sensors

MDPI

Article

Capturing Conversational Gestures for Embodied Conversational Agents Using an Optimized Kaneda–Lucas–Tomasi Tracker and Denavit–Hartenberg-Based Kinematic Model

Grega Močnik *, Zdravko Kačič, Riko Šafarič and Izidor Mlakar

Faculty of Electrical Engineering and Computer Science, University of Maribor, Koroška c. 46, 2000 Maribor, Slovenia

* Correspondence: grega.mocnik@um.si

Abstract: In order to recreate viable and human-like conversational responses, the artificial entity, i.e., an embodied conversational agent, must express correlated speech (verbal) and gestures (non-verbal) responses in spoken social interaction. Most of the existing frameworks focus on intent planning and behavior planning. The realization, however, is left to a limited set of static 3D representations of conversational expressions. In addition to functional and semantic synchrony between verbal and non-verbal signals, the final believability of the displayed expression is sculpted by the physical realization of non-verbal expressions. A major challenge of most conversational systems capable of reproducing gestures is the diversity in expressiveness. In this paper, we propose a method for capturing gestures automatically from videos and transforming them into 3D representations stored as part of the conversational agent's repository of motor skills. The main advantage of the proposed method is ensuring the naturalness of the embodied conversational agent's gestures, which results in a higher quality of human-computer interaction. The method is based on a Kanade–Lucas–Tomasi tracker, a Savitzky–Golay filter, a Denavit–Hartenberg-based kinematic model and the EVA framework. Furthermore, we designed an objective method based on cosine similarity instead of a subjective evaluation of synthesized movement. The proposed method resulted in a 96% similarity.

Keywords: conversational gestures; 3D gestures; motor skills; gesture reconstruction; kinematics; embodied conversational agents; Kanade–Lucas–Tomasi tracker; Denavit–Hartenberg

Citation: Močnik, G.; Kačič, Z.; Šafarič, R.; Mlakar, I. Capturing Conversational Gestures for Embodied Conversational Agents Using an Optimized Kaneda–Lucas–Tomasi Tracker and Denavit–Hartenberg-Based Kinematic Model. *Sensors* **2022**, *22*, 8318. https://doi.org/10.3390/s22218318

Academic Editors: Roberto Vezzani, Mohamed Daoudi, Guido Borghi, Marcella Cornia, Claudio Ferrari, Federico Becattini and Andrea Pilzer

Received: 8 September 2022
Accepted: 27 October 2022
Published: 29 October 2022

1. Introduction

Visual articulation of information through embodied behavior plays an important role in spoken social interaction [1]. Speech and gestures (including hand gestures, facial expressions, posture, and gazing) originate from the same representation, but are not necessarily based solely on the speech production process; i.e., "speech affects what people produce in a gesture, and that gesture, in turn, affects what people produce in speech" [2] (p. 260). In fact, more than 50 percent of visual articulation (i.e., embodied behavior) in spoken interaction adds non-redundant information to the common ground of the conversation [3]. Moreover, over 70% of the social meaning of a conversation or an interaction is transmitted through concepts other than words [4] (p. 27).

Recently, face-to-face interaction has been gaining attention, especially in interfaces where personalization is one of the key drivers, such as eHealth [5] and support for the elderly in their interaction with information technology [6]. Face-to-face interaction has been shown to elicit user engagement and stimulate the use of conversational interfaces [7], where the non-verbal, visual components drive the elicitation of affect and social awareness in human partners [8]. Overall, embodied conversational agents (ECAs) are becoming indispensable tools in personalizing and personifying everyday scenarios, where non-verbal behavior plays a crucial role in both representations of information and its understanding [9].

However, animating ECAs, whose non-verbal behavior is perceived as believable, is quite challenging, especially considering the complexity of the underlying bio-mechanical system [10]. The perceived believability of synthetic behavior and its plausibility depend on appearance, awareness, personality, emotional state, liveliness, illusion of life, consistency, diversity, and social fluency [11]. The dis-synchrony (unnaturalness) between verbal and non-verbal elements is most noticeable in synthesized non-verbal forms (i.e., shapes and poses), and especially through kinetics and "prosody" (i.e., fluidity, internal dynamics, movement phases, etc.) synthesis of movement [12].

Thus, two main challenges exist for the synthesized multimodal gestures to be perceived as believable [13]. The first challenge, i.e., the 'symbolics' of gesture, is related to the contextual alignment of visualized 'shapes' to speech and situational context, i.e., determining what kind of shapes a character should display in the given 'semantic' context: 'What to display to visualize the given communicative intent?'. The symbolic alignment is, in general, implemented as a rule-based [14] or as a conversational behavior generation system with a machine learning (ML) baseline [15,16]. The second challenge, i.e., the 'prosody' gesture, is then related to the inner and outer fluidity (dynamics) of physical realization (animation) of selected visualizations, i.e., determining the trajectories and fluidity of movement in the given "non-semantic" context: "How to display the sequence of shapes to match conversational intent, i.e., acoustic and linguistic properties of spoken content?".

The first challenge we tackled successfully in [16]. To tackle the second challenge, a wide range of techniques (data- and prosody-driven approaches) were introduced, to cope with significant requirements related to the believable fluidity of non-verbal expressions [17]. The main drawback of data/speech/prosody-driven approaches is that they are generated based on a small set of signals related mainly to the speech signal (e.g., pitch and prosody). Thus, they cannot facilitate "symbolics" [18].

The main idea behind the proposed method is to create contextually relevant resources that can be re-used when an embodied conversational agent generates a viable conversational sequence. The verbal and non-verbal context of an observed sequence, to be fed to behavior planning, is pre-annotated, and the role of the proposed method is to extract a possible visual articulation, including inner fluidity. Overall, systems utilizing gesture templates (e.g., procedural/physical animation) show the capacity to align movement with momentary context, as well as the context in the planned near future [19]. However, synthetic gestures still lack believability. To cope with the challenge of addressing the liveliness, diversity, and consistency of synthetic gestures adequately, we propose to exploit gesture tracking and 3D reconstruction to deliver a system capable of recording gestures expressed in the video during face-to-face conversation automatically, and storing them as gesture prototypes of the so-called "motor skills" [20]. We built the concept based on the following assumption: ECAs with diverse sets of resources, which preserve the dynamics and complexity of human movement, will be more successful in their attempt to mimic human-like conversational behavior. Such entities will be perceived as more believable virtual entities with human-like (and not human) attributes. Instead of subjective evaluation through human observation, we implement an objective measure to evaluate the naturalness of synthesized movement based on cosine similarity.

With the goal of reconstructing conversational gestures as natural as possible, we present our choice of suitable methods, announced in the title of the paper and our successful connection of stated methods in an efficient conversational gestures reconstruction system. We propose a measure based on cosine similarity for objectively evaluating the naturalness of synthesized hand movements generated by the proposed method instead of subjective evaluation through human observation, which is what, to the extent of our knowledge, was being done to evaluate gestures until now. In addition, we present the results of our system that were evaluated objectively on an embodied conversational agent called EVA (An EVA is an embodied conversational agent, developed in the Laboratory for Digital Signal Processing, Faculty of Electrical Engineering and Computer Science, University of Maribor [21]).

The paper is structured as follows. Section "Related Work" provides an overview of related work on gesture tracking and reconstruction techniques. Section 2 outlines the formalism used to implement the proposed tracking and reconstruction system. Section 3 shows the results of the system evaluated objectively on an embodied conversational agent EVA [21]. Section 4 provides the discussion, and the conclusions follow.

Related Work

Synthetic responses are often perceived as lacking in consistency, diversity, and vividness. Gestures perceived as "random", "non-aligned", or "no-sense" distort the perception of eloquence, competence, human-likeness, and vividness of conversational responses. Creating believable movements is challenging, since the movements must be meaningful and natural, reflecting the coupling between gestures and speech [22]. In some cases, it has been suggested that having no gestures will achieve better persuasion and perceived believability than inadequate gesturing [23]. The main drawback most systems face in the context of believability is how to cope with diversity, i.e., how to create a repository of non-verbal expressions large enough to adequately represent both intent and thought [21].

Much of the complexity associated with the reconstruction process is related to the "posing" phase, in which the animators must handle a large number of on-screen handles associated with the character's virtual skeleton. These handles allow direct or indirect modeling of the available degrees of freedom (DOFs) of all the individual character's joints [24]. Moreover, to generate believable animation using CAD environments, the animators must understand and tackle many modeling and animation techniques (e.g., polygonal modeling, modeling with NURBS or subdivision, UV skinning, forward and inverse kinematics, rigging, etc.). The resulting animations are realistic in a given context. However, they may decrease believability significantly when the internal fluidity (dynamics) is changed due to different intent or prosodic context. To increase the versatility of the gestural morphology and to decrease the complexity of designing the movements a conversational agent can reproduce, we propose to build a 3D corpus of gesture prototypes [20]. For the modeling environment, we exploited the DAZ3D studio, which simplifies the CAD controls, and, through a vast repertoire of conversational resources, minimizes the human animator's need for 'artistic' skills. Each prototype is expressively adjustable, and the EVA realizer [21] can mitigate the general issues of fluidity. However, since prototypes are generated manually by observing 'real-life' conversational expressions, the inner fluidity and the trajectory are not captured, but rather defined artificially through a set of orientational in-between points used to model the forward kinematics. As a result, the increasing complexity of gestures decreases the observed fluidity and perceived believability significantly.

