$  es llamado *plano tangente*. Para poder comenzar a tomar medidas sobre estas variedades es necesario dotarlas de una métrica, la cual se conoce como *métrica Riemanniana* y resulta ser una correspondencia entre cada elemento  $P \in \mathcal{M}$  y un producto interno  $\langle \cdot, \cdot \rangle_P$  en el plano tangente  $T_P \mathcal{M}$  que varía diferenciablemente (esto es, la función  $P \mapsto \langle \cdot, \cdot \rangle_P$  es diferenciable <sup>12</sup>). Una *variedad de Riemann* se define entonces como una variedad diferenciable con una métrica Riemanniana.

Con esta métrica, dado un vector  $V \in T_P \mathcal{M}$  se puede calcular entonces  $\|V\|_P = \sqrt{\langle V, V \rangle_P}$  y dada una curva  $\gamma : [a, b] \rightarrow \mathcal{M}$ , se define la longitud de la curva como

$$L(\gamma) = \int_a^b \|\dot{\gamma}(t)\|_{\gamma(t)} dt,$$

donde  $\dot{\gamma}(t)$  denota el vector tangente a la curva en el punto  $\gamma(t)$ . Como se menciona en <sup>17</sup>, por lo anterior se puede definir una distancia entre dos puntos  $P, Q \in \mathcal{M}$ , la cual se conoce como *distancia Riemanniana* y viene dada por

$$d(P, Q) = \inf\{L(\gamma) \mid \gamma : [a, b] \rightarrow \mathcal{M} \text{ sea diferenciable por partes y } \gamma(a) = P, \gamma(b) = Q\}. \quad (1)$$

Con la anterior métrica,  $(\mathcal{M}, d)$  es un espacio métrico. Existen unas curvas especiales  $\gamma : [a, b] \rightarrow \mathcal{M}$  llamadas *geodésicas* que son las curvas que minimizan  $d(P, Q)$ . En particular como se menciona en <sup>12</sup>, estas curvas tienen la propiedad de que la longitud de sus vectores tangentes es constante. Estas no son necesariamente únicas (por ejemplo en la esfera, superficie de dimensión 2 podemos observar que existen infinitas geodésicas conectando el polo norte y el polo sur, ver <sup>12</sup>).

**Teorema 1.** (1.4.2 de <sup>13</sup>) Sea  $\mathcal{M}$  una variedad de Riemann,  $P \in \mathcal{M}$ ,  $V \in T_P \mathcal{M}$ .

---

<sup>17</sup> H. Quang Minh y Vittorio Murino. *Covariances in Computer Vision and Machine Learning*. Morgan & Claypool Publishers, 2017.

Entonces existe un  $\epsilon > 0$  y una única geodésica  $\gamma : [0, \epsilon] \rightarrow \mathcal{M}$  tal que  $\gamma(0) = P$  y  $\dot{\gamma}(0) = V$ . Note que  $\gamma$  depende diferencialmente de  $P$  y  $V$ , denote  $\gamma$  entonces como  $\gamma_V$ .

La función  $\exp_P : V_P \rightarrow \mathcal{M}$ , donde  $V_P = \{V \in T_P\mathcal{M} \mid \gamma_V \text{ está definido en } [0, 1]\}$  viene dada por  $\exp_P(V) = \gamma_V(1)$  y es llamada la *aplicación exponencial* de  $\mathcal{M}$  en  $P$ . Por ser  $\gamma_V(t)$  una geodésica se tiene entonces que  $\|\dot{\gamma}_V(t)\|_{\gamma_V(t)}$  es constante y así

$$L(\gamma_V) = \int_0^1 \|\dot{\gamma}_V(t)\|_{\gamma_V(t)} dt = \int_0^1 \|\dot{\gamma}_V(0)\|_{\gamma_V(0)} dt = \|\dot{\gamma}_V(0)\|_{\gamma_V(0)} = \|V\|_P. \quad (2)$$

Antes de continuar, se usará  $S^n$  para denotar el conjunto de las matrices simétricas de dimensión  $n$ ,  $S_+^n$  para denotar el conjunto de las matrices simétricas semi-definidas positivas de dimensión  $n$  y  $S_{++}^n$  para el conjunto de las matrices simétricas definidas positivas de dimensión  $n$ . Las siguientes definiciones y resultados son necesarios para conectar la teoría que se viene mencionando con el espacio formado por el conjunto  $S_{++}^n$  (ver más en <sup>13</sup>).

Una variedad de Riemann  $\mathcal{M}$  es *geodésicamente completa* si para todo  $P \in \mathcal{M}$ , la aplicación exponencial  $\exp_P$  está definida en todo  $T_P\mathcal{M}$  y cada geodésica  $\gamma_V$  se extiende infinitamente, es decir, cualquier geodésica  $\gamma(t)$  con  $\gamma(0) = P$  es definida para todo  $t \in \mathbb{R}$ . Propiedades importantes se pueden deducir del siguiente teorema.

**Teorema 2. (Hopf-Rinow).** *Para una variedad de Riemann  $\mathcal{M}$  las siguientes afirmaciones son equivalentes:*

1. *El espacio  $(\mathcal{M}, d)$  con la distancia mencionada en (1) es un espacio métrico completo.*
2. *Los subconjuntos cerrados y acotados de  $\mathcal{M}$  son compactos.*
3.  *$\mathcal{M}$  es geodésicamente completo.*

Más aún, 1. o 3. implican que todo par de puntos  $P, Q \in \mathcal{M}$  se puede unir por una geodésica de longitud  $d(P, Q)$ .

Esto último quiere decir que si la variedad es geodésicamente completa (entonces, dados dos puntos siempre hay una geodésica que los une, sin embargo, esta no necesariamente es única pero si se cumple que su longitud coincide con la distancia Riemanniana vista en (1), es decir es la de menor longitud).

**Teorema 3.** (1.4.3 de <sup>13</sup>) *La aplicación exponencial  $\exp_P$  envía vecindades de  $0 \in T_P\mathcal{M}$  mediante difeomorfismos a vecindades de  $P \in \mathcal{M}$ .*

De esto se tiene que existe una vecindad de  $P \in \mathcal{M}$  que se puede enviar mediante un difeomorfismo a una vecindad de  $0 \in T_P\mathcal{M}$ . Se puede definir entonces la aplicación logaritmo como la función inversa de la aplicación exponencial  $\log_P := \exp_P^{-1}$ .

Vamos ahora a enunciar el Teorema de Cartan-Hadamard que va a garantizar que la aplicación  $\exp_P$  esta globalmente definida y por la tanto podremos definir la aplicación inversa o  $\log_P$ . Para un enunciado auto-contenido de este resultado debemos revisar conceptos como el de cuadratura asociado a un variedad. Encontramos que una revisión completa de este concepto esta fuera del alcance de este trabajo y recomendamos la lectura referencia <sup>12</sup>. Antes de continuar se debe recordar que un espacio topológico es *conexo por caminos* si dados dos puntos  $P, Q \in \mathcal{M}$  existe un camino continuo  $\gamma : [0, 1] \rightarrow \mathcal{M}$  tal que  $\gamma(0) = P$  y  $\gamma(1) = Q$ , además  $\mathcal{M}$  es *reducible a un punto* si dado un lazo (es decir una función continua  $\gamma : [0, 1] \rightarrow \mathcal{M}$  tal que  $\gamma(0) = \gamma(1) = P \in \mathcal{M}$ ) en cualquier punto  $P \in \mathcal{M}$  existe una homotopía  $H : [0, 1] \times [0, 1] \rightarrow \mathcal{M}$  tal que  $H(x, 0) = \gamma(x)$  y  $H(x, 1) = P$ . Si un espacio topológico (en este caso la variedad  $\mathcal{M}$ ) satisface que es conexo por caminos y toda curva continua es reducible a un punto se dice entonces que es *simplemente conexo*.

**Teorema 4. (Cartan-Hadamard).** *Sea  $\mathcal{M}$  una variedad de Riemann geodésicamente completa con curvatura seccional no positiva. Si  $\mathcal{M}$  es simplemente conexo, en-*

tonces la aplicación exponencial  $\exp_P : T_P\mathcal{M} \rightarrow \mathcal{M}$  es un difeomorfismo para todo  $P \in \mathcal{M}$ .

Una variedad que satisfaga las condiciones del Teorema (4) se llama una variedad de *Cartan-Hadamard*, es decir debe ser geodésicamente completa, simplemente conexa y con curvatura seccional no positiva. Al ser geodesicamente completa, la variedad viene determinada con geodésicas únicas cuya longitud coincide con la distancia Riemanniana. Además, por el difeomorfismo va a haber un único  $V \in T_P\mathcal{M}$  que defina  $\exp_P(V) = Q$  y a la geodesica

$$\exp_P(tV), \quad \text{donde } \log_P(Q) = V,$$

así que teniendo en cuenta (2) la distancia geodésica de  $P$  a  $Q$  viene dada por

$$d(P, Q) = \|V\|_P = \|\log_P(Q)\|_P. \quad (3)$$

Para ver que  $S_{++}^n$  es una variedad diferenciable basta probar que  $S_{++}^n$  es un subconjunto abierto de  $S^n$ . Definiendo la función  $f : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^n$  donde  $f(A) = (\det_1(A), \dots, \det_n(A))^T$ , se puede ver que  $f$  es continua pues las funciones  $\det_i : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$  con  $1 \leq i \leq n$  son continuas, por tanto  $S_{++}^n$  va a ser abierto ya que  $S_{++}^n = f^{-1}(\mathbb{R}_+^n)$ . Por otra parte, se puede probar que para todo  $P \in S_{++}^n$  se cumple que  $T_P(S_{++}^n) = S^n$ . Note que por ejemplo en  $\mathbb{R}^3$  se cumple que  $T_P R \subset \mathbb{R}^3$ , es decir, todo plano tangente en un punto  $P$  a una superficie regular  $R$  en  $\mathbb{R}^3$  está contenido en  $\mathbb{R}^3$ , esto mismo sucede con  $S_{++}^n \subset S^n$ , es decir  $T_P(S_{++}^n) \subset S^n$  para todo  $P \in S_{++}^n$ , falta probar entonces que dado  $V \in S^n$  y  $P \in S_{++}^n$ ,  $V \in T_P(S_{++}^n)$ , definiendo  $\gamma : (-\epsilon, \epsilon) \rightarrow S_{++}^n$  como  $\gamma(t) = P + tV$  por ser  $S_{++}^n$  abierto de  $S^n$  va a existir un  $\delta > 0$  tal que  $\gamma(t)|_{(-\delta, \delta)} \subset S_{++}^n$ , entonces lo que se tiene es una curva  $\beta(t) = \gamma(t)|_{(-\delta, \delta)}$  en  $S_{++}^n$  tal que  $\beta(0) = P$  y  $\beta'(0) = V$ , probando así que  $V \in T_P(S_{++}^n)$ .

Pennec introduce en <sup>7</sup> una métrica riemanniana en  $S_{++}^n$  definiendo el siguiente pro-

ducto interno en cada espacio tangente  $T_P(S_{++}^n)$

$$\langle A, B \rangle_P = \langle P^{-\frac{1}{2}}AP^{-\frac{1}{2}}, P^{-\frac{1}{2}}BP^{-\frac{1}{2}} \rangle_F, \quad (4)$$

donde  $A, B \in T_P(S_{++}^n)$  y  $\langle A, B \rangle_F = \text{tr}(A^T B)$  es el producto de Frobenius. Este producto interno define una métrica Riemanniana conocida como *métrica afín invariante*. De esto se puede ver que

$$\begin{aligned} \|A\|_P &= \sqrt{\langle A, A \rangle_P} \\ &= \sqrt{\langle P^{-\frac{1}{2}}AP^{-\frac{1}{2}}, P^{-\frac{1}{2}}AP^{-\frac{1}{2}} \rangle_F} \\ &= \|P^{-\frac{1}{2}}AP^{-\frac{1}{2}}\|_F. \end{aligned} \quad (5)$$

De donde se puede observar que  $\|A\|_{Id} = \|A\|_F$ , es decir en la identidad la distancia viene dada por la norma de Frobenius. El producto interno definido en (4) resulta ser invariante a transformaciones afines, es decir

$$\langle CAC^T, CBC^T \rangle_{CPC^T} = \langle A, B \rangle_P, \quad \text{para todo } C \in GL(n), \quad (6)$$

donde  $GL(n)$  son las matrices no singulares de dimensión  $n$ . Un resultado importante que se desarrolla en el capítulo 12 de <sup>15</sup> es que la variedad diferenciable  $S_{++}^n$  con esta métrica *afín invariante* resulta ser una variedad Riemanniana y a la vez es una variedad Cartan-Hadamard, es decir, es simplemente conexa, geodésicamente completa y tiene curvatura seccional no positiva.

Como viene explicado en 4 la geodésica que pasa por  $\Sigma \in S_{++}^n$  con vector tangente  $w \in T_\Sigma(S_{++}^n)$  viene dada por

$$\Gamma_{(\Sigma, V)}(t) = \Sigma^{\frac{1}{2}} \exp \left( t \Sigma^{-\frac{1}{2}} w \Sigma^{-\frac{1}{2}} \right) \Sigma^{\frac{1}{2}},$$

entonces por la definición de la aplicación exponencial y teniendo en cuenta el teo-



rema (4), se tiene que la aplicación exponencial  $\exp_P : T_P(S_{++}^n) \rightarrow S_{++}^n$  está definida en todo  $T_P(S_{++}^n)$  y está dada por

$$\exp_P(V) = P^{\frac{1}{2}} \exp \left( P^{-\frac{1}{2}} V P^{-\frac{1}{2}} \right) P^{\frac{1}{2}}, \quad \text{donde } V \in T_P(S_{++}^n),$$

análogamente la aplicación logaritmo  $\log_P : S_{++}^n \rightarrow T_P(S_{++}^n)$  está definida en todo  $S_{++}^n$  y viene dada por

$$\log_P(Q) = P^{\frac{1}{2}} \log \left( P^{-\frac{1}{2}} Q P^{-\frac{1}{2}} \right) P^{\frac{1}{2}}, \quad \text{donde } Q \in S_{++}^n.$$

Por ser  $S_{++}^n$  de Cartan-Hadamard, la distancia riemanniana entre dos puntos está determinada por la longitud de la única geodésica que hay entre ellos. Entonces la distancia Riemanniana afín invariante está dada por

$$\begin{aligned} d(P, Q) &= \|\log_P(Q)\|_P \\ &= \|P^{\frac{1}{2}} \log \left( P^{-\frac{1}{2}} Q P^{-\frac{1}{2}} \right) P^{\frac{1}{2}}\|_P, \quad \text{donde } Q \in S_{++}^n. \\ &= \|\log \left( P^{-\frac{1}{2}} Q P^{-\frac{1}{2}} \right)\|_F \\ &= \sqrt{\text{tr} \left( \log \left( P^{-\frac{1}{2}} Q P^{-\frac{1}{2}} \right)^2 \right)} \end{aligned} \tag{7}$$

### 3. FORMULACIÓN VARIACIONAL DE LA MEDIA GEOMÉTRICA

Para definir una variedad riemanniana hace falta previamente definir una métrica riemanniana que en nuestro caso es la métrica afín invariante, la cual ha sido estudiada previamente por Pennec en <sup>7,5</sup> y además es investigada en varios artículos de medias geométricas, ver <sup>18,19,20</sup>. Si se tiene entonces una variedad riemanniana con una distancia asociada <sup>5,8</sup>, la idea es utilizar el enfoque variacional de Fréchet <sup>21,22</sup>, donde nota que la media  $\bar{x}$  de una variable aleatoria  $x \in \mathbb{R}^n$  minimiza a la varianza, es decir

$$\bar{x} = \arg \min_{y \in \mathcal{M}} \sigma_x^2(y)$$

donde si  $x$  son mediciones entonces  $\sigma_x^2(y) = \frac{1}{n} \sum_{i=1}^n d(y, x_i)^2$ . Al ser este un problema global de optimización la media de Fréchet esto puede acarrear algunas dificultades de existencia y unicidad y por ello Karcher en <sup>23</sup> propone utilizar mínimos

- 
- <sup>18</sup> Vincent Arsigny y col. "Geometric Means in a Novel Vector Space Structure on Symmetric Positive-Definite Matrices". En: *SIAM J. Matrix Analysis Applications* 29 (ene. de 2006), págs. 328-347. DOI: 10.1137/050637996.
- <sup>19</sup> P. Fletcher y Sarang Joshi. "Principal Geodesic Analysis on Symmetric Spaces: Statistics of Diffusion Tensors". En: vol. 3117. Ene. de 2004, págs. 87-98.
- <sup>20</sup> Maher Moakher. "A Differential Geometric Approach to the Geometric Mean of Symmetric Positive Definite Matrices". En: *SIAM J. Matrix Anal. Appl.* 26 (mar. de 2005), págs. 735-747. DOI: 10.1137/S0895479803436937.
- <sup>21</sup> M. Fréchet. *L'intégrale abstraite d'une fonction abstraite d'une variable abstraite et son application a la moyenne d'un élément aléatoire de nature quelconque*. Revue Scientifique, 1944, págs. 483-512.
- <sup>22</sup> Maurice René Fréchet. "Les éléments aléatoires de nature quelconque dans un espace distancié". fr. En: *Annales de l'institut Henri Poincaré* 10.4 (1948), págs. 215-310.
- <sup>23</sup> H. Karcher. "Riemannian center of mass and mollifier smoothing". En: *Communications on Pure and Applied Mathematics* 30.5 (1977), págs. 509-541.

locales y muestra además que en una variedad con curvatura seccional no positiva esta media está definida de forma única. Como menciona Fletcher en <sup>8</sup> se puede ver que al ser  $S_{++}^n$  una variedad riemanniana con curvatura seccional no positiva (ver <sup>15</sup>) entonces su media va a ser la solución única que optimice el problema de optimización mencionado.

El material presentado en esta sección es basado en gran parte al trabajo realizado en <sup>20</sup> y pretende inicialmente mostrar algunos resultados previos basados en la definición de la derivada matricial, algunas derivadas e integrales de matrices particulares, funciones especiales y propiedades, para posteriormente desarrollar una caracterización variacional de la media geométrica con base en el problema de optimización y al final llevar el problema de optimización a un problema matricial. Algunos definiciones y resultados utilizados en esta sección se presentan en detalle en el 3.

### 3.1. PROBLEMA DE OPTIMIZACIÓN

En <sup>20</sup> se menciona cómo las medias se han visto como la solución de un problema de optimización, por ejemplo dado un conjunto  $\{x_i\}_{1 \leq i \leq m} \subset \mathbb{R}$  el valor

$$\bar{x} = \arg \min_{x > 0} \sum_{k=0}^m d(x, x_k)^2,$$

va a ser la media aritmética cuando  $d(x, y) = |x - y|$  y la media geométrica cuando  $d(x, y) = |\log(x) - \log(y)|$ . Si ahora se considera el anterior problema pero con un conjunto de matrices  $\{X_1, X_2, \dots, X_m\}$  en la variedad de Riemann  $S_{++}^n$  con la

distancia definida en (7) entonces el valor apropiado para la media sería

$$\begin{aligned}\bar{X} &= \arg \min_{X \in S_{++}^n} \sum_{k=0}^m d(X, X_k)^2 \\ &= \arg \min_{X \in S_{++}^n} \sum_{k=0}^m \|\log \left( P^{-\frac{1}{2}} Q P^{-\frac{1}{2}} \right)\|_F^2.\end{aligned}$$

A continuación se mostrarán algunos resultados necesarios para plantear este problema de optimización.

