



Article

The Optimization of Numerical Algorithm Parameters with a Genetic Algorithm to Animate Letters of the Sign Alphabet

Sergio Hernandez-Mendez ¹ , Carlos Hernández-Mejía ² , Delia Torres-Muñoz ³
and Carolina Maldonado-Mendez ^{4,*}

¹ Artificial Intelligence Research Institute, Universidad Veracruzana, Xalapa 91097, Mexico; sergihernandez@uv.mx

² Tecnológico Nacional de México/ITS de Misantla, Misantla 93850, Mexico; cmahernandez@gmail.com

³ Instituto Tecnológico Superior de San Martín Texmelucan, San Martín Texmelucan, Puebla 74120, México; delets@gmail.com

⁴ Ingeniería en Computación, Instituto de Agroingeniería, Universidad del Papaloapan Loma Bonita, Loma Bonita 68400, México

* Correspondence: carolinamaldonadomendez@gmail.com

Abstract: At present, the development of animation-based works for human–computer interaction applications has increased. To generate animations, actions are pre-recorded and animation flows are configured. In this research, from two images of letters of the sign language alphabet, intermediate frames were generated using a numerical traced algorithm based on homotopy. The parameters of a homotopy curve were optimized with a genetic algorithm to generate intermediate frames. In the experiments performed, sequences where a person executes pairs of letters in sign language were recorded and animations of the same pairs of letters were generated with the proposed method. Subsequently, the similarity of the real sequences to the animations was measured using Dynamic Time Wrapping. The results obtained show that the images obtained are consistent with their execution by a person. Animation files between sign pairs were created from sign images, with each file weighing an average of 18.3 KB. By having sequences between pairs of letters it is possible to animate words and sentences. The animations generated by this homotopy-based animation method optimized with a genetic algorithm can be used in various deaf interaction applications to provide assistance. From several pairs of letters a file base was generated using the animations between pairs of letters; with these files you can create animations of words and sentences.

Keywords: genetic algorithm; Dynamic Time Wrapping; homotopy; numerical traced algorithm; sign animation



Citation: Hernandez-Mendez, S.; Hernández-Mejía, C.; Torres-Muñoz, D.; Maldonado-Mendez, C. The Optimization of Numerical Algorithm Parameters with a Genetic Algorithm to Animate Letters of the Sign Alphabet. *Multimodal Technol. Interact.* **2024**, *8*, 61. <https://doi.org/10.3390/mti8070061>

Academic Editors: Alexey Karpov and Richard Jiang

Received: 28 February 2024

Revised: 26 June 2024

Accepted: 4 July 2024

Published: 10 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Deaf people communicate through sign language, which consists of a series of gestural signs articulated with the hands and accompanied by facial expressions, intentional gaze and body movement, endowed with linguistic function [1]. According to data from the World Health Organization (WHO), 1.5 billion people live with some degree of hearing loss. According to the National Institute of Statistics and Geography (INEGI), in Mexico, 1.3% of the population aged three years or older cannot hear. Mexican Sign Language (MSL) is officially recognized as a national language and is part of the linguistic heritage of the Mexican nation [2,3]. On the other hand, this form of communication has not yet been disseminated throughout the entire population of Mexico, since there are less than 500,000 people who communicate with this language [4]. Therefore, it is important to develop a tool for deaf people to communicate, otherwise this limits their development, access to information, social inclusion and participation in everyday life [5].

In recent years, several works have been conducted on hand gesture recognition based on visual information [6–8] in order to develop human–computer applications. In these

works [9], machine learning algorithms are used to recognize static and moving signs in various applications of human–computer interactions, such as controlling a robot. However, there are fewer works in which an animation or avatar is developed to communicate with deaf people. The authors of [10] implemented an avatar that performs signs; motion capture (MoCap) was employed to capture body, limb and head movements in 3D space, consequently, these movements had to be corrected during the post-production process and additional animations had to be made by moving each finger bone to the required sign position.

In [11], facial animations are created from two images with a numerical traced algorithm. Their methodology uses the homotopy curve path to generate intermediate frames for different λ values. The intermediate frames are the deformations from the initial image to the final image. A hyperspherical tracking method establishes deformations with visually consistent and smooth changes. In the experiments, the radius of the hypersphere is constant. This method showed good results in the examples presented.

In this research, we are interested in creating animations between pairs of sign language letters using the method proposed in [11]. The original contribution of this research is to use a genetic algorithm [12] to optimize the radius and its increment to plot the homotopy curve of the numerical traced algorithm to calculate the animation between pairs of letters of the sign language alphabet. In this way, from a base of images of letters of the alphabet (<https://acortar.link/1KWigu> (accessed on 8 February 2024)), animations can be generated to spell words in sign language. One of the advantages of generating animations with this algorithm is that once an animation is generated between two pairs of letters, such as (a,b), this same animation can be used for pairs of gestures (b,a) that are to be executed in the reverse order. The files containing the animations between pairs of letters weigh on average 18.3 KB, and they are executed in Matlab (R2024a).

The manuscript is organized as follows: Section 2 describes the hand gesture animation system. In Section 3, the homotopy-based animation method is introduced. In Section 4, optimization with a genetic algorithm is explained. After that, in Section 5, the experimental design and the obtained results are presented. In Section 6, a brief discussion is presented. Finally, Section 7 summarizes the findings of this research and sets up future work.

2. The Hand Gesture Animation System

The hand gesture animation system proposed in this research consists of the following stages (Figure 1):

- **Hand joint position detection:** Google's Mediapipe library is used to recognize the positions of the hand joints in the starting and the ending frame. The Mediapipe library is used in hand tracking work [13], while the MediaPipe Hands library provides only a 2.5D pose estimation. In [14], a simple calibration and the concept of perspective projection were proposed to obtain the 3D position of the hands relative to a smartphone. Figure 2 shows the 21 joints that are detected in a hand with Google's Mediapipe library. Each joint has three coordinates (j_x, j_y, j_z) . The library was configured to detect the hands and their joints with a confidence of 0.5.
- **Calculate transitions:** the initial and end image of the joints' position are used with the proposed method in [11] and optimized with a genetic algorithm to calculate the transition images. The lower part of Figure 1 shows the 15 transitions calculated from a given pair of images. These transitions are the animation between pairs of signing gestures. Matlab was used to implement the numerical traced algorithm and its parameter optimization with a genetic algorithm.

