

Development of Communication Model for Social Robots based on Mobile Service

Ji-Dong Yim*, Sungkuk Chun[†], Keechul Jung[†], Christopher D. Shaw*

*SIAT, Simon Fraser University, Surrey, Canada

[†]HCI lab. Soongsil University, Seoul, Korea

jdyim@sfu.ca, k612051@ssu.ac.kr, kcjung@ssu.ac.kr, shaw@sfu.ca

Abstract—This paper describes an interactive social agent platform which examines anthropomorphic robot features in the mobile phone usage context. Our system is smart phone based robot agent that operates on a mobile network and co-located ad-hoc networks. It helps remote users communicate interactively with each other through the robotic interface which utilizes facial expressions and body gestures. In this paper, we introduce the mobile environment as a service platform for social robots, and discuss design considerations for such a communication system. Then we illustrate the development of our system followed by its network protocols built on existing mobile services such as Telephony and Short Messaging Service (SMS). Usage scenarios and working prototypes of the implemented system are also presented. We are hopeful that our research will open a new discussion on socially interactive robot platforms, and thus, that such efforts will enrich the telecommunication and personal robot services in the near future.

Keywords—Mobile phone, network packet design, social robot, telecommunication, anthropomorphic expressions

I. INTRODUCTION

People use many kinds of interaction media such as tone of voice, facial expressions and gestures, whereas computer systems are limited to visual display and synthetic audio when they communicate to human users. Nevertheless, a mobile phone is an interesting platform for researchers and designers as it has a variety of features that are valuable in developing and evaluating new HCI technologies. To cite a parallel example, when a new speech recognition engine is enabled on a phone, one can test it in real settings and collect a lot of data, because people may use it very often, anywhere, any time. One would expect a long-term experiment to be easily available with a cell phone, because it is usually owned for time ranging from months to years. Personalization or customization issues can also be investigated, since users add accessories and use different ring-tones for each contact group. It might be a good choice even for evaluating aesthetics, since people carry the phone in a pocket or in a purse - more of a fashion item than a laptop in the backpack.

In order to explore new expressive modalities for handheld computers, we investigate Robotic User Interface (RUI) as an alternative interaction medium. Our prototypes, CALLY and

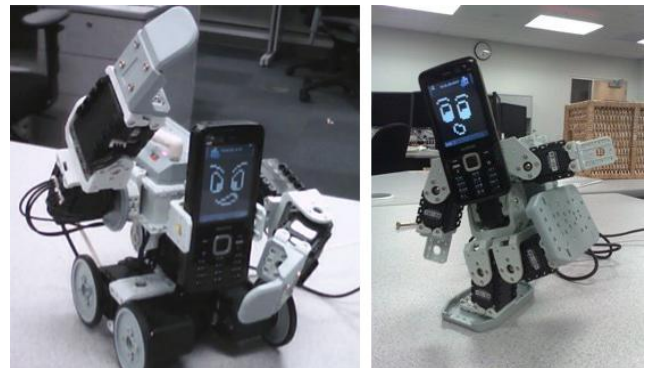


Figure 1. Developed prototypes; CALLY (left); and CALLO (right).

CALLO, are *functionally designed* robots [19] that are built on a mobile phone technology (Figure 1). The robotic interface is physically combined with the phone device, and controlled by the phone applications. The robot's anthropomorphic features, thus, add more means of social abilities to the phone device in conjunction with mobile network services. In this paper, we first look into current mobile phone use paradigms to explore HCI issues for our robotic social media system, and present how those concerns are considered during implementation. A full cycle of gesture-based communication model is illustrated with the software system architecture, applications and detailed user interface issue.

II. RELATED WORK

Adding to traditional use of telephones, recent mobile phones are enhanced with a variety of new communication services, such as SMS, email, IM (instant messaging), blogs, video call and social networking applications [1][2][3]. HCI researchers and designers have explored other expressive and more tangible means of interaction including phonic signals [4], tactile vibration [5][6] and force feedback [7][8].

A few systems have used actuators or life-like robot expressions in mobile phone use contexts. The Apple iPhone is able to be a remote control or teleoperation console for a navigation robot [9]. Some prototype systems, for example, Ambient Life [10], proposed a mobile phone that displays

device status in life-like signals, like breath and pulse. Life-like appearance and movements have been long discussed in the fields of computer agent and Human-Robot Interaction (HRI) [11], in terms of social mediators having virtual presence and non-verbal conversational cues [12][13][14].

Once a mobile phone is equipped with a robotic body, the system should provide an easy interface for robot animation, usable by experts and non-experts alike. However, standard GUI techniques for computer animation do not port well to handheld displays, and motion tracking equipment similar to that used in the video game industry is clearly overkill. Thus, the robot cell phone needs a new input system that is also tangibly interactive and lightly equipped, such as direct manipulation with/without kinetic memory [15][16][17], audio-driven [12], or vision-based [18] methods. Ogawa et al. [12] and Li et al. [18] pointed out an interesting and valuable aspect of such tracking systems; for avatar-based communication, quick response and adequate accuracy to the user’s gesture are more important than precise estimation.