**Teorema 5.** *Si  $A$  es simétrica definida positiva y  $s \in \mathbb{R}$ , las matrices  $((A - Id)s + Id)^{-1}$  y  $\log(A)$  conmutan siempre y cuando existan.*

*Demostración.* Denotando  $B = ((A - Id)s + Id)^{-1}$ , se quiere ver que  $B$  y  $\log(A)$  conmutan. Sea  $A = QDQ^T$  con  $Q^T Q = Id$  y  $D = \text{diag}(\lambda_i)_{i=1}^n$  una matriz diagonal. Sean  $D_1 = \text{diag}(\log(\lambda_i))_{i=1}^n$  y  $D_2 = \text{diag}\left(\frac{1}{(\lambda_i - 1)s + 1}\right)_{i=1}^n$ . Note que  $D_1$  y  $D_2$  conmutan por ser diagonales, además  $\log(A) = QD_1Q^T$  y  $B = QD_2Q^T$ . Por tanto

$$\begin{aligned}\log(A)B &= QD_1Q^T QD_2Q^T \\ &= QD_1D_2Q^T \\ &= QD_2D_1Q^T \\ &= QD_2Q^T QD_1Q^T \\ &= B\log(A),\end{aligned}$$

con lo que queda demostrado que  $B = ((A - Id)s + Id)^{-1}$  y  $\log(A)$  conmutan.  $\square$

**Teorema 6.** *Si  $A$  es simétrica definida positiva entonces*

$$\int_0^1 [(A - Id)s + Id]^{-2} ds = A^{-1}$$

*Demostración.* En esta prueba se usará definiciones presentadas en 3. Sea

$$M(s) = (Id - A)^{-1}[(A - Id)s + Id]^{-1}.$$

Por un razonamiento similar al Teorema anterior, se puede ver que

$$M(s) = (Id - A)^{-1}[(A - Id)s + Id]^{-1} = [(A - Id)s + Id]^{-1}(Id - A)^{-1}. \quad (8)$$

Entonces  $M(s) [(A - Id)s + Id] = (Id - A)^{-1}$  y derivando esto se tiene

$$M'(s) [(A - Id)s + Id] + M(s)(A - Id) = 0,$$

y por tanto

$$M'(s) = M(s)(Id - A)[(A - Id)s + Id]^{-1}, \quad \text{usando (8)}$$

$$M'(s) = [(A - Id)s + Id]^{-2}.$$

De donde

$$\begin{aligned} \int_0^1 [(A - Id)s + Id]^{-2} &= M(1) - M(0) \\ &= (Id - A)^{-1}[(A - Id) + Id]^{-1} - (Id - A)^{-1} \\ &= (Id - A)^{-1}A^{-1} - (Id - A)^{-1} \\ &= (Id - A)^{-1} [A^{-1} - Id] \\ &= (Id - A)^{-1} [A^{-1} - AA^{-1}] \\ &= (Id - A)^{-1}(Id - A)A^{-1} \\ &= A^{-1}. \end{aligned}$$

Probando así que la integral  $\int_0^1 [(A - Id)s + Id]^{-2} = A^{-1}$ . □

**Lema 1.** Supongamos que  $F, G : S_{++}^n \rightarrow S_n$  son funciones matriciales y definen

$$p(X) = \langle F(X), G(X) \rangle_{S_n} = \text{tr}(F(X)G(X)).$$

Si  $F$  y  $G$  son Frechét diferenciables (ver 3) entonces  $p$  es diferenciable y

$$\lim_{t \rightarrow 0} \frac{p(X + tV) - p(X)}{t} = \text{tr}(DF(X, V)G(X)) + \text{tr}(F(X)DG(X, V)),$$

para toda matriz simétrica  $V$ .

*Demostración.* En primer lugar note que

$$\begin{aligned} p(X + tV) &= \text{tr}\left(F(X + tV)G(X + tV)\right) \\ &= \text{tr}\left([F(X) + tDF(X, V) + o(t)][G(X) + tDG(X, V) + o(t)]\right) \\ &= \text{tr}\left(F(X)G(X)\right) + t\left[\text{tr}\left(DF(X, V)G(X)\right) + \text{tr}\left(F(X)DG(X, V)\right)\right] \\ &\quad + o(t) \\ &= p(X) + t\left[\text{tr}\left(DF(X, V)G(X)\right) + \text{tr}\left(F(X)DG(X, V)\right)\right] + o(t), \end{aligned}$$

con lo anterior se tiene entonces que

$$\frac{p(X + tV) - p(X)}{t} = \left[\text{tr}\left(DF(X, V)G(X)\right) + \text{tr}\left(F(X)DG(X, V)\right)\right] + \frac{o(t)}{t},$$

y así cuando  $t \rightarrow 0$  se obtiene el resultado. □

**Corollary 6.1.** Si  $F : S_{++}^n \rightarrow S_n$  es Frechét diferenciable y

$$\begin{aligned} p(X) &= \text{tr}\left(F(X)^2\right) \\ &= \|F(X)\|_F^2, \end{aligned}$$

entonces  $p$  es diferenciable y

$$\lim_{t \rightarrow 0} \frac{p(X + tV) - p(X)}{t} = 2\text{tr}(DF(X, V)F(X)).$$

**Teorema 7.** (Teorema 11.1 de <sup>10</sup>). Para  $X \in S_{++}^n$  se tiene que

$$\log(X) = \int_0^1 (X - Id)[(X - Id)s + Id]^{-1} ds.$$

*Demostración.* Como  $A \in S_{++}^n$ , se puede representar  $A = Q\text{diag}(\lambda_i)Q^T$  con  $Q$  ortogonal. Además el hecho de que  $\lambda_i > 0$  implica que  $(\lambda_i - 1)s + 1 > 0$  para  $s \in [0, 1]$  y así pueda calcularse

$$\begin{aligned} \log(A) &= Q\text{diag}(\log(\lambda_i))Q^T \\ &= Q\text{diag}\left(\int_0^1 \frac{\lambda_i - 1}{(\lambda_i - 1)s + 1} ds\right)Q^T, \quad \text{ver 3} \\ &= \int_0^1 Q\text{diag}\left(\frac{\lambda_i - 1}{(\lambda_i - 1)s + 1}\right)Q^T ds \\ &= \int_0^1 (A - Id)[(A - Id)s + Id]^{-1} ds, \end{aligned}$$

esto se tiene usando la definición de integral en una matriz que se menciona en 3. □

**Lema 2.** Si  $F(X) = X^{-1}$  entonces  $DF(X, V) = -X^{-1}VX^{-1}$ .

*Demostración.* Como  $XF(X) = Id$  se tiene que  $(X + tV)F(X + tV) = Id$  y así

$$XF(X + tV) + tVF(X + tV) = Id,$$

restando  $XF(X) = Id$  de lo anterior se tiene que

$$X[F(x + tV) - F(X)] + tVF(X + tV) = 0,$$

y así

$$\frac{F(x + tV) - F(X)}{t} = -X^{-1}VF(X + tV).$$

Note que cuando  $t \rightarrow 0$  como  $F(X) = X^{-1}$  se tiene que

$$\lim_{t \rightarrow 0} \frac{F(x + tV) - F(X)}{t} = -X^{-1}VX^{-1},$$

mostrando así que  $DF(X, V) = -X^{-1}VX^{-1}$ . □

**Teorema 8.** Si  $F(X) = \log(X)$  se tiene que

$$DF(X, E) = \int_0^1 [(X - Id)s + Id]^{-1} E [(X - Id)s + Id]^{-1} ds.$$

*Demostración.* Por el Teorema 7

$$F(X) = (X - Id) \int_0^1 [(X - Id)s + Id]^{-1} ds,$$

y también

$$\begin{aligned} F(X + tV) &= (X + tV - Id) \int_0^1 [(X + tV - Id)s + Id]^{-1} ds \\ &= (X - Id) \int_0^1 [(X + tV - Id)s + Id]^{-1} ds \\ &\quad + tV \int_0^1 [(X + tV - Id)s + Id]^{-1} ds, \end{aligned}$$



usando estos dos resultados se tiene entonces que

$$\begin{aligned} F(X + tV) - F(X) &= (X - Id) \int_0^1 [(X + tV - Id)s + Id]^{-1} - [(X - Id)s + Id]^{-1} ds \\ &\quad + tV \int_0^1 [(X + tV - Id)s + Id]^{-1} ds. \end{aligned}$$

Usando el anterior lema y como  $F(X + tV) = F(X) + tDF(X, V) + o(t)$

$$\begin{aligned} F(X + tV) - F(X) &= (X - Id) \int_0^1 \left[ -t[(X - Id)s + Id]^{-1}V[(X - Id)s + Id]^{-1}s \right. \\ &\quad \left. + o(t) \right] ds + tV \int_0^1 [(X + tV - Id)s + Id]^{-1} ds \\ &= (Id - X) \int_0^1 ts[(X - Id)s + Id]^{-1}V[(X - Id)s + Id]^{-1} ds \\ &\quad + (X - Id) \int_0^1 o(t) ds + tV \int_0^1 [(X + tV - Id)s + Id]^{-1} ds, \end{aligned}$$

note que cuando  $t \rightarrow 0$

$$\begin{aligned} \lim_{t \rightarrow 0} \frac{F(X + tV) - F(X)}{t} &= (Id - X) \int_0^1 s[(X - Id)s + Id]^{-1}V[(X - Id)s + Id]^{-1} ds \\ &\quad + V \int_0^1 [(X - Id)s + Id]^{-1} ds \\ &= \int_0^1 \left( s(Id - X)[(X - Id)s + Id]^{-1} + Id \right) V[(X - Id)s + Id]^{-1} ds \\ &= \int_0^1 \left( s(Id - X) + [(X - Id)s + Id] \right) \\ &\quad [(X - Id)s + Id]^{-1}V[(X - Id)s + Id]^{-1} ds \\ &= \int_0^1 [(X - Id)s + Id]^{-1}V[(X - Id)s + Id]^{-1} ds, \end{aligned}$$

mostrando así que  $DF(X, E) = \int_0^1 [(X - Id)s + Id]^{-1}E[(X - Id)s + Id]^{-1} ds$ .  $\square$

**Corollary 8.1.** Si  $F(X) = \log(X)$  entonces

$$\text{tr}(DF(X, E)) = \text{tr}(X^{-1}E).$$

*Demostración.* Por el anterior Teorema se tiene que

$$\begin{aligned} \text{tr}(DF(X, E)) &= \text{tr}\left(\int_0^1 [(X - Id)s + Id]^{-1} E [(X - Id)s + Id]^{-1} ds\right) \\ &= \int_0^1 \text{tr}\left([(X - Id)s + Id]^{-1} E [(X - Id)s + Id]^{-1}\right) ds \\ &= \int_0^1 \text{tr}\left([(X - Id)s + Id]^{-2} E\right) ds \\ &= \text{tr}\left(\left[\int_0^1 [(X - Id)s + Id]^{-2} ds\right] E\right), \quad \text{Por Teorema 6} \\ &= \text{tr}\left(X^{-1}E\right), \end{aligned}$$

mostrando así lo que se quería. □

**Corollary 8.2.** Si  $F(X) = \log(X)$  y  $W$  conmuta con  $[(X - Id)s + Id]^{-1}$  entonces

$$\text{tr}(DF(X, E)W) = \text{tr}(X^{-1}EW).$$

*Demostración.* Usando el Teorema 8

$$\begin{aligned} \text{tr}(DF(X, E)W) &= \text{tr}\left(\int_0^1 [(X - Id)s + Id]^{-1} E [(X - Id)s + Id]^{-1} ds W\right) \\ &= \int_0^1 \text{tr}\left([(X - Id)s + Id]^{-1} E W [(X - Id)s + Id]^{-1}\right) ds \\ &= \int_0^1 \text{tr}\left([(X - Id)s + Id]^{-2} E W\right) ds, \quad \text{Por Teorema 6} \\ &= \text{tr}\left(X^{-1}EW\right), \end{aligned}$$

mostrando lo que se quería. □

Usando el anterior Corolario y el Corolario 6.1 se tiene el siguiente Corolario.

**Corollary 8.3.** Si  $F(X) = \log(X)$  y  $p(X) = \text{tr}(F(X)^2) = \text{tr}(\log(X)^2)$ . Entonces  $p$  es diferenciable y

$$\lim_{t \rightarrow 0} \frac{p(X + tV) - p(X)}{t} = 2\text{tr}(X^{-1}V\log(X)).$$

*Demostración.* Por el Corolario 6.1 se tiene que

$$\begin{aligned} \lim_{t \rightarrow 0} \frac{p(X + tV) - p(X)}{t} &= 2\text{tr}(DF(X, V)F(X)) \\ &= 2\text{tr}(DF(X, V)\log(X)), \quad \text{por el Corolario 8.2} \\ &= 2\text{tr}(X^{-1}V\log(X)), \end{aligned}$$

mostrando lo que se pretendía. □

En 1 en el Teorema 9 se menciona la propiedad de que  $f(AB) = Af(BA)A^{-1}$  donde  $A \in \mathbb{C}^{m \times n}$ ,  $B \in \mathbb{C}^{n \times m}$  y la función  $f$  está definida en el espectro de  $AB$  y  $BA$ . Usando esto se puede ver que

$$\begin{aligned} \log(X_K^{-1}X) &= \log\left(X_k^{-\frac{1}{2}}X_k^{-\frac{1}{2}}X\right) \\ &= X_k^{-\frac{1}{2}}\log\left(X_k^{-\frac{1}{2}}XX_k^{-\frac{1}{2}}\right)X_k^{\frac{1}{2}}, \end{aligned} \tag{9}$$

y por tanto

$$\begin{aligned} \text{tr}\left(\log(X_K^{-1}X)^2\right) &= \text{tr}\left(X_k^{-\frac{1}{2}}\log\left(X_k^{-\frac{1}{2}}XX_k^{-\frac{1}{2}}\right)X_k^{\frac{1}{2}}X_k^{-\frac{1}{2}}\log\left(X_k^{-\frac{1}{2}}XX_k^{-\frac{1}{2}}\right)X_k^{\frac{1}{2}}\right) \\ &= \text{tr}\left(\log\left(X_k^{-\frac{1}{2}}XX_k^{-\frac{1}{2}}\right)^2\right). \end{aligned}$$

Dadas  $X_1, X_2, \dots, X_n \in S_{++}^n$  sea  $\sigma(X) = \sum_{i=1}^n \|\log(X_i^{-\frac{1}{2}}XX_i^{-\frac{1}{2}})\|^2$  la suma de las

distancias de un punto  $X$  a los puntos  $X_1, X_2, \dots, X_n$  elevadas al cuadrado. Note que

$\sigma(X) = \sum_{i=1}^n \text{tr} \left( \log \left( X_i^{-\frac{1}{2}} X X_i^{-\frac{1}{2}} \right)^2 \right)$ , entonces del Corolario (8.3) definiendo  $p_i(X)$  como  $p_i(X) = p(X_i^{-\frac{1}{2}} X X_i^{-\frac{1}{2}})$  se tiene que  $\sigma(X) = \sum_{i=1}^n p_i(X)$  y además

$$\begin{aligned}
 \lim_{t \rightarrow 0} \frac{p_i(X + tV) - p_i(X)}{t} &= \lim_{t \rightarrow 0} \frac{p \left( X_i^{-\frac{1}{2}} X X_i^{-\frac{1}{2}} + t X_i^{-\frac{1}{2}} V X_i^{-\frac{1}{2}} \right) - p \left( X_i^{-\frac{1}{2}} X X_i^{-\frac{1}{2}} \right)}{t} \\
 &= 2 \text{tr} \left( \left( X_i^{-\frac{1}{2}} X X_i^{-\frac{1}{2}} \right)^{-1} \left( X_i^{-\frac{1}{2}} V X_i^{-\frac{1}{2}} \right) \log \left( X_i^{-\frac{1}{2}} X X_i^{-\frac{1}{2}} \right) \right) \\
 &= 2 \text{tr} \left( X_i^{\frac{1}{2}} X^{-1} X_i^{\frac{1}{2}} \left( X_i^{-\frac{1}{2}} V X_i^{-\frac{1}{2}} \right) \log \left( X_i^{-\frac{1}{2}} X X_i^{-\frac{1}{2}} \right) \right) \\
 &= 2 \text{tr} \left( X^{-1} V X_i^{-\frac{1}{2}} \log \left( X_i^{-\frac{1}{2}} X X_i^{-\frac{1}{2}} \right) X_i^{\frac{1}{2}} \right), \quad \text{aplicando (9)} \\
 &= 2 \text{tr} \left( X^{-1} V X_i^{-\frac{1}{2}} X_i^{\frac{1}{2}} \log \left( X_i^{-1} X \right) \right) \\
 &= 2 \text{tr} \left( X^{-1} V \log \left( X_i^{-1} X \right) \right).
 \end{aligned}$$

Denotando  $\partial_V \sigma(X) = \lim_{t \rightarrow 0} \frac{\sigma(X+tV) - \sigma(X)}{t}$  se quiere que  $\partial_V \sigma(X) = 0$  para todo  $V \in S_n$ , utilizando el calculo anterior se tiene entonces que

$$\begin{aligned}
 \partial_V \sigma(X) &= \lim_{t \rightarrow 0} \frac{\sigma(X + tV) - \sigma(X)}{t} \\
 &= \lim_{t \rightarrow 0} \sum_{i=1}^m \frac{p_i(X + tV) - p_i(X)}{t} \\
 &= \sum_{i=1}^m \lim_{t \rightarrow 0} \frac{p_i(X + tV) - p_i(X)}{t} \\
 &= \sum_{i=1}^m 2 \text{tr} \left( X^{-1} V \log \left( X_i^{-1} X \right) \right) \\
 &= \sum_{i=1}^m 2 \text{tr} \left( \log \left( X_i^{-1} X \right) X^{-1} V \right),
 \end{aligned} \tag{10}$$

y por tanto se quiere que  $\sum_{i=1}^m 2\text{tr}(\log(X_i^{-1}X)X^{-1}V) = 0$  para todo  $V \in S_n$ . Como

$$\log(X_i^{-1}X)X^{-1} = X_i^{-\frac{1}{2}}\log(X_i^{-\frac{1}{2}}XX_i^{-\frac{1}{2}})X_i^{-\frac{1}{2}} = \log_{X_i}(X)$$

pertenece a  $S_n$  y  $\text{tr}(ZW)$  es un producto interno en  $S_n$  se concluye que la condición de optimalidad es

$$2\left(\sum_{i=1}^n \log(X_i^{-1}X)\right)X^{-1} = 0,$$

o que es lo mismo

$$\sum_{i=1}^n \log(X_i^{-1}X) = 0. \quad (11)$$

En lo anterior se mostraron entonces resultados previos para al final plantear en (11) de manera matricial el problema de optimización con el que se puede obtener la matriz  $X$  que está mas cercana a todas las matrices  $\{X_1, X_2, \dots, X_m\}$  con la distancia riemanniana definida en (7). Note que si esta media se viera en  $\mathbb{R}^+$  coincidiría con la media geométrica pues sería el  $x \in \mathbb{R}$  que cumple

$$\sum_{i=1}^n \log(x_i^{-1}x) = 0,$$

que efectivamente es la media geométrica de  $\{x_1, x_2, \dots, x_n\}$ .

## 4. ALGORITMOS ESTUDIADOS

Como se mencionó el inicio de la Sección 3, el problema de calcular una media ha sido estudiado utilizando la media de Fréchet que en  $S_{++}^n$  coincide con la media de Karcher. Dado un conjunto  $\{C_i\}_{i=1}^k \subset S_{++}^n$ , Pennec en Pennec, Fillard y Ayache, “A Riemannian Framework for Tensor Computing” plantea un algoritmo de gradiente descendiente de Newton para resolver el problema de minimización

$$\arg \min_{C \in S_{++}^n} \sum_{i=1}^k d(C, C_i)^2.$$

Este algoritmo que propone da cómo estimación de la media en la iteración  $t + 1$ :

$$\mu_{t+1} = \exp_{\mu_t} \left( \frac{1}{k} \sum_{i=1}^k \log_{\mu_t}(C_i) \right). \quad (12)$$

En base a este se han planteado otros algoritmos como el de Fletcher (ver Thomas Fletcher y Joshi, “Riemannian Geometry for the statistical analysis of diffusion tensor Data”) que presenta algunas variaciones. En esta sección presentaremos diferentes algoritmos existentes y se plantearán dos algoritmos, uno usando autovalores generalizados y otro una descomposición de Cholesky.

### 4.1. ALGORITMO PRINCIPAL DE GRADIENTE DESCENDIENTE: SOLUCIÓN DE PENNEC

Como se mencionó anteriormente, el algoritmo que se muestra en 12 fue planteado por Pennec en <sup>7</sup>. Haciendo algunos cálculos, el algoritmo de Pennec se puede ver

de la siguiente forma

$$\begin{aligned}
\mu_{t+1} &= \exp_{\mu_t} \left( \frac{1}{k} \sum_{i=1}^k \log_{\mu_t} (C_i) \right) \\
&= \exp_{\mu_t} \left( \frac{1}{k} \sum_{i=1}^k \mu_t^{\frac{1}{2}} \log(\mu_t^{-\frac{1}{2}} C_i \mu_t^{-\frac{1}{2}}) \mu_t^{\frac{1}{2}} \right) \\
&= \mu_t^{\frac{1}{2}} \exp \left( \mu_t^{-\frac{1}{2}} \left[ \frac{1}{k} \sum_{i=1}^k \mu_t^{\frac{1}{2}} \log(\mu_t^{-\frac{1}{2}} C_i \mu_t^{-\frac{1}{2}}) \mu_t^{\frac{1}{2}} \right] \mu_t^{-\frac{1}{2}} \right) \mu_t^{\frac{1}{2}} \\
&= \mu_t^{\frac{1}{2}} \exp \left( \frac{1}{k} \sum_{i=1}^k \log(\mu_t^{-\frac{1}{2}} C_i \mu_t^{-\frac{1}{2}}) \right) \mu_t^{\frac{1}{2}},
\end{aligned} \tag{13}$$

así es mas sencillo calcular la aproximación  $\mu_{t+1}$ . Recuerde que  $\log_{\mu_t}$  está definido en  $S_{++}^n$ .