To capture and preserve a high resolution of inner fluidity, performance-driven animation can be exploited to create 3D resources [25]. In this concept, an actor's physical performance is transferred interactively to the virtual character to be animated. The method requires specialized equipment (e.g., a sensor suit) and specially trained experts, making it highly expensive and less suitable for non-professional animators. The mapping between the performer's and character's motion is also a complex task, since both entities operate in different spaces. Thus, the process requires sophisticated configuration steps and automatic retargeting [26]. In [27], the authors outline a sophisticated system consisting of multiple cameras and passive sensors to compensate for the lack of naturalness and capture movement generated during the conversation. Most multi-view methods utilize multiple cameras and exploit the image depth and shape from silhouette cues to capture the moving actor [28], or reconstruct gestures via a multi-view photometric stereo approach [29]. These methods typically require a high-resolution scan of the person as an input.

With advances in deep learning (DL) and image processing and the availability of depth camera sensors, new opportunities arise that could enable end-to-end reconstruction and capture of 3D resources. Methods integrating Kinect or similar depth sensors [30–32] or multi-view data [33,34] achieve impressive reconstructions, but do not register all frames to the same canonical template, and require complicated capture setups. Moreover, to

represent conversational movement viably, the captured resources must originate from real-life situations integrating spontaneous behavior [35]. Recreation, even when performed by professional actors, will always reflect artificialness, resulting in less spontaneous and less diverse responses [36].

If we want to create a sufficiently large inventory of gestures that will enable the generation of natural gestures in interaction, and if we want to achieve a time-efficient generation of such an inventory, it makes sense to use a multitude of existing video recordings, and, consequently, it makes sense to use a method based on the use of video materials recorded with one camera. Most conversational corpora consist of TV interviews and theatrical plays that have shown themselves to be an appropriate resource of spontaneous conversational expressions, and are significantly more suitable for research in wider 'discourse concepts' than any artificially recorded material [37]. Most methods related to 3D Pose and Shape Estimation from monocular sources refer to (Deep) Convolutional Neural Networks ((D) CNNs) and leverage 2D joint tracking and predict 3D joint poses in the form of stick figures [38–42]. The major challenge with deep learning and similar probabilistic approaches is that the tracking process involves predicting the most probable configuration of the artificial skeleton. Thus, the captured conversational movement will approximate something known to the model rather than an exact replication of what is observed. Overall, the DNN-based approaches work well within the constraints of the known context (i.e., a fixed environment and known classes). However, in uncertainty, the models tend to underperform and require retraining [43]. The inconsistency and uncertainty of deep models (e.g., Pose Net, Open Pose) in many cases result in issues such as incoherence in fluidity (e.g., sudden shifts) and over smoothing of actual movement [44], leading to a decrease in believability when replicated as part of synthetic conversational behavior.

With our main motivation in mind, i.e., to capture conversational expressions from monocular video as similar to the original as possible, and by preserving the 'prosody of movement', i.e., fluidity and dynamics, we designed a novel system, which consists of a Kanade–Lucas–Tomasi tracker (KLT) [45] to track the observed body parts based on optical flow, and a Denavit–Hartenberg-based kinematic model [46] to reconstruct tracked features as 3D templates and store them as part of the EVA's [20] motor skills repository. However, as with any image processing algorithm, mismatches in either tracking or reconstruction will always appear. Thus, in addition to the non-predictive method, it is crucial to have an objective measure to evaluate this effect. Instead of subjective evaluation of believability through perceptive experiments, as generally utilized in the field of embodied conversational agents, we propose a new method that allows for easy assessment of the mismatch generated in the tracking and reconstruction process.

2. Materials and Methods

2.1. Materials

The material we used in our work is a video signal with FullHD resolution and a different density of frames per second (frame rate). To test our system, conversational gestures were recorded with a video camera in a laboratory environment with a relatively impoverished background with only one actor. The resolution of the laboratory videos was FullHD (1920 × 1080) with H.264 compression and a frame rate of 30 FPS. In addition to laboratory videos, video clips from a video podcast were also selected; their content included spontaneously created conversational gestures and a diverse conversation with two performing actors. We used video content with a large number of spontaneously generated gestures. Such video content usually consists of videos with conversations. It is necessary to be aware that certain video content with professional actors (talk shows, evening news, etc.) does not offer a large amount of spontaneous and/or naturally created gestures. Professional actors know how to create conversational gestures that are not created spontaneously, but are acted out. We subjectively selected video content with gestures created spontaneously as our experimental example. The podcast videos were streamed from a social network, where they were published with the purpose of sharing video

content. The resolution of the obtained video was also FullHD (1920 × 1080) with a different frame rate (25 FPS) than the laboratory video sources. Between 10 and 19 conversational gestures for each type of movement were analyzed, to evaluate the conversational gesture reconstruction from the EVAPose system. The analysis was performed for laboratory and spontaneously generated conversational gestures. We captured the spontaneous gestures from the video podcast, Gospoda [47].

2.2. Laboratory Set-Up

The core idea of the proposed method is to capture conversational expressions from different human collocutors engaged in interaction contained in the EVA-Corpus Video dataset, and store them back as "motor-skills" in the EVA-Corpus MotorSkills dataset, i.e., 3D artefacts to be re-used by embodied conversational agents during human-machine interaction. The workflow with individual steps is outlined in Figure 1.

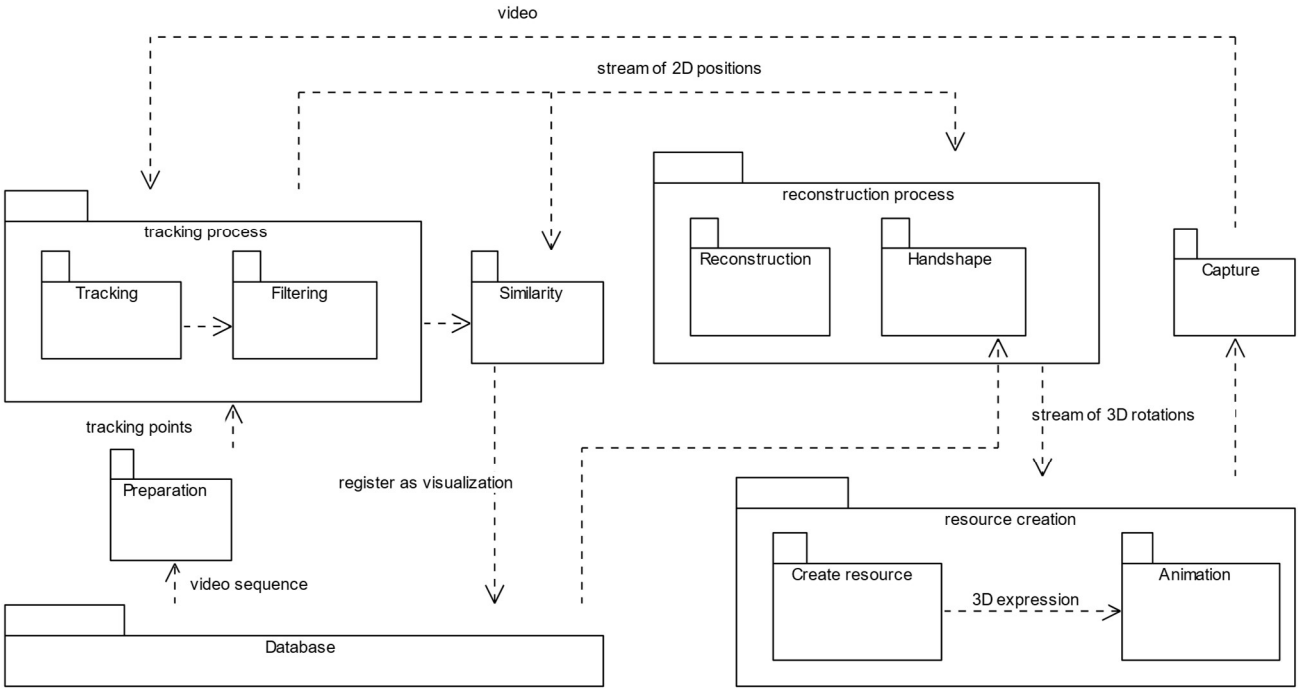

Figure 1. Workflow to capture conversational behavior in spontaneous discourse automatically, store it as gesture templates, and interconnect the captured templates with other verbal and nonverbal features of the observed sequence. The hand shape is not tracked; a CNN model was used to select the shape from a dictionary of possible shapes based on the HamNoSys notation system [48].

The input to the proposed method was a color video stream, a conversational sequence contained in the EVA Corpus Video dataset. In the preparation phase, the best-fitting tracking points are selected automatically. The tracked points were filtered to reduce noise and possible inaccuracy in tracking—visualized as 'jitter' or sudden and instant jumps of the observed object from one position to another. The tracked geometry is sent to the Denavit–Hartenberg-based kinematic model and transformed into (Reconstruction in Figure 1) Euler angles (yaw, pitch, and roll) stored as a procedural animation (Create Resource in Figure 1). To validate the captured conversational expression (including the articulated shapes and inner fluidity) and compare it against the original, the expression was synthesized on our proprietary ECA realizer [21] by its in-scene recorder (Animation and Capture in Figure 1) functionality. If the synthetic system is recognized as similar (similarity index above 70%), the realization is registered as a possible visualization of the conversational concept. The following sections highlight the individual steps in more detail.