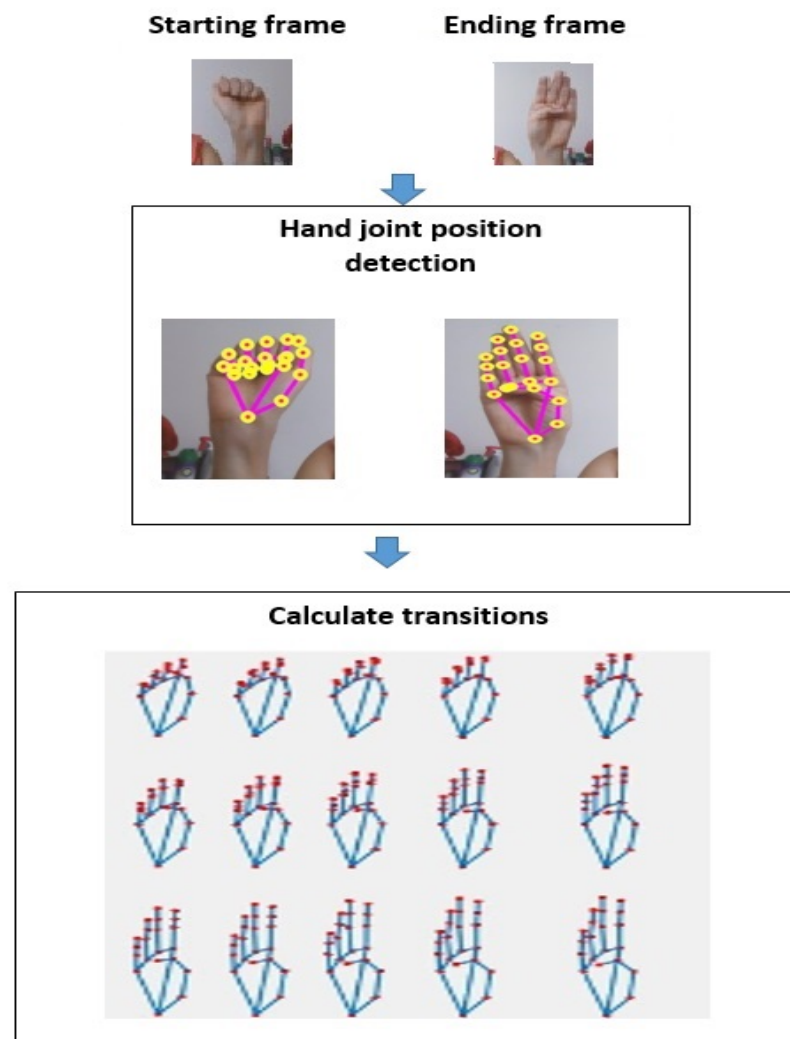


Figure 1. First, the position of the hand joints in each image is obtained. Then, the transitions between the initial and the final hand signal are calculated. In the end, 15 calculated transitions are shown, which are the animation between the signs given to the hand gesture animation system.



Figure 2. The black dots correspond to the joints detected in a hand using Google's Mediapipe library.

3. Homotopy-Based Animation Method

Homotopy continuation methods [15] are based on the insertion of a homotopy parameter λ into non-linear algebraic equations in order to obtain a continuous deformation

from a trivial state to a non-linear state. In (1) n is the number of variables and $\mathbf{x}$ is the set of variables from the system of equations

$$\begin{aligned} \mathbf{H} : \mathbf{R}^{n+1} &\rightarrow \mathbf{R}^n, \\ \mathbf{x} &\in \mathbf{R}^n, \\ \lambda &\in [0, 1] \end{aligned} \quad (1)$$

Transitions between the starting and ending hand gesture are calculated by applying the homotopy-based animation method (HAM) explained in [11]. Following the same notation, the initial hand gesture is named G1 and the final hand gesture is named G2. At the stage of hand joint position detection (Figure 1), each gesture is stored in a matrix of 21 rows and 3 columns, since 21 joints with 3 components are detected in each hand (see the first column of Table 1). In (2), the initial G1 and ending G2 gesture hands are introduced to the system of equations:

$$H(\mathbf{x}, \lambda) = \lambda * G_2(\mathbf{y}) + (1 - \lambda) * G_1(\mathbf{x}) \quad (2)$$

where λ represents the homotopy parameter and G1 and G2 are $\mathbf{Ax} = \mathbf{B}$ and $\mathbf{Cy} = \mathbf{D}$ system equations respectively. Each joint corresponds to variables $x_1, x_2, x_3, \dots, x_{63}$ for the initial hand gesture (second column in Table 1) and to $y_1, y_2, y_3, \dots, y_{63}$ for the end hand gesture (third column in Table 1).

Table 1. Mapping of joint position to $\mathbf{x}$ and $\mathbf{y}$ variables .

Joint	Variable in G1	Variable in G2
(j_{1x}, j_{1y}, j_{1z})	x_1, x_2, x_3	y_1, y_2, y_3
(j_{2x}, j_{2y}, j_{2z})	x_4, x_5, x_6	y_4, y_5, y_6
(j_{3x}, j_{3y}, j_{3z})	x_7, x_8, x_9	y_7, y_8, y_9
.	.	.
.	.	.
.	.	.
$(j_{19x}, j_{19y}, j_{19z})$	x_{55}, x_{56}, x_{57}	y_{55}, y_{56}, y_{57}
$(j_{20x}, j_{20y}, j_{20z})$	x_{58}, x_{59}, x_{60}	y_{58}, y_{59}, y_{60}
$(j_{21x}, j_{21y}, j_{21z})$	x_{61}, x_{62}, x_{63}	y_{61}, y_{62}, y_{63}

According to $\mathbf{Ax} = \mathbf{B}$ and $\mathbf{Cy} = \mathbf{D}$, the starting and ending hand gesture can be established as follows:

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ . \\ . \\ . \\ b_{63} \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \dots a_{1,63} \\ a_{2,1} & a_{2,2} & a_{2,3} \dots a_{2,63} \\ a_{3,1} & a_{3,2} & a_{3,3} \dots a_{3,63} \\ . & . & . \\ . & . & . \\ . & . & . \\ a_{63,1} & a_{63,2} & a_{63,3} \dots a_{63,63} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ x_{63} \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ . \\ . \\ . \\ d_{63} \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \dots c_{1,63} \\ c_{2,1} & c_{2,2} & c_{2,3} \dots c_{2,63} \\ c_{3,1} & c_{3,2} & c_{3,3} \dots c_{3,63} \\ . & . & . \\ . & . & . \\ . & . & . \\ c_{63,1} & c_{63,2} & c_{63,3} \dots c_{63,63} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ . \\ . \\ . \\ y_{63} \end{bmatrix} \quad (4)$$