**4.1.1. EJEMPLO MATRICES DE 2X2** Desde un punto de vista práctico, podemos considerar matrices SPD de baja dimensionalidad que nos permitirán ilustrar el algoritmo para el cálculo de la media. Considerando el conjunto

$$S_2^{++} = \left\{ \begin{bmatrix} a & b \\ b & c \end{bmatrix} : ac - b^2 \geq 0, a \geq 0, c \geq 0 \right\},$$

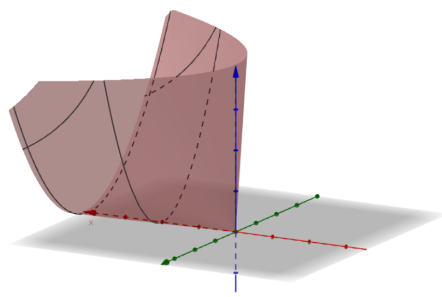
este puede ser visto en  $\mathbb{R}^3$  como

$$S_2^{++} = \{(x, y, z) \in \mathbb{R}^3 : xz - y^2 \geq 0, x \geq 0, z \geq 0\}.$$

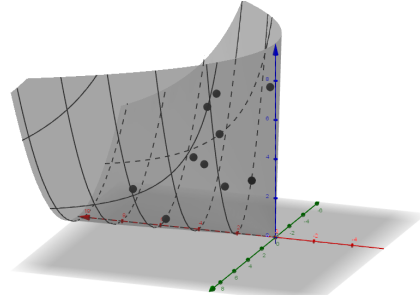
Usando esta representación se puede ilustrar el conjunto  $S_2^{++}$  como se ve en la Figura 4(a).

A continuación se mostrará como funciona el algoritmo propuesto por Pennec en <sup>7</sup> para el cálculo de la media del conjunto de matrices  $C = \{C_1, C_2, \dots, C_{10}\}$ , donde

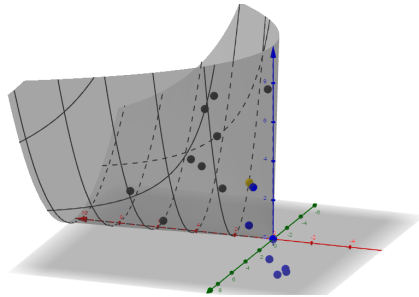
**Figura 3.** Ilustración algoritmo de la media para matrices  $2 \times 2$



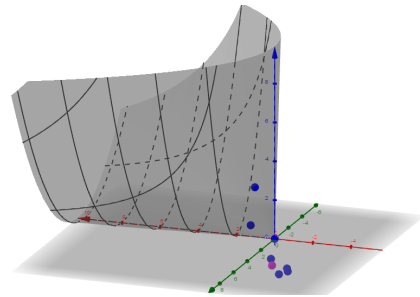
(a) Cono matrices SPD  $2 \times 2$ .



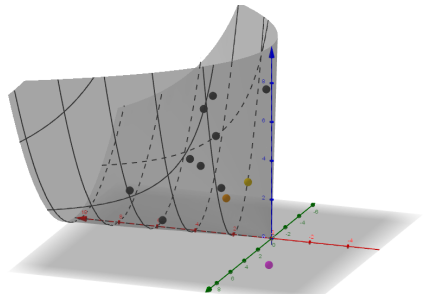
(b) Puntos del conjunto  $C$  (en negro).



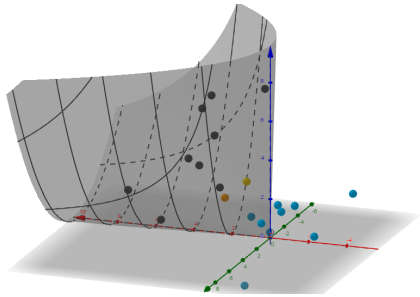
(c) Punto inicial  $M_0 = C_1$  (amarillo), conjunto  $C$  (negro) y los puntos  $\log_{C_1}(C_i)$  (azul).



(d) Puntos  $\log_{C_1}(C_i)$  (azul) y  $M_{A_1}$  (morado).

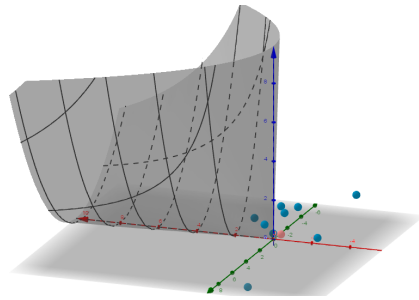


(e) Conjunto inicial  $C$  (negro),  $M_0$  (amarillo),  $M_{A_1}$  (morado) y los puntos  $\log_{M_1}(C_i)$  (naranja).

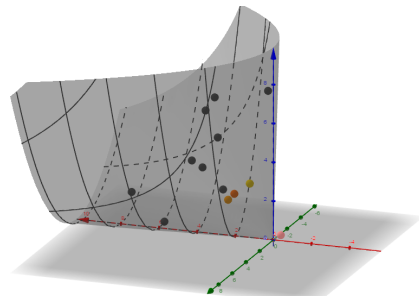


(f) Conjunto inicial  $C$  (negro) incluyen- do  $C_1$  (amarillo),  $M_1$  (naranja) y los puntos  $\log_{M_1}(C_i)$  (azul claro).

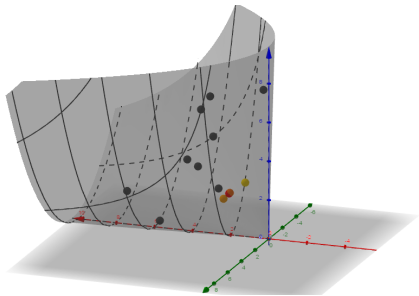




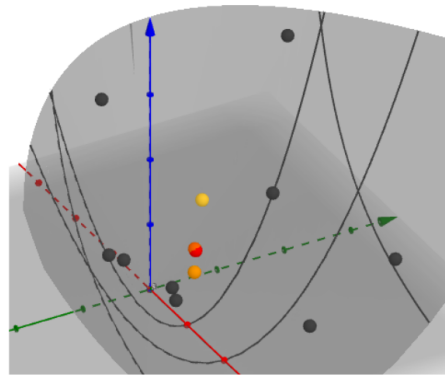
(g) Puntos  $\log_{M_1}(C_i)$  (azul claro) y  $M_{A_2}$  (rosado) .



(h) Conjunto inicial  $C$  (negro) incluyendo  $C_1$  (amarillo),  $M_1$  (naranja),  $M_{A_2}$  (rosado) y  $M_2$  (naranja oscuro) .



(i) Conjunto inicial  $C$  (negro) incluyendo  $C_1$  (amarillo),  $M_1$  (naranja),  $M_{A_2}$  (rosado) y  $M_2$  (naranja oscuro) y la media  $M$  (rojo) .



(j) Otra vista del conjunto inicial  $C$  (negro) incluyendo  $C_1$  (amarillo),  $M_1$  (naranja),  $M_{A_2}$  (rosado) y  $M_2$  (naranja oscuro) y la media  $M$  (rojo) .

$$C_1 = \begin{pmatrix} 1 & 1 \\ 1 & 3 \end{pmatrix}, C_2 = \begin{pmatrix} 2 & 3 \\ 3 & 8 \end{pmatrix}, C_3 = \begin{pmatrix} 3 & 2 \\ 2 & 4 \end{pmatrix}, C_4 = \begin{pmatrix} 5 & 2 \\ 2 & 1 \end{pmatrix}, C_5 = \begin{pmatrix} 1 & -2 \\ -2 & 7 \end{pmatrix},$$

$$C_6 = \begin{pmatrix} 4 & -3 \\ -3 & 4 \end{pmatrix}, C_7 = \begin{pmatrix} 5 & -2 \\ -2 & 3 \end{pmatrix}, C_8 = \begin{pmatrix} 3 & -1 \\ -1 & 2 \end{pmatrix}, C_9 = \begin{pmatrix} 5 & -4 \\ -4 & 5 \end{pmatrix}, \text{ y}$$

$$C_{10} = \begin{pmatrix} 6 & 4 \\ 4 & 3 \end{pmatrix} \text{ serían los puntos ilustrados en la Figura 4(b).}$$

Conocido el conjunto de matrices, se elige una matriz inicial  $M_0$  que puede ser por ejemplo  $M_0 = C_1$  y la cual se puede visualizar de amarillo en la Figura 4(c). Posteriormente se calcula  $\log_{C_1}(C_i)$  para cada  $i \in \{1, 2, \dots, 10\}$ , ver Figura 4(c). Ahora, con los puntos  $\log_{C_1}(C_i)$  se calcula una media aritmética  $M_{A_1} = \frac{1}{10} \sum_{i=1}^{10} \log_{C_1}(C_i)$  ilustrada en la Figura 4(d) que es

$$M_{A_1} = \begin{pmatrix} 0,4687 & -0,9092 \\ -0,9092 & -1,6788 \end{pmatrix}.$$

Una vez calculada esta media aritmética, se lleva al cono nuevamente con respecto a  $C_1$  mediante  $\exp_{C_1}(M_{A_1})$  a lo que se denotará como  $M_1$  (Ver Figura 4(e)) y que equivale a

$$M_1 = \begin{pmatrix} 2,3745 & 0,00756 \\ 0,00756 & 1,8244 \end{pmatrix}.$$

Todo lo anterior corresponde a la primera iteración del algoritmo para el cálculo de la media del conjunto  $C$ . Para continuar, se repite el proceso pero ahora la matriz inicial es  $M_1$ , se debe entonces primero calcular los puntos  $\log_{M_1}(C_i)$  (Ver Figura 4(f)), luego hallar una media aritmética de estos  $M_{A_2} = \log_{M_1}(C_i)$  (Ver Figura 4(g)) y finalmente calcular  $\exp_{M_1}(M_{A_2})$  (Ver Figura 4(h)) para obtener un nuevo punto en  $S_2^{++}$  que se denotará como  $M_2$

$$M_2 = \begin{pmatrix} 1,9595 & 0,2481 \\ 0,2481 & 2,2423 \end{pmatrix}.$$

Estos puntos que se están agregando al cono ( $M_1$  y  $M_2$ ) son iteraciones que se van aproximando a un punto de manera que al hacer más iteraciones se obtiene una mejor aproximación de la media del conjunto  $C$ . Haciendo mas iteraciones en un programa computacional se calculó como media del conjunto  $C$  a la siguiente matriz que se denotará como  $M$  y que está en las Figuras 4(i) y 4(j)

$$M = \begin{pmatrix} 1,9919 & 0,2571 \\ 0,2571 & 2,2081 \end{pmatrix}.$$

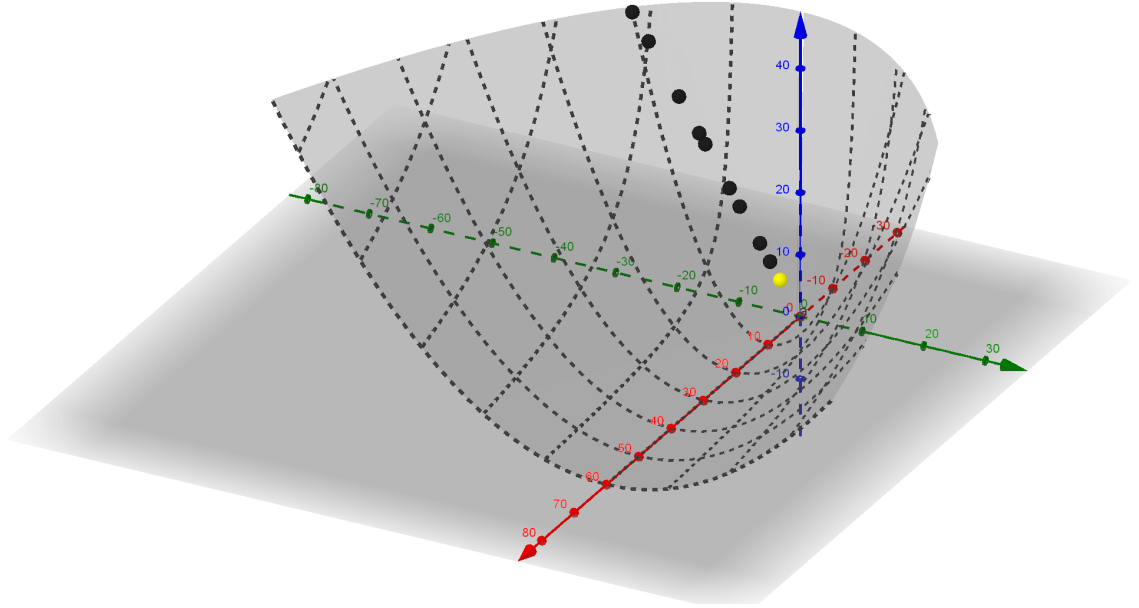
Es de notar que en este caso las primeras dos iteraciones que se calcularon estaban bastante próximas a la media que se obtiene al final de algoritmo.

**4.1.2. CASO PARTICULAR: MEDIA GEOMÉTRICA** Entendido el anterior ejemplo resulta mucho mas fácil ahora manipular el algoritmo para el calculo de una media. Veamos ahora un caso en el que este algoritmo resulta ser una media geométrica. Sea  $A \in S_+^n$  y  $A_1, A_2, \dots, A_n$  matrices producto escalares de  $A$ , es decir,  $A_1 = \lambda_1 A, A_2 = \lambda_2 A, \dots, A_n = \lambda_n A$  con  $\lambda_1, \lambda_2, \dots, \lambda_n \in \mathbb{R}^+$  (Ver Figura 4).

Aplicando el algoritmo para el cálculo de la media tomando como matriz inicial a  $A$ , primero se debe hallar una media aritmética  $M_{A_1}$  de los  $\log_A(A_i)$ , lo que sería

$$\begin{aligned} M_{A_1} &= \frac{1}{n} \sum_{i=1}^n \log_A(A_i) \\ &= \frac{1}{n} \sum_{i=1}^n A^{\frac{1}{2}} \log(A^{-\frac{1}{2}} A_i A^{-\frac{1}{2}}) A^{\frac{1}{2}} \\ &= \frac{1}{n} \sum_{i=1}^n A^{\frac{1}{2}} \log(A^{-\frac{1}{2}} \lambda_i A A^{-\frac{1}{2}}) A^{\frac{1}{2}} \end{aligned}$$

**Figura 4.** Matriz  $A$  de color amarillo y las demás múltiplos escalares de ella  $A_1 = \lambda_1 A, A_2 = \lambda_2 A, \dots, A_n = \lambda_n A$  en color negro.



$$\begin{aligned}
 &= \frac{1}{n} \sum_{i=1}^n A^{\frac{1}{2}} \log(\text{diag}(\lambda_i)) A^{\frac{1}{2}} \\
 &= \frac{1}{n} \sum_{i=1}^n A^{\frac{1}{2}} \text{diag}(\log(\lambda_i)) A^{\frac{1}{2}} \\
 &= A^{\frac{1}{2}} \left[ \frac{1}{n} \sum_{i=1}^n \text{diag}(\log(\lambda_i)) \right] A^{\frac{1}{2}} \\
 &= A^{\frac{1}{2}} \left[ \text{diag} \left( \log \left( \left[ \prod_{i=1}^n \lambda_i \right]^{\frac{1}{n}} \right) \right) \right] A^{\frac{1}{2}}
 \end{aligned}$$

Se debe hallar ahora  $\exp_A(M_{A_1})$  que sería

$$\begin{aligned}
 \exp_A(M_{A_1}) &= A^{\frac{1}{2}} \exp \left( A^{-\frac{1}{2}} A^{\frac{1}{2}} \left[ \text{diag} \left( \log \left( \left[ \prod_{i=1}^n \lambda_i \right]^{\frac{1}{n}} \right) \right) \right] A^{\frac{1}{2}} A^{-\frac{1}{2}} \right) A^{\frac{1}{2}} \\
 &= A^{\frac{1}{2}} \text{diag} \left( \left[ \prod_{i=1}^n \lambda_i \right]^{\frac{1}{n}} \right) A^{\frac{1}{2}}
 \end{aligned}$$

Note entonces que si se toma  $A = Id$  se obtiene una matriz diagonal donde todas sus entradas son la media geométrica de los factores  $\lambda_1, \lambda_2, \dots, \lambda_n$ .

## 4.2. CONTROL SOBRE EL GRADIENTE: ALGORITMO DE FLETCHER

Thomas Fletcher en <sup>8</sup> se basa en la media de Karcher <sup>23</sup> y el trabajo de Pennec <sup>7</sup> para plantear su algoritmo. Sabemos que si se toma una variedad  $\mathcal{M}$  con su distancia geodésica  $d$ , en un conjunto  $C = \{x_i\}_{i=1}^k$  la media es

$$\mu = \arg \min_{x \in \mathcal{M}} \rho_C(x),$$

donde  $\rho_C(x) = \frac{1}{2k} \sum_{i=1}^k d(x, x_i)^2$ . Flétcher usa que el gradiente de la función objetivo  $\rho_C(x)$  es

$$\nabla \rho_C(x) = -\frac{1}{k} \sum_{i=1}^k \log_x(x_i),$$

y con este propone controlar el paso en su algoritmo de gradiente descendiente con el fin de ser mas robusto en el caso que los elementos del conjunto  $C = \{x_i\}_{i=1}^k$  estén bastante dispersos.

---

**Algorithm 1** Algoritmo de Fletcher (ver <sup>24</sup>), compara la norma de los gradientes con la iteración anterior y controla con el paso con  $\tau$ .

---

$$\mu_0 = C_1$$

$$\tau = 1$$

**repeat**

$$X_i = \frac{1}{N} \sum_{k=1}^N \log(\mu_i^{-\frac{1}{2}} C_k \mu_i^{-\frac{1}{2}})$$

$$\mu_{i+1} = \mu_i^{\frac{1}{2}} \exp(\tau X_i) \mu_i^{\frac{1}{2}}$$

**if**  $\|X_i\| > \|X_{i-1}\|$  **then**

$$\tau = \frac{\tau}{2}, X_i = X_{i-1}$$

**end if**

**until**  $\|X_i\| > \epsilon$

---

### 4.3. OTRA MÉTRICA PARA EL CÁLCULO DE LA MEDIA: LOG-EUCLIDEAN

Pennec junto a otros autores plantean en <sup>18</sup> una nueva familia de medias llamadas “*Log-Euclidean*” que surgen al dar nuevas estructuras a  $S_{++}^n$ . En <sup>18</sup> se define el producto logarítmico de dos matrices  $A, B \in S_{++}^n$  como

$$A * B = \exp(\log(A) - \log(B)),$$

de forma que  $(S_{++}^n, *)$  es un grupo al cual se le puede construir una estructura de grupo de Lie (ver <sup>18</sup>) y se le puede asignar una métrica bi-invariante que se conoce como “*métrica Log-Euclidean*” ya que resulta ser una métrica Euclidiana en el plano tangente  $S^n$ , además el grupo de Lie  $S_{++}^n$  resulta ser isomorfo y difeomorfo a su algebra de Lie, plano tangente y grupo aditivo  $S^n$ . Con la norma asociada a la métrica se puede calcular la distancia entre dos matrices  $A, B \in S_{++}^n$  como

$$d_{le}(A, B) = \|\log(A) - \log(B)\|_{Id}. \quad (14)$$

Además extienden el isomorfismo de grupos a un isomorfismo como espacios vectoriales definiendo la multiplicación logarítmica escalar de un escalar  $\lambda \in \mathbb{R}$  y una matriz  $A \in S_{++}^n$

$$\lambda \odot A = \exp(\lambda \log(A)).$$

Finalmente lo que se logra es definir una estructura Euclidiana en  $S_{++}^n$  a partir de su algebra de Lie  $S^n$ , esto con el fin de hacer cálculos aritméticos entre matrices SPD de forma mas rápida y sencilla, y a su vez poder calcular la media sin un alto costo computacional como con la métrica afín invariante propuesta previamente por Pennec en <sup>7</sup>. El teorema 3.13 de <sup>18</sup> menciona cómo la media de Fréchet de un

conjunto  $C = \{x_i\}_{i=1}^k$  es

$$\mu = \frac{1}{k} \sum_{i=1}^k \log(x_i), \quad (15)$$

independiente de la métrica *Log-Euclidean* elegida. Con la distancia definida en (14) esto se tiene, pues  $\mu$  es el  $x$  que minimice lo siguiente

$$\begin{aligned} \mu &= \arg \min \frac{1}{k} \sum_{i=1}^k d_{le}(x, x_i)^2 \\ &= \arg \min \frac{1}{k} \sum_{i=1}^k \|\log(x) - \log(x_i)\|_{Id}^2, \end{aligned}$$

de donde se obtiene la condición de optimalidad

$$\frac{1}{k} \sum_{i=1}^k (\log(x) - \log(x_i)) = 0,$$

y obteniendo así la media presentada en (15). A continuación el algoritmo de su implementación

---

**Algorithm 2** Media Log-Euclidean (ver <sup>25</sup>) que surge al dar una estructura Euclidiana a  $S_{++}^n$  a partir de su álgebra de Lie  $S^n$ . Sin importar la métrica Log-Euclidean la media se calcula de la misma manera.

