Modeling and Composing Gestures for Human-Robot Interaction

Junyun Tay
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213, USA
junyun@cmu.edu

Manuela Veloso
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
veloso@cmu.edu

***Abstract*— We formalize the representation of gestures and present a model that is capable of synchronizing expressive and relevant gestures with text-to-speech input. A gesture consists of gesture primitives that are executed simultaneously. We formally define the gesture primitive and introduce the concept of a spatially targeted gesture primitive, i.e., a gesture primitive that is directed at a target of interest. The spatially targeted gesture primitive is useful for situations where the direction of the gesture is important for meaningful human-robot interaction. We contribute an algorithm to determine how a spatially targeted gesture primitive is generated. We also contribute a process to analyze the input text, determine relevant gesture primitives from the input text, compose gestures from gesture primitives and rank the combinations of gestures. We propose a set of criteria that weights and ranks the combinations of gestures. Although we illustrate the utility of our model, algorithm and process using a NAO humanoid robot, our contributions are applicable to other robots.**

I. Introduction and Background

Humans communicate via speech and gestures with each other. Gestures are used based on experiences without any misunderstandings. Robots that interact with people are usually used as service robots, entertainment or health care. To enable effective communication with people, robots expressing themselves through similar gestures can help greatly. When anthropomorphic robots display emotions, humans respond to them in predictable ways [1].

Input text provides context for human-robot interaction as robots express the meanings of the text through appropriate emotional speech and relevant gestures at the right moments. We are interested in automating the task of modeling gestures and composing gestures based on the analysis of the input text. The automatically generated gestures should satisfy several goals. First, gestures should be dynamically stable and safe (the robot should not collide with itself). Second, gestures should reflect the emotions and meanings determined from the input text. Third, gestures to be conveyed to a target of interest should be automatically generated given the target's pose (position and orientation). Finally, gestures generated should be ranked and synchronized to the speech generated from the input text. Ranked gestures can enable other viable options to be probabilistically selected.

Gestures have been generally organized into a few categories [2], [3], [4]:

- *Emblems* are commonly understood without speech, self-explanatory, but can be culturally-specific.
- *Iconics* depict the characteristics of physical concrete things with the motion of the hands.
- *Metaphorics* describe abstract concepts, e.g., referring to both sides of the arguments using both hands.
- *Deictics* point to items that are being described.
- *Beats* are small, short movements matching the rhythm of speech to convey emphasis, emotion and personality.

There are other types of gestures that are less commonly used, such as turn-taking gestures known as *regulators*, e.g., a person wanting to speak raises an arm or *affect displays* that display emotions.

To address the goals of automatically generating gestures, we define gesture primitives as building blocks for a gesture, as combinations of gesture primitives enable a greater variety of gestures. For example, the gesture of shaking one's head and waving two hands, indicating no, can be made up of three gesture primitives and these gesture primitives can be reused for other situations as shown in Fig. 1. Besides deictics that are used to point at things, existing formalization of gestures do not capture the notion of direction such that certain gestures have to be directed at something in the process of communication. For instance, Character A is waving goodbye to Character B. Hence, A waves goodbye to B with A's palm facing B. Therefore, we introduce a new gesture primitive, spatially targeted gesture primitive, as compared to general gesture primitives with no directional needs. We also classify gesture primitives by body parts so as to allow simultaneous selection of gestures primitives during the planning of the gestures to execute. We describe the parametrization of gesture primitives and explain the purpose of each parameter. Next, we contribute an algorithm to instantiate a parametrized gesture primitive based on the target's pose and determine the resulting pose of the robot.

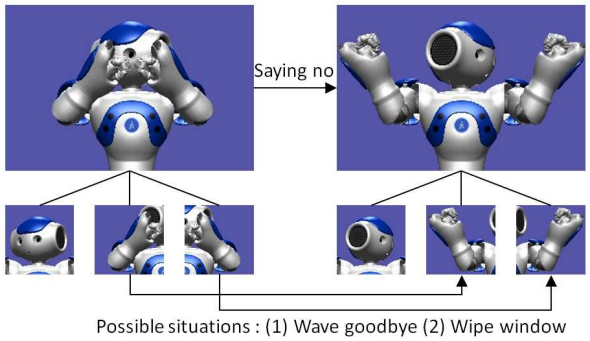
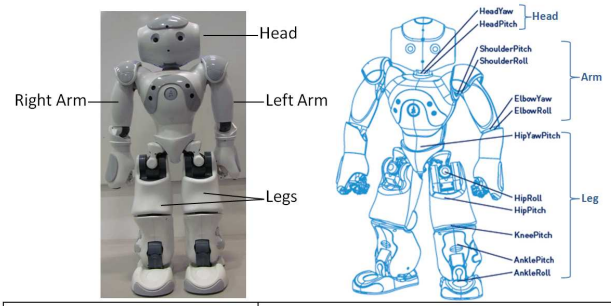