The systems of the equations shown in (3) and (4) are substituted into (2) to obtain a global system of equations that contains a combination of the systems from the starting and

ending hand gesture in order to create the animation. To achieve deformations or transition from the initial gesture hand G_1 when $\lambda = 0$, to the end gesture hand G_2 , where $\lambda = 1$, it is necessary to track the homotopic curve using a numerical traced algorithm. For this purpose, the hypersphere equation [16] is introduced.

$$(x_1 - C_1)^2 + (x_2 - C_2)^2 + \dots + (x_{64} - C_{64})^2 - r^2 = 0 \quad (5)$$

where x_{64} is the value of λ in each transition. To start the tracing of the curve [17], the value of λ is 0; $x_1, x_2, x_3, \dots, x_{63}$ are the dimensions of the hypersphere; $C_1, C_2, C_3, \dots, C_{64}$ are the coordinates of the center of the hypersphere; and r is the radius of the hypersphere. Therefore, the system of equations to be solved to calculate the transitions from a starting hand gesture to a final hand gesture contains (2) and (5).

The numerical traced algorithm calculates the transitions between the hand gestures G_1 and G_2 as follows:

1. Matrix **A** and **C** are created with random values; for this research **A** and **C** are equal to simplify the calculations.
2. Matrix **B** is calculated using the values of the initial hand gesture joints G_1 and matrix **A**. Matrix **D** is calculated using the values of the joints of the end hand gesture G_2 and matrix **A**. **B** and **D** are kept constant during the execution of the algorithm.
3. Since G_1 is $Ax = B$ and G_2 is $Ay = D$, and thus G_1 is $Ax - B$ and G_2 is $Ay - D$, these equations are substituted into (2) to obtain (6).

$$H(x, \lambda) = \lambda * (Ay - D) + (1 - \lambda) * (Ax - B) \quad (6)$$

4. In (6), x and y correspond to the joint positions in G_1 and G_2 , respectively, and both sets of variables correspond to the same joint positions in the intermediate gestures of the hand. Therefore, x and y correspond to the same joint, and thus the variable y is changed to x . Then, to calculate the intermediate transitions, (6) is changed as follows:

$$H(x, \lambda) = \lambda * (Ax - D) + (1 - \lambda) * (Ax - B) \quad (7)$$

5. Thus, solving the system of Equations (8), x contains the transitions needed to obtain animations between pairs of hand gestures.

$$H(x, \lambda) = \lambda * (Ax - D) + (1 - \lambda) * (Ax - B) \quad (8)$$

$$(x_1 - C_1)^2 + (x_2 - C_2)^2 + \dots + (x_{63} - C_{63})^2 + (\lambda - 0)^2 - r^2 = 0$$

6. The centers $C_1, C_2, C_3, \dots, C_{63}$ of the hypersphere are substituted by the values of G_1 and C_{64} is equal to the initial value of λ , which has a value of 0. A value is assigned to r . The system of Equations (8) is solved iteratively with the Newton–Raphson method [18].

Figure 3 shows the hand gestures corresponding to the letters a, b, c and d. The transitions between pairs of hand gestures (a,b), (b,c), (d,c) and (b,d) were calculated. For each pair of gestures the numerical traced algorithm was run 7 times, and the value of the radius r was set as shown in Table 2.

Table 2. Radius value set for each run.

Number of Run	r
1	0.1
2	0.4
3	0.7
4	1.0
5	1.3

Table 2. Cont.

Number of Run	r
6	1.6
7	1.9

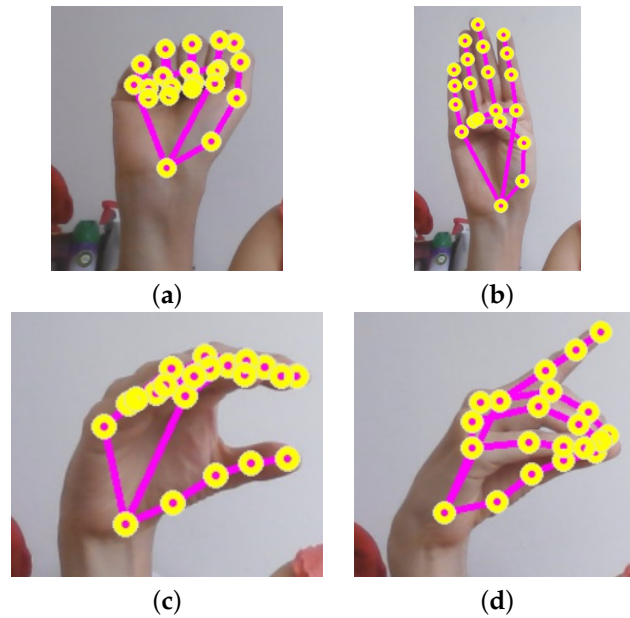


Figure 3. Images of the hand gestures corresponding to the letters (a–d) are shown. The detected joints are also shown.

Figure 4 shows a graph for each pair of the following hand gestures: (a,b), (b,c), (d,c) and (b,d).