---

$$X = \frac{1}{N} \sum_{k=1}^N \log(C_k)$$

$$\mu = \exp(X)$$


---

#### 4.4. ALGORITMO DE UNA LIBRERÍA DE GEOMETRÍA RIEMANNIANA APLICADA: PYRIEMANN

Alexandre Barachant en su repositorio <sup>26</sup> tiene una librería de python llamada “*Pyriemann*” con funciones para la manipulación y clasificación de matrices de covarianza

---

<sup>26</sup> A. Barachant y J-R. King. *pyRiemann v0.2.2*. Jun. de 2015. DOI: 10.5281/zenodo.18982.

con geometría riemanniana. Su trabajo se puede ver en sus publicaciones <sup>27,28</sup>, donde junto a la teoría presenta también algunas aplicaciones.

---

**Algorithm 3** Algoritmo planteado por Alexandre Barachant en su librería “Pyriemann” (Ver<sup>29</sup>).

---

$\mu_0$  {Media aritmética de  $\{C_i\}_{i=1}^k$ }

$\tau = 1$

$\delta \gg 0$

**repeat**

$X_i = \frac{1}{N} \sum_{k=1}^N \exp(\mu_i^{-\frac{1}{2}} C_k \mu_i^{-\frac{1}{2}})$

$h = \tau \|X_i\|$

$\mu_{i+1} = \mu_i^{\frac{1}{2}} \exp(\tau X_i) \mu_i^{\frac{1}{2}}$

**if**  $h < \delta$  **then**

$\tau = 0,95\tau$

$\delta = h$

**else if** **then**

$\tau = 0,5\tau$

**end if**

**until**  $\|X_i\| > \epsilon$  **and**  $\tau > \epsilon$

---



---

<sup>27</sup> Alexandre Barachant y col. “Multiclass Brain-Computer Interface Classification by Riemannian Geometry”. En: *IEEE Transactions on Biomedical Engineering* 59.4 (mar. de 2012), págs. 920-928. DOI: 10.1109/TBME.2011.2172210.

<sup>28</sup> Alexandre Barachant y col. “Classification of Covariance Matrices Using a Riemannian-Based Kernel for BCI Applications”. En: *Neurocomput.* 112 (jul. de 2013), 172–178. DOI: 10.1016/j.neucom.2012.12.039.



#### 4.5. MODIFICACIONES DEL ALGORITMO MEDIANTE AUTOVALORES GENERALIZADOS

Como se puede ver en 2  $f_{\Sigma}(V) = \Sigma Q f(D) Q^T \Sigma$  donde  $Q$  y  $D$  son los autovectores y autovalores del problema de autovalor generalizado  $\Sigma^{-1}V$ , respectivamente. A continuación se usará esto para calcular (12). Tenemos

$$\begin{aligned}\mu_{t+1} &= \exp_{\mu_t} \left( \frac{1}{k} \sum_{i=1}^k \log_{\mu_t}(C_i) \right) \\ &= \exp_{\mu_t} \left( \frac{1}{k} \sum_{i=1}^k \mu_t Q_i f(D_i) Q_i^T \mu_t \right) \\ &= \exp_{\mu_t} \left( \mu_t \left[ \frac{1}{k} \sum_{i=1}^k Q_i f(D_i) Q_i^T \right] \mu_t \right),\end{aligned}\tag{16}$$

para continuar se denotará  $J = \frac{1}{k} \sum_{i=1}^k Q_i f(D_i) Q_i^T$  y por tanto para el calculo de la exponencial  $\exp_{\mu_t} \left( \mu_t J \mu_t \right)$  se debe hacer el problema de autovalor  $J \mu_t$ , así

$$\begin{aligned}\mu_{t+1} &= \exp_{\mu_t} \left( \mu_t J \mu_t \right) \\ &= \mu_t Q \exp(D) Q^T \mu_t,\end{aligned}$$

donde  $J \mu_t = Q \mu_t Q^{-1}$ . Si ahora a los anteriores cálculos se les aplica el Corolario 9.1 donde se afirma que  $f_{\Sigma}(V) = \Sigma f(\Sigma^{-1}V)$ , se puede calcular (12) de esta otra forma.

Tenemos

$$\begin{aligned}
\mu_{t+1} &= \exp_{\mu_t} \left( \frac{1}{k} \sum_{i=1}^k \log_{\mu_t}(C_i) \right) \\
&= \exp_{\mu_t} \left( \mu_t \frac{1}{k} \sum_{i=1}^k \log(\mu_t^{-1} C_i) \right) \\
&= \mu_t \exp \left( \mu_t^{-1} \mu_t \frac{1}{k} \sum_{i=1}^k \log(\mu_t^{-1} C_i) \right) \\
&= \mu_t \exp \left( \frac{1}{k} \sum_{i=1}^k \log(\mu_t^{-1} C_i) \right) \\
&= \mu_t \exp \left( \left[ \frac{1}{k} \sum_{i=1}^k Q_i \log(D_i) Q_i^T \right] \mu_t \right),
\end{aligned} \tag{17}$$

donde se resuelve el problema de autovalor generalizado  $\mu_t^{-1} C_i = Q_i D_i Q_i^T \mu_t$  para calcular  $\log(\mu_t^{-1} C_i)$ . Note que en la implementación de Pennec en (13) se requiere calcular  $k$  veces  $\log_{\mu_t}(C_i) = \mu_t^{\frac{1}{2}} \log(\mu_t^{-\frac{1}{2}} C_i \mu_t^{-\frac{1}{2}}) \mu_t^{\frac{1}{2}}$ , donde se requiere resolver dos problemas de autovalor clásico, el de  $\mu_t$  para calcular  $\mu_t^{\frac{1}{2}}$  y  $\mu_t^{-\frac{1}{2}}$ , y el de  $\mu_t^{-\frac{1}{2}} C_i \mu_t^{-\frac{1}{2}}$  para calcular  $\log(\mu_t^{-\frac{1}{2}} C_i \mu_t^{-\frac{1}{2}})$ . Posteriormente hay que calcular  $\mu_t^{\frac{1}{2}} \exp\left(\frac{1}{k} \sum_{i=1}^k \log(\mu_t^{-\frac{1}{2}} C_i \mu_t^{-\frac{1}{2}})\right) \mu_t^{\frac{1}{2}}$ , entonces en total se requiere solucionar  $2k+1$  problemas de autovalor clásico por iteración. Con las modificaciones que se proponen en (16) requieren  $k$  problemas de autovalor clásico para calcular  $\log_{\mu_t}(C_i) = \mu_t Q_i f(D_i) Q_i^T \mu_t$  y un problema de autovalor clásico en  $J\mu_t = Q\mu_t Q^{-1}$ . En las modificaciones que se presentan en (17) también se requieren  $k$  problemas de autovalor generalizado y uno de autovalor clásico pero se hacen menos productos en comparación con (16). A continuación se presenta el algoritmo implementado en las experimentaciones.

---

**Algorithm 4** Algoritmo implementado en las experimentaciones y planteado en (16), se reducen los  $2k + 1$  problemas de autovalor clásico presentados en el algoritmo de Pennec a  $k$  problemas de autovalor generalizado y uno clásico.

---

$\mu_0$  {Media aritmética de  $\{C_i\}_{i=1}^k$ }

**repeat**

$Q_i, D_i$  {Autovectores y autovalores de  $\mu_i^{-1}C_i$ }

$$J = \frac{1}{k} \sum_{i=1}^k Q_i f(D_i) Q_i^T$$

$$X_i = \mu_i J \mu_i$$

$Q, D$  {Autovectores y autovalores de  $J \mu_t$ }

$$\mu_{i+1} = \mu_i Q \exp(D) Q^T \mu_i$$

**until**  $\|X_i\|_F > \epsilon$  and  $k < \text{maxiter}$

---

#### 4.6. MODIFICACIONES DEL ALGORITMO MEDIANTE EL USO DE LA DESCOMPOSICIÓN DE CHOLSKY

Se usará ahora una descomposición de Cholesky aplicada al calculo con los autovalores generalizados. Se tenía que  $\exp_Y(X) = YQ\exp(D)Q^TY$  donde  $Q$  resuelve  $Y^{-1}X$ , es decir  $XQ = DYQ$ . Usando la descomposición de Cholesky se tiene que  $Y = R^TR$ , donde  $R$  es una matriz triangular superior. Entonces se tendría que  $XQ = DR^TRQ$  o lo que es lo mismo  $XR^{-1}RQ = DR^TRQ$ , tomando a  $\psi = RQ$  resulta  $R^{-T}XR^{-1}\psi = D\psi$ . Es decir  $\psi$  resuelve el problema de autovalor clásico de la matriz  $V = R^{-T}XR^{-1}$  con los autovalores  $D$ , los cuales también son autovalores del problema inicial  $Y^{-1}X$ . Note que  $Q = R^{-1}\psi$  y por tanto  $Q^T = \psi^T R^{-T}$ , entonces

$$\begin{aligned} \exp_Y(X) &= YQ\exp(D)Q^TY \\ &= R^T R R^{-1} \psi \exp(D) \psi^T R^{-T} R^T R \\ &= R^T \psi \exp(D) \psi^T R, \end{aligned}$$

Se mostrará ahora como se puede aplicar lo anterior para calcular (12). Vea que

$$\begin{aligned}\mu_{t+1} &= \exp_{\mu_t} \left( \frac{1}{k} \sum_{i=1}^k \log_{\mu_t}(C_i) \right) \\ &= \exp_{\mu_t} \left( \frac{1}{k} \sum_{i=1}^k R^T \psi_i \log(D_i) \psi_i^T R \right),\end{aligned}\tag{18}$$

donde  $\psi_i$  resuelve el problema de autovalor clásico de la matriz  $V_i = R^{-T} C_i R^{-1}$  con los autovalores  $D_i$ . Denotando  $J = \frac{1}{k} \sum_{i=1}^k \psi_i \log(D) \psi_i^T$  se puede continuar con (18) de forma que

$$\begin{aligned}\mu_{t+1} &= \exp_{\mu_t}(R^T J R) \\ &= R^T \psi' \exp(D') \psi'^T R,\end{aligned}\tag{19}$$

donde  $\psi'$  resuelve el problema de autovalor clásico de la matriz  $V' = R^{-T}(R^T J R)R^{-1} = J$  con los autovalores  $D'$ . Note que con esta factorización de Choleskyse reducen los  $2k + 1$  problemas de autovalor clásico presentados en el algoritmo de Pennec a una descomposición de Cholesky, calculo de la inversa de una matriz y  $k$  problemas de autovalor clásico.

---

**Algorithm 5** Algoritmo de Cholesky implementado en las experimentaciones y planteado en (18) y (19), se reducen los  $2k + 1$  problemas de autovalor clásico presentados en el algoritmo de Pennec a una descomposición de Cholesky, calculo de la inversa de una matriz y  $k$  problemas de autovalor clásico.

---

$\mu_0$  {Media aritmética de  $\{C_i\}_{i=1}^k$ }

**repeat**

$X_i$  {Matriz de ceros del mismo tamaño de  $\mu_0$ }

$R$  {Matriz superior de  $\mu_i$ }

$R^T$  {Transpuesta de  $R$ }

$R^{-1}$  {Solución de  $XR = Id$ }

$R^{-T}$  {Transpuesta de  $R^{-1}$ }

**for**  $X \in \{C_i\}_{i=1}^k$  **do**

$X = R^{-T} X R^{-1}$

$Q_i, D_i$  {Autovectores y autovalores de la matriz  $X$ }

$X_i = X_i + \frac{1}{k} Q_i \log(D_i) Q_i^T$

**end for**

$J = X_i$

$X_i = R^T X_i R$

$Q, D$  {Autovectores y autovalores de la matriz  $J$ }

$\mu_{i+1} = R^T Q \exp(D) Q^T R$

**until**  $\|X_i\|_F > \epsilon$

---

## 5. RECONOCIMIENTO DE ACCIONES USANDO MEDIAS RIEMANNIANAS

Una aplicación potencial de la media en un espacio de Riemann es la descripción compacta de características de bajo nivel que describen un objeto en una imagen o una acción específica en un video. En este sentido, la media puede resumir la relación lineal entre características, y esta representación constituye el descriptor de bajo nivel, que relaciona un concepto de alto nivel. Por ejemplo, las medidas obtenidas a nivel del pixel, pueden resumirse en una covarianza media, y para videos, puede constituir el descriptor relacionado con caminar, correr, entre otras acciones. En este trabajo, utilizaremos medidas de velocidad local (flujo óptico) para describir relaciones de bajo nivel, las cuales servirán como descriptor en un problema de reconocimiento de acciones.

### 5.1. APLICACIÓN DEL FLUJO ÓPTICO A ACCIONES

En secuencias de video, el flujo óptico describe el movimiento local y aparente de una escena desde un punto relativo, en este caso una cámara. Como resultado, entre un par de cuadros consecutivos, se estima el desplazamiento de cada píxel, constituyendo un campo vectorial de desplazamiento. Existen diversos métodos computacionales para estimar el flujo óptico, siendo el mas conocido, el algoritmo de Lukas Kannade. Este algoritmo establece las condiciones y premisas comunes en la mayoría de enfoques de flujo óptico.

Supongamos que se tiene una secuencia de video que se denotará por  $D$  donde todas las imágenes que lo componen tienen la misma dimensión,  $W$  de ancho por  $H$  de alto. En una imagen cualquiera, el valor  $I(x, y)$  representa la luminosidad del pixel  $(x, y)$ , entonces realmente un vídeo se puede entender como una sucesión de matrices  $I(x, y, t)$ . Para empezar a introducir la definición de flujo óptico, se

asume inicialmente que un pixel tiene un desplazamiento corto en un instante de tiempo  $\Delta_t$ . En este desplazamiento, se asume que la luminosidad del pixel no cambia:  $I(x, y, t) = I(x + \Delta_x, y + \Delta_y, t + \Delta_t)$ . Entonces, esta función que representa el desplazamiento en la imagen, se puede aproximar como

$$\frac{\partial I \Delta x}{\partial x \Delta t} + \frac{\partial I \Delta y}{\partial y \Delta t} + \frac{\partial I}{\partial t} = 0.$$

Ya que la variación total es nula o sea si  $I(x(t), y(t), t) = 0$  para todo  $t$  entonces derivando se tiene que  $\frac{\partial I x'(t)}{\partial x} + \frac{\partial I y'(t)}{\partial y} + \frac{\partial I}{\partial t} = 0$ . Normalizando con respecto a  $\Delta_t$ , podemos definir el vector de desplazamiento ( $V_x = \frac{\Delta x}{\Delta t}$ ,  $V_y = \frac{\Delta y}{\Delta t}$ ). Obteniendo de esta forma la ecuación de desplazamiento

$$I_x(q_i)V_x + I_y(q_i)V_y = -I_t(q_i). \quad (20)$$

La idea para esto es analizar un arreglo rectangular de pixeles (lo cual como se conoce como ventana) que pueden ser  $q_1, q_2, \dots, q_n$  cuyo centro es el pixel de interés  $p$ . En la ecuación (20)  $I_x(q_i)$  denota el cambio de brillo horizontal del pixel  $q_i$ ,  $I_y(q_i)$  el cambio vertical y  $I_t(q_i)$  el cambio en brillo en el tiempo, el signo menos es debido a la relación inversa entre la dirección del movimiento y la variación del brillo. Se obtiene entonces un sistema de ecuaciones lineales de la forma  $Av = B$ , o sea,

$$\begin{pmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{pmatrix} \begin{pmatrix} V_x \\ V_y \end{pmatrix} = \begin{pmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{pmatrix}.$$

Multiplicando lo anterior por  $A^T$  (Para obtener la solución de mínimos cuadrados),

$$\begin{pmatrix} \sum_{i=1}^n I_x(q_i) & \sum_{i=1}^n I_x(q_i)I_y(q_i) \\ \sum_{i=1}^n I_y(q_i)I_x(q_i) & \sum_{i=1}^n I_y(q_i) \end{pmatrix} \begin{pmatrix} V_x \\ V_y \end{pmatrix} = \begin{pmatrix} -\sum_{i=1}^n I_x(q_i)I_t(q_i) \\ -\sum_{i=1}^n I_y(q_i)I_t(q_i) \end{pmatrix}.$$

De allí se obtiene la velocidad  $(V_x, V_y)$ . Note que por pixel se obtiene un vector  $V = (V_x, V_y)$ , entonces estas características describen toda la imagen y vienen siendo realmente dos matrices:  $V_x = [V_x(i, j)]_{i=1, j=1}^{W, H}$  y  $V_y = [V_y(i, j)]_{i=1, j=1}^{W, H}$ . Con estas características y operaciones matriciales punto a punto se pueden hacer más cálculos como por ejemplo la magnitud de la velocidad  $\|V\|$  en cada punto,

$$\begin{aligned} \|V\| &= [\|V\|(i, j)]_{i=1, j=1}^{W, H} \\ &= \left[ \sqrt{V_x(i, j)^2 + V_y(i, j)^2} \right]_{i=1, j=1}^{W, H}. \end{aligned}$$

De la misma forma se puede calcular la tangente que al ser un vector será también dos matrices:  $T_x = \frac{V_x}{\|V\|}$  y  $T_y = \frac{V_y}{\|V\|}$ . La normal será:  $N_x = \frac{T'_y}{\|T'\|}$  y  $N_y = \frac{T'_x}{\|T'\|}$  y como la aceleración es la derivada de la velocidad

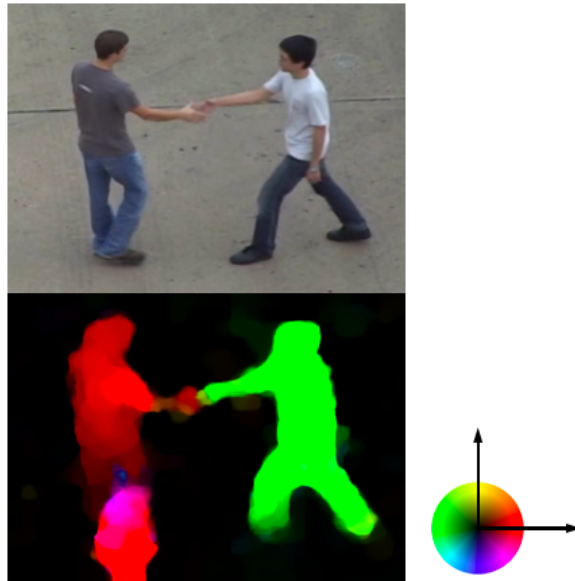
$$\begin{aligned} a &= \frac{d}{dt} T \|V\| \\ &= T' \|V\| + T \frac{d}{dt} \|V\| \\ &= N \|T'\| \|V\| + T \frac{d}{dt} \|V\| \\ &= a_N N + a_T T, \end{aligned}$$

donde  $a_N = \|T'\| \|V\|$  y  $a_T = \frac{d}{dt} \|V\|$  son la aceleración tangencial y normal respectivamente. Al ser  $N$  y  $T$  unitarios, la magnitud de la aceleración sería  $\|a\| = \sqrt{\|a_N\|^2 + \|a_T\|^2}$  y al igual que la tangente y la normal, la aceleración serán las matrices:  $a_x = a_N N_x + a_T T_x$  y  $a_y = a_N N_y + a_T T_y$ . En la Figura 5 se muestran dos imágenes, una del data-set y su correspondiente imagen de flujo óptico, de acuerdo



al círculo de color que se describe en la imagen, en ella se observa que el sujeto de la derecha realiza un desplazamiento hacia la izquierda mientras el sujeto de la izquierda realiza un desplazamiento hacia la derecha.

**Figura 5.** Imagen real e imagen del flujo óptico que como indica el círculo de color, el color depende de la dirección del flujo y la intensidad del color depende de la magnitud del vector del flujo.