Fig. 1. Gesture Composition and possible combinations for other situations

Finally, we describe a process with three phases to automatically select gesture primitives and generate gestures from

the combinations of gesture primitives. We propose a set of criteria to rank the possible combinations and include higher weights for criteria that are more important to users. We also illustrate how the process works with a NAO humanoid robot (Fig. 2) gesturing with an example of a text input. While we use the NAO robot, our formalization, algorithm and process are applicable to other robots as well.



Body Part Categories	Joints
Head	HeadYaw, HeadPitch
Left arm (LArm)	LShoulderPitch, LShoulderRoll, LElbowYaw, LElbowRoll
Right arm (RArm)	RShoulderPitch, RShoulderRoll, RElbowYaw, RElbowRoll
Legs	LHipYawPitch, LHipRoll, LHipPitch, LKneePitch, LAnklePitch, LAnkleRoll, RHipRoll, RHipPitch, RKneePitch, RAnklePitch, RAnkleRoll

Fig. 2. NAO humanoid robot's body parts and joints

II. Related Work

To express a variety of emotions, gestures have been modified in several ways. [5] classifies different gestures with fixed keyframes for a particular emotion, whereas [3] designs gesture templates that consists of a basic trajectory shape which are stored as a set of key points for each parameter value. [3] varies the gesture templates using Kochanek-Bartels (TCB) cubic splines, a type of interpolation method, where TCB (tension-continuity-bias) splines allows control of how smoothly or tightly the trajectories follow the key points to express different emotions. For our formalization of a gesture primitive as compared to a gesture by [3], we do not cap the maximum time between keyframes, but define a minimum time using the fastest possible joint speed. We also allow the use of any interpolation methods but note that different interpolation methods to affect the trajectory of the gesture may increase the number of parameters in the formalization of a gesture. The increase of parameters may also cause difficulties for gesture designers to visualize the shape of the trajectories and determine the appropriate parameters for different emotions.

Research has also been done on how movements can affect the perception of emotions [5], [6]. Timing and velocity of movements are associated with different emotions, e.g., fast velocity is associated with anger and happiness, while a slow velocity is associated with sadness [6]. Our gesture primitives allow the manipulation of the timings of gestures that will affect the velocity and hence express varied emotions. Emotional rules can be defined for the timings of the gestures.

The BEAT gesture system selects gestures using an extensible rule set and uses beat gestures as a default gesture when no other gestures are possible [7]. [3] extends the BEAT gesture model by determining the relative probability for the occurrence of each gesture type (emblem, metaphoric, iconic, deictic and beat gestures) with an expressivity parameter, and also includes an option to do nothing to prevent excessive gestures. Instead of selecting gestures based on the gesture category, we propose a number of criteria and weight the criteria to rank the combinations of gestures.

To facilitate effective human-robot interaction, in the area of appropriate proxemic behaviors, [8], [9] conducted studies of appropriate distances for non-verbal behaviors, such as gazes. To allow users to control proxemic behavior, we include proxemic parameters in our gesture primitives.

III. Formalization

We formally define gestures, keyframes, gesture primitives, and gesture primitive categories. We explain the need to have various types of keyframes and gesture primitives and how various parameters and categories can effectively reduce the number of expressive gesture primitives defined.

A. Modeling Gestures

Gestures are movements that express an idea or meaning. To execute gestures on a robot for human-robot interaction, the joints of the robot have to be actuated. A robot can only actuate its joints within the angular joint limits and speeds.

Definition 3.1: A robot has a series of j actuated joints with the corresponding joint limits and velocities, $\{(J_1, L_{1,\min}, L_{1,\max}, V_{1,\max}), \ldots, (J_j, L_{j,\min}, L_{j,\max}, V_{j,\max})\}$, $J_i \neq J_j$. The joint index is J_i, the minimum and maximum angle is $L_{i,\min}$, $L_{i,\max}$ and the maximum velocity is $V_{i,\max}$.