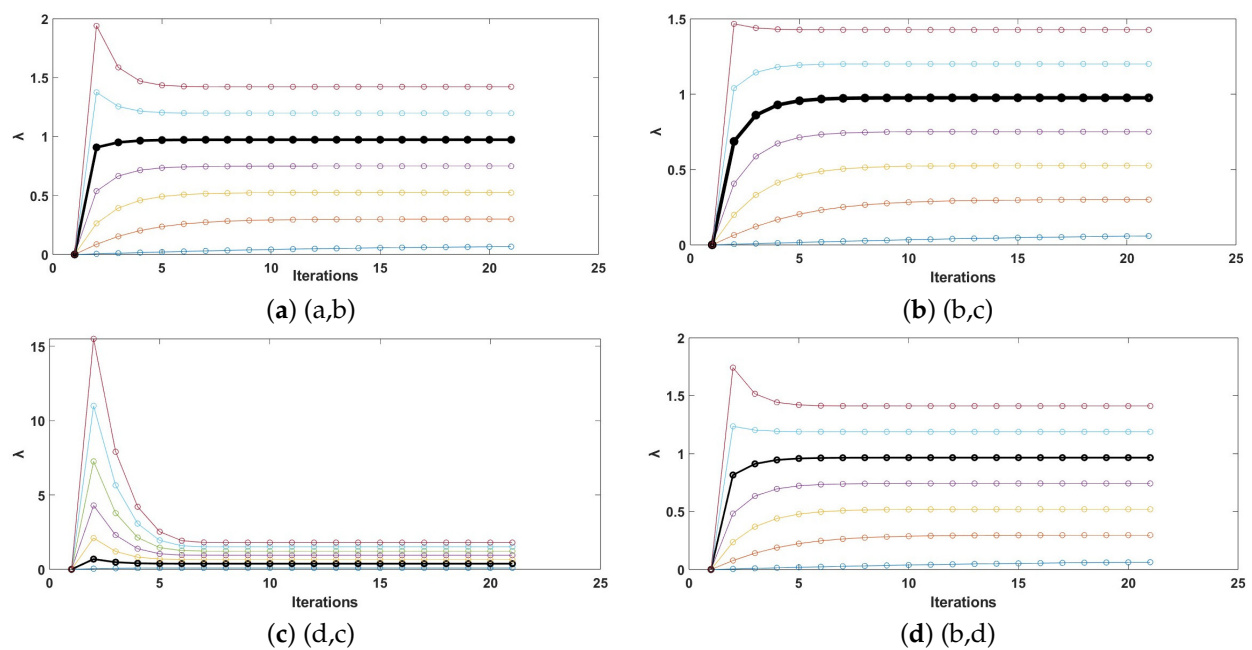


Figure 4. Each graph shows the execution of the numerical traced algorithm for 7 different values of the hypersphere radius for each line to calculate the transitions between pairs of letters. The best execution is the one shown in black, in which the λ starts at a value of zero and reaches a value close to 1. The circles show the calculated transitions.

Each graph shows the 7 runs of the numerical traced algorithm, with each line corresponding to an animation. The x -axis shows the iterations executed to solve the system of Equations (8) and the y -axis shows the calculated values of λ corresponding to each iteration. In each graph it is observed that the lines start at a λ equal to 0, which corresponds to G1. As the iterations are executed to solve the system of equations, the value of λ must increase, and then, according to (2), when λ is equal to 1 the algorithm calculated the transition from G1 to G2 successfully.

Circles in the transition's line indicate the transition gestures that were calculated for each value of r . In each graph, the line that reached a value of λ close to 1 is highlighted in black. For λ in the interval $[0,1]$, it is observed from the transition line in black that for (a,b) it has 5 circles, while it has 7 circles for (b,c), 2 circles for (d,c) and 5 circles for (b,d). No transitions were created for (d,c), since the solution to the system of equations is the initial gesture G1 when λ is equal to 0 and the final gesture G2 when λ is equal to 1 and, in the next iteration, it is observed that the value of λ decreases. Figure 5 shows the animations created for each letter pair (a,b), (b,c), (d,c) and (b,d).

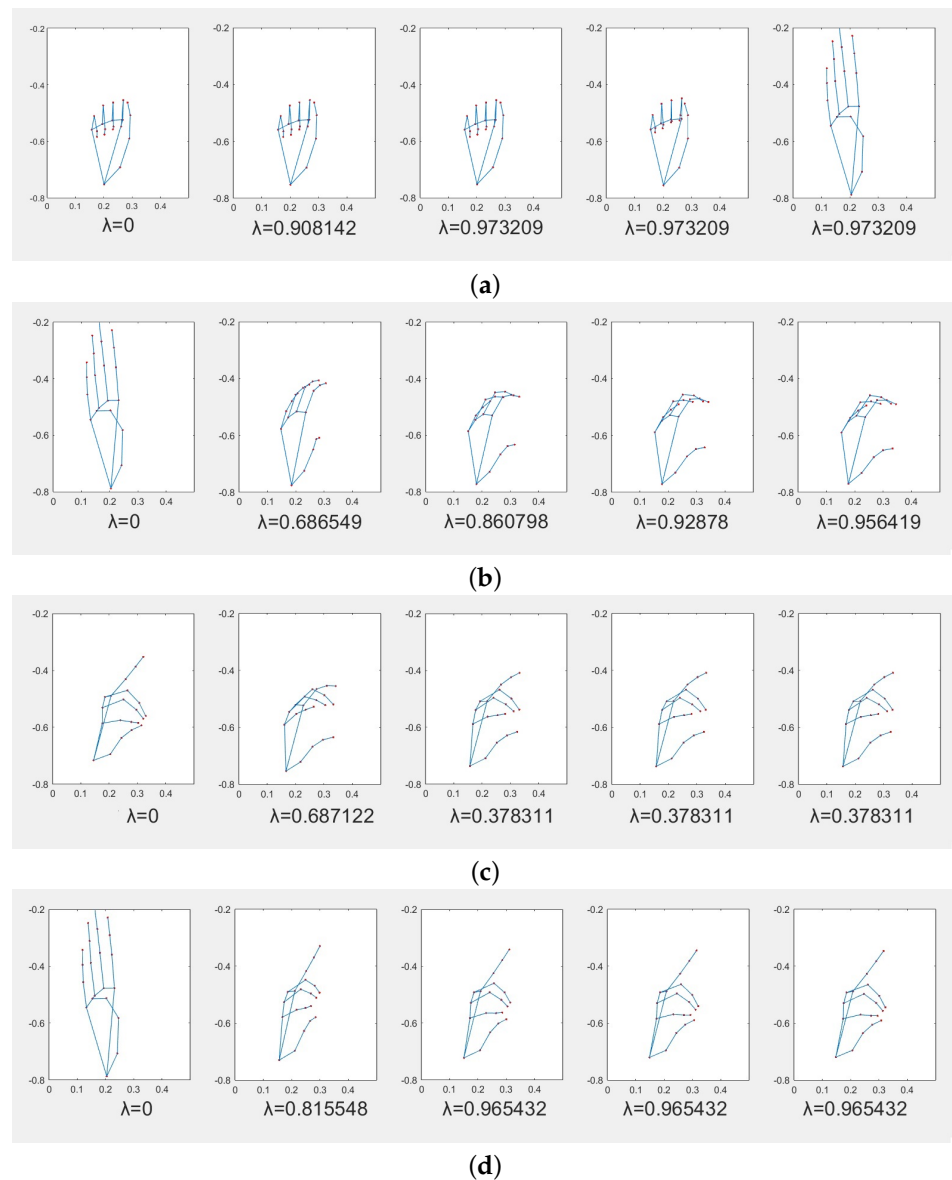


Figure 5. The animation for a pair of letters is shown in each line: (a) animation of the letters (a,b), (b) animation of the letters (b,c), (c) animation of the letters (d,c) and (d) animation of the letters (b,d). In each transition the λ value corresponding to the calculated transition is shown.