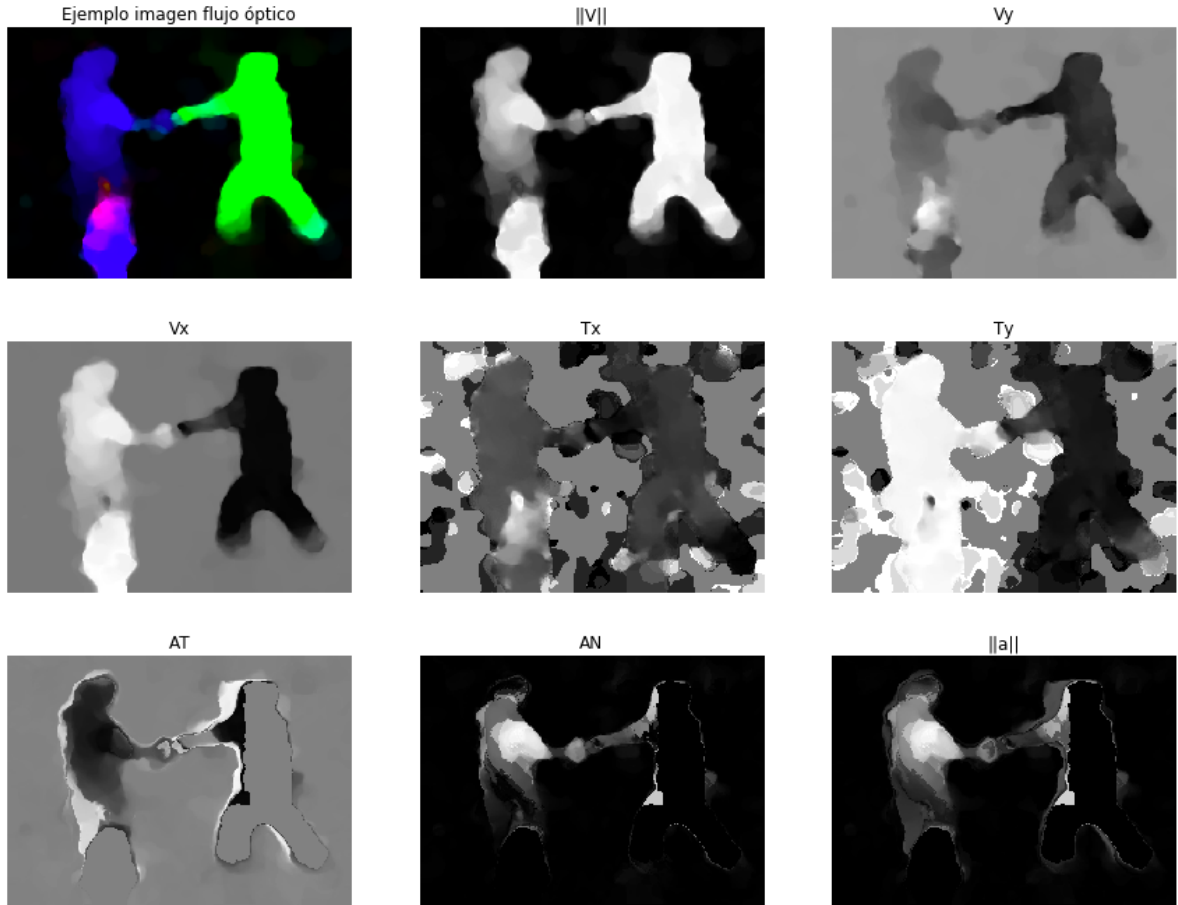
En la Figura 6 se presenta nuevamente este ejemplo de flujo óptico y algunas características cinemáticas calculadas sobre este patrón. Para el mapa cinemático de la magnitud  $\|V\|$ , los valores están escalados entre el máximo (blanco) y el mínimo (negro). En el ejemplo en particular se observa el sujeto de la derecha se mueve más rápido, con cierta importancia de rapidez en la pierna del sujeto de la izquierda. En cuanto a los mapas que representan los componentes de  $V_x$ ,  $V_y$ ,  $T_x$  y  $T_y$ , los mapas están representados entre  $[-val, val]$  y por lo tanto el valor cero se representa como un valor intermedio en gris. Estos mapas discriminan la dirección de la velocidad según cada uno de sus componentes, observándose mas acentuado en los mapas unitarios. Para la características  $AN$  y  $\|a\|$  los mapas cinemáticos están escalados entre  $[0-1]$  siendo blanco el máximo valor que se representa con el color

blanco y negro para los valores cero. Como se puede observar, estos mapas dan información complementaria a la velocidad, donde las regiones con mayor aceleración están acentuadas en el brazo, lo cual esta altamente correlacionado con la acción que desarrolla (dar la mano).

Además de estas, se pueden calcular más características y por tanto en un video que se puede ver como un conjunto de  $k$  imágenes  $\{I_1, I_2, \dots, I_k\} \subset \mathbb{R}^{W \times H}$ , cada imagen va a ser representada por  $N$  características  $\{F^{(1)}, F^{(2)}, \dots, F^{(N)}\} \subset \mathbb{R}^{W \times H}$ . Como se mencionó en la Sección 1 de la introducción y planteamiento del problema, una vez están definidas las características, a cada imagen  $I_m$  le va corresponder una matriz de covarianza  $C_{(m)} \in \mathbb{R}^{N \times N}$  que se construyó con las  $N$  características y en donde  $m \in \{1, 2, \dots, k\}$ .

Lo que se hizo entonces fue pasar de tener el video como un conjunto de  $k$  imágenes secuenciales  $\{I_1, I_2, \dots, I_k\} \subset \mathbb{R}^{W \times H}$  a tenerlo como un conjunto de matrices de covarianza  $\{C_1, C_2, \dots, C_k\} \subset \mathbb{R}^{N \times N}$ , pero esto se puede compactar más si se logra calcular la media del conjunto  $\{C_1, C_2, \dots, C_k\} \subset S_+^N$ .

**Figura 6.** Ejemplo de una imagen de flujo óptico de donde se extraen las características velocidad y tangente por componentes ( $V_x, V_y, T_x, T_y$ ), también se extrae la magnitud de la velocidad ( $\|V\|$ ), la aceleración tangencial ( $a_T$  que se menciona como  $AT$ ), la aceleración normal ( $a_N$  que se menciona como  $AN$ ) y la magnitud de la aceleración ( $\|a\|$ ). La imagen de flujo es tomada del conjunto de datos <sup>30</sup>.



## 5.2. REGULARIZACIÓN DE MATRICES

A la hora de trabajar con las matrices de covarianza sus autovalores pueden resultar ser valores cercanos a cero o incluso negativos por errores numéricos de computación, lo cual da inconvenientes a la hora de calcular funciones como log. A continuación se menciona el siguiente lema para posteriormente poder trabajar con los algoritmos para calcular la media de un conjunto de matrices como descriptor de

video.

**Lema 3.** Si  $v, \lambda$  son el autovector y el autovalor respectivamente de una matriz  $A$  entonces  $v, \lambda + \alpha$  son el autovector y el autovalor de la matriz  $A + \text{diag}(\alpha)$

*Demostración.* Se tiene por hipótesis que  $Av = \lambda v$ , entonces

$$\begin{aligned}(A + \text{diag}(\alpha))v &= (A + \alpha I)v \\ &= Av + \alpha v \\ &= \lambda v + \alpha v \\ &= (\lambda + \alpha)v,\end{aligned}$$

mostrando así que efectivamente al sumar una matriz diagonal  $\text{diag}(\alpha)$ , se incrementan los autovalores en un  $\alpha$ . □

Como se mencionó en la Sección 1, el conjunto  $C = \{C_1, C_2, \dots, C_n\} \subset S_+^n$  y se requiere que esté contenido en  $S_{++}^n$  para que todas las matrices estén en esta variedad y a su vez los autovalores de cada  $C_i$  sean estrictamente mayores que cero para todo índice  $i = 1, \dots, n$  y de esta forma no hayan problemas al trabajar con la función logaritmo. Para lograr esto se hace uso del Lema 3 de forma que a cada  $C_i$  se le suma una matriz  $\text{diag}(\alpha)$  de forma que  $C_i + \text{diag}(\alpha) \succ 0$  para  $i = 1, 2, \dots, n$ . Note entonces que el conjunto  $\{C_1 + \text{diag}(\alpha), C_2 + \text{diag}(\alpha), \dots, C_n + \text{diag}(\alpha)\} \subset S_{++}^n$  y por tanto se le puede calcular la media de forma que el video queda entonces descrito por una sola matriz  $\mu \in S_{++}^N$ .

### 5.3. CLASIFICACIÓN DE ACCIONES: CLASIFICADOR BAYESIANO INGENUO

En el reconocimiento de acciones, una vez se describe cada secuencia de video con una matriz media de covarianza, se procede a entrenar un algoritmo de aprendizaje de máquina para determinar la acción mas probable dado el descriptor computado

para la secuencia. En este caso, se seleccionan un conjunto de videos  $V_1, V_2, \dots, V_m$ , descrito por sus respectivas medias de covarianza  $\{\mu_1, \mu_2, \dots, \mu_m\}$ , para determinar cuales son las fronteras que separan las matrices media de covarianza según la acción que corresponde. Para este proceso de entrenamiento es necesario que los descriptores estén en un espacio Euclidiano por tanto en este caso es necesario llevar el conjunto  $\{\mu_1, \mu_2, \dots, \mu_m\}$  a  $T_{id}S_{++}^N$  con la función  $\log : S_{++}^N \rightarrow T_{id}S_{++}^N = S^N$  y posteriormente vectorizar la parte triangular superior de cada  $\log(\mu_i)$  ya que  $\log(\mu_i) \in S^N$  y así obtener un vector  $X_i \in \mathbb{R}^{\frac{N(N+1)}{2}}$ . Entonces cada  $\mu_i \in S_{++}^N$  es ahora  $X_i \in \mathbb{R}^{\frac{N(N+1)}{2}}$  y por tanto nuestros videos  $V_1, V_2, \dots, V_m$  están ahora descritos por los vectores  $\{X_1, X_2, \dots, X_m\} \subset \mathbb{R}^{\frac{N(N+1)}{2}}$ , donde cada uno de ellos tiene asociada la etiqueta de la acción correspondiente, formando la tupla:  $(X_i, y_i)$ . En la literatura existen diversos algoritmos para aproximar este problema, cada uno de ellos, enfocado en una hipótesis de separación diferente. En nuestro caso particular, utilizamos el algoritmo Bayesiano ingenuo que se fundamenta en la caracterización como distribuciones normales e independientes de cada una de los valores de los vectores  $X_i$  y resuelve la probabilidad de pertenecer a cada clase como una regla de Bayes, es decir, para cada uno de los  $\frac{N(N+1)}{2}$  valores, se tienen  $m$  datos provenientes de los vectores  $\{X_1, X_2, \dots, X_m\}$  con los que se pueden construir  $\frac{N(N+1)}{2}$  distribuciones normales e independientes con las que se busca mediante un proceso de clasificación asignarle la clase mas probable a un nuevo valor  $X$ .

Formalmente, la clasificación es una función  $\gamma : \mathbb{R}^m \rightarrow \{C_1, C_2, \dots, C_n\}$  que asigna a un vector  $X \in \mathbb{R}^m$  una de las  $n$  clases existentes  $\{C_1, C_2, \dots, C_n\}$ , en este caso es determinar a cuál de las clases  $C_1, C_2, \dots, C_n$  pertenece un vídeo descrito como un vector  $X = (x_1, x_2, \dots, x_\ell) \in \mathbb{R}^\ell$  donde  $\ell = \frac{N(N+1)}{2}$ . En la aplicación que se trabaja, con el clasificador Bayesiano ingenuo. Del teorema de Bayes se puede obtener la probabilidad de que dado este nuevo valor  $X \in \mathbb{R}^\ell$  este pertenezca a una clase  $C_k$ ,

donde  $k \in \{1, 2, \dots, n\}$

$$P(C_k|X) = \frac{P(C_k)P(X|C_k)}{P(X)}, \quad (21)$$

o que es lo mismo

$$P(C_k|x_1, x_2, \dots, x_\ell) = \frac{P(C_k)P(x_1, x_2, \dots, x_\ell|C_k)}{P(x_1, x_2, \dots, x_\ell)}, \quad (22)$$

donde  $P(X) = \sum_{i=1}^n P(C_i)P(X|C_i)$ . Como las clases son mutuamente independientes y además  $P(C_k)P(x_1, x_2, \dots, x_\ell|C_k) = P(C_k, x_1, x_2, \dots, x_\ell)$  entonces

$$P(C_k, x_1, x_2, \dots, x_\ell) = P(C_k)P(x_1|C_k)P(x_2|C_k) \cdots P(x_\ell|C_k),$$

y así (22) sería

$$P(C_k|x_1, x_2, \dots, x_\ell) = \frac{1}{P(x_1, x_2, \dots, x_\ell)} P(C_k) \prod_{i=1}^m P(x_i|C_k).$$

Con lo anterior

$$\begin{aligned} \gamma(X) &= \arg \max_{k \in \{1, 2, \dots, n\}} P(C_k|X) \\ &= \arg \max_{k \in \{1, 2, \dots, n\}} P(C_k|x_1, x_2, \dots, x_\ell) \\ &= \arg \max_{k \in \{1, 2, \dots, n\}} \frac{1}{P(x_1, x_2, \dots, x_\ell)} P(C_k) \prod_{i=1}^m P(x_i|C_k) \\ &= \arg \max_{k \in \{1, 2, \dots, n\}} P(C_k) \prod_{i=1}^n P(x_i|C_k), \end{aligned}$$

pues note que  $P(x_1, x_2, \dots, x_\ell)$  es un valor escalar que depende de  $x_1, x_2, \dots, x_\ell$  y no influye en el cálculo de  $C$ , por otra parte note que se requiere calcular cada  $P(x_i|C_k)$  y para esto se asume que los valores de cada clase se distribuyen de

manera normal, entonces la distribución de probabilidad de la clase  $C_k$  será

$$P(x = v|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}},$$

donde  $\mu_k$  y  $\sigma_k^2$  son la media y la varianza de la clase  $C_k$  respectivamente.

Un descriptor es un proceso que convierte los datos a objetos en  $\mathbb{R}^m$ , usualmente se usan conjuntos de datos existentes para aplicar el descriptor que se haya planteado y posteriormente se usa el método de clasificación elegido.

#### 5.4. PROCESO DE VALIDACIÓN CRUZADA

Una forma de medir el proceso de clasificación elegido es mediante una validación, la cual consiste en elegir previamente un conjunto de datos (data-set) etiquetados  $\{(X_i, y_i)\}_{i=1}^m$  el cual se particiona en dos conjuntos: conjunto de test  $T \subset \{(X_i, y_i)\}_{i=1}^m$  y el conjunto de prueba  $P \subset \{(X_i, y_i)\}_{i=1}^m$  donde usualmente  $|T| > |P|$ . Lo anterior con el fin de tomar los elementos de  $T$  para generar las distribuciones del proceso de clasificación y luego ver para cada  $X \in P$  si  $\gamma(X_i) = y_i$ , el porcentaje de exactitud (accuracy) será el valor con el que se medirá la clasificación realizada.

En este proyecto se utilizó una validación cruzada de orden  $K$  (K-Fold) que consiste en primer lugar en desordenar el conjunto  $\{(X_i, y_i)\}_{i=1}^m$ , posteriormente decidir el  $K$  que es la cantidad de particiones en la que se quiere dividir el conjunto de datos de forma que en cada iteración se tomen  $K - 1$  bloques de test y el bloque restante de prueba de forma que habrán  $K$  iteraciones, en cada iteración se hace una validación como la mencionada en 5.4, de forma que se obtiene un porcentaje de exactitud (accuracy) por iteración y al final estos porcentajes se promedian para dar el porcentaje de acierto del método de clasificación elegido mediante esta validación cruzada de orden  $K$ .

## 6. DISEÑO EXPERIMENTAL

En esta sección se mostrará el diseño de dos experimentos para la validación de las estrategias estudiadas y propuestas durante este trabajo. La primera con simulaciones de matrices aleatorias y la segunda con un conjunto de datos reales.

### 6.1. SIMULACIONES CON MATRICES ALEATORIAS

Para hacer pruebas con matrices SPD aleatorias se usó la función `make_spd_matrix` del paquete `scikit - learn` <sup>31</sup>. Con este generador de matrices se creó una nueva función que permitía generar un conjunto de matrices SPD teniendo en cuenta parámetros como la dimensión y la cantidad de matrices. Con estos conjuntos mas adelante se comparará el comportamiento de los algoritmos estudiados en cuanto al acierto en el proceso de clasificación y en cuanto al tiempo que toman en calcular la media variando el tamaño o la cantidad de las matrices.

**6.1.1. EJEMPLO DE CLASIFICACIÓN CON MATRICES DE BAJA DIMENSIONALIDAD** Partiendo del ejemplo que se hizo en la Sección 4.1.1 se pretende ahora ilustrar cómo se vería un proceso de clasificación en  $S_{++}^2$ . Con la función que genera matrices aleatorias SPD mencionada en la Sección 6.1 se pueden generar dos subconjuntos disjuntos  $A$  y  $B$  de  $S_{++}^2$  con 30 elementos cada uno (Ver figuras 8(a) y 8(b)), al generador de matrices SPD se le hizo una modificación de forma que

$$A \subset \{(x, y, z) \in \mathbb{R}^3 : xz - y^2 \geq 0, x \geq 0, z \geq 0, y > 0\},$$

---

<sup>31</sup> F. Pedregosa y col. "Scikit-learn: Machine Learning in Python". En: *Journal of Machine Learning Research* 12 (2011), págs. 2825-2830.

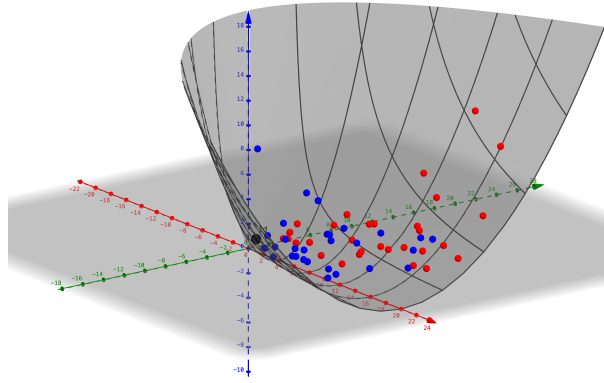


y

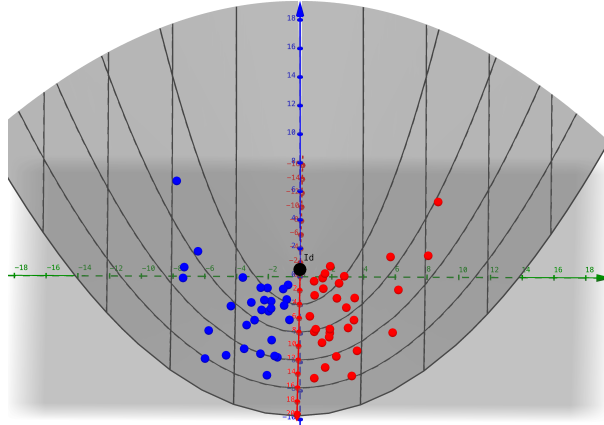
$$B \subset \{(x, y, z) \in \mathbb{R}^3 : xz - y^2 \geq 0, x \geq 0, z \geq 0, y < 0\}.$$

Como se mencionó en la Sección 5.3, es necesario calcular  $\log_{Id}(A)$  y  $\log_{Id}(B)$  previamente para poder aplicar el clasificador Bayesiano, ver Figura 7. Con  $\log_{Id}(A)$  y  $\log_{Id}(B)$  se hizo una validación cruzada de 30 y se obtuvo un 100 % de acierto.

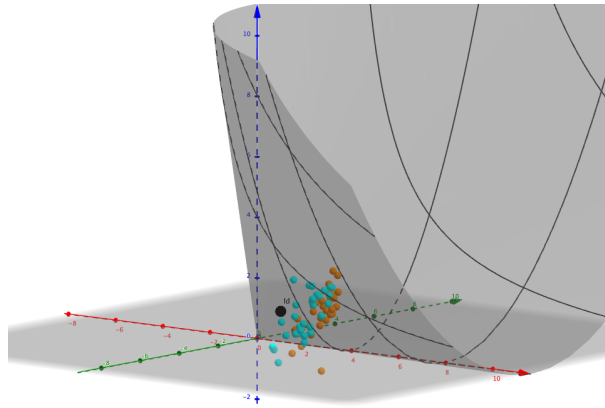
**Figura 7.** Ejemplo Clasificación matrices 2x2



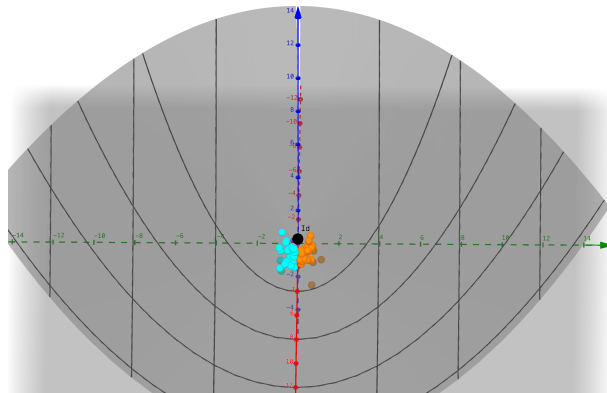
(a) Espacio  $S^2_{++}$  visto en  $\mathbb{R}^3$ , grupo de matrices  $A$  con  $y > 0$  en rojo y grupo  $B$  con  $y < 0$  en azul.