B. Keyframe

A keyframe (static pose) stores the joints and corresponding angles at a particular time step. For a robot to perform a gesture, several keyframes are stored at different time steps and interpolated to form a continuous motion.

Definition 3.2: A keyframe with fixed joint angles, $k_d = \{(J_1, \theta_1), \ldots, (J_n, \theta_n)\}, J_i \neq J_j$ and $n \leq j$. Let the set of keyframes with fixed joint angles be $K_d = \bigcup k_d$.

While having keyframes that have clearly defined joint angles enable gesture designers or users to know the exact motion of a gesture, it does not enable flexibility in defining gestures that have different starting positions. For example, Fig. 3 shows a gesture of nodding the head at different yaw angles with the same pitch angle changes. If keyframes of

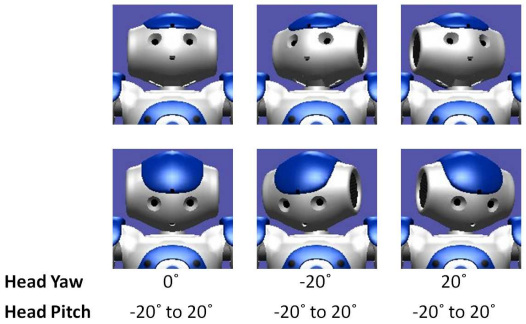


Fig. 3. Head nods - different yaw angles, same pitch angle changes

type k_d are used, all combinations of different yaw angles have to be defined. This problem can be solved by defining relative changes for certain joints to the previous keyframe.

Definition 3.3: A variable keyframe, $k_v(\alpha) = \{(J_1, \theta_{1,\text{min}}, \theta_{1,\text{max}}), \ldots, (J_n, \theta_{n,\text{min}}, \theta_{n,\text{max}})\}$ where $\theta_{i,\text{min}}$ and $\theta_{i,\text{max}}$ contains the minimum and maximum relative change for the joint with index J_i, $J_i \neq J_j$ and $n \leq j$. Let $K_v = \bigcup k_v$ be the set of all variable keyframes.

To determine the joint angle in k_v for joint index J_i, we use a parameter $\alpha \in [0, 1]$, to determine the amplitude of the relative change of J_i, $\theta_{J_i} = \alpha \cdot (\theta_{i,\text{max}} - \theta_{i,\text{min}}) + \theta_{i,\text{min}}$. Hence, with α, a variable keyframe k_v becomes a clearly defined keyframe k_d, so k_v is a parameterized form of k_d. Therefore, we have 2 different types of keyframes, where $K = K_d \cup K_v$ and K is the entire set of keyframes.

C. Gesture Primitive

A gesture is made up of several gesture primitives, which can be a general or a spatially targeted gesture primitive.

1) General Gesture Primitive:

Definition 3.4: A general gesture primitive is composed of keyframes, $g_p(\beta, N) = (k_1, \beta T_1, k_2, \ldots, k_{f-1}, \beta T_{f-1}, k_f)$, where f is the number of keyframes in g_p, and T_{f-1} is the minimum time that it takes to interpolate from one keyframe, k_{f-1} to the next keyframe k_f and is determined by the joints' maximum speed. Let $G_p = \bigcup g_p$ be the set of all general gesture primitives.

We parameterize the gesture primitive with β, where $\beta \in \mathbb{R}$ and $\beta \geq 1$. β is determined by the duration required to complete the gesture primitive based on factors such as the duration of the word, or emotional rules, and is used as a multiplying factor. As some gestures can be repetitive, the parameter N indicates the number of times to execute.

2) Spatially Targeted Gesture Primitive (STG): Most gestures are directed at a point of interest or target. However, to our knowledge, existing formalizations of gestures cannot automatically direct the gestures at a target based on the parameters of the gesture. Hence, we define another type of gesture primitive, spatially targeted gesture (STG), g_{st}, that includes more parameters to define the direction. A STG can be directed at a point or a vector in a particular direction, e.g., to look at a bird in the sky, the robot directs its head to look at a point in space. However, in the case of facing someone, the target's orientation is defined as a vector, instead of a point, since the robot has to look at the face of the person.