According to [19] if the radius of the hypersphere varies, more transitions can be calculated. To prove the above, the letters (d,c) were chosen, because only two transitions were obtained with a fixed radius value in each run. In each run, the value of the radius was increased when it was observed that there was no change in the value of λ between the current and previous iteration. Several tests were performed, and one of the best results is shown in Figure 6. Five transition lines are shown with initial radius values of 0.05, 0.1, 0.15, 0.20 and 0.25, respectively; in each run the radius increment was set to a value of 0.05, and all runs reached the value of λ equal to 1. The line highlighted in black corresponds to an initial radius of 0.1 and has 20 circles, which means that the animation has 20 transitions.

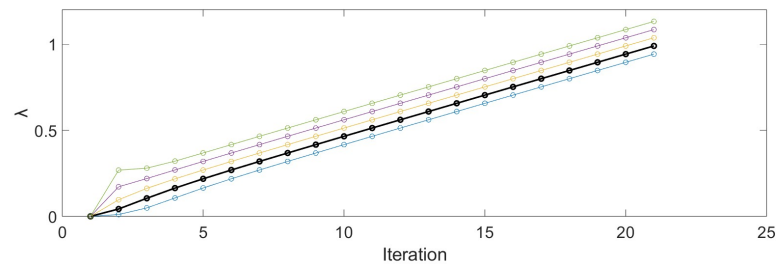


Figure 6. The graph shows the execution of the numerical traced algorithm when using 5 different values of hypersphere radius to calculate the transitions between pairs of letters (d,c). In each run the radius increment was set to a value of 0.05; all runs reached a value of λ equal to 1.

Figure 7 shows the 20 transitions calculated for the letters (d,c) using an initial radius of 0.1 that increased by a value of 0.05.

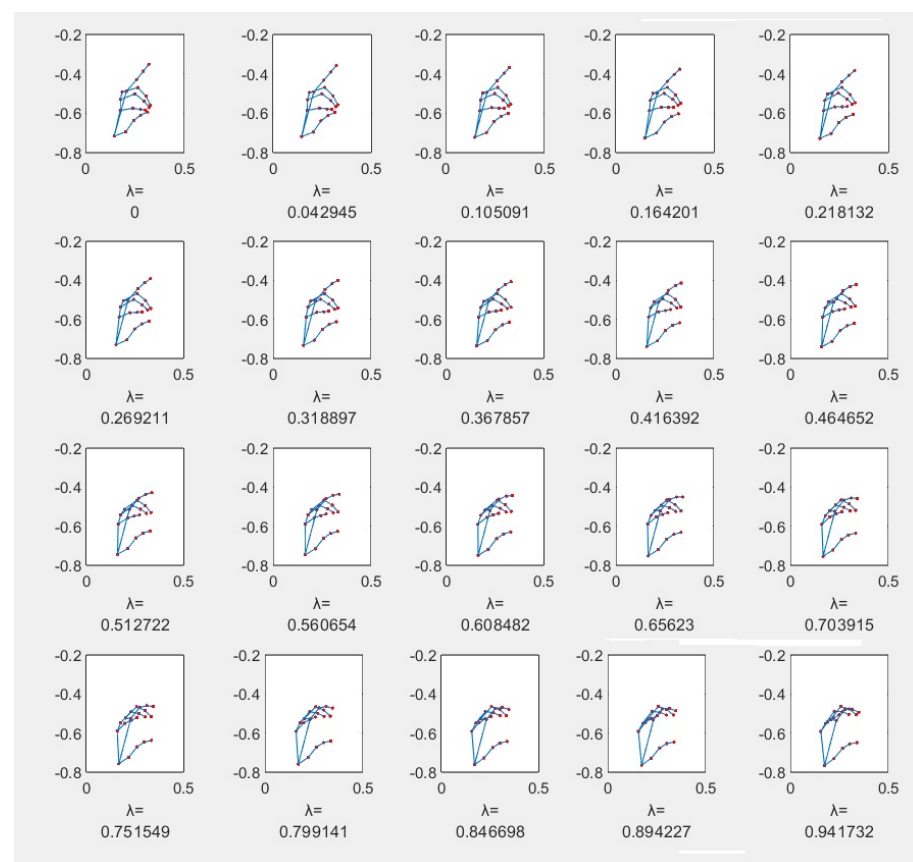


Figure 7. The 20 transitions calculated for the letters (d,c) are shown. The initial value of the radius is 0.1 and it increased by a value of 0.05. Each image shows the λ value corresponding to the calculated transition.