Yet, as far as we know, no social robot platform has detailed a full interaction cycle of expressive gesture RUIs that augment anthropomorphism with existing mobile phone native networks such as Telephony, SMS, and video telephony services.

III. SOCIAL AGENT PLATFORM ON MOBILE SERVICE

Our social robot system is “functionally designed” on the basis of mobile phone use context rather than “biologically inspired” [19]. Thus, in order to find design considerations for our bi-directional communication model, the current mobile phone environment must be examined. In this section, we describe our framework for combining a mobile device and its related services with a mobile robot to develop a physically interactive social agent platform.

A. Interactions in Mobile Service

There are many types of interactions and social entities involved in using mobile telecommunications service. As our research team seeks design considerations for developing social intermediates in the mobile phone use context, first we classify characteristics of mobile phones based on how a cell phone works with a user in the existing mobile services. Inspired by Breazeal’s human-robot interaction paradigms [20], we suggest three categories; *tools*, *avatars*, and *smart agents*.

The simplest relationship between a cell phone and a user is found when we see a phone as a static *tool* that is not connected to a network. A mobile device in this case is in charge of simple tasks, such as managing a phone book or playing multimedia files in local storage. Functioning well without disturbing the user are the first requirements of such a product. A very basic example of this case is of a phone as an alarm clock.

When a user is connected to a remote party, a cell phone becomes an *avatar* (Figure 2, middle). A phone device is possessed by the owner, although it *represents* the counter party. Other one-on-one telecommunication services, such as paging, messaging and video call, can be placed in this

category. In fact, via a video call, we can see more clearly that a mobile phone turns into an avatar that shows a live portrait of a remote person. Interestingly, from a user’s view, it seems as if only three important entities are involved in this interaction – the user, user’s phone and the counter person – while there are actually four including the other’s phone device. That allows us first to consider the issues of co-located peer-to-peer interaction between the user and his/her phone.

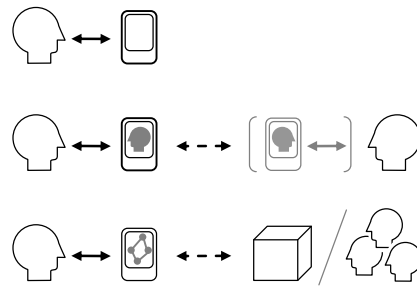


Figure 2. Three types of interaction with a cell phone; one-on-one relationship between a user and a device (top); one-on-one interaction between users in a traditional mobile phone network (middle); interactions between a user and a service in a multi-user networking environment (bottom).

In a networked environment, at least in the near future, a phone or a mobile application becomes an intelligent *agent* that handles back-end data to bring selective information to the human user. Car navigation systems use GPS to help way finding. Location-based services draw geographically useful information on a map. An instant messenger links to group networks in order to help users stay connected on-line. Social networking services broadcast live news to people around the world.

B. Mobile Device as a Robot Platform

A mobile phone provides robot researchers with a variety of features for developing and evaluating new robot capabilities. Besides a wide range of characteristics of its usage context, the technological capabilities of mobile devices are becoming more feasible for robot research. Many phone devices have a camera, a far distance microphone, a loud speaker, and a touch screen display, which can allow a robot to communicate with a human user. Embedded sensors such as the accelerometer, the Infra Red light detector, and the GPS module enable a robot to perceive its environment. Tele-operation comes available by using wireless networks like Bluetooth and Wi-Fi. On the top of these technical components, a smart phone device has computing power and open-development environments on which researchers can develop robot intelligence. A robot application can control parameters of test attributes by using mobile phone’s Application Programming Interface (API) and can access device resources such as hardware features, file system and user data (e.g. contact list). Many cell phone manufacturers provide developers with operating systems and APIs, but openness and use policy vary.

C. Communication Loops of Mobile Phone Robot

We define a mobile phone robot as a system that integrates three communication loops as shown in [Figure 3]. First, we see it as a technological system that deals with the interface between a phone and a motor system. Considering the technologies currently available in the market and the fact that we are aiming to illustrate the future applications and interaction styles, we developed our system by combining existing phones and robot kits rather than building a new robot mechanism from scratch. This bottom level interface is accomplished by realizing communication protocols between two co-located devices.