Definition 3.5: A spatially targeted gesture is: $g_{st}(\beta, N, P_s, P_e) = (k_{v,1}, \beta T_1, k_{v,2}, \ldots, k_{v,f}, D_{\text{min}}, D_{\text{max}})$, where P_s, P_e are two ego-centric coordinates, used to define the vector $\mathcal{V}$, a direction the robot's STG's first keyframe is at. Let $G_{st} = \bigcup g_{st}$ be the set of spatially targeted gestures.

With $\mathcal{V}$, we calculate the pose of the robot so as to execute the STG that is directed at its desired target. Proximity studies can help to define these 2 parameters, $D_{\text{min}}, D_{\text{max}}$, the minimum and maximum distance the STG can be at, so that if the STG's distance to the target is within the defined range, the STG is executed. Otherwise, the robot's position is adjusted so that the STG can be executed. This is also useful for gestures that require a certain distance to the target, e.g., shaking hands with someone.

D. Gesture Primitive Categories

We categorize gestures and gesture primitives as shown in Fig. 4. Gesture primitives, GT $= G_p \cup G_{st}$, are made up of general gesture primitives and spatially targeted gesture primitives. Gesture primitives are categorized according to a group of joints that actuates independently of other groups of joints. E.g., for humanoid robots, we group gestures according to body parts. For the NAO, we group the joints into 4 categories: head, left arm (LArm), right arm (RArm) and legs (Fig. 2). Hence, our gesture primitives are labelled with g_c where $c \in$ Head, LArm, RArm, Legs. With these categorizations, we select gesture primitives to execute simultaneously and emphasize what the robot is expressing. For example, with a left arm gesture primitive shaking the fist angrily and a right arm gesture primitive shaking the fist angrily, we combine both gesture primitives to emphasize anger. We also note that a gesture primitive may be categorized into more than one category given that the gesture primitive cannot be separated. E.g., a single gesture primitive that expresses anger by staring at someone is composed of a head movement and each arm moving to the side of the hips.

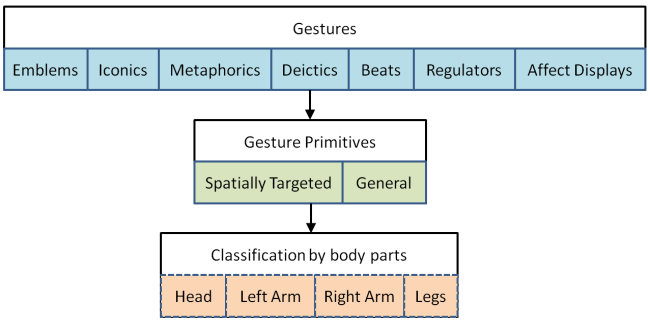


Fig. 4. Classification of Gestures and Gesture Primitives

Gesture primitives are associated with different bags of words to enable gesture primitives selection, so that we can use the input text to drive the selection process by automatically narrowing down a list of gesture primitives. Some gesture primitives have multiple meanings, e.g., a gesture primitive, g, with a left hand waving side to side action, is associated with these bags of words, $B_1 = wave, goodbye, B_2 = no, B_3 = wipe, window$. Being associated with different bags of words will reduce the number of gesture primitives defined. GetBagsOfWords(g) returns $B_1, \ldots, B_w$, where each B_i contains a list of words, the first word being the most relevant word and $w \in \mathbb{Z}^+$ is the total number of bags of words associated with g.

IV. Algorithms

In this section, we present an algorithm to generate a STG and determine the robot's pose. We introduce a process to generate relevant, executable gestures based on the input text.

A. Generating Spatially Targeted Gestures (STGs)

To generate a spatially targeted gesture, we need the target's pose and the robot's current pose. Fig. 5 illustrates examples of adjusting the robot's pose based on the target.

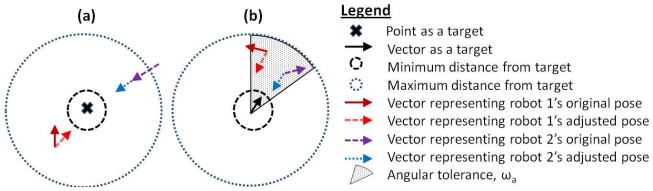


Fig. 5. Examples of robot's adjusted poses to face a point or vector target.