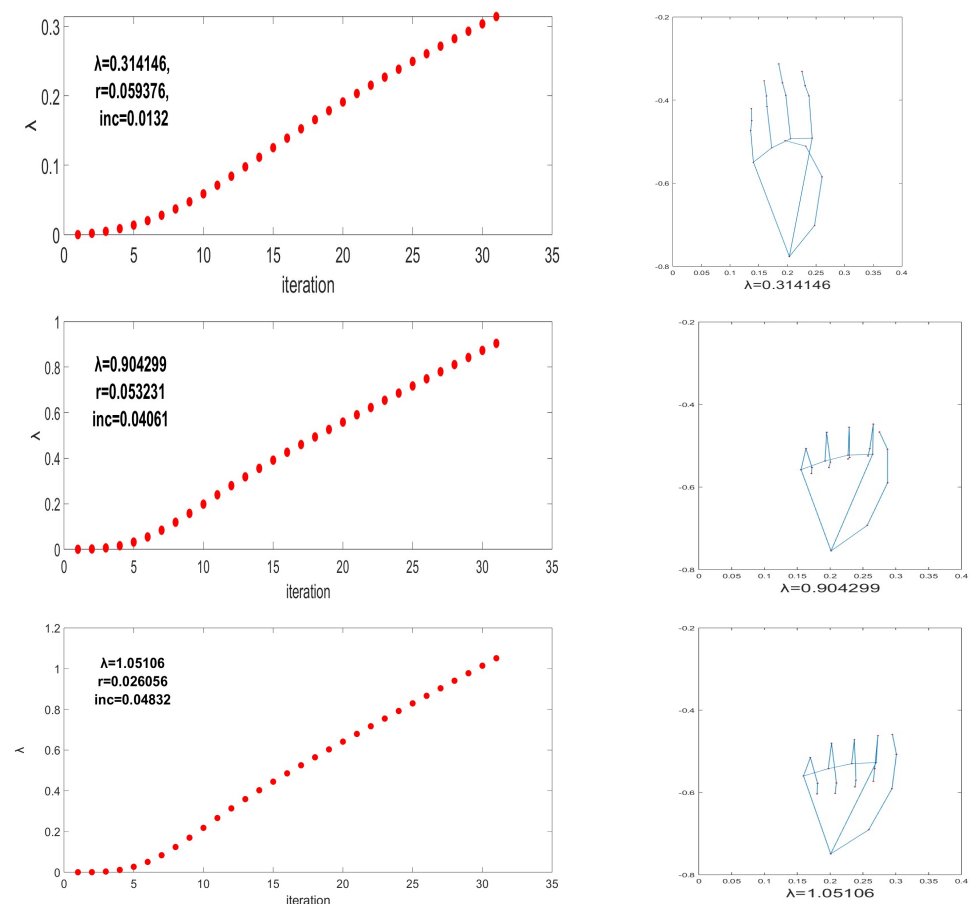


Figure 8. The execution of the numerical traced algorithm for 3 individuals is shown. The left column shows the value of λ for each individual as the 30 iterations are run and the right column shows the final gesture that was calculated for each individual.

5. Experiments and Results

The experiments designed to evaluate the animations created with the numerical traced algorithm optimized with a GA are as follows:

- Three videos were recorded in which a person spells the following pairs of letters: (h,o), (o,l) and (l,a).
- With the Mediapipe library and python, the positions of the joints were obtained and recorded in a .txt file. The file structure is as follows: the first column corresponds to the x-coordinate, the second to the y-coordinate and the third to the z-coordinate of the joints; the first 21 rows correspond to the 21 joints in the first frame, the next 21 rows to the second frame and so on. The videos, text files and matlab program that show the animation have been uploaded to the following link <https://acortar.link/YI1ajV> (accessed on 8 February 2024).
- The positions of the joints in the first and last frame of each .txt file were used to create the animation with the numerical traced algorithm optimized with a GA. The GA was run 30 times (10 for each pair of letters). Table 4 shows the statistical results. Figure 9 shows the execution of the three best individuals, one for calculating the animation of (h,o), another for (o,l) and the last one for (l,a).
- The similarity between the animations created with the numerical traced algorithm and the recorded sequence was measured using Dynamic Time Warping (DTW) [23]. DTW [24,25] is useful because lets us to compare time series with different numbers of frames.

Table 4. GA's statistical results.

Metric	Value	Individual
Mean	0.7471	
Median	0.8489	
Standard Deviation	0.2494	
Best Fit	1.0732	$r = 0.0542$ inc = 0.0367
Worst Fit	0.1867	$r = 0.0464$ inc = 0.0059

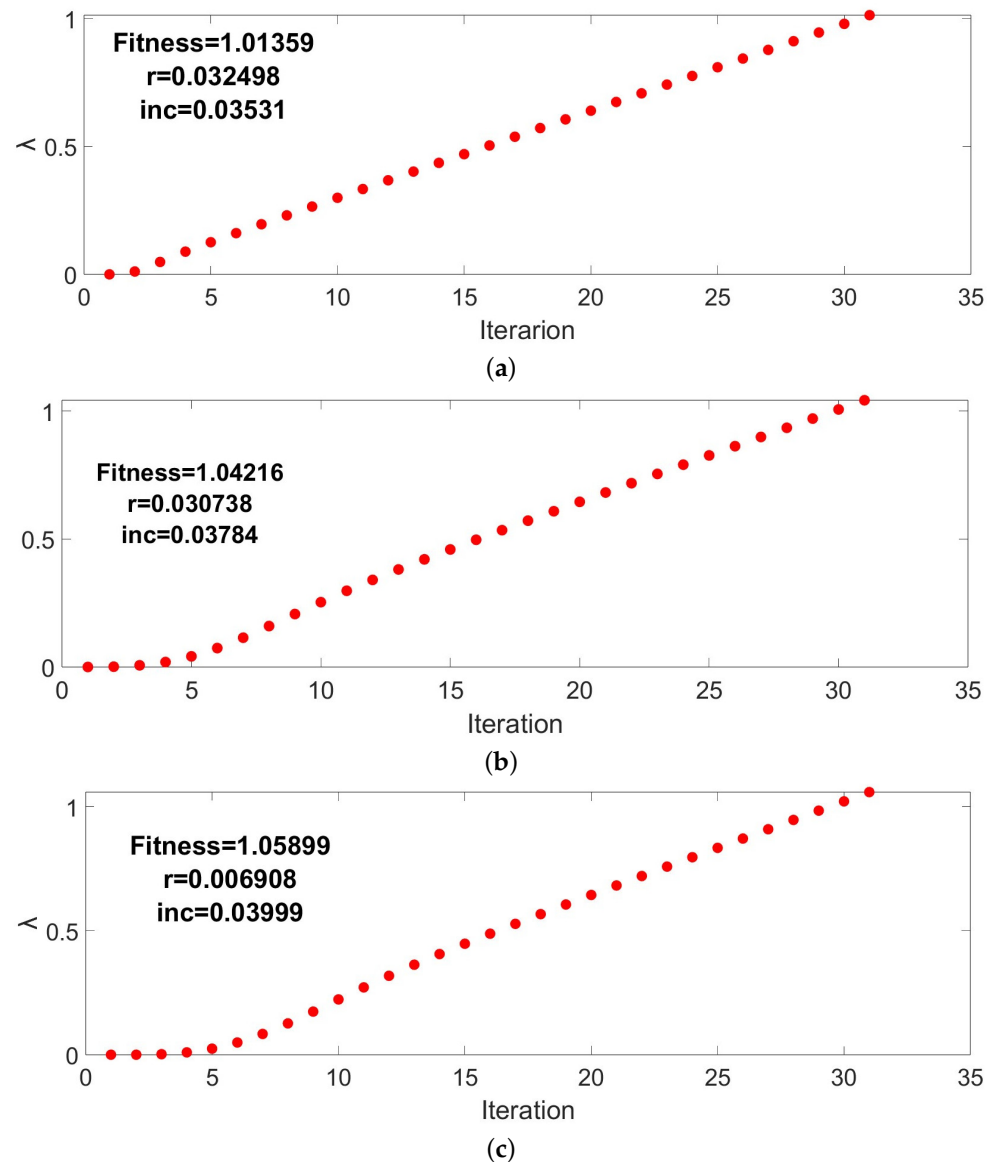
**Figure 9.** Each line shows the performance of the 3 best individuals, (a) one for calculating the animation of (h,o), (b) another for (o,l) and (c) the last one for (l,a).