2.3. Preparation Step

Tracking arbitrary objects consistently and accurately in video sequences is challenging. Selecting robust features that best correspond to physical points and can, at the same time, be tracked well (e.g., mitigate occlusions, disocclusions and features that do not correspond to points in the world) is the first step in delivering an effective tracker. Shi-Tomasi's implementation [45,49] represents a robust method to select "good features" and can, at the same time, compensate for lack of naturalness if and when these features are lost due to occlusion or "loss of visibility"; a common occurrence when tracking the movement of hands in multiparty discourse. Unlike the Harris Corner Detector, the Shi-Tomasi implementation proposes a variation in the selection of corners, and proposes a pixel to be considered as a corner by comparing the eigenvalues, i.e.:

$$R = \min(\lambda_1, \lambda_2) > \lambda, \tag{1}$$

where λ_1 and λ_2 are two eigenvalues of a symmetric matrix and λ is the predefined threshold. The pixel is considered a corner when both λ_1 and λ_2 are above the threshold. Figure 2 highlights the selection of the N strongest corners as defined by Shi-Tomasi (a) and selected tracking points (b), to be tracked and used as input in the reconstruction.

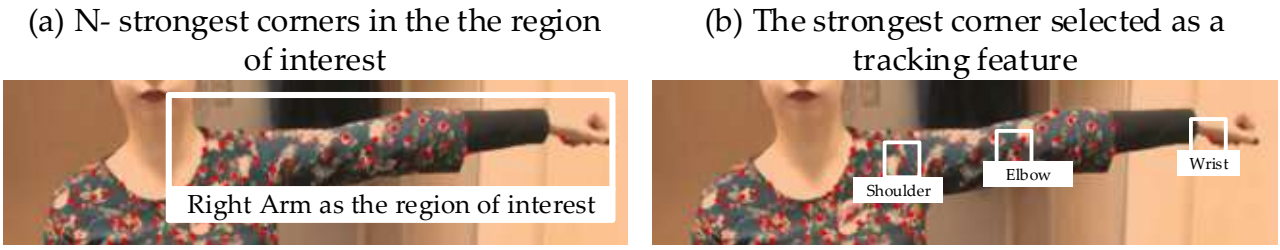

Figure 2. Example of (**a**) Good features as extracted by the Shi-Tomasi detector and (**b**) Tracking points as the strongest corners in a specific region, representing the "tracked" joints in the human skeleton.

As outlined in Figure 2, the algorithm operates over grayscale images. The "quality" of corners (i.e., the λ threshold) is specified as a value between 0 and 1. All the corners below the threshold are rejected. Since we wanted to track only specific artefacts representing the shoulder, elbow, and wrist "joints", the user selects the regions of interest. Based on the manual selection of the region of interest, the algorithm selects the strongest corner automatically (i.e., the "green" circles in Figure 2b) as the final tracking point, and "rejects" all nearby corners of interest. In the tracking process, the tracking points are regarded as features.

2.4. KLT Feature Tracker

The KLT feature tracker [50] computes the displacement of features between consecutive frames by aligning a second image J to an input image I, where $I(x,y)$ represent the intensity of the image at $[x\ y]^{\mathrm{T}}$.

Let:

$$u = \left[\mathrm{u}_x\ \mathrm{u}_y\right]^{\mathbf{T}}, \tag{2}$$

where u represents the point at coordinates (x, y) in the first image I. The goal of tracking is to find point v in the second image J, where the displacement d is minimal; thus $I(u)$ and $J(v)$ are similar:

$$v = u + d = \left[\mathrm{u}_x + d_x\ \ \mathrm{u}_y + d_y\right]^{\mathbf{T}}. \tag{3}$$

The displacement $d = \begin{bmatrix} d_x \; d_y \end{bmatrix}^T$ represents the image velocity (optical flow) at u. The minimal difference is computed as the mean squared error function:

$$\varepsilon(d) = \sum_{x=\mathrm{u}_x-\mathrm{w}_x}^{\mathrm{u}_x+w_x} \sum_{y=\mathrm{u}_y-\mathrm{w}_y}^{\mathrm{u}_y+w_y} \big(I(x,y) - J\big(x+d_x,\; y+d_y\big)\big)^2 \tag{4}$$

where w_x, w_y represent the integration window size parameter of the template window of size $(2w_x+1) \times (2w_y+1)$.

The intensity of the image is represented by a small template window of $n \times n$, centered at one of the feature points. $J(x)$ is the same window in the next frame. d represents the displacement vector, and η represents the error introduced due to the shape change.

During the tracking process, the goal is to find point v in image J that corresponds to point u in the image I:

$$\overline{v}_{opt} = G^{-1}\overline{b}, \tag{5}$$

where:

$$G \doteq \sum_{x=p_x-\mathrm{w}_x}^{p_x+w_x} \sum_{y=p_y-\mathrm{w}_y}^{p_y+w_y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{6}$$

I_x and I_y represent image derivatives.

The image mismatch vector $\overline{b}$ is defined as:

$$\overline{b} \doteq \sum_{x=p_x-\mathrm{w}_x}^{p_x+w_x} \sum_{y=p_y-\mathrm{w}_y}^{p_y+w_y} \begin{bmatrix} \delta I \; I_x \\ \delta I \; I_y \end{bmatrix} \tag{7}$$

δI represents the image difference. In standard optical flow computation, the goal is to find $\overline{v}_{opt}$ as displacement $\overline{v}$, which minimizes the matching function $\varepsilon(\overline{v})$:

$$\varepsilon(\overline{v}) = \varepsilon\big(v_x,\; v_y\big) \sum_{x=p_x-\mathrm{w}_x}^{p_x+w_x} \sum_{y=p_y-\mathrm{w}_y}^{p_y+w_y} \big(A(x,y) - B\big(x+v_x,\; y+v_y\big)\big)^2 \tag{8}$$

where optimum $\overline{v}$ is calculated through Taylor expansion:

$$\left.\frac{\partial \varepsilon(\overline{v})}{\partial \overline{v}}\right|_{\overline{v}=\overline{v_{opt}}} = \frac{1}{2}\left[\frac{\partial \varepsilon(\overline{v})}{\partial \overline{v}}\right] \approx G\overline{v} - \overline{b} \tag{9}$$

Because of the first-order Taylor approximation, this is only valid when the pixel displacement is small. Thus, the standard optical flow computation is performed in k steps and defined by:

$$\overline{v} = d^L = \overline{v}^K = \sum_{k=1}^{K} \overline{\eta}^k \tag{10}$$

where $\overline{v}$ represents the final optical flow and d^L the displacement, K the number of iterations to reach convergence, and new pixel displacement (i.e., one step in the LK optical flow computation) $\overline{\eta}^k$ is defined as:

$$\overline{\eta}^k = G^{-1}\overline{b}_k \tag{11}$$

Derivatives $Ix,\; Iy$ in the image mismatch vector are computed at the beginning, and only δI is recomputed at each step k. The overall iteration completes when $\overline{\eta}^k$ is smaller than the threshold, or the maximum number of iterations is reached.

The implementation of the KLT tracker used in our research is highlighted in Figure 3. The preparation step, definition of feature points and tracking points according to the defined process, is described in Section 2.1. Feature points were selected according to Shi-Tomasi's approach, and the tracking points were set to regions representing the "shoulder", "elbow", and "wrist" joints. We used a 5×5 points integration window.

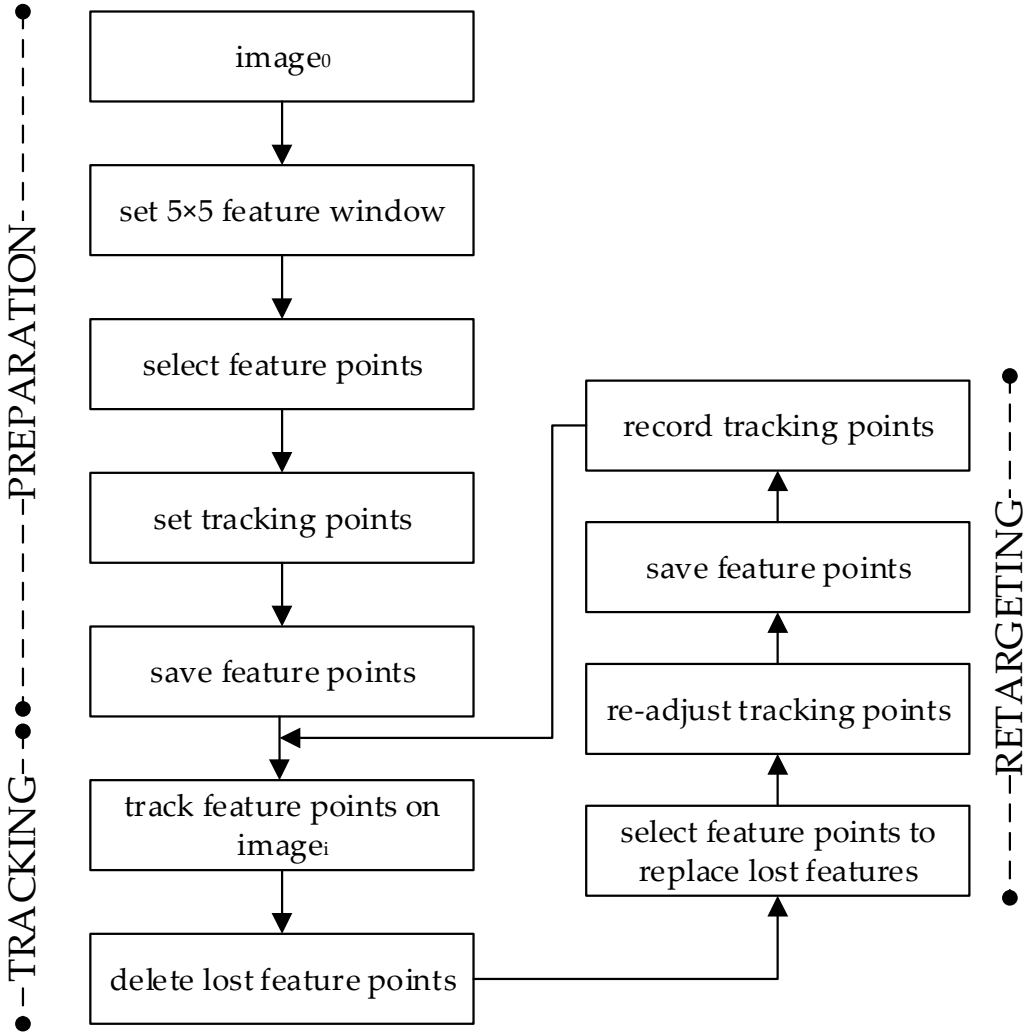

Figure 3. Overview of the implementation of the pyramidal KLT Tracker.