Algorithm 1 determines the robot's global pose using its original global pose, R_p, given a known STG, g_{st}, and a target, t, so as to direct the robot to face the target. A target can be a point, t_s, or a vector, $\overrightarrow{t_s t_e}$, defined in global coordinates. ω_a in Fig. 5 provides an angular tolerance, where $|O_{gt} - O_t| \leq \omega_a$, where O_{gt} and O_t are the current and desired STG orientations respectively. The function convertRelativeToGlobal(P) converts any point relative to the robot to global coordinates. The function canUpdateSTG(g_{st}, O_t) performs several checks and updates to determine the final robot position P_f and orientation O_f: (a) It checks if g_{st} can be updated to face the target at a global orientation angle of O_t and this function returns True if it is possible and updates g_{st}, otherwise it returns False and the robot's orientation will be updated. (b) Since g_{st} includes variable keyframes, canUpdateSTG(g_{st}, O_t) determines if the gesture is able to execute with the parameters specified. (c) canUpdateSTG(g_{st}, O_{gt}) also checks if the joint angular changes stay within the joints' angular limits. For example, if the knee pitch joint only actuates from $-20°$ to $20°$, and the current knee pitch angle is $-10°$ and the variable keyframe specifies a relative change of $-15°$, the knee pitch joint cannot turn to $-25°$. Thus, the robot's orientation is updated.

After determining the global orientation, we check if the position of the robot needs to be changed given the minimum and maximum distance the gesture primitive, g_{st}, can be executed. If the robot's position has to be updated, the robot will be placed at a distance of $D_{\text{mid}} = \frac{D_{\text{min}} + D_{\text{max}}}{2}$. Algorithm 1 is written for a 2-dimensional space scenario, but can be extended to a 3-dimensional space.

Algorithm 1 Determines the final pose of the robot.

DeterminePoseForSTG$(g_{st}, t, R_p, \omega_a)$

$P_{gs} \leftarrow$ convertRelativeToGlobal(P_s) // P_s is from g_{st}.
$P_{ge} \leftarrow$ convertRelativeToGlobal(P_e) // P_e is from g_{st}.
$O_{gt} \leftarrow \text{atan2}(P_{ge}.y - P_{gs}.y, P_{ge}.x - P_{gs}.x)$
if t is a point **then**
  $O_t \leftarrow \text{atan2}(t_s.y - P_{gs}.y, t_s.x - P_{gs}.x)$
else if t is a vector **then**
  $O_t \leftarrow 2\pi - \text{atan2}(t_e.y - t_s.y, t_e.x - t_s.x)$
if $|O_{gt} - O_t| \leq \omega_a$ **then**
  $O_f \leftarrow R_p.\theta$
else if canUpdateSTG(g_{st}, O_t) **then**
  $O_f \leftarrow R_p.\theta$
else
  $O_f \leftarrow R_p.\theta + (O_t - O_{gt})$
dist $\leftarrow \sqrt{(P_{gs}.x - t_s.x)^2 + (P_{gs}.y - t_s.y)^2}$
$D_{\text{mid}} \leftarrow \frac{D_{\text{max}} + D_{\text{min}}}{2}$
if t is a point **then**
  if dist $\geq D_{\text{min}}$ and dist $\leq D_{\text{max}}$ **then**
    $P_f \leftarrow R_p$
  else
    $P_f.x \leftarrow (t_s.x - D_{\text{mid}} * \cos(O_t)) - P_{gs}.x + R_p.x$
    $P_f.y \leftarrow (t_s.y - D_{\text{mid}} * \sin(O_t)) - P_{gs}.y + R_p.y$
else if t is a vector **then**
  if dist $\geq D_{\text{min}}$ and dist $\leq D_{\text{max}}$ and $|\text{atan2}(t_s.y - P_{gs}.y, t_s.x - P_{gs}.x) - O_t| \leq \omega_a$ **then**
    $P_f \leftarrow R_p$
  else
    $\gamma \leftarrow \frac{t_e.y - t_s.y}{t_e.x - t_s.x}$
    $P_f.x \leftarrow (t_s.x - \frac{D_{\text{mid}}^2}{\gamma^2+1}) - P_{gs}.x + R_p.x$
    $P_f.y \leftarrow (t_s.y - \gamma\frac{D_{\text{mid}}^2}{\gamma^2+1}) - P_{gs}.y + R_p.y$
return P_f, O_f

B. Composing Gestures

We describe a process to analyze the input text, select relevant gesture primitives, and rank combinations of executable gesture primitives generated to form gestures based on a weighted list of criteria proposed.