Table 5 shows the similarity value between the recordings made and the animations created. The diagonal corresponds to the similarity when comparing the recording of the pair of letters to the corresponding animation. For each row, the cells on the diagonal have the smallest value, so the recordings and their corresponding animations are more similar when compared to themselves and not others. Table 6 shows the similarity value between the recordings and Table 7 shows the similarity value between the animations.

Table 5. Similarity between animations and real sequences.

		Animations		
		ho	ol	la
Real	ho	24.78	43.90	82.92
	ol	41.07	6.94	43.22
	la	68.68	55.91	7.10

When comparing the real sequences and the animations there is a greater difference between the sequences (h,o) and (l,a), in second place are the sequences (o,l) and (l,a) and in third place are the sequences (h,o) and (o,l), so the similarity measures between the real recordings and the simulations maintain the same order of similarity.

Table 6. Similarity between real sequences.

		ho	Real ol	la
Real	ho	0	58.96	86.74
	ol	58.96	0	65.22
	la	86.74	65.22	0

Table 7. Similarity between animations

		ho	Animation ol	la
Animation	ho	0	10.41	15.87
	ol	10.41	0	12.56
	la	15.87	12.56	0

For the last experiment, for each of the images that have been uploaded to the following link <https://acortar.link/1KWigu>, accessed on 8 February 2024, the positions (x,y,x) of 21 joints were obtained and stored in a .txt file. Subsequently, 156 animations, calculated using the numerical algorithm optimized by a GA from pairs of gestures taken from the 20 .txt files containing the positions (x,y,z) of the 21 hand translations, were loaded into the animation folder. When creating the animations we realized that having the animation of, for example, (b,c) means that it is not necessary to create a file for the animation (c,b); we can just run the animation (b,c) in reverse order. In this way, it is not necessary to record animations between all pairs of letters. A file that uses 30 frames to create an animation weighs on average 18.3 KB.

For the 156 animations created, it was measured whether the last frame corresponds to the letter indicated in the sequence. For example, in the sequence “lt” we want to know whether the last frame corresponds to the letter t. For the 156 sequences that were created, we compared, using Euclidean distances, the position of the joints of the last frame in the sequence with the positions of the joints of the 20 gestures a, b, c, d, e, f, g, h, i, l, m, n, o, p, r, s, t, u, v and w. The gesture is identified as the one that is closest in distance. Table 8 shows the result of this classification. The first column shows the final letter of an animation, and the second column shows the number of sequences that end with the letter indicated in the first column and are correctly classified, while the third column shows that only two sequences that end with the letter v were confused with the letter u. The accuracy of creating animations between two pairs of letters in which the final letter is the desired letter is 98.8%.

Finally, in the animation folder, two files—phrase1.m and phrase2.m—were uploaded to show the animation of the phrases “We eat some bananas” and “The table is big”. In these

files it was observed that to spell the word banana, the animation of “an” was used to spell “an” and “na”; in the program, the order of the execution of this sequence is indicated.

Table 8. Classification results.

Final Sign	Correctly Classified	Misclassified
a	7	
b	2	
c	4	
d	5	
e	3	
f	0	
g	15	
h	9	
i	7	
l	7	
m	17	
n	15	
o	0	
p	15	
r	3	
s	11	
t	16	
u	7	
v	5	2
w	6	

6. Discussion

Using the numerical traced algorithm proposed in [11], gesture transitions between pairs of letters of the sign alphabet were calculated. The transitions have an associated λ value in the interval of [0,1]. The calculated transitions are smooth changes from an initial gesture G1 to a final gesture G2. Calculating the final gesture and the number of transitions depends on assigning values to the radius of a hypersphere and increasing that radius. Figure 5c shows that, with a fixed radius value in all iterations, two transitions were calculated for the letter pair (d,c) that had different values of λ in [0,1]; subsequently, by increasing the radius, 20 transitions were created (Figure 7).

In this research, we proposed the use of a GA to set the value of the radius and its increment to calculate the animation between pairs of letters. For each individual, the numerical algorithm was executed 30 times and the best individual had a fitness of λ equal to 1 at the end of the iterations. In this manner, the best individual must contain at least 30 different gestures. To decrease the number of gestures in the animation, fewer iterations must be performed, and to increase the number of gestures, the number of iterations must be increased. DTW was used to measure the similarity between the animations created between three pairs of letters and sequences recorded by a person; the distance between the recordings and the animations corresponding to the same pairs of letters is smaller, so the animations created can be used as patterns for dynamic sign recognition.

7. Conclusions

In this research, the parameters of the radius and its increment were optimized to obtain animations between pairs of letters of the sign alphabet using the numerical traced algorithm presented in [11]. We performed experiments with the proposed method and observed the following: a value is assigned to the radius of the hypersphere, and the number of intermediate images calculated depends on this radius value. There is no guarantee that a final image will be calculated [19]; better results are obtained if the value of this radius is increased to plot homotopy curves while calculating the deformations between the initial and the final image. The number of transitions can be changed by changing the number of iterations that the numerical algorithm executes for each individual.

The animations created with the proposed optimization were compared with the real recordings and were most similar to their corresponding recordings. With the proposal made in this research, it is concluded that animations can be generated between pairs of sign language letters to implement applications that communicate with and provide assistance to deaf people.

From 20 gestures of the sign alphabet, pairs of animations were created to form sentences. The advantage is that each animation and three pairs of letters weighs 18.3 KB and that the same animation, such as (a,b), can be used to execute its inverse animation, (b,a); only the direction in which the animation is executed is changed. This can be seen in the files loaded in the link <https://acortar.link/1KWigu>, accessed on 8 February 2024, which makes it easy to create animations of words and phrases.

Future work based on this research should focus on implementing these animations in avatars that are controlled by joint movement. Additionally, in [26], actions were recognized from a set of key poses and using DTW. The animations calculated by the method proposed in this research can be used as key poses to recognize dynamic sign language.

Author Contributions: Conceptualization, S.H.-M., D.T.-M. and C.M.-M.; Formal analysis, S.H.-M., C.H.-M., D.T.-M. and C.M.-M.; Investigation, S.H.-M., C.H.-M., D.T.-M. and C.M.-M.; Methodology, S.H.-M., C.H.-M., D.T.-M. and C.M.-M.; Resources, S.H.-M. and C.M.-M.; Writing—original draft, S.H.-M., C.H.-M., D.T.-M. and C.M.-M.; Writing—review and editing, S.H.-M., C.H.-M., D.T.-M. and C.M.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable, as no participants involved.