The tracking process implements the iterative optical flow computation and matches tracking features between the current $image_i$ *(J)* and the previous $image_{i-1}$ *(I)* by tracking feature points, where i is on the interval [1, n] and n is the last frame of the video recording of the conversational sequence. If point v in $image_i$ that corresponds to point u in $image_{i-1}$ cannot be found (i.e., the feature falls outside of the image), and/or the image path around the track point varies too much (i.e., the cost function is larger than a threshold), the feature point is regarded as lost and is deleted. To recreate the expression, all tracking points must be registered in all frames, and the missing feature points must be replaced. The process exploits the Shi-Tomasi detector to create a new set of good features to replace lost features. The tracking points are relocated automatically to the closest new best feature point. The new feature points and 2D coordinates of each tracking point are saved, and the algorithm may proceed with the next frame. The tracking process completes when the last frame of the video is reached.

2.5. Filtering

The designed tracker implements "tracking-by-detection" and does not implement feature descriptors, such as SIFT [51]. This means that the tracking accuracy varies depending on the rotation, scale, and image perspective distortions (including lighting changes). The inaccuracy results in a "long-distance" move of a tracking point instead of a small shift in position (i.e., noise). While reconstructing the movement on the ECA, the jumps will be observed as instant "jumps"—movements which cannot be expected in real life. To avoid this,

we implemented a digital filter based on the Savitzky–Golay algorithm [52]. The Savitzky–Golay filter belongs to the family of FIR filters, and provides an estimate of the derivative of the smoothed signal using convolutional sets derived from least-squares formulas coefficients. Savitzky–Golay filters minimize the least-squares error in fitting a polynomial to a sequence of noisy data. Consequently, the precision of data increases without distorting the signal tendency. Thus, the method is suitable for signal smoothing [53,54].

Let us consider the captured tracked points as a compositum of captured movements, i.e., the main signal $f(l)$, corrupted randomly by distortions, i.e., $w(l)$, thus the real signal (stream of tracking points) is defined as:

$$x(l) = f(l) + w(l), \qquad l = 0, \ldots, L \tag{12}$$

where $x(l)$ indicates the lth tracking point (i.e., the lth frame) in the signal with L points (sequence of data). The goal is to smooth the $x(l)$ to reduce the level of the remaining noise to as low as possible, i.e., to minimize the following MSE:

$$\varepsilon_n = \sum_{i=-M}^{M} (P(i) - x(i))^2 = \sum_{i=-M}^{M} \left(\sum_{k=0}^{n} a_k i^k - x(i) \right)^2 \tag{13}$$

where the smoothing is carried out with a symmetric window with width $N = 2M + 1$ samples around the "reconstruction point". In this case, smoothing can be represented as a polynomial with the order $n(P(i)) = \sum_{k=0}^{n} a_k i^k; k = 0, \ldots, n,$ and a_k is the kth coefficient of the polynomial.

The filter output is then equal to the value of polynomial (n) in the central point $y(0)$; $y(0) = p(0) = a_0$. To calculate the next point, the window N is shifted by 1 unit. Savitzky and Golay [52] showed that the above process of 'filtering' is equivalent to convolving samples in windows with a fixed impulse response:

$$y(k) = \sum_{i=-M}^{M} w_i x(k-1) \tag{14}$$

To select the SG parameters used for smoothing optimally, we applied the power spectrum analysis. A power spectrum analysis was performed on an arbitrarily selected area of an unfiltered input signal. The analysis showed the level of the signal power spectrum and noise. Such analysis was also calculated for a filtered input signal with SG parameters of choice. In the area of the high frequency harmonics, we checked at what distance from the densification of the signal spectrum there was still a greater change in the spectrum of the individual filtered signal at the noise level. It was found that, with a window width of N = 9 and a polynomial degree 3, we achieved a sufficient smoothing effect. Figure 4 shows the power spectrum of the smoothed signal, the smoothed signal with a third-degree polynomial, and a window width of 9 is highlighted. It can be seen that the level of the power spectrum of the smoothed signal is less intense at slow transitions of the input signal than at faster ones. From this, we can conclude that the parameters selected preserve the signal's slower jumps, while the smoothing increases in the areas with faster jumps (a part of the signal with a higher frequency). Figure 5 shows the smoothing results using an SG filter with a window width N = 9 and a polynomial of the 3rd order. For the objective weight function w, a cubic function was used.

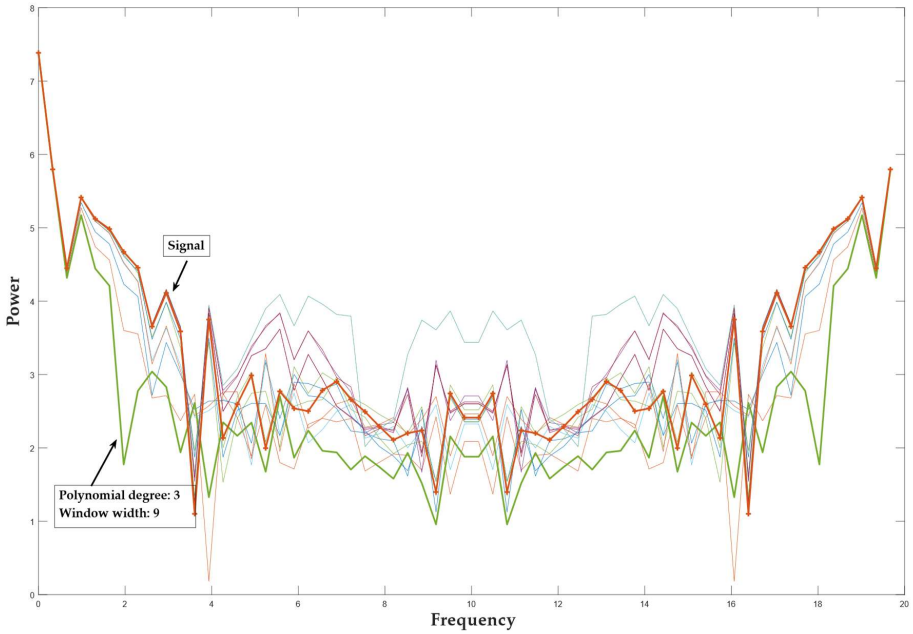

Figure 4. Power spectrum analysis of a filtered and nonfiltered signal. The green curve represents the optimal filtered signal; the orange curve represents the nonfiltered signal.

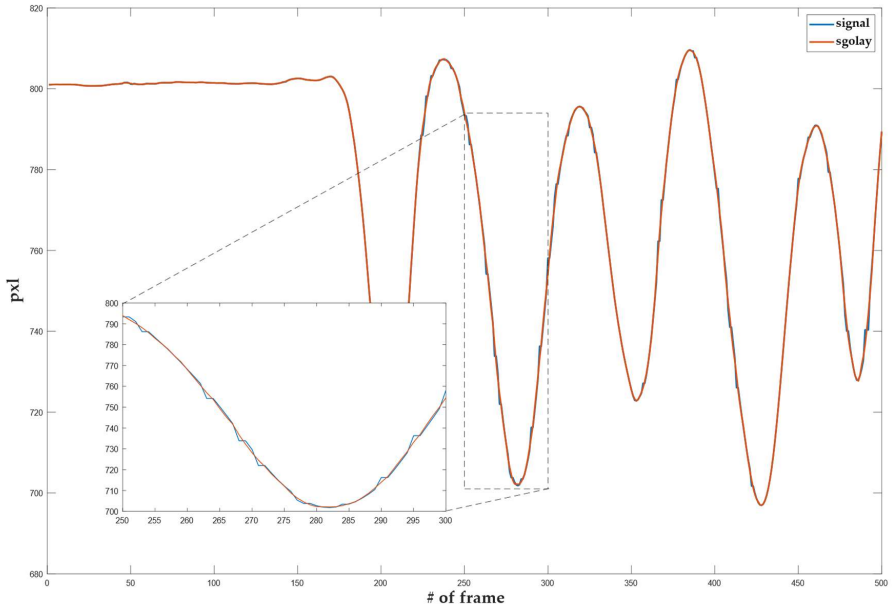

Figure 5. Smoothing the raw tracking results with an SG filter.

2.6. Denavit–Hartenberg Based Reconstruction

As highlighted in Figure 1, the tracking points are sent to the reconstruction phase. In the first step, internal angles are calculated from tracking points using basic angle functions based on the player's position in the video scene. The internal angle is calculated between the starting point o_a (in our case the player's shoulder on the video signal) and the end of the player's arm (end-effector) o_{ef}. In the case when the player is facing us, the internal angle is calculated as $q_8 = \text{arctan2}\left(o_a,\ o_{ef}\right)$. For each joint that rotates, we can write $q_i = \text{arctan2}(o_a,\ o_i)$. Signal tracking points for each joint are marked as o_i.