We divide the process into three phases as shown in Fig. 6:

1) Phase 1: Text Analysis: A textual emotion recognition system, e.g., Synesketch [10], identifies an emotion (one of Paul Ekman's six basic emotions: happy, sad, surprise, fear, angry and disgust) and the intensity of the emotion, a value between 0 to 1 from the input text (a sentence). Next, the sentence, the emotion and intensity are passed into an emotional text-to-speech system, e.g., Festival [11] and MBROLA [12].The emotional text-to-speech system can also provide the timings of each word and also allow pauses to be inserted in between words.

2) Phase 2: Gesture Primitives Analysis: Given the sequence of words, $W_1 \ldots W_n$, from the input text, we can compare each word with the bags of words associated with the gesture primitives in the database. For each word, there can be different number of gesture primitives found.

Next, after selecting the relevant gesture primitives, we generate values for each gesture primitive's parameters. The values are filled in by the rules for gesture primitives and the duration of each word. The rules for gesture primitives can include the number of times a gesture primitive should be repeated and the target's information for a spatially targeted gesture primitive. If the gesture primitive cannot be completed within the duration of each word, a pause can be inserted after the word to give the gesture primitive enough time to complete. We generate the gesture primitive and determine the robot's pose using Algorithm 1.

3) Phase 3: Gesture Ranking: With the list of gestures generated for each word in the input text, we also include the choice to do nothing for each word in the input text. From the

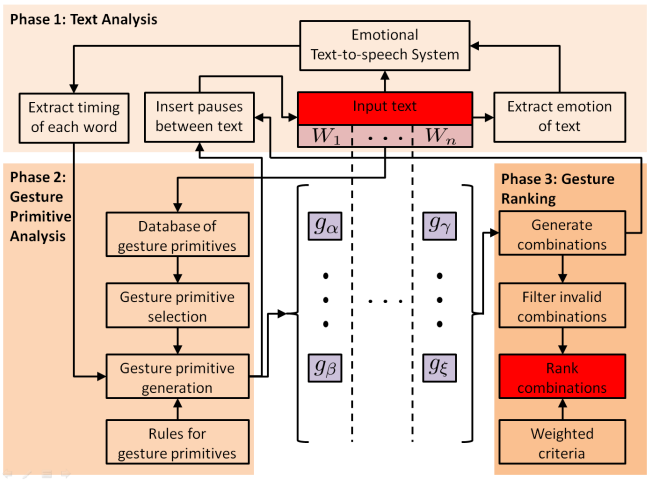


Fig. 6. Process of composing gestures from text input and gesture primitives. The process starts from the red box, "Input text" and ends at the red box, "Rank combinations".

list of choices for each word, we generate all combinations. We note that multiple gesture primitives for each word can be executed simultaneously if they are from different gesture primitives categories, $c \in$ Head, LArm, RArm, Legs. Hence, multiple gesture primitives can be considered as a choice for a particular word. Once we determine the gesture primitives for each word, the gesture primitives form a gesture and can be categorized by the purpose or meaning shown in Fig. 4.

Between each gesture, if the time between two words is shorter than the time needed to interpolate from one gesture to another, pauses will be inserted. Moreover, the pose of the robot can change due to the execution of spatially targeted gestures which may also require more time for the robot to move. After that, we filter the combinations for invalid combinations by performing collision checking when the robot interpolates from one gesture to another so that the robot's gestures will not collide with its own body parts. We also discard combinations that cause the robot to fall.

After finding the combinations that are valid and executable, we propose the following criteria to rank them:

- Length of pauses inserted - Pauses are inserted to allow time to execute the gestures. Inserting more pauses may decrease the flow of speech and continuity of gestures. Hence, we penalize the ranking of a combination if the length of pauses inserted is longer than another.
- Intensity of emotion - Given the intensity of an emotion, the number of times a gesture primitive is repeated and the number of gesture primitives that can be executed simultaneously can be varied. We rank gestures that reflect the intensity of the emotion better higher. E.g., if the emotion intensity is high, the higher the number of times a gesture primitive is repeated and/or the higher the number of gesture primitives that can be executed simultaneously, the intensity of the emotion is better reflected and the combination is ranked higher.
- Bags of words - Each gesture primitive is associated with different bags of words. Gesture designers can manually define the bags of words associated with the gesture primitive. Based on the text input, there may be different overlaps of words with different bags of words for different gesture primitives. With more overlap of words, we assume that the gesture primitives will be more relevant. If the exact words cannot be found, synonyms can be used instead to determine if there are matches. However, gesture primitives with bags of words as synonyms are ranked lower.
- Spatially targeted versus general gesture primitive - When the input text references a target of interest, a gesture with a spatially targeted gesture primitive is ranked higher than one with a general gesture primitive.
- Expressivity - [3] uses a expressivity parameter to determine the distribution of gesture occurrence for each gesture category. The user can determine the distribution of gesture occurrence for each gesture category and the distribution of gesture occurence for each combination that matches the user's requirements is ranked higher.