(b) Espacio  $S^2_{++}$  visto en  $\mathbb{R}^3$ , grupo de matrices  $A$  con  $y > 0$  en rojo y grupo  $B$  con  $y < 0$  en azul.



(c) Espacio  $S^2_{++}$  visto en  $\mathbb{R}^3$ , grupo de matrices  $\log_{Id}(A)$  en naranja y  $\log_{Id}(B)$  en celeste.



(d)  $\log_{Id}(A)$  en naranja y  $\log_{Id}(B)$  en celeste.

## 6.2. RECONOCIMIENTO DE ACCIONES

Una vez hechas las simulaciones se pretende ahora trabajar con videos reales, esto con el fin de mas adelante poder comparar diferentes algoritmos para el cálculo de la media en cuanto al acierto en el proceso de clasificación y el tiempo utilizado para calcular la media. Para hacer pruebas con conjuntos de datos reales se hizo uso del conjunto de datos *UT-Interaction* <sup>32</sup>, el cual está conformado de dos conjuntos de vídeos, uno grabado en un parqueadero con el fondo estático (Conjunto 1) y otro con fondo un parque (Conjunto 2) ver Figura 8.

**Figura 8.** Presentación Data-Set.



(a) Ejemplo de video de la secuencia 1 del data-set UT-Interacion <sup>(34)</sup> con fondo sólido. (b) Ejemplo de video de la secuencia 2 del data-set UT-Interacion <sup>(36)</sup> con fondo un parque.

Cada conjunto tiene 60 videos de forma que hay 10 videos de cada clase las cuales son “Dar la mano”, “Abrazar”, “Patear”, “Apuntar”, “Dar un puñoz”, “Empujar”. Los videos del conjunto 1 son de aproximadamente 120-150 imágenes de dimensión  $480 \times 320$  y los del conjunto 2 son de aproximadamente 70-120 frames de dimensión  $300 \times 250$ . En ambas secuencias de videos del Data-Set *UT-Interaction*, a los 60

---

<sup>32</sup> M. S. Ryoo y J. K. Aggarwal. *UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA)*. 2010.

vídeos se les calculó el campo de vectores de velocidad con ayuda del flujo óptico descrito en <sup>37</sup> que al igual que el flujo óptico mencionado en la Sección 5.1 este devuelve por cada imagen tres matrices:  $V_x$ ,  $V_y$  y  $\|V\|$  como primeras características de esta imagen, para este trabajo se decidió utilizar las características cinemáticas:  $[V_x, V_y, \|V\|, T_x, T_y, A_T, A_N, \|a\|, N_x, N_y, a_x, a_y, \|T'\|, \|N\|]$ . Con estas características se describió cada imagen de cada video con una matriz de covarianza de dimensión  $14 \times 14$  y posteriormente se calculó la media de estas matrices. Luego estas matrices se pasan al plano tangente en la identidad y se ordenan en un vector de dimensión  $\frac{14 \times 15}{2} = 105$ .

---

<sup>37</sup> Thomas Brox y Jitendra Malik. "Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation". En: *IEEE transactions on pattern analysis and machine intelligence* 33 (mar. de 2011), págs. 500-13. DOI: 10.1109/TPAMI.2010.143.

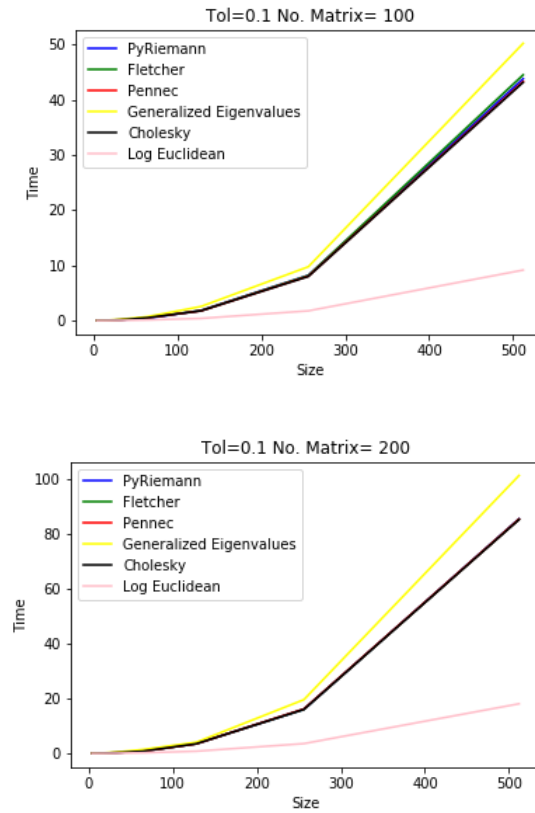
## 7. EVALUACIÓN Y RESULTADOS

Este trabajo tenía como objetivo principal estudiar y analizar el potencial que tiene calcular la media de un conjunto de matrices de covarianza como descriptor compacto de videos en problemas de reconocimientos de acciones. Los algoritmos que se estudiarán en las gráficas de acá en adelante son el de Pennec 4.1, el de Fletcher 4.2, el de Alexandre Barachant 4.4 que denotamos como “*Pyriemann*”, el propuesto usando autovalores generalizados 4.5 denotado por “*Generalized Eigenvalues*”, el propuesto usando descomposición de Cholesky 4.6 y la iteración Log-Euclidean 4.3. Inicialmente se presentarán simulaciones con conjuntos aleatorios de matrices de covarianza para evaluar el comportamiento de los diferentes algoritmos estudiados para el calculo de una media, posteriormente se usarán los algoritmos en un problema real de reconocimiento de acciones para poder llevar a cabo un proceso de clasificación y evaluar los resultados en cuanto a porcentaje de exactitud (accuracy).

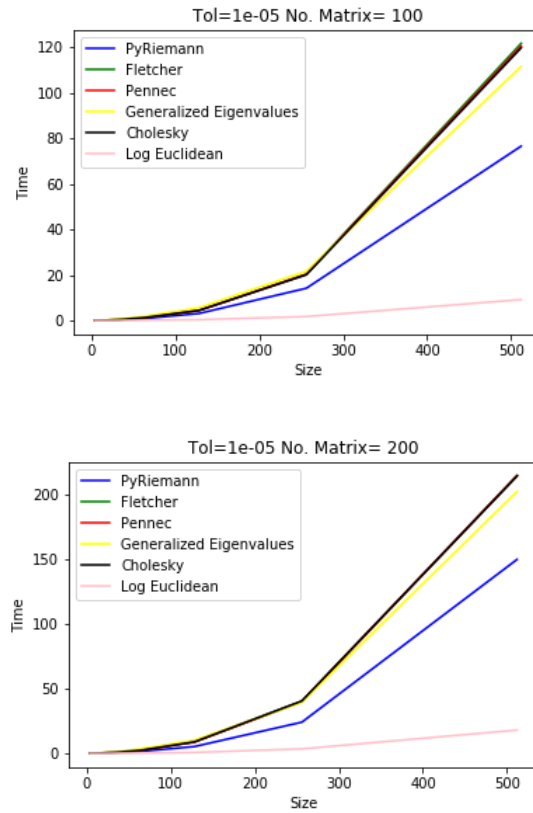
### 7.1. RESULTADOS EN SIMULACIONES CON MATRICES ALEATORIAS

En las Figuras 9, 10 y 11 se hace en cada una dos experimentos, uno con 100 matrices y otro con 200, a la vez en cada figura se varía la tolerancia de los algoritmos primero con tolerancia de  $10^{-2}$ , luego  $10^{-6}$  y posteriormente  $10^{-9}$ , lo que presentan los gráficos es el tiempo que demoran en hallar la media los diferentes algoritmos mientras se varía el tamaño desde matrices de dimensión  $2 \times 2$  hasta matrices de  $512 \times 512$ .

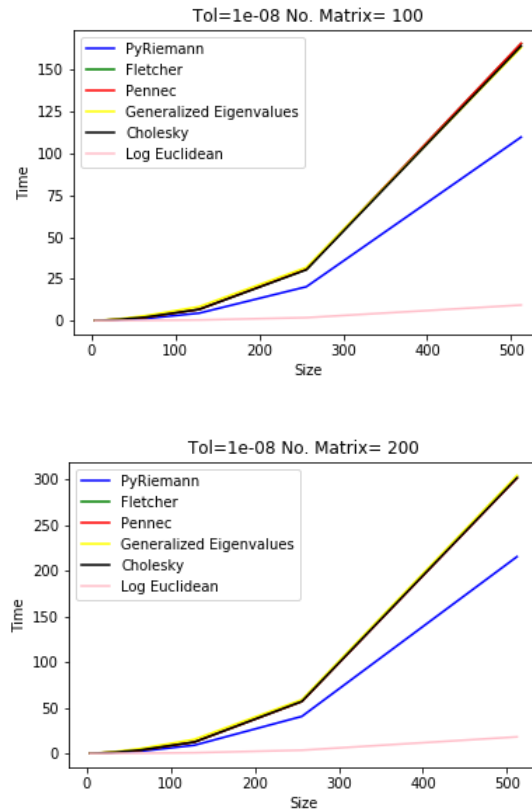
**Figura 9.** Comparación del tiempo que toman los diferentes algoritmos al cambiar el tamaño de las matrices (desde  $2 \times 2$  hasta  $512 \times 512$ ) bajo una tolerancia de  $10^{-2}$ . En la primera subfigura se realiza con 100 y en la segunda con 200.



**Figura 10.** Comparación del tiempo que toman los diferentes algoritmos al cambiar el tamaño de las matrices (desde  $2 \times 2$  hasta  $512 \times 512$ ) bajo una tolerancia de  $10^{-6}$ . En la primera subfigura se realiza con 100 y en la segunda con 200.

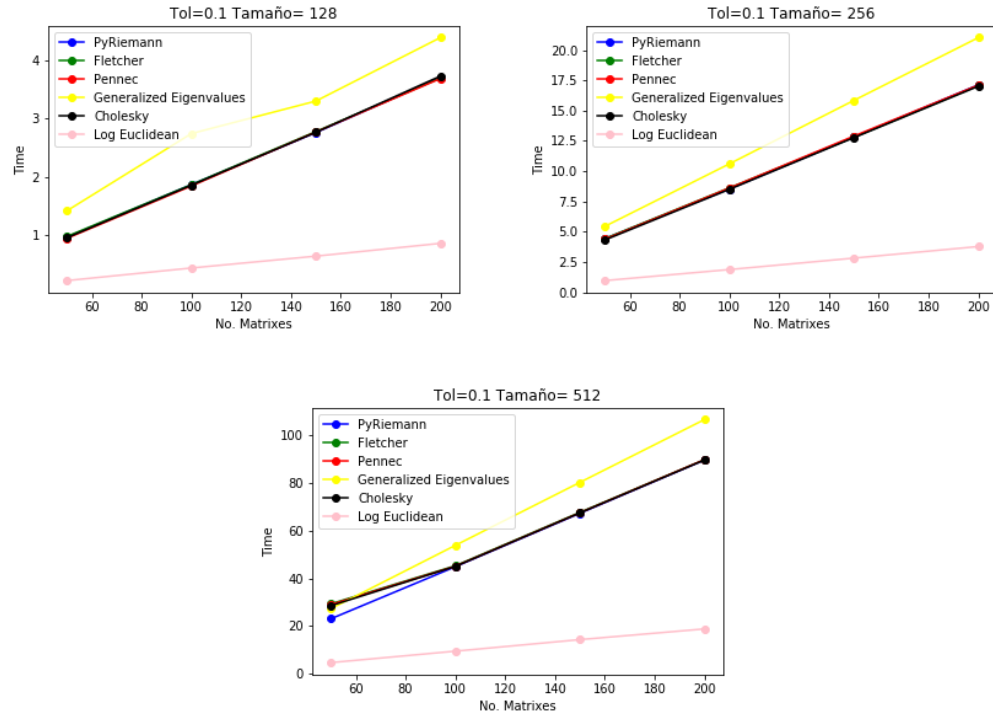


**Figura 11.** Comparación del tiempo que toman los diferentes algoritmos al cambiar el tamaño de las matrices (desde  $2 \times 2$  hasta  $512 \times 512$ ) bajo una tolerancia de  $10^{-9}$ . En la primera subfigura se realiza con 100 y en la segunda con 200.

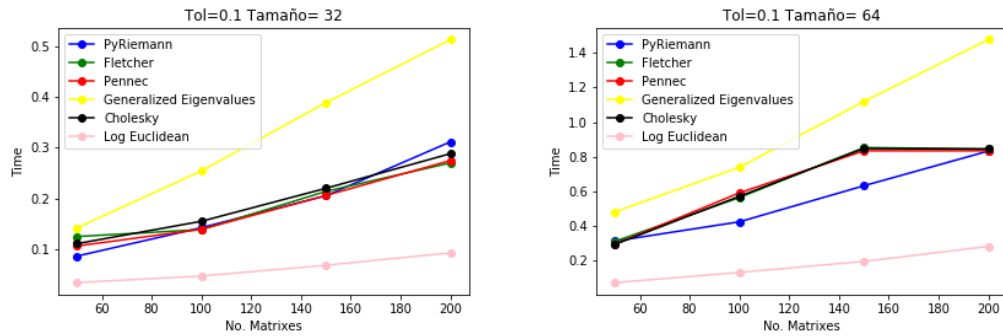


En las Figuras 12, 13 y 14 se hace en cada una cinco experimentos, con matrices de  $32 \times 32$ ,  $64 \times 64$ ,  $128 \times 128$ ,  $256 \times 256$  y  $512 \times 512$ . A la vez en cada figura se varía la tolerancia de los algoritmos primero con tolerancia de  $10^{-2}$ , luego  $10^{-6}$  y posteriormente  $10^{-9}$ , lo que presentan los gráficos es el tiempo que demoran en hallar la media los diferentes algoritmos mientras se varía la cantidad de matrices, desde 50 hasta 200 matrices.

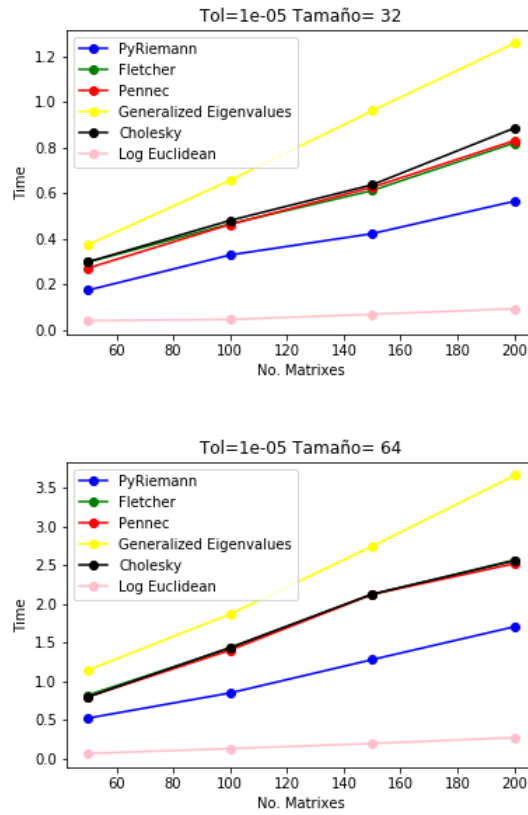


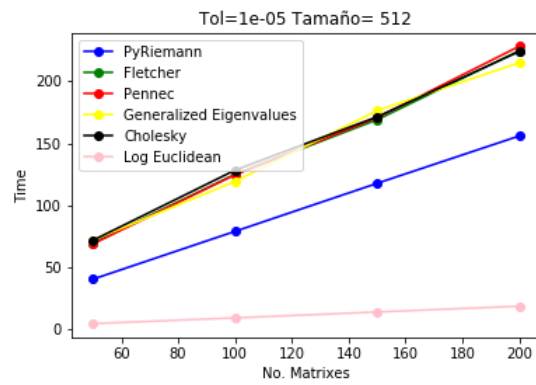
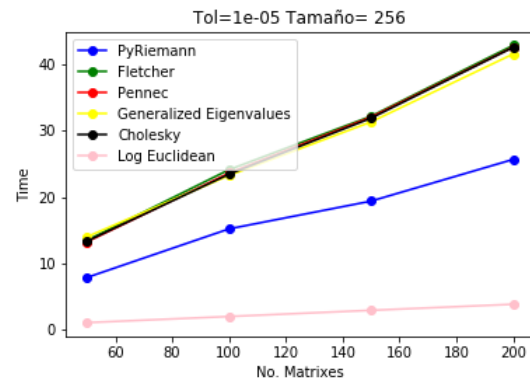
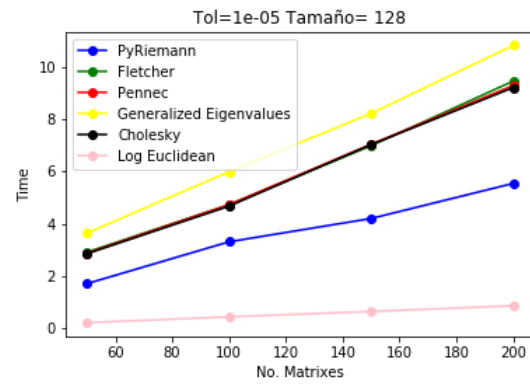


**Figura 12.** Comparación del tiempo que toman los diferentes algoritmos al cambiar la cantidad de matrices (desde 50 hasta 200) bajo una tolerancia de  $10^{-2}$ . Entre cada subfigura se varía el tamaño de las matrices desde  $32 \times 32$  hasta  $512 \times 512$ .

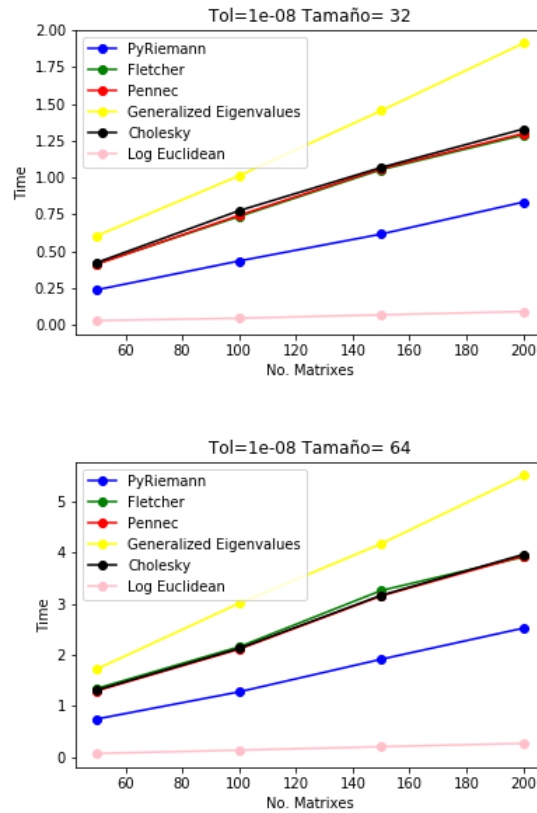


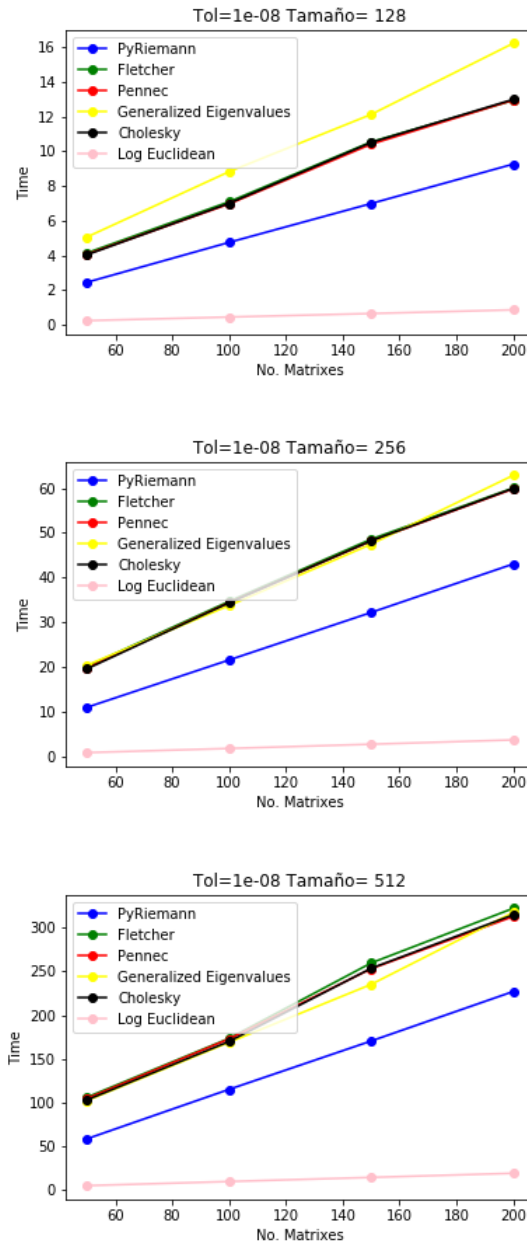
**Figura 13.** Comparación del tiempo que toman los diferentes algoritmos al cambiar la cantidad de matrices (desde 50 hasta 200) bajo una tolerancia de  $10^{-6}$ . Entre cada subfigura se varía el tamaño de las matrices desde  $32 \times 32$  hasta  $512 \times 512$ .