In this phase, 2D coordinates of tracked points are converted into Euler angles, which can be animated by our conversational agent. The arm is deconstructed into a manipulator consisting of three spherical joints: the shoulder, the elbow, and the wrist joint. Figure 6

outlines the designed manipulator. The end-effector is placed at the far end of the arm (e.g., the tip of the hand) as a reference point used in the automatic kinematic analysis algorithm.

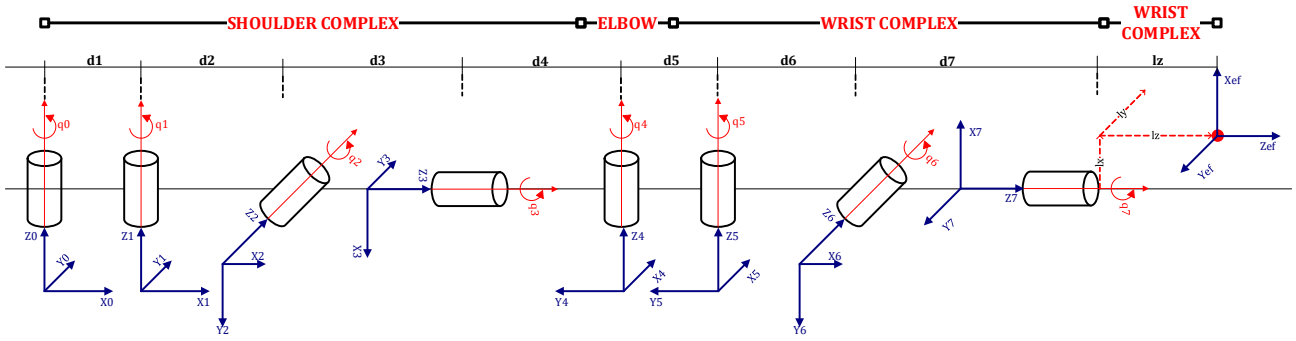

Figure 6. Visualization of the kinematic model, i.e., the arm manipulator consisting of complex cylindrical joints, implementing degrees of freedom of a spherical joint utilized in the skeleton of the realization entity.

The goal is to determine the rotation of each joint in the mechanism of human arm movement from the positions of the tracking points. As outlined in Figure 6, the proposed kinematic model assumes each spherical joint is represented by multiple revolute joints that permit linear motion along a single axis. Using a single degree of freedom allows us to represent each angle of rotation of the spherical joint with a single real number, and the rotation in the spherical joint as a composition of single-axis rotation. This allows us to determine the position, and, more importantly, the orientation of tracking points in a systematic way, where the cumulative effect (A_i) is calculated using the Denavit–Hartenberg (D–H) convention [46].

We assumed the proposed model consisted of eight revolute joints and ten links. We assumed that $joint_i$ connects $link_{i-1}$ with $link_i$. Thus, when $joint_i$ was actuated, $link_i$ and further links in the kinematic chain of the robot arm moved. To perform the kinematic analysis, a coordinate frame was attached to each link, i.e., $x_iy_iz_i$ to $link_i$, represented by the tracking point. Using the D–H convention, we assumed A_i was a homogeneous transformation matrix which expresses the position and orientation of $x_iy_iz_i$ in respect to $x_{i-1}y_{i-1}z_{i-1}$. The A_i varied as the configuration of the manipulator was changed; however, since we assumed the use of revolute joints, A_i is a function of a single joint variable and can be represented as the product of four basic transformations, rotation, and translation around the z_{i-1} and x_i axes:

$$A_i^{i-1} = Trans(z_{i-1}, d_i)Rot(z_{i-1},\ \theta_i)Trans(x_i, a_i)Rot(x_i,\ \alpha_i) =$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

where θ_i, d_i, a_i, α_i are parameters associated with $link_i$ and $joint_i$ denoted as joint angle, link offset, link length, and link twist. The parameters mentioned above are shown in Figure 6.

Since revolute joints were used, the A_i^{i-1} is a function of a single variable θ_i and the other three parameters are denoted as D–H parameters and are constant for a given link. The D–H parameters for each link of the proposed kinematic model are shown in Table 1.

Table 1. The Denavit–Hartenberg parameters for the kinematic model.

Link$_i$	a_i	d_i	α_i	θ
1	a_1	a_1	$\pi/2$	q_1
2	0	0	$-\pi/2$	q_2
3	0	0	$\pi/2$	q_3
4	0	0	$-\pi/2$	q_4
5	a_5	a_5	0	q_5
6	0	0	$-\pi/2$	q_6
7	0	0	$\pi/2$	q_7
8	0	0	0	q_8
		End-Effector		
9	$-l_x$	l_z	0	0
10	$-l_y$	0	0	0

Where link lengths a_1, $a_5, -l_x, -l_y$ and link offsets d_4, l_y, were calculated as the average values of measurements performed over multiple human arms:

$$(a_n, d_n, l_n) = \left(\frac{\sum_1^{n_p} a_n}{n_p}, \frac{\sum_1^{n_p} d_n}{n_p}, \frac{\sum_1^{n_p} l_n}{n_p} \right) \tag{16}$$

where n_p represents the number of candidates participating in the measurements ($n_p = 10$) and the variable l_n represents the end-effector position. For our case, we calculated the D–N parameters as $a_1 = 30\ cm$, $a_5 = 30\ cm$, $d_4 = 30\ cm$, $l_x = 1,5\ cm$, $l_y = 2.0\ cm$ and $l_z = 12\ cm$.

Using the D–H parameters in Table 1, we calculated the homogeneous transformation matrices A_i^{i-1} for each joint, and created a reference transformation matrix for the forward kinematics of the proposed kinematic model:

$$A_9^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 A_6^5 A_7^6 A_8^7 A_9^8, \tag{17}$$

where the last two matrices A_8^7 and A_9^8 are the matrices of the end effector.

However, it is not trivial to represent any arbitrary homogeneous transformation using only four parameters [46,55]. Given two consecutive frames, 0 and 1, with given coordinate frames $x_0y_0z_0$ and $x_1y_1z_1$ respectively, we assumed there exists an A_0^1 homogeneous transformation matrix. Moreover, we assumed the axis x_1 was perpendicular to z_0, and x_1 intersects the z_0 axis. Under these conditions, there exist unique numbers a, d, θ, α, and Equation (15) can also be written as:

$$A_i^{i-1} = \begin{bmatrix} & R & & p \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

where R describes the rotation matrix of joints and p describes the displacement. We can write the rotation in a homogeneous transform matrix as:

$$R_i = R_{i,z}(\gamma) R_{i,y}(\beta) R_{i,x}(\alpha) = \underbrace{\begin{bmatrix} cos\gamma_i & -sin\gamma_i & 0 \\ sin\gamma_i & cos\gamma_i & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{yaw} \underbrace{\begin{bmatrix} cos\beta_i & 0 & sin\beta_i \\ 0 & 1 & 0 \\ -sin\beta_i & 0 & cos\beta_i \end{bmatrix}}_{pitch} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\alpha_i & -sin\alpha_i \\ 0 & sin\alpha_i & cos\alpha_i \end{bmatrix}}_{roll} \tag{19}$$

where the angles α, β, and γ represent the Euler rotation. The names of each rotation angles are roll, pitch, and yaw. Since the embodied conversation agent EVA [21] has a default position different from our kinematic model, the calculated data must be adjusted to be suitable for the EVA-Script Template using a 90-degree rotation of α angle (roll):

$$R_i \xrightarrow{adjustment} R_{i,EVA} = R_{i,z}(\gamma) R_{i,y}(\beta) R_{i,x}(\alpha + 90) \tag{20}$$

The above adjustment represents a method to normalize scene results to any given space of any embodied conversational agent with an underlying three joint-based skeletal structure. The calculated data can now be used for 3D conversational resources, as defined by the final component, i.e., the Resource Generator.

2.7. Resource Generator

In this step, the captured and reconstructed stream of Euler angles is transformed into a procedural animation, an EVA-Script Template compatible with the repository of "motor skills". In the EVA Framework, a conversational expression or gesture is defined as a function of conversational intent and its realization through visualization (i.e., movement model) [16], i.e.:

$$\hat{G} = T_m^{-1}\hat{H} \tag{21}$$

where $\hat{G}$ represents gesture T_m^{-1} contextual interpretation based on conversational intent, and $\hat{H}$ is the movement model used to "visualize" the conversational intent. The movement model is then defined as a transition between the pose at the beginning (P_S) and the pose at the end (P_E), via a trajectory (J), carried out over time t.

$$\hat{H} = J(P_S,\ P_E)_{|t} \tag{22}$$

Since the realization engine utilizes procedural animation and forward kinematics, the trajectory J is specified as a sequence of in-between frames, i.e., $J = \{P_{S+1}, \ldots, P_{E-1}\}$. The attribute t is used to "optimize" the number of in-between-frames given the time constraint t, and the realizers targeted frame rate f:

$$N = round\left(\frac{n_J + 1}{t \times f}\right) \tag{23}$$

Here, N represents the number of in-between frames in J to skip, n_J represents the total number of frames captured by J. We added 1 frame to preserve the number of in-between frames, since the first transformation from configuration P_{S-1} to P_S (i.e., the *start pose*) is captured by J. The $t \times f$ represents the maximum number of frames the realizer can implement without any impact on the inner fluidity. Namely, adding too many in-between frames will "slow" down the gesture synthesis. The remaining series of configurations are, afterwards, transformed into an EVA template [22]. The overall process is outlined by the following pseudocode:

Algorithm 1. Proposed algorithm for creating resources.

```
t = calculate_from timestamps
f = from_config
N = round_up (size(frame) + 1)/(t × f)
if N > 1 then H = array (t × f)
else H = array (size(frame) + 2)
H [0] = to_unit(frame [0])
for i = 1 to size(frame) − 2:
if i mod N == 0 then:
append(toUnit(frame[i]), H)
else:
continue
append(toUnit(frame[size(frame) − 1]), H)
createGesture (H, t)
```

The algorithm always takes the first and the last frame as the start and end poses, respectively, and every ith in-between frame, such as *i mod N equals zero*. Other frames are dropped. The function *toUnit* maps the 3D definition of each frame into the EVAScript's unit notation, i.e., *{key, list}* pair where *key* represents the articulated joint (e.g., collar, shoulder, elbow, forearm, or wrist) and *list* represents a sequence of 3D configurations for

the joint (i.e., the sequence of yaw, pitch, roll configurations) that constitute the 3D transformation from pose *i* to pose *j*. The function *createGesture* then defines $\hat{G}$ as a procedural fragment written in EVAScript markup as shown in Figure 7.

```
<sequence><parallel>
<UNIT name = "abdomenLower" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "354.5, 358.4, 0.8" />
<UNIT name = "chestLower" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "14.7, 2.0, 359.8" />
<UNIT name = "neckUpper" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "0.0, 0.1, 0.5" />
<UNIT name = "lCollar" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "0.0, 0.1, 0.5" />
<UNIT name = "lShldrBend" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "353.7, 353.7, 80.8" />
<UNIT name = "lForearmBend" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "3.3, 14.6, 0.8" />
<UNIT name = "rCollar" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "0.0, 0.1, 0.5" />
<UNIT name = "rShldrBend" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "349.2, 353.8, 278.8" />
<UNIT name = "rForearmBend" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "352.3, 347.3, 1.6" />
</parallel></sequence>
<sequence><parallel>
<UNIT name = "abdomenLower" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "354.6, 358.4, 0.9" />
<UNIT name = "chestLower" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "14.7, 2.0, 359.7" />
<UNIT name = "neckUpper" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "0.0, 0.1, 0.5" />
<UNIT name = "lCollar" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "0.0, 0.1, 0.5" />
<UNIT name = "lShldrBend" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "353.7, 353.7, 80.8" />
<UNIT name = "lForearmBend" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "3.4, 14.5, 0.8" />
<UNIT name = "rCollar" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "0.0, 0.1, 0.5" />
<UNIT name = "rShldrBend" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "349.3, 354.1, 278.7" />
<UNIT name = "rForearmBend" durationUp = "0.04" durationDown = "0.01" persistent = "0" transition ="linear" type = "HPR" value = "352.3, 347.5, 1.5" />
</parallel></sequence>
```

Figure 7. An example of procedural animation formulated in EVAScriptMarkup, each <*sequence*><*parallel*> represents configurations P_i, P_{i+1} as the transition between two consecutive frames adjusted to the frame rate scaling. *durationUp* represents the duration of the transition, and is calculated as $\frac{sizeof(H)}{t}$ and the value represents the 3D configuration of the "joint" (movement controller) in Euler angles expressed in roll–pitch–yaw (HPR) notation.

3. Results

In this section, we will present the content related to evaluating the reconstructed motion. Aspects will be presented of the evaluation of the reconstructed movement, evaluation procedures, and the characteristics used in such procedures. Here, we should keep in mind that the main objective is creating natural movement from sources whose content is naturally generated conversation. The existing corpora of reconstructed conversational gestures suffer from a relatively small number of artificially generated gestures. The use of social networks and their video content sharing allows us to obtain resources with the content of naturally generated conversational gestures. Although we can create a large number of reconstructed conversational gestures in this way, we have to evaluate these gestures objectively and subjectively. In our evaluation, we focused on objective evaluation, which allows us to remove poorly reconstructed gestures before evaluating them subjectively.

As already mentioned, the assessment of the similarity between actual and reconstructed movement in most cases comes from the context of use. In some cases, only the rough course of the movement may be of interest, while, in others, the accuracy of the reconstruction of the movement of each joint is necessary. Usually, a 3D reconstructed movement is considered to be similar to the real one if the only difference is a global geometric transformation such as translation or rotation, and the speed of conversational gestures can also be taken into account [56]. In the case of conversational gestures, such an aspect is not satisfactory, since logical and numerical similarities are missing. Aspects taking into account logical and numerical similarities define the logical similarities of movement as several versions of the same action or sequence of actions. In most cases, the algorithms used to evaluate logical and numerical similarities are based on quantitative characteristics [57]. However, it is necessary to consider that a logically similar movement can be numerically different, and vice versa. In many contexts of use, partial similarities are important, and we reconstructed the movement of some parts of the body differently from others. In this aspect, it is necessary to be aware that considering the extracted characteristics in the similarity measure of "unimportant" parts of the body can impact the result negatively [58].

In our case, the similarity between the input signal, tracked using the Kanade–Lucas–Tomasi tracker, and the reconstructed signal, tracked with the same tracker, was calculated using local similarity assessment features. This approach aimed to address the partial, logical, and numerical aspects of the similarity between reconstructed and actual human

movement. The cosine distance features were used to compare the two signals on the entire time axis (marked as sim1) to assess the similarity. At the same time, the similarity of both signals was also assessed at selected time points, representing the moment of the conversational gesture position (*sim2*) with the greatest similarity. An important feature that must be considered in our case is the number of displayed images per second (frame rate) measured in FPS. With this feature, we added a new dimension to the similarity assessment, which captured the aspect of the global transformation with time to a certain extent. By taking into account the time feature, we can evaluate the directly reconstructed gestures quantitatively. We performed an experimental evaluation for two methods: OpenPose and our proposed EVAPose. Due to the fact that the OpenPose method failed to provide results for the original frame rate of the recording, we defined the condition sim2, which enabled us to compare the results of the methods at a similar frame rate. In this, we implemented the proposed EVAPose method in a way that it generated the same frame rate as the OpenPose method. A similarity assessment was made between the input video signal and the reconstructed signal using the proposed EVAPose and OpenPose systems. The similarity score marked with sim1 means the average of the scores of individual gestures. The sim2 rating contains the maximum value from the average of the ratings of individual gestures. Both systems measured both ratings. Similarity scores for the cases used in the evaluation are given in Table 2.

Table 2. Similarity scores for the considered cases. *sim2* in both cases contains moments of the reconstructed signal from the EVAPose with the highest similarity values $\max(sim1)$.

	System Type	**Type of Movement**	**30 FPS**		**Adjusted FPS**	
			sim1	***sim2***	***sim1***	***sim2***
An ideal example, a laboratory environment	EVAPose	Up/down (vertical movement) (46s)—10 gestures	86.85	98.30	91.25 ⊗	98.30
		Left/right (horizontal movement) (46s)—10 gestures (single gesture measured in seconds)	82.3	92.45	89.65 ⊗	96.88
	OpenPose [42]	Up/down (vertical movement)	-	97.2	97.50 ∅	97.19
		Left/right (horizontal movement)	-	79.31	86.75 ∅	90.31
Real case from Gospoda ([47])	EVAPose	Example of up/down gesture (vertical movement) (19 gestures)	82.45	93.56	90.01 ⊗	93.56
		Example of a left/right gesture (horizontal movement) (12 gestures)	79.9	89.72	85.14 ⊗	89.72
	OpenPose [42]	Example of up/down gesture (vertical movement) (19 gestures)	-	95.01	95.33 ∅	96.12
		Example of a left/right gesture (12 gestures)	-	85.91	89.87 ∅	90.33

⊗ 19 FPS; ∅ 11 FPS.

The results of the reconstructed motion from video clips created in the laboratory are shown in the upper part of Table 2. The content of such videos was created to test the system, and does not show naturally generated conversational gestures. The lower part of Table 2 shows the results of conversational gestures, which are gesticulated in a spontaneous and natural way. Table 2 lists one of the tested sources with such content, where the content was, in most cases, a content-varied conversation between two actors with different and spontaneous gestures. The language used by the two actors was Slovenian. The results show how the proposed system compared with the OpenPose system [42]. Conversational gestures containing the most movement in the vertical and horizontal directions were selected for comparison. We considered such types of conversational gestures as the geometric basis for more complex gestures, which consist of a combination of horizontal and vertical movements.

To assess naturalness, we monitored events that occurred during the reconstructed gesture. We noticed that the signal from the OpenPose system did not have a smooth

continuous transition compared to the EVAPose system. At some moments, an unnatural movement occurred, triggered by a sudden jump in the angle of a single joint. To detect such jumps, we analyzed the signal by differentiating the reconstructed signal (position as a function of time) three times (Figure 8). The third derivative showed us the jerks (jumps) and their number. Figure 8 shows the third derivative of the reconstructed signal (position as a function of the number of samples) as a function of the number of samples.