The ranking for each criterion is weighted based on the user's needs. To determine the ranking of a combination, $R_i = \sum_{j=1}^{|C|} W_j R_{i,j}$, where i is the ith combination, $|C|$ is the total number of criteria, j is the index of the criterion, $R_{i,j}$ is the ranking for the combination i under criterion j and W_j is the weight of the criterion j. The higher the weighting for a criterion, the more important the criterion is to the user. The best combination has a ranking of the least R_i. Other ranked combinations can also be chosen probabilistically to learn the preferences of the audience.

V. Results

To demonstrate how our model, algorithm and process works, we used the text input, "Little Red Riding Hood looked at her grandmother and gasped out in surprise, 'Oh! Grandmother, what a big mouth you have!' " as an example. We shall now walk through each phase in the process:

A. Phase 1: Text Analysis

We pass the input text into Synesketch [10] and determine the emotion of the text to be "surprise" and the intensity to be 1.0. Given the emotion, "surprise", an emotional intensity of 1.0, and the sentence, we determine the starting times (in seconds) of each word using an emotional text-to-speech system, Festival [11] and MBROLA [12] (Table I).

TABLE I
Timings of words in text input

Little	Red	Riding	Hood	looked	at	her	grandmother
0.18	0.54	0.80	1.20	1.42	1.72	1.83	2.02
and	gasped	out	in	surprise,	"Oh!	Grandmother,	what
2.97	3.14	3.65	3.84	3.96	4.77	4.93	5.92
a	big	mouth	you	have!"			
6.092	6.15	6.39	6.76	6.92	7.12		

B. Phase 2: Gesture Primitive Analysis

After extracting the timings of each word from the input text, we select gesture primitives from the database that contains words from the sentence in the bags of words associated

with each gesture primitive. In Table II, the gesture primitives found are listed with other relevant information such as the bags of words associated with the gesture primitive.

TABLE II

GESTURE PRIMITIVES SELECTED

Word	Gesture primitive	Bags of words	Total minimum duration (s)	Body part categorization
looked	$g_{st,1}$	look stare	0.06	Head
looked	$g_{st,2}$	peer	0.1	Head, left and right arms
surprise	$g_{p,1}$	surprise	1.5	Head, left and right arms
surprise	$g_{p,2}$	surprise	0.5	Head and Legs
big	$g_{st,3}$	big	0.3	Left and right arms
big	$g_{st,4}$	big	1	Legs

Given the durations of words and the target of interest, in this case, the vector representing the pose of "Grandmother", and the requirement that each gesture primitive selected is only performed once, each gesture primitive is generated by determining the values for the parameters using Algorithm 1. To generate gesture primitives that have a longer total minimum duration compared to the duration of the word, we insert pauses after the word to allow time for executing the gesture primitive.

C. Phase 3: Gesture Ranking

After we generate each gesture primitive, we determine the list of combinations. For this example, we have a total of $3 \times 3 \times 4 = 36$ combinations as we include the choice to do nothing for each word, and for the word, "big", we can execute $g_{st,3}$ and $g_{st,4}$ simultaneously, hence adding another choice. After generating all combinations, we filter for invalid combinations by checking for collisions. We discard 4 combinations that involve $g_{st,2}$ and $g_{p,1}$ as the arms collide with the head. We also discard 3 combinations that include $g_{p,2}$ and $g_{st,4}$, and 3 combinations that consist of $g_{p,2}$ and $g_{st,3}, g_{st,4}$ as the robot will fall. There are 6 other combinations that cause instability of the robot. Hence, we are left with only $36 - 4 - 3 - 3 - 6 = 20$ possible combinations. Lastly, we rank each gesture combination based on the criteria listed in Section IV-B.3 and use a weighting of 1 for each criteria since all the criteria are equally important in this case. Fig. 7 shows snapshots of NAO executing the highest ranked gesture combination. The NAO looks in the direction where the character "Grandmother" is at, expresses surprises and expresses how big her mouth is.