**Figura 14.** Comparación del tiempo que toman los diferentes algoritmos al cambiar la cantidad de matrices (desde 50 hasta 200) bajo una tolerancia de  $10^{-9}$ . Entre cada subfigura se varía el tamaño de las matrices desde  $32 \times 32$  hasta  $512 \times 512$ .

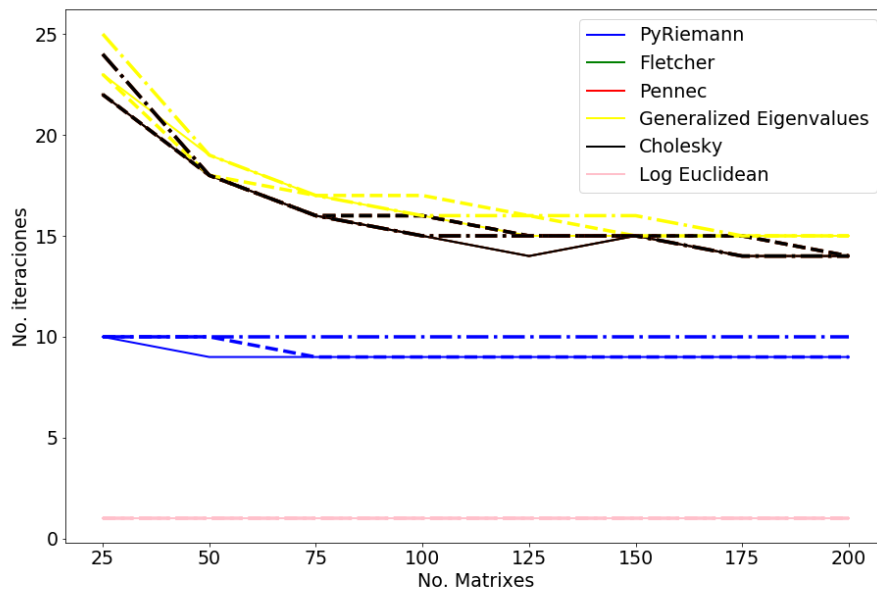




En la Figura 15 se hacen 3 experimentos, el primero con tolerancia de  $10^{-2}$ , luego  $10^{-6}$  y posteriormente  $10^{-9}$ . A la vez en cada uno las líneas continuas representan matrices de  $32 \times 32$ , las discontinuas matrices de  $64 \times 64$  y las discontinuas con puntos representan matrices de  $128 \times 128$ . Lo que presentan los gráficos es la cantidad de

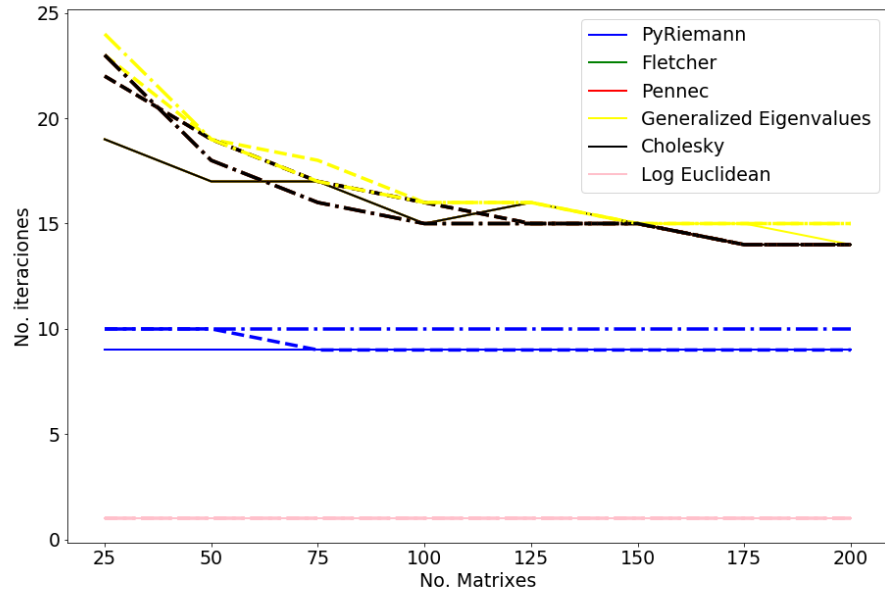
iteraciones que toman en hallar la media los diferentes algoritmos mientras se varía la cantidad de matrices, desde 50 hasta 200 matrices.

**Figura 15.** Comparación del número de iteraciones que toman los diferentes algoritmos al cambiar la cantidad de matrices (desde 50 hasta 200). Entre cada subfigura se varía el la tolerancia, la primera de  $10^{-2}$ , la segunda de  $10^{-6}$  y la última de  $10^{-9}$ . Las líneas continuas representan matrices matrices de  $32 \times 32$ , las discontinuas matrices de  $64 \times 64$  y las discontinuas con puntos matrices de  $128 \times 128$ .

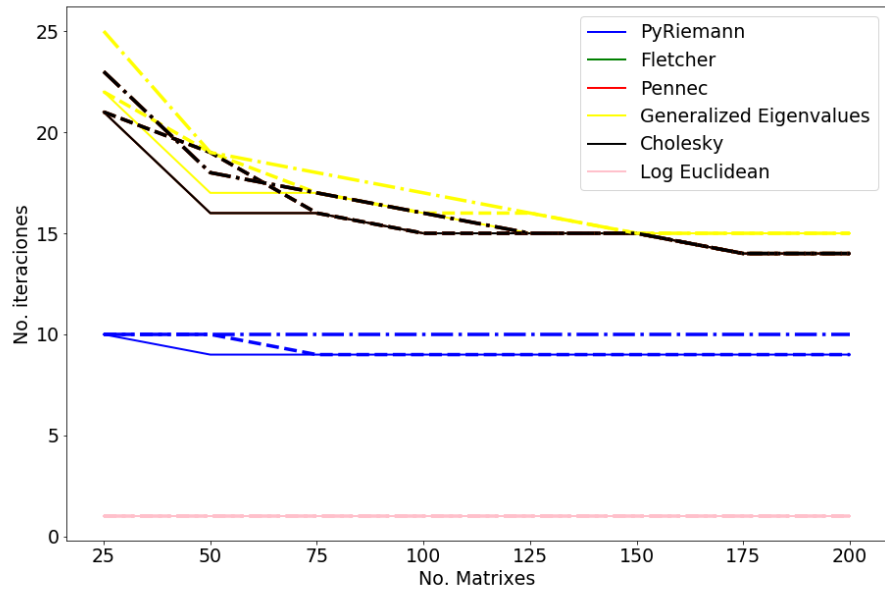


(a) Tolerancia de  $10^{-2}$ .

En todas las simulaciones se evidencia que Log Euclidean resulta ser la mas rápida y la que menos iteraciones requiere, esto se debe a que su computo es simplemente una iteración, sobre las Figuras 9, 10 y 11 donde se analiza el tiempo con respecto a la dimensión de las matrices, se evidencia que ante al disminuir la tolerancia, el tiempo de convergencia aumenta. Además, los demás algoritmos a parte de Log Euclidean y el propuesto por Alexandre Barachant ("*PyRiemann*") toman tiempos muy similares y mayores a estos dos.



(b) Tolerancia de  $10^{-6}$ .



(c) Tolerancia de  $10^{-9}$ .

Analizando el tiempo con respecto a la cantidad de matrices, en las Figuras 12,13 y 14 se evidencia que ante matrices de mayor dimensión el tiempo de convergencia es mayor. Además, la media propuesta que involucra autovalores generalizados ("*Generalized Eigenvalues*") en general es la que mas demora pero mejora los tiempos con respecto a las demás al incrementar el tamaño. Cuando la tolerancia es de  $10^{-2}$ , las medias diferentes a la "*Log-Euclidean*" y "*Generalized Eigenvalues*" tienen un comportamiento similar, pero cuando se disminuye la tolerancia a  $10^{-6}$  y  $10^{-9}$  el algoritmo propuesto por Alexandre Barachant ("*PyRiemann*") mejora el tiempo.

Analizando la cantidad de iteraciones con respecto a la cantidad de matrices, en la Figura 15 se evidencia claramente que el algoritmo "*Log Euclidean*" siempre toma una sola iteración y que al aumentar la tolerancia la cantidad de iteraciones que toman los demás algoritmos no cambia, se mantiene entre 9 y 25. Además, se evidencia que al aumentar la cantidad de matrices, la cantidad de iteraciones requerida por los algoritmos se mantiene o disminuye. El incremento del tamaño de las matrices tampoco cambia la cantidad de iteraciones requerida y el algoritmo "*PyRiemann*" es el que menos iteraciones toma para converger.

## 7.2. RESULTADOS EN RECONOCIMIENTO DE ACCIONES

Con las 14 características descritas en la Sección 6.2 se describió cada imagen de cada video con una matriz de covarianza de dimensión  $14 \times 14$  y posteriormente se calculó la media de estas matrices con los seis algoritmos mencionados al inicio de esta sección. De forma que con cada algoritmo se tienen 60 matrices de  $14 \times 14$  describiendo los 60 videos. Luego, cada una de estas matrices se pasa al plano tangente en la identidad y se ordena en un vector de dimensión  $\frac{14 \times 15}{2} = 105$  como se menciona en la Sección 5.3, este vector es el descriptor de un video y en este caso resulta ser bastante compacto en comparación con descriptores presentados



en el estado del estado del arte que pueden llegar a tener dimensión 73,728 <sup>38</sup>. Como con cada algoritmo se obtiene una descripción distinta de los videos y como se conoce la clase de cada video, se procede entonces a realizar un proceso de clasificación Bayesiano usando una validación cruzada de orden 60, esto con cada algoritmo para posteriormente comparar el acierto en la clasificación. En las figuras 16 y 17 se comparan los diferentes algoritmos por su acierto clasificando los videos de la secuencia 1 y la secuencia 2 del Data-Set *UT-Interaction* respectivamente y En las Figuras 18 y 19 se comparan los diferentes algoritmos con respecto a su tiempo.

**Figura 16.** Exactitud de los diferentes algoritmos en la clasificación de los 60 videos de la secuencia 1 del Data-Set *UT-Interaction* usando un clasificador Bayesiano y un proceso de validación cruzada de orden 60.

	Log Euclidean	Pennec	Fletcher	PyRiemann	Autovalores Generalizados	Cholesky
<b>Accuracy</b>	59.32%	65.25%	66.10%	65.25%	66.95%	65.25%

**Figura 17.** Exactitud de los diferentes algoritmos en la clasificación de los 60 videos de la secuencia 2 del Data-Set *UT-Interaction* usando un clasificador Bayesiano y un proceso de validación cruzada de orden 60.

	Log Euclidean	Pennec	Fletcher	PyRiemann	Autovalores Generalizados	Cholesky
<b>Accuracy</b>	52.54%	55.08%	54.24%	54.24%	55.93%	55.08%

**Figura 18.** Tiempos promedio de los diferentes algoritmos por cada clase en el cálculo de la media de los 60 videos de la secuencia 1 del Data-Set *UT-Interaction*.

	Log Euclidean	Pennec	Fletcher	PyRiemann	Autovalores Generalizados	Cholesky
<b>Dar la mano</b>	0.015	0.635	0.579	0.206	0.910	0.652
<b>Abrazar</b>	0.016	0.660	0.629	0.223	1.050	0.636
<b>Patear</b>	0.008	0.374	0.337	0.137	0.674	0.377
<b>Apuntar</b>	0.015	0.694	0.640	0.186	0.835	0.720
<b>Puño</b>	0.008	0.305	0.275	0.132	0.529	0.332
<b>Empujar</b>	0.010	0.469	0.428	0.151	0.731	0.469

<sup>38</sup> Mihir Jain, Hervé Jégou y Patrick Bouthemy. "Better Exploiting Motion for Better Action Recognition". En: jun. de 2013, págs. 2555-2562. DOI: 10.1109/CVPR.2013.330.

**Figura 19.** Tiempos promedio de los diferentes algoritmos por cada clase en el cálculo de la media de los 60 videos de la secuencia 2 del Data-Set *UT-Interaction*.

	<b>Log Euclidean</b>	<b>Pennec</b>	<b>Fletcher</b>	<b>PyRiemann</b>	<b>Autovalores Generalizados</b>	<b>Cholesky</b>
<b>Dar la mano</b>	0.012	0.701	0.643	0.196	0.848	0.710
<b>Abrazar</b>	0.013	0.807	0.718	0.251	0.992	0.790
<b>Patear</b>	0.008	0.383	0.362	0.098	0.482	0.392
<b>Apuntar</b>	0.019	0.659	0.505	0.418	0.718	0.628
<b>Puño</b>	0.006	0.329	0.302	0.096	0.479	0.337
<b>Empujar</b>	0.012	0.531	0.466	0.133	0.626	0.515

Con respecto al acierto en el proceso de clasificación tanto en la secuencia 1 como en la secuencia 2 (Figuras 16 y 17) se evidencia que el algoritmo “*Log-euclidean*” tiene el acierto mas bajo, esto puede deberse a su cálculo simple de una sola iteración. Los demás algoritmos presentan un acierto similar esto puede deberse a que la media calculada por estos es muy similar, sin embargo se destaca mínimamente el algoritmo implementado por nosotros (“*Autovalores generalizados*”), esto puede deberse a que se hace un calculo mas directo que puede llevar a una solución mas precisa en el calculo de la media como problema de optimización.

Con respecto al tiempo que tomó cada algoritmo en el calculo de la media tanto en la secuencia 1 como en la secuencia 2 (Figuras 18 y 19) se observa un comportamiento similar al de las simulaciones hechas con el generador de matrices SPD en la Sección 6.1, donde “*Log-euclidean*” es la que toma menor tiempo, luego le sigue “*PyRiemann*”, luego los demás tienen un comportamiento similar excepto (“*Autovalores generalizados*”) que es el algoritmo que más tiempo toma.

## 8. CONCLUSIONES Y PERSPECTIVAS

En este trabajo se hizo un estudio del cálculo de medias geométricas en el cono de las matrices SPD. Para ello se abordaron fundamentos relacionados con conceptos de geometría diferencial, que permitieron formular el problema variacional involucrado al calcular una media en el espacio de las matrices SPD. También se realizó una revisión bibliográfica con la fundamentación matemática necesaria en el tratamiento de matrices SPD. En 8 se introdujo la definición de funciones de matrices necesaria para el cálculo de las funciones log y exp en la variedad  $S_{++}^n$ . Además, se presentaron propiedades de las funciones de matrices que permitieron proponer dos algoritmos, uno utilizando autovalores generalizados y otro una descomposición de Cholesky. En este trabajo también se abordó el trabajo "*Riemannian Geometry for the Statistical Analysis of Diffusion Tensor Data*" <sup>(7)</sup> que permitió entender el problema del cálculo de la media como un proceso iterativo, resultado de aplicar el método del gradiente descendiente para solucionar un problema de optimización con una métrica riemanniana.

A partir de este trabajo, se pudieron entender los elementos principales del cálculo de la media y relacionarlo con otros trabajos como el propuesto por Fletcher *et. al* ?? y Alexandre Barachant *et. al* ?? . El análisis de estos componentes permitieron formular dos propuestas para el cálculo de la media en matrices SPD, que en la práctica, sobre secuencias de video, permiten estabilizar el cálculo de la media, así como también reducen el número de iteraciones del algoritmo. Tanto los trabajos estudiados en el estado del arte, como las propuestas descritas en este trabajo fueron evaluados en datos sintéticos y secuencias de video reales.

Específicamente, en este trabajo se estudió el uso de la media de matrices de covarianza como un descriptor compacto para el reconocimiento de acciones. Una vez se entendió la fundamentación matemática se hizo una implementación de los

diferentes algoritmos estudiados para calcular la media geométrica en el contexto del reconocimiento de acciones. Los algoritmos primero fueron evaluados usando simulaciones con matrices aleatorias y luego con un conjunto de videos públicos. En estas implementaciones se evidencia que el algoritmo propuesto usando autovalores generalizados a pesar de ser el que más tiempo toma presenta la mejor exactitud (69.95 %).

En el área del reconocimiento de acciones el estudio del cálculo de una media de un conjunto de matrices de covarianza abre la puerta a nuevos estudios para la descripción de un conjunto de datos, pues además de la media se pueden hacer otros estudios estadísticos en la variedad riemanniana que forman estas matrices de covarianza, como por ejemplo el análisis de componentes principales (PCA) y nuevas métricas específicas que permitan analizar la variedad que conforma el conjunto de covarianzas que representan el video.

## BIBLIOGRAFÍA

- Arsigny, Vincent y col. "Geometric Means in a Novel Vector Space Structure on Symmetric Positive-Definite Matrices". En: *SIAM J. Matrix Analysis Applications* 29 (ene. de 2006), págs. 328-347. DOI: 10.1137/050637996 (vid. págs. 28, 48, 49).
- Barachant, A. y J-R. King. *pyRiemann v0.2.2*. Jun. de 2015. DOI: 10.5281/zenodo.18982 (vid. págs. 49, 50, 85).
- Barachant, Alexandre y col. "Classification of Covariance Matrices Using a Riemannian-Based Kernel for BCI Applications". En: *Neurocomput.* 112 (jul. de 2013), 172–178. DOI: 10.1016/j.neucom.2012.12.039 (vid. pág. 50).
- Barachant, Alexandre y col. "Multiclass Brain-Computer Interface Classification by Riemannian Geometry". En: *IEEE Transactions on Biomedical Engineering* 59.4 (mar. de 2012), págs. 920-928. DOI: 10.1109/TBME.2011.2172210 (vid. pág. 50).
- Brox, Thomas y Jitendra Malik. "Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation". En: *IEEE transactions on pattern analysis and machine intelligence* 33 (mar. de 2011), págs. 500-13. DOI: 10.1109/TPAMI.2010.143 (vid. pág. 70).
- Carmo, Manfredo do. *Riemannian Geometry*. Birkhäuser Base, 1992 (vid. págs. 20-22, 24).
- Fletcher, P. y Sarang Joshi. "Principal Geodesic Analysis on Symmetric Spaces: Statistics of Diffusion Tensors". En: vol. 3117. Ene. de 2004, págs. 87-98 (vid. pág. 28).

Forsyth, David A. y Jean Ponce. *Computer Vision - A Modern Approach, Second Edition*. Pitman, 2012, págs. 1-791 (vid. págs. 13, 20).

Fréchet, Maurice René. “Les éléments aléatoires de nature quelconque dans un espace distancié”. fr. En: *Annales de l’institut Henri Poincaré* 10.4 (1948), págs. 215-310 (vid. pág. 28).

Fréchet, M. *L’intégrale abstraite d’une fonction abstraite d’une variable abstraite et son application a la moyenne d’un élément aléatoire de nature quelconque*. Revue Scientifique, 1944, págs. 483-512 (vid. pág. 28).

Geronimo, D. y col. “Survey of Pedestrian Detection for Advanced Driver Assistance Systems”. En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.7 (2010), págs. 1239-1258. DOI: 10.1109/TPAMI.2009.122 (vid. págs. 13, 18).

Higham, Nicholas J. *Functions of Matrices: Theory and Computation (Other Titles in Applied Mathematics)*. Philadelphia, PA, USA: Society for Industrial y Applied Mathematics, 2008 (vid. págs. 18, 33, 91, 96, 99).

Jain, Mihir, Hervé Jégou y Patrick Bouthemy. “Better Exploiting Motion for Better Action Recognition”. En: jun. de 2013, págs. 2555-2562. DOI: 10.1109/CVPR.2013.330 (vid. pág. 83).