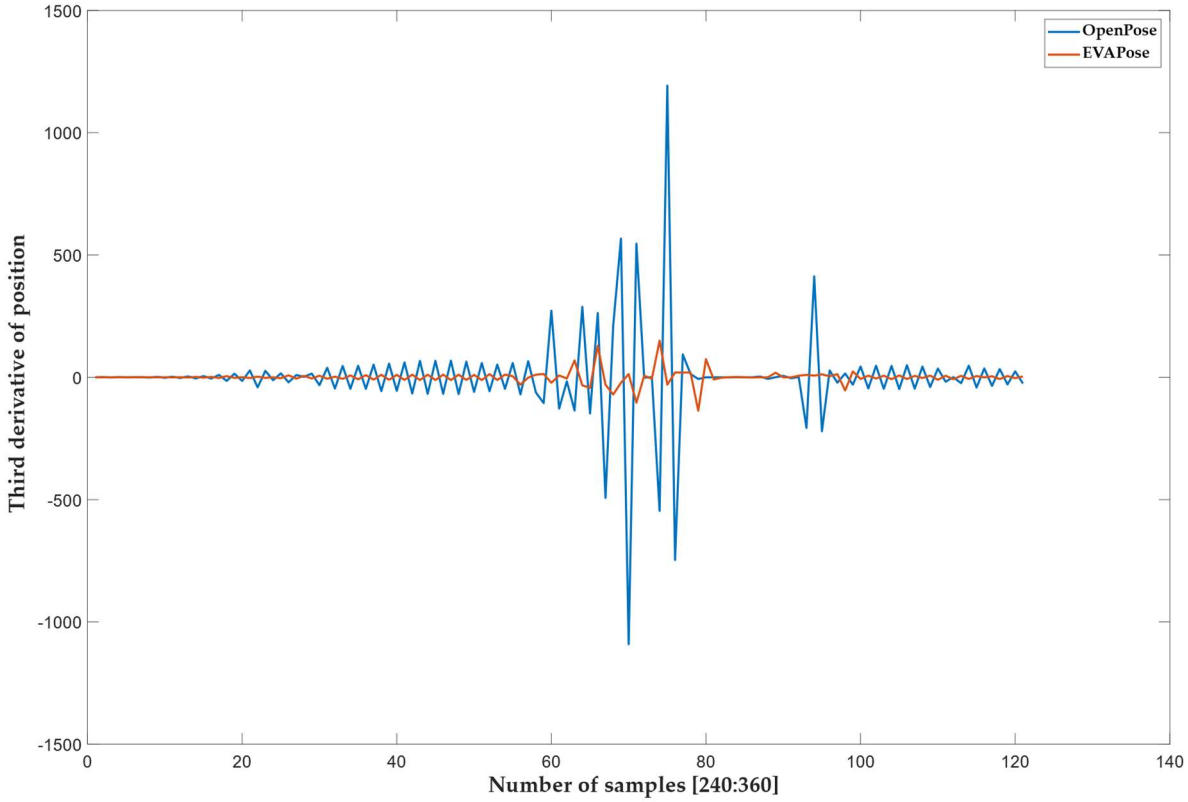

Figure 8. The red and blue curves represent the third derivative (jerk) of the reconstructed signal (position) from EVAPose and OpenPose, respectively. The third derivative is shown as a function of the number of samples. The number and amplitude of jerks in this type of analysis show unnatural and high−energy concentrated spikes on the reconstructed signal. Only a section of the entire signal (120 samples/1380 samples) is shown for easier and better presentation of the reconstructed signal's third derivative.

4. Discussion

It can be seen from Table 2 that, in absolute terms, the similarity score using the OpenPose system was better than the score using the proposed system. The highest frame rate achieved by the OpenPose system was 11 FPS. To achieve comparability, we adjusted the frame rate of the proposed system by increasing the degree of the polynomial in the smoothing process (described in previous chapters). When we reached a frame rate of 19 FPS, the similarity did not improve anymore, but the computational complexity of the proposed method increased tremendously. The same gesture was reconstructed in all videos containing multiple vertical and horizontal movements. By reducing the frame rate in the EVAPose system, it was shown that the similarity had improved. In the case of horizontal movement from the video content created in a laboratory environment, the score exceeded that of the OpenPose system. The laboratory video content with a length of 46 s, which was given to the input of the proposed system, contained 10 gestures that were reconstructed completely successfully.

The results show that, despite the lower average score, a gesture was reconstructed better (in the sense of naturalness) with the proposed system than with the OpenPose

system. The better reconstruction in terms of naturalness was manifested mainly in the jumps of the reconstructed signal. It can be seen from Figure 8 that the EVAPose system had a smaller number of jerks with a larger amplitude. It is also important that the EVAPose system jerks were not aligned with the OpenPose system jerks, which can undoubtedly be attributed to the fact that the jerks are not the result of naturally generated jumps in conversational gestures, but are a statistical error in the algorithms.

The results show that, for the real case from the Gospoda dataset, the proposed system did not reach the similarity level achieved by the OpenPose system; however, the results were comparable. On the other hand, the EVAPose system can maintain the frame rates. In the case with an adjusted frame rate, the video content with a lower frame rate was chosen only to test the EVAPose system at different frame rates. Here, it is worth emphasizing that the preliminary laboratory experiments in the subjective similarity evaluation showed that the objective assessment could reach the subjective level of assessment in all cases.

Despite the better evaluation, the reconstruction with the OpenPose system did not provide a sufficiently large degree of naturalness. Even though the gesture was captured from video content in which a spontaneously created conversational gesture appeared, the OpenPose reconstruction system introduced an error in the reconstruction of the rotation of individual joints, which a human cannot create. We concluded that the cause of such errors comes from the learning process in the learning technologies. Unlike our system, the OpenPose system does not have clearly defined areas of rotation of individual joints in the human body. In our case, we defined the rotation areas of individual joints using the Denavit–Hartenberg notation before calculating forward kinematics, thus ensuring that unnatural joint rotations did not occur. The choice of forward kinematics and the notation of the kinematic model according to Denavit–Hartenberg also allowed a continuous transition of a specific movement.

As already stated, the assessment of similarity is relative in nature, because the systems can be used in different ways, in some of which the similarity assessment is important, and computational complexity of gesture reconstruction does not matter, nor how long the reconstruction process takes. In other cases, however, we wanted to preserve the frame rate, and can settle for a lower similarity score of the speech gesture reconstruction. In this case, we must be aware that the frame rate was preserved at the expense of lower similarity, which was conditioned by the acceptability of the gestures generated in this way, and by the perception of their naturalness and other important characteristics of successful spoken social interaction.

Author Contributions: Conceptualization: G.M.; methodology: G.M., I.M. and R.Š.; software: G.M.; validation: I.M. and Z.K.; formal analysis: Z.K.; writing—original draft preparation: G.M.; writing—review and editing: G.M. and Z.K.; supervision: Z.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Slovenian Research Agency (Research Core Funding) No. P2-0069, Young Researcher Funding 6316-3/2018-255, 603-1/2018-16.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Trujillo, J.P.; Simanova, I.; Bekkering, H.; Özyürek, A. Communicative intent modulates production and comprehension of actions and gestures: A Kinect study. *Cognition* **2018**, *180*, 38–51. [CrossRef] [PubMed]
2. Kelly, S.D.; Özyürek, A.; Maris, E. Two Sides of the Same Coin: Speech and Gesture Mutually Interact to Enhance Comprehension. *Psychol. Sci.* **2010**, *21*, 260–267. [CrossRef] [PubMed]
3. Cassell, J. Embodied Conversational Agents: Representation and Intelligence in User Interfaces. *AI Mag.* **2001**, *22*, 67. [CrossRef]