VI. CONCLUSIONS

We explain the need for various types of keyframes and gesture primitives. We also show how the parametrization of keyframes and gesture primitives are used and contribute an algorithm to generate spatially targeted gesture primitives. We categorize gesture primitives and show how various categorizations can help in forming a gesture. We contribute a process to analyze the text input, select the relevant gesture primitives based on the analysis of the input, generate the gesture primitives and combine them to form gestures. The gestures are synchronized to speech and the valid combinations of gestures are ranked based on the user's

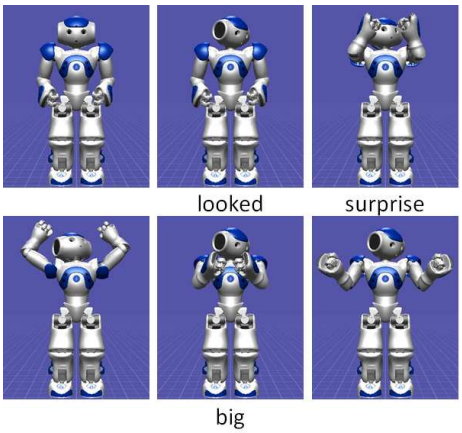


Fig. 7. Snapshots of the NAO executing the highest ranked gesture combination, corresponding to certain words of the output text

requirements. We propose a set of criteria that is weighted to rank the gesture combinations. We show that our goals of automatically generating gestures are accomplished and the validity is demonstrated on a NAO humanoid robot. Our contributions are also applicable to other robots.

Selection of relevant gestures are highly dependent on the accuracy of the text analysis performed, the richness of the gesture primitives database and the associated bags of words. For future work, we will look into using the time available between gesture primitives, instead of adding pauses, so that gesturing with speech is more natural and smooth.

ACKNOWLEDGEMENTS

We thank Somchaya Liemhetcharat for the feedback on the paper. The views and conclusions contained in this document are those of the authors only. Junyun Tay is in the Carnegie Mellon University-Nanyang Technological University Dual PhD Programme and is co-advised by Associate Professor Chen I-Ming from Nanyang Technological University.

REFERENCES

[1] I. R. Nourbakhsh, C. Kunz, and T. Willeke, "The mobot museum robot installations: A five year experiment," in *IEEE Int. Conf. Intelligent Robots and Systems*, 2003, pp. 3636–3641.
[2] D. McNeill, *Hand and Mind: What Gestures Reveal about Thought*. Chicago: University of Chicago Press, 1996.
[3] V. Ng-Thow-Hing, P. Luo, and S. Okita, "Synchronized gesture and speech production for humanoid robots," in *IEEE Int. Conf. Intelligent Robots and Systems*, 2010, pp. 4617–4624.
[4] G. Beattie, *Visible thought: The new psychology of body language*. New York: Routledge, 2004.
[5] M. Haring, N. Bee, and E. Andre, "Creation and evaluation of emotion expression with body movement, sound and eye color for humanoid robots," in *IEEE Int. Symp. RO-MAN*, 2011, pp. 204–209.
[6] A.-A. Samadani, B. DeHart, K. Robinson, D. Kulic, E. Kubica, and R. Gorbet, "A study of human performance in recognizing expressive hand movements," in *IEEE Int. Symp. RO-MAN*, 2011, pp. 93 –100.
[7] J. Cassell, H. H. Vilhjlmsson, and T. W. Bickmore, "Beat: the behavior expression animation toolkit," in *SIGGRAPH*, 2001, pp. 477–486.
[8] D. S. S. Michael L. Walters, Mohammedreza A. Oskoei and K. Dautenhahn, "A long-term human-robot proxemic study," in *IEEE Int. Symp. RO-MAN*, 2011, pp. 137–142.
[9] J. Mumm and B. Mutlu, "Human-robot proxemics: physical and psychological distancing in human-robot interaction," in *Proceedings of the 6th international conference on HRI*, 2011, pp. 331–338.
[10] "Synesketch," http://www.synesketch.krcadinac.com/wiki/index.php.
[11] "Festival," http://www.cstr.ed.ac.uk/projects/festival/.
[12] "Mbrola," http://mambo.ucsc.edu/psl/mbrola/.