Informed Consent Statement: Informed consent was obtained from all subjects who were shown in the figures.

Data Availability Statement: Publicly available datasets were analyzed in this study. Their data can be found here: <https://acortar.link/YI1ajV> (accessed on 8 February 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. de Fleischmann, M.E.S.; Pérez, R.G. *Manos con Voz*; Consejo Nacional para Prevenir la Discriminación: Ciudad de Mexico, Mexico, 2011.
2. de Fleischmann, M.E.S. *Diccionario de Lenguaje de Señas Mexicana*; Editorial Trillas: Ciudad de Mexico, Mexico, 2021.
3. Trujillo-Romero, F.; García-Bautista, G. Mexican Sign Language corpus: Towards an automatic translator. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2023**, *22*, 1–24. [\[CrossRef\]](#)
4. Moreno Vite, I.; Martínez Olvera, W.; Martínez Cervantes, Y.; González Miy, D. Reflexiones sobre la primera etapa del programa Modelos Lingüísticos Sordos. *Rev. Iberoam. Educ.* **2022**, *89*, 93–110. [\[CrossRef\]](#)
5. Fernandes, J.K.; Myers, S.S. Inclusive deaf studies: Barriers and pathways. *J. Deaf. Stud. Deaf. Educ.* **2010**, *15*, 17–29. [\[CrossRef\]](#) [\[PubMed\]](#)
6. De Smedt, Q.; Wannous, H.; Vandeborre, J.P. Skeleton-based dynamic hand gesture recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 11–12 June 2016; pp. 1–9.
7. Tan, C.K.; Lim, K.M.; Lee, C.P.; Chang, R.K.Y.; Alqahtani, A. SDViT: Stacking of Distilled Vision Transformers for Hand Gesture Recognition. *Appl. Sci.* **2023**, *13*, 12204. [\[CrossRef\]](#)

8. Miah, A.S.M.; Shin, J.; Hasan, M.A.M.; Fujimoto, Y.; Nobuyoshi, A. Skeleton-based Hand Gesture Recognition using Geometric Features and Spatio-Temporal Deep Learning Approach. In Proceedings of the 2023 11th European Workshop on Visual Information Processing (EUVIP), Gjøvik, Norway, 11–14 September 2023; pp. 1–6.
9. Parihar, S.; Shrotriya, N.; Thakore, P. Hand Gesture Recognition: A Review. In Proceedings of the International Conference on Mathematical Modeling and Computational Science, Malaysia, 23–25 February 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 471–483.
10. Sosa-Jiménez, C.O.; Ríos-Figueroa, H.V.; Solís-González-Cosío, A.L. A Prototype for Mexican Sign Language Recognition and Synthesis in Support of a Primary Care Physician. *IEEE Access* **2022**, *10*, 127620–127635. [\[CrossRef\]](#)
11. Torres-Muñoz, D.; Hernández-Mejía, C.; Maldonado-Mendez, C.; Hernández-Mendez, S. Exploring a novel facial animation technique using numerical traced algorithm. *Multimed. Tools Appl.* **2022**, *81*, 30961–30976. [\[CrossRef\]](#)
12. Kar, A.K. Bio inspired computing—A review of algorithms and scope of applications. *Expert Syst. Appl.* **2016**, *59*, 20–32. [\[CrossRef\]](#)
13. Chunduru, V.; Roy, M.; Chittawadigi, R.G. Hand Tracking in 3D Space using MediaPipe and PnP Method for Intuitive Control of Virtual Globe. In Proceedings of the 2021 IEEE 9th Region 10 Humanitarian Technology Conference (R10-HTC), Bangalore, India, 30 September–2 October 2021; pp. 1–6. [\[CrossRef\]](#)
14. Sreenath, S.; Daniels, D.I.; Ganesh, A.S.D.; Kuruganti, Y.S.; Chittawadigi, R.G. Monocular Tracking of Human Hand on a Smart Phone Camera using MediaPipe and its Application in Robotics. In Proceedings of the 2021 IEEE 9th Region 10 Humanitarian Technology Conference (R10-HTC), Bangalore, India, 30 September–2 October 2021; pp. 1–6. [\[CrossRef\]](#)
15. Li, T.Y. Numerical solution of multivariate polynomial systems by homotopy continuation methods. *Acta Numer.* **1997**, *6*, 399–436. [\[CrossRef\]](#)
16. Allgower, E.L.; Georg, K. Numerical path following. *Handb. Numer. Anal.* **1997**, *5*, 3–207.
17. Yamamura, K. Simple algorithms for tracing solution curves. *IEEE Trans. Circuits Syst. Fundam. Theory Appl.* **1993**, *40*, 537–541. [\[CrossRef\]](#)
18. Akram, S.; Ann, Q.U. Newton raphson method. *Int. J. Sci. Eng. Res.* **2015**, *6*, 1748–1752.
19. Torres-Muñoz, D.; Hernandez-Martinez, L.; Vázquez-Leal, H. Spherical continuation algorithm with spheres of variable radius to trace homotopy curves. *Int. J. Appl. Comput. Math.* **2016**, *2*, 421–433. [\[CrossRef\]](#)
20. Kramer, O.; Kramer, O. *Genetic Algorithms*; Springer: Berlin/Heidelberg, Germany, 2017.
21. Coello, C.A.C.; Zacetenco, C.S.P. *Introducción a la Computación Evolutiva*; Ediciones CINVESTAV-IPN; Instituto Politécnico Nacional, México: Ciudad de Mexico, Mexico, 2003.
22. Goldberg, D.E. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013; Volume 7.
23. Senin, P. *Dynamic Time Warping Algorithm Review*; Information and Computer Science Department University of Hawaii: Honolulu, HI, USA, 2008; pp. 1–23.
24. Li, T.; Wu, X.; Zhang, J. Time Series Clustering Model based on DTW for Classifying Car Parks. *Algorithms* **2020**, *13*, 57. [\[CrossRef\]](#)
25. Sempena, S.; Maulidevi, N.U.; Aryan, P.R. Human action recognition using dynamic time warping. In Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, Bandung, Indonesia, 17–19 July 2011; pp. 1–5.
26. Chaaraoui, A.A.; Padilla-López, J.R.; Climent-Pérez, P.; Flórez-Revuelta, F. Evolutionary joint selection to improve human action recognition with RGB-D devices. *Expert Syst. Appl.* **2014**, *41*, 786–794. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.