4. Birdwhistell, R.L. *Kinesics and Context: Essays on Body Motion Communication*; University of Pennsylvania Press: Philadelphia, PA, USA, 2010. [CrossRef]
5. ter Stal, S.; Kramer, L.L.; Tabak, M.; op den Akker, H.; Hermens, H. Design Features of Embodied Conversational Agents in eHealth: A Literature Review. *Int. J. Hum.-Comput. Stud.* **2020**, *138*, 102409. [CrossRef]
6. Philip, P.; Dupuy, L.; Auriacombe, M.; Serre, F.; de Sevin, E.; Sauteraud, A.; Micoulaud-Franchi, J.-A. Trust and acceptance of a virtual psychiatric interview between embodied conversational agents and outpatients. *NPJ Digit. Med.* **2020**, *3*, 2. [CrossRef]
7. Ruttkay, Z. *From Brows to Trust: Evaluating Embodied Conversational Agents*; Human-Computer Interaction Series; Kluwer Academic Publisher: Dordrecht, The Netherlands, 2004; ISBN 978-1-4020-2729-1.
8. Malatesta, L.; Asteriadis, S.; Caridakis, G.; Vasalou, A.; Karpouzis, K. Associating gesture expressivity with affective representations. *Eng. Appl. Artif. Intell.* **2016**, *51*, 124–135. [CrossRef]
9. Graesser, A.C.; Cai, Z.; Morgan, B.; Wang, L. Assessment with computer agents that engage in conversational dialogues and trialogues with learners. *Comput. Hum. Behav.* **2017**, *76*, 607–616. [CrossRef]
10. Lamberti, F.; Paravati, G.; Gatteschi, V.; Cannavò, A.; Montuschi, P. Virtual Character Animation Based on Affordable Motion Capture and Reconfigurable Tangible Interfaces. *IEEE Trans. Vis. Comput. Graph.* **2017**, *24*, 1742–1755. [CrossRef]
11. Bogdanovych, A.; Trescak, T.; Simoff, S. What makes virtual agents believable? *Connect. Sci.* **2016**, *28*, 83–108. [CrossRef]
12. Carreno, P.; Gibet, S.; Marteau, P.-F. Perceptual Validation for the Generation of Expressive Movements from End-Effector Trajectories. *ACM Trans. Interact. Intell. Syst.* **2018**, *8*, 1–26. [CrossRef]
13. Neff, M. Hand Gesture Synthesis for Conversational Characters. In *Handbook of Human Motion*; Springer: Berlin/Heidelberg, Germany, 2018; p. 11.
14. Lee, J.; Marsella, S. *Nonverbal Behavior Generator for Embodied Conversational Agents*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 243–255.
15. Bozkurt, E.; Erzin, E.; Yemez, Y. Affect-expressive hand gestures synthesis and animation. In Proceedings of the 2015 IEEE International Conference on Multimedia and Expo (ICME), Turin, Italy, 29 June–3 July 2015; pp. 1–6.
16. Rojc, M.; Mlakar, I.; Kačič, Z. The TTS-driven affective embodied conversational agent EVA, based on a novel conversational-behavior generation algorithm. *Eng. Appl. Artif. Intell.* **2017**, *57*, 80–104. [CrossRef]
17. Ding, Y.; Huang, J.; Pelachaud, C. Audio-Driven Laughter Behavior Controller. *IEEE Trans. Affect. Comput.* **2017**, *8*, 546–558. [CrossRef]
18. Larboulette, C.; Gibet, S. I Am a Tree: Embodiment Using Physically Based Animation Driven by Expressive Descriptors of Motion. In Proceedings of the 3rd International Symposium on Movement and Computing, Thessaloniki, Greece, 5–6 July 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 1–8.
19. Neff, M.; Pelachaud, C. Animation of Natural Virtual Characters. *IEEE Comput. Graph. Appl.* **2017**, *37*, 14–16. [CrossRef]
20. Mlakar, I.; Kacic, Z.; Borko, M.; Markus, A.; Rojc, M. *Development of a Repository of Virtual 3D Conversational Gestures and Expressions*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 105–110. ISBN 978-3-030-21506-4.
21. Mlakar, I.; Kačič, Z.; Borko, M.; Rojc, M. A Novel Realizer of Conversational Behavior for Affective and Personalized Human Machine Interaction—EVA U-Realizer. *WSEAS Trans. Environ. Dev.* **2018**, *14*, 15.
22. Sadoughi, N.; Busso, C. Speech-driven Animation with Meaningful Behaviors. *arXiv* **2017**. [CrossRef]
23. Bergmann, K.; Kopp, S.; Eyssel, F. Individualized Gesturing Outperforms Average Gesturing—Evaluating Gesture Production in Virtual Humans. In *Intelligent Virtual Agents*; Allbeck, J., Badler, N., Bickmore, T., Pelachaud, C., Safonova, A., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6356, pp. 104–117. ISBN 978-3-642-15891-9.
24. Jacobson, A.; Panozzo, D.; Glauser, O.; Pradalier, C.; Hilliges, O.; Sorkine-Hornung, O. Tangible and modular input device for character articulation. *ACM Trans. Graph.* **2014**, *33*, 82:1–82:12. [CrossRef]
25. Liang, H.; Deng, S.; Chang, J.; Zhang, J.J.; Chen, C.; Tong, R. Semantic framework for interactive animation generation and its application in virtual shadow play performance. *Virtual Real.* **2018**, *22*, 149–165. [CrossRef]
26. Rhodin, H.; Tompkin, J.; In Kim, K.; Varanasi, K.; Seidel, H.-P.; Theobalt, C. Interactive motion mapping for real-time character control. *Comput. Graph. Forum* **2014**, *33*, 273–282. [CrossRef]
27. Nirme, J.; Haake, M.; Gulz, A.; Gullberg, M. Motion capture-based animated characters for the study of speech–gesture integration. *Behav. Res. Methods* **2020**, *52*, 1339–1354. [CrossRef]
28. Zhang, Y.; Luo, X.; Yang, W.; Yu, J. Fragmentation Guided Human Shape Reconstruction. *IEEE Access* **2019**, *7*, 45651–45661. [CrossRef]
29. Vlasic, D.; Popovic, J.; Peers, P.; Baran, I.; Debevec, P.; Matusik, W. Dynamic Shape Capture using Multi-View Photometric Stereo. *ACM Trans. Graph.* **2009**, *28*, 1–11. [CrossRef]
30. Lin, J.-L.; Hwang, K.-S. Balancing and Reconstruction of Segmented Postures for Humanoid Robots in Imitation of Motion. *IEEE Access* **2017**, *5*, 17534–17542. [CrossRef]
31. Dou, M.; Khamis, S.; Degtyarev, Y.; Davidson, P.; Fanello, S.R.; Kowdle, A.; Escolano, S.O.; Rhemann, C.; Kim, D.; Taylor, J.; et al. Fusion4D: Real-time performance capture of challenging scenes. *ACM Trans. Graph.* **2016**, *35*, 1–13. [CrossRef]
32. Slavcheva, M.; Baust, M.; Cremers, D.; Ilic, S. KillingFusion: Non-rigid 3D Reconstruction without Correspondences. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5474–5483.

33. Leroy, V.; Franco, J.-S.; Boyer, E. Multi-view Dynamic Shape Refinement Using Local Temporal Integration. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 3113–3122.
34. Aliakbarpour, H.; Prasath, S.; Palaniappan, K.; Seetharaman, G.; Dias, J. Heterogeneous Multi-View Information Fusion: Review of 3-D Reconstruction Methods and a New Registration with Uncertainty Modeling. *IEEE Access* **2016**, *4*, 8264–8285. [CrossRef]
35. Pelachaud, C. Greta, an Interactive Expressive Embodied Conversational Agent. In Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Istanbul, Turkey, 4–8 May 2015.
36. Sun, X.; Lichtenauer, J.; Valstar, M.; Nijholt, A.; Pantic, M. *A Multimodal Database for Mimicry Analysis*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 367–376.
37. Knight, D. *Multimodality and Active Listenership: A Corpus Approach*; Research in Corpus and Discourse; Continuum: London, UK, 2011; ISBN 978-1-4411-6723-1.
38. Rogez, G.; Weinzaepfel, P.; Schmid, C. LCR-Net: Localization-Classification-Regression for Human Pose. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1216–1224. [CrossRef]
39. Zhou, X.; Huang, Q.; Sun, X.; Xue, X.; Wei, Y. Towards 3D Human Pose Estimation in the Wild: A Weakly-Supervised Approach. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 398–407. [CrossRef]
40. Habermann, M.; Xu, W.; Zollhöfer, M.; Pons-Moll, G.; Theobalt, C. LiveCap: Real-Time Human Performance Capture From Monocular Video. *ACM Trans. Graph.* **2019**, *38*, 14:1–14:17. [CrossRef]
41. Liang, G.; Zhong, X.; Ran, L.; Zhang, Y. An Adaptive Viewpoint Transformation Network for 3D Human Pose Estimation. *IEEE Access* **2020**, *8*, 143076–143084. [CrossRef]
42. Cao, Z.; Simon, T.; Wei, S.-E.; Sheikh, Y. Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1302–1310.
43. Marcus, G. Deep Learning: A Critical Appraisal. *arXiv* **2018**. [CrossRef]
44. Kim, T.T.; Zohdy, M.A.; Barker, M.P. Applying Pose Estimation to Predict Amateur Golf Swing Performance Using Edge Processing. *IEEE Access* **2020**, *8*, 143769–143776. [CrossRef]
45. KLT: Kanade-Lucas-Tomasi Feature Tracker. Available online: https://cecas.clemson.edu/~{}stb/klt/ (accessed on 1 August 2022).
46. Denavit, J.; Hartenberg, R.S. A kinematic notation for lower-pair mechanisms based on matrices. *Trans ASME E J. Appl. Mech.* **1955**, *22*, 215–221. [CrossRef]
47. Godler, J.; Urankar, D. Gospoda. Available online: https://www.youtube.com/c/Gospodapodcast (accessed on 1 August 2022).
48. Hanke, T. HamNoSys—Representing Sign Language Data in Language Resources and Language Processing Contexts. In Proceedings of the 4th International Conference on Language Resources and Evaluation, Lisbon, Portugal, 26–28 May 2004.
49. Shi, J.; Tomasi. Good features to track. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR-94, Seattle, WA, USA, 21–23 June 1994; pp. 593–600.
50. Quan, M.; Mu, B.; Chai, Z. IMRL: An Improved Inertial-Aided KLT Feature Tracker. In Proceedings of the 2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), Bangkok, Thailand, 18–20 November 2019. [CrossRef]
51. Lv, G. Self-Similarity and Symmetry With SIFT for Multi-Modal Image Registration. *IEEE Access* **2019**, *7*, 52202–52213. [CrossRef]
52. Savitzky, A.; Golay, M.J.E. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Anal. Chem.* **1964**, *36*, 1627–1639. [CrossRef]
53. Jahani, S.; Setarehdan, S.K.; Boas, D.A.; Yücel, M.A. Motion artifact detection and correction in functional near-infrared spectroscopy: A new hybrid method based on spline interpolation method and Savitzky-Golay filtering. *Neurophotonics* **2018**, *5*, 015003. [CrossRef] [PubMed]
54. Schafer, R.W. What Is a Savitzky-Golay Filter? [Lecture Notes]. *IEEE Signal Process. Mag.* **2011**, *28*, 111–117. [CrossRef]
55. Atique, M.M.U.; Sarker, M.R.I.; Ahad, M.A.R. Development of an 8DOF quadruped robot and implementation of Inverse Kinematics using Denavit-Hartenberg convention. *Heliyon* **2018**, *4*, e01053. [CrossRef]
56. Röder, T. Similarity, Retrieval, and Classification of Motion Capture Data. Ph.D. Thesis, Rheinische Friedrich-Wilhelms-Universität, Bonn, Germany, 2007.
57. Kovar, L.; Gleicher, M. Automated extraction and parameterization of motions in large data sets. *ACM Trans. Graph.* **2004**, *23*, 559–568. [CrossRef]
58. Chen, S.; Sun, Z.; Li, Y.; Li, Q. Partial Similarity Human Motion Retrieval Based on Relative Geometry Features. In Proceedings of the 2012 Fourth International Conference on Digital Home, Guangzhou, China, 23–25 November 2012; pp. 298–303.