Jost, Jürgen. *Riemannian Geometry and Geometric Analysis*. Springer-Verlag, 2008 (vid. págs. 20, 22-24).

Karcher, H. “Riemannian center of mass and mollifier smoothing”. En: *Communications on Pure and Applied Mathematics* 30.5 (1977), págs. 509-541 (vid. págs. 28, 47).

Lang, Serge. *Fundamentals of Differential Geometry*. Vol. 191. Springer, 1999 (vid. págs. 20, 26, 29).

Lee, John M. *Introduction to Smooth Manifolds*. University of Washinton, Department of Mathematics, 2000 (vid. pág. 20).

Lima, Elon. *Análise Real Volume 2*. Ene. de 2004 (vid. pág. 21).

Minh, H. Quang y Vittorio Murino. *Covariances in Computer Vision and Machine Learning*. Morgan & Claypool Publishers, 2017 (vid. pág. 22).

Moakher, Maher. “A Differential Geometric Approach to the Geometric Mean of Symmetric Positive Definite Matrices”. En: *SIAM J. Matrix Anal. Appl.* 26 (mar. de 2005), págs. 735-747. DOI: 10.1137/S0895479803436937 (vid. págs. 28, 29).

Moakher, Maher y Mourad Zerai. “The Riemannian Geometry of the Space of Positive-Definite Matrices and Its Application to the Regularization of Positive-Definite Matrix-Valued Data”. En: *Mathematical Imaging and Vision* 40 (jun. de 2011), págs. 171-187. DOI: 10.1007/s10851-010-0255-x (vid. pág. 98).

Moreno, Wilson y Fabio Martinez. “Frame-Level Covariance Descriptor for Action Recognition Riemannian Geometry for the statistical analysis of diffusion tensor Data”. En: *Colombian Conference on Computing* (sep. de 2018), págs. 276-290 (vid. pág. 18).

Pedregosa, F. y col. “Scikit-learn: Machine Learning in Python”. En: *Journal of Machine Learning Research* 12 (2011), págs. 2825-2830 (vid. pág. 66).

Pennec, Xavier. “Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements”. En: *Journal of Mathematical Imaging and Vision* 25.1

- (2006). A preliminary appeared as INRIA RR-5093, January 2004., págs. 127-154.  
DOI: 10.1007/s10851-006-6228-4 (vid. págs. 14, 28).
- Pennec, Xavier. “Probabilities and Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements”. En: ene. de 1999, págs. 194-198 (vid. pág. 17).
- Pennec, Xavier, Pierre Fillard y Nicholas Ayache. “A Riemannian Framework for Tensor Computing”. En: *International Journal of Computer Vision* 66.1 (2006), págs. 41-66. DOI: 10.1007/s11263-005-3222-z (vid. págs. 17-19, 25, 28, 40, 41, 47, 48, 85, 98).
- Ryoo, M. S. y J. K. Aggarwal. *UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA)*. 2010 (vid. págs. 14, 16, 61, 69).
- Thomas Fletcher, P y Sarang Joshi. “Riemannian Geometry for the statistical analysis of diffusion tensor Data”. En: *Signal Processing* 87 (feb. de 2007), págs. 250-262.  
DOI: 10.1016/j.sigpro.2005.12.018 (vid. págs. 17, 18, 28, 29, 40, 47, 85).
- Wang, Y., K. Huang y T. Tan. “Human Activity Recognition Based on R Transform”. En: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. 2007, págs. 1-8. DOI: 10.1109/CVPR.2007.383505 (vid. págs. 13, 18).



## ANEXOS

Se presentarán algunos conceptos y resultados teóricos de álgebra, funciones de matrices y geometría utilizados en el trabajo necesarios para una mejor sustentación teórica. Inicialmente se definen las funciones de matrices y algunos resultados para realizar el calculo

$$f_{\Sigma}(V) = \Sigma^{\frac{1}{2}} f(\Sigma^{-\frac{1}{2}} V \Sigma^{-\frac{1}{2}}) \Sigma^{\frac{1}{2}},$$

necesario para poder calcular las funciones riemannianas log y exp, posteriormente se desarrolla el calculo con autovalores generalizados que se menciona en la Sección 4.5, luego se mostrarán varias definiciones y teoremas variacionales de matrices utilizados en la Sección 3 y por último se deducirá cómo calcular la geodésica en la variedad  $S_{++}^n$ .

### Anexo A. Funciones de matrices

El término funciones de matrices surge cuando dada una función escalar  $f$  y una matriz  $A \in \mathbb{C}^{n \times n}$  se quiere calcular  $f(A)$  de forma que  $f(A)$  resulte siendo una matriz de la misma dimensión que  $A$ . Por otra parte si una matriz  $A$  es diagonalizable se puede descomponer de la forma  $A = Z D Z^{-1}$ , donde  $D = \text{diag}(\lambda_i)$  es una matriz diagonal cuyas entradas son los autovalores asociados a  $A$  (si  $A \in S_{++}^n$ , los autovalores son positivos) y las columnas de  $Z$  son los respectivos autovectores de  $A$ . Se calcula entonces  $f(A)$  de la siguiente forma

$$f(A) = Z f(D) Z^{-1} = Z f(\text{diag}(\lambda_i)) Z^{-1}. \quad (23)$$

Para otras alternativas vea <sup>10</sup>.

**Teorema 9** (Corolario 1.34 de <sup>10</sup>). Sean  $A, B \in \mathbb{C}^{n \times n}$  y  $f$  definida en el espectro de

$AB$  y  $BA$ . Entonces

$$Af(BA) = f(AB)A.$$

Este resultado puede ser aplicado a la definición de  $\exp_{\Sigma}(V)$  and  $\log_{\Sigma}(v)$ . En general, dada una matriz  $\Sigma \in \mathbb{R}^{n \times n}$  simétrica definida positiva, se definirá

$$f_{\Sigma}(V) = \Sigma^{\frac{1}{2}} f(\Sigma^{-\frac{1}{2}} V \Sigma^{-\frac{1}{2}}) \Sigma^{\frac{1}{2}}.$$

En general el calculo eficiente de  $f_{\Sigma}$  no es sencillo. Usando el Teorema previo se pueden encontrar varias formas de escribir  $f_{\Sigma}(V)$ .

**Corollary 9.1.** *Asuma que  $\Sigma \in S_{++}^n$  y  $V \in S^n$ . Se tiene que*

1.  $f_{\Sigma}(V) = \Sigma f(\Sigma^{-1}V).$
2.  $f_{\Sigma}(V) = f(V\Sigma^{-1})\Sigma.$

*Demostración.* Tomando  $B = \Sigma^{-\frac{1}{2}}V$  y  $A = \Sigma^{-\frac{1}{2}}$  en el Teorema 9. Se tiene

$$\Sigma^{-\frac{1}{2}} f(\Sigma^{-\frac{1}{2}} V \Sigma^{-\frac{1}{2}}) = f(\Sigma^{-\frac{1}{2}} \Sigma^{-\frac{1}{2}} V) \Sigma^{-\frac{1}{2}}$$

y por tanto

$$\begin{aligned} f_{\Sigma}(V) &= \Sigma^{\frac{1}{2}} f(\Sigma^{-\frac{1}{2}} V \Sigma^{-\frac{1}{2}}) \Sigma^{\frac{1}{2}} \\ &= \Sigma^{\frac{1}{2}} \left[ \Sigma^{\frac{1}{2}} f(\Sigma^{-\frac{1}{2}} \Sigma^{-\frac{1}{2}} V) \Sigma^{-\frac{1}{2}} \right] \Sigma^{\frac{1}{2}} \\ &= \Sigma f(\Sigma^{-1}V) \end{aligned}$$

Analogamente se obtiene 2 tomando  $A = \Sigma^{\frac{1}{2}}$  y  $B = \Sigma^{-\frac{1}{2}}V\Sigma^{-1}$ .

□

## Anexo B. Problema de autovalor generalizado

Dada  $\Sigma$  simétrica definida positiva y  $V$  simétrica, se puede considerar el problema de autovalor generalizado  $V\phi = \lambda\Sigma\phi$  que escrito de forma matricial es

$$\Sigma^{-1}V = QDQ^{-1}, \quad (24)$$

donde  $Q^{-1} = Q^T\Sigma$  pues los autovectores de  $\Sigma^{-1}V$  son ortogonales en el producto interno generado por  $\Sigma$ , es decir  $Q^T\Sigma Q = Id$  con el producto  $\langle u, v \rangle_\Sigma = v^T\Sigma u$ . Entonces 24 puede ser visto como

$$\Sigma^{-1}V = QDQ^T\Sigma, \quad (25)$$

o lo que es lo mismo

$$\Sigma^{-\frac{1}{2}}V\Sigma^{-\frac{1}{2}} = \Sigma^{\frac{1}{2}}QDQ^T\Sigma^{\frac{1}{2}}. \quad (26)$$

Tomando  $\Psi = \Sigma^{\frac{1}{2}}Q$  se tiene que  $\Psi^{-1} = Q^T\Sigma^{\frac{1}{2}}$  ya que  $Q^T\Sigma^{\frac{1}{2}}\Sigma^{\frac{1}{2}}Q = Id$ . Entonces 26 sería

$$\Sigma^{-\frac{1}{2}}V\Sigma^{-\frac{1}{2}} = \Psi D \Psi^{-1}. \quad (27)$$

De (24) y (27) note que los autovalores de  $\Sigma^{-1}V$  y  $\Sigma^{-\frac{1}{2}}V\Sigma^{-\frac{1}{2}}$  son los mismos. Aplicando la función matricial  $f$  en la ecuación (27) se tiene

$$f\left(\Sigma^{-\frac{1}{2}}V\Sigma^{-\frac{1}{2}}\right) = \Psi f(D) \Psi^{-1}, \quad (28)$$

y así

$$\begin{aligned} f_\Sigma(V) &= \Sigma^{\frac{1}{2}}f\left(\Sigma^{-\frac{1}{2}}V\Sigma^{-\frac{1}{2}}\right)\Sigma^{\frac{1}{2}} \\ &= \Sigma Q f(D) Q^T \Sigma, \end{aligned} \quad (29)$$

es decir  $f_{\Sigma}(V)$  puede ser calculado usando el problema de autovalor generalizado  $\Sigma^{-1}V$ . Con el Corolario 9.1 también se puede obtener el anterior resultado, pues usando el hecho de que  $f_{\Sigma}(V) = \Sigma f(\Sigma^{-1}V)$  y la descomposición (24)

$$\begin{aligned} f_{\Sigma}(V) &= \Sigma f(QVQ^{-1}) \\ &= \Sigma Q f(D) Q^{-1} \\ &= \Sigma Q f(D) Q^T \Sigma. \end{aligned}$$

**Teorema 10.** *Los autovalores generalizados de  $f_{\Sigma}(V)$  con respecto a  $\Sigma$  son de la forma  $f(\lambda_i)$  donde  $\lambda_i$  es un autovalor generalizado de  $V$  con respecto a  $\Sigma$ .*

*Demostración.* Los autovalores de  $f_{\Sigma}(V)$  con respecto a  $\Sigma$  son los autovalores de  $\Sigma^{-1}f_{\Sigma}(V)$ , pero usando el Corolario 9.1

$$\begin{aligned} \Sigma^{-1}f_{\Sigma}(V) &= f(\Sigma^{-1}V) \\ &= f(QDQ^{-1}) \\ &= Qf(D)Q^{-1}, \end{aligned}$$

es decir, los autovalores de  $\Sigma^{-1}f_{\Sigma}(V)$  forman la matriz diagonal  $f(D)$  donde  $D$  es una matriz diagonal con los autovalores de  $V$  con respecto a  $\Sigma$ . □

## Anexo C. Definiciones y resultados variacionales en matrices

En la Sección 3 se planteó el problema de optimización de forma matricial que debe resolverse para el cálculo de la media de un conjunto de matrices SPD. Esta subsección es un complemento teórico en el que se mencionarán algunas propiedades de la traza y varias definiciones variacionales para trabajar con matrices.

Siendo  $A \in \mathbb{R}^{n \times n}$  de la forma  $A = [a_{ij}]_{i=1,j=1}^{n,n}$ , se define la traza como

$$\text{tr}(A) = \sum_{i=1}^n a_{ii}.$$

**Teorema 11.** *Dadas dos matrices  $A, B \in \mathbb{R}^{n \times n}$ , se cumple que*

$$\text{tr}(AB) = \text{tr}(BA).$$

*Demostración.* Note que

$$\begin{aligned} \text{tr}(AB) &= \sum_{i=1}^n (AB)_{ii} \\ &= \sum_{i=1}^n \sum_{k=1}^n a_{ik} b_{ki} \\ &= \sum_{k=1}^n \sum_{i=1}^n b_{ik} a_{ik} \\ &= \sum_{k=1}^n (BA)_{kk} \\ &= \text{tr}(BA), \end{aligned}$$

mostrando así que  $\text{tr}(AB) = \text{tr}(BA)$ . □

**Corollary 11.1.** Dadas  $k$  matrices  $A_1, A_2, \dots, A_k \in \mathbb{R}^{n \times n}$ , se cumple que

$$\text{tr}(A_1 A_2 \cdots A_{k-1} A_k) = \text{tr}(A_k A_1 A_2 \cdots A_{k-1}).$$

*Demostración.* Esta invarianza ante permutaciones cíclicas resulta de aplicar el anterior teorema a las matrices  $A_1 A_2 \cdots A_{k-1}$  y  $A_k$ .  $\square$

**Definición 1.** Sea  $M(t) \in \mathbb{R}^{n \times n}$  una matriz para cada  $t \in [a, b]$ , es decir,  $M(t)$  es la matriz  $M(t) = [m_{ij}(t)]_{i=1, j=1}^{n, n}$  donde  $m_{ij} : [a, b] \rightarrow \mathbb{R}$  para todo  $1 \leq i, j \leq n$ . Se define la integral de  $M(t)$  como

$$\int_a^b M(s) ds := \left[ \int_a^b m_{ij}(s) ds \right]_{i=1, j=1}^{n, n}$$

**Teorema 12.** Dada  $M(t) = [m_{ij}(t)]_{i=1, j=1}^{n, n}$  donde  $m_{ij} : [a, b] \rightarrow \mathbb{R}$  para todo  $1 \leq i, j \leq n$ . Se tiene que

$$\text{tr} \left( \int_a^b M(s) ds \right) = \int_a^b \text{tr}(M(s)) ds$$

*Demostración.* Sea  $N = \int_a^b M(s) ds$ .

$$\begin{aligned} \text{tr}(N) &= \sum_{i=1}^n N_{ii} \\ &= \sum_{i=1}^n \int_a^b m_{ii}(s) ds \\ &= \int_a^b \left( \sum_{i=1}^n m_{ii}(s) \right) ds \\ &= \int_a^b \text{tr}(M(s)) ds \end{aligned}$$

$\square$

En el capítulo 3.1 de <sup>10</sup> se define una noción apropiada para la derivada de funciones de matrices que se introducirá a continuación.

**Definición 2.** La derivada de Frechét de una función matricial  $F : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$  en el punto  $X \in \mathbb{R}^{n \times n}$  es una aplicación lineal  $L : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$  que envía  $E$  en  $L(X, E)$ , tal que para todo  $E \in \mathbb{R}^{n \times n}$  se cumple

$$F(X + E) = F(X) + L(X, E) + o(\|E\|),$$

esta aplicación  $L(X, E)$  suele denotarse como  $DF(X, E)$ . En particular si  $E = tV$  se tiene

$$F(X + tV) = F(X) + tL(x, v) + o(t).$$

## Anexo D. Cálculo de geodésicas

Como menciona Pennec en <sup>7</sup> una forma de poder definir una distancia invariante es a través de una métrica riemanniana, es decir, si se tiene una acción  $\star$  cualquiera sobre el espacio  $S_{++}^n$  se quiere que dados dos vectores  $V_1, V_2 \in T_\Sigma S_{++}^n$  se cumpla que

$$\langle V_1, V_2 \rangle_\Sigma = \langle A \star V_1, A \star V_2 \rangle_{A \star \Sigma}.$$

Esto es cierto para  $A = \Sigma^{-\frac{1}{2}}$  mediante la acción  $\star : GL_n^+ \times S_{++}^n \rightarrow S_{++}^n$  donde  $A \star \Sigma = A \Sigma A^T$ . Donde se ve que en este caso la acción es precisamente la métrica riemanniana afín invariante

$$\langle V_1, V_2 \rangle_\Sigma = \langle \Sigma^{-\frac{1}{2}} V_1 \Sigma^{-\frac{1}{2}}, \Sigma^{-\frac{1}{2}} V_2 \Sigma^{-\frac{1}{2}} \rangle_{id}.$$

En el artículo <sup>39</sup> el *Teorema 3* menciona que las geodésicas con la métrica riemanniana afín invariante vienen dadas por  $P(t) = P_0^{\frac{1}{2}} \exp \left( t P_0^{-\frac{1}{2}} S P_0^{-\frac{1}{2}} \right) P_0^{\frac{1}{2}}$  donde  $t \in \mathbb{R}$ ,  $P_0 \in S_{++}^n$  y  $S \in T_{P_0} S_{++}^n$  es el vector que indica la dirección. Moakher y Mourad en este artículo calculan la geodésica como la solución de una ecuación diferencial de segundo orden, sin embargo Pennec con esta métrica invariante y resultados de geometría diferencial menciona que la geodésica que pasa por  $\Sigma \in S_{++}^n$  con vector tangente  $V \in T_\Sigma(S_{++}^n)$  debe tener la forma:

$$\Gamma_{(\Sigma, V)}(t) = \exp(tA) \Sigma \exp(tA)^T \quad \text{donde } V = A \Sigma + \Sigma A^T.$$

En el caso particular en el que  $\Sigma = id$ , una solución simétrica sería  $V = 2A$  y por

---

<sup>39</sup> Maher Moakher y Mourad Zerai. "The Riemannian Geometry of the Space of Positive-Definite Matrices and Its Application to the Regularization of Positive-Definite Matrix-Valued Data". En: *Mathematical Imaging and Vision* 40 (jun. de 2011), págs. 171-187. DOI: 10.1007/s10851-010-0255-x.



tanto

$$\begin{aligned}\Gamma_{(id,V)}(t) &= \exp\left(t\frac{1}{2}V\right)\exp\left(t\frac{1}{2}V\right)^T \\ &= \exp(tV).\end{aligned}$$

La longitud de la geodésica que va desde  $id$  hasta  $\Gamma_{(id,V)}(1) = \Sigma$  sería  $\|V\|$ . Esto es

$$\begin{aligned}\int_0^1 \|\Gamma'(t)\|_{\Gamma(t)}^2 dt &= \int_0^1 \langle \Gamma'(t), \Gamma'(t) \rangle_{\Gamma(t)} dt \\ &= \int_0^1 \langle \Gamma(t)^{-\frac{1}{2}} \Gamma'(t) \Gamma(t)^{-\frac{1}{2}}, \Gamma(t)^{-\frac{1}{2}} \Gamma'(t) \Gamma(t)^{-\frac{1}{2}} \rangle_{id} dt \\ &= \int_0^1 \langle V, V \rangle_{id} dt \\ &= \int_0^1 \|V\|_{id}^2 dt \\ &= \|V\|_{id}^2,\end{aligned}$$

pues

$$\begin{aligned}\Gamma(t)^{-\frac{1}{2}} \Gamma'(t) \Gamma(t)^{-\frac{1}{2}} &= \exp\left(-\frac{1}{2}tV\right) [\exp(tV)V] \exp\left(-\frac{1}{2}tV\right) \\ &= \exp\left(\frac{1}{2}tV\right) V \exp\left(-\frac{1}{2}tV\right) \\ &= V,\end{aligned}\tag{30}$$

y por el teorema <sup>10</sup> se tiene que  $\exp\left(-\frac{1}{2}tV\right) V = V \exp\left(-\frac{1}{2}tV\right)$  y por tanto se tiene la igualdad 30. Ahora por la invarianza de la métrica se puede obtener  $\Gamma_{(\Sigma,V)}(t)$ , pues note que

$$\begin{aligned}\Gamma_{(\Sigma,V)}(t) &= \Sigma^{\frac{1}{2}} \star \Gamma_{(id, \Sigma^{-\frac{1}{2}} \star V)}(t) \\ &= \Sigma^{\frac{1}{2}} \star \exp\left(t \Sigma^{-\frac{1}{2}} \star V\right) \\ &= \Sigma^{\frac{1}{2}} \exp\left(t \Sigma^{-\frac{1}{2}} V \Sigma^{-\frac{1}{2}}\right) \Sigma^{\frac{1}{2}}.\end{aligned}$$

Calculando así lo que se quería, la geodésica que pasa por  $\Sigma$  con vector tangente  $V$ .