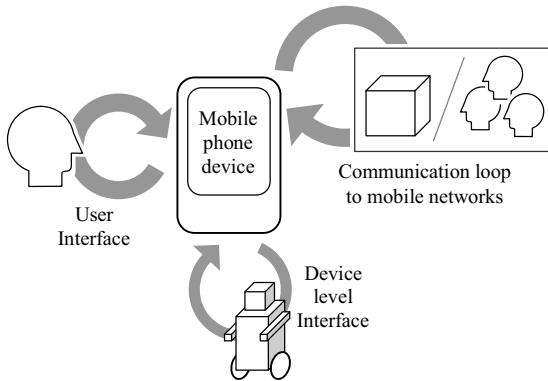


Figure 3. Three communication loops in our mobile robot system

The second communication loop is the user interface of the system. We are interested in how easily a computing machine learns gestures and talks to a human user by using its physical attributes. We think of our robot system as an animated creature that partly inherits human shapes and gestures. To explore the tradeoffs of the robotic user interface, we implemented a full cycle of a teach-and-learn task where a user creates gestural expressions by directly manipulating the postures of the robot.

Third, the system communicates to other systems over mobile phone networks. It is a mobile phone in a symbolic and anthropomorphic shape which surrogates a remote party, no matter if the commander is a human user, another device or an autonomous service. To illustrate the last communication loop, we considered existing mobile phone networks and services such as Telephony, SMS, Wi-Fi Instant Messaging, or GPS.

IV. OVERVIEW OF SYSTEM DEVELOPMENT

We have developed a mobile phone based social robot platform which consists of two main parts; a cell-phone head and a robot body. First, a cell-phone device in our system shapes the robot's head and acts as robot's brain as well. Mounted on the robot body, it displays symbolic facial expressions, actuates the robot's motor system, reads the environment by utilizing sensors, accesses user's personal data in the phone file system, and communicates to other devices or services. Second, the robot part deals with control commands from/to the phone to give the system physical abilities such as spatial mobility and/or body gestures.

A. Head – the Robot Brain

The main software is built on a Nokia N82 phone. N82 is a 2G and 3G compatible smart phone which runs on the Symbian S60 3rd Edition Operating System. We display robot's facial expressions on its 2.4 inches TFT color LCD in 240×320 pixels. The front camera is used for face detection at 30fps in 352×288 pixels. Mobile phone native SMS and wireless LAN 802.11 b/g help the robot receive commands from a remote operator. In order to control the robot body from the phone device, a serial port over Bluetooth v2.0 is used. A Dual ARM 11 332 MHz processor runs the OS with 128 MB RAM. Several built-in features such as a GPS module, a microphone, an IR transceiver and an accelerometer are not used under the current robot system but may be considered for future use.

The latest versions of phone applications, which are the robot AI in other words, were built on the Symbian C++ environments. We utilized various kinds of Symbian C++ APIs from the S60 platform SDKs that handles Telephony, Messaging (e.g. SMS and IM), Networking (e.g. Bluetooth and Wi-Fi), File and User Data, Graphics and Multimedia, Symbian native User Interface, and so forth. An open source program titled FreeCaller [21] was also used for a part of our project to prototype Telephony functionalities (Figure 4). The selection of the device and the software platform was found to be suitable for our social robot project, because it enabled us to integrate the full capabilities of the phone device and native mobile phone services into our system. Many of other mobile phone platforms, as far as we knew at the time, do not provide developers with easy access to paid services such as incoming /outgoing calls or SMS, due to potential abuse cases. Symbian, of course, requires a very strict process for running those high-risk applications; each compilation has to be registered on-line before it is installed onto a single phone device.



Figure 4. Examples of CALLO's call indicator actions; "lover's dance" (left), "happy friends" (center), and "feeling lazy when called from work" (right). The phone application handles telephony routines and controls the robot body.

Early prototypes on CALLY were quickly developed on the Java™ Mobile Edition 2 (J2ME) to show the robot's look-and-feel and general ideas of possible future applications. Since phone native services are not allowed with J2ME, the main software was developed on a PC that communicated with the phone via TCP/IP. Our current applications in CALLO, built on Symbian C++, now talk directly to the Symbian OS and access to mobile phone services.

B. Robot Body

The robot body is implemented using the Bioloid robotic kit [22]. The kit consists of an ATmega128 microcontroller board, multiple servo motor modules, sensor modules and various

types of joint assemblies. The microcontroller is programmable to contain a series of preset gestures and simple triggering logic. The program includes customized sets of the robot components and is built to transmit primitive motor values from the robot to our phone application, and vice versa. The bidirectional data communication between the robot and the phone is realized by replacing the built-in Zigbee module with an extra Bluetooth embedded module in the controller board. Once the wireless module is mounted and powered on the board, it links to the designated RF host in the phone and virtually creates a serial port over Bluetooth v2.0.

V. OVERVIEW OF SOFTWARE STRUCTURE

The communication model of our mobile phone based robot system consists of three levels of software as shown in [Figure 5]. The bottom level of the structure supports hardware interface built on device specific functions such as motor system commands, serial port protocol, Bluetooth RF driver, and timer control for data transfer routines. This base level structure is described in the first subsection. On top of the device level, the middle level software manages the data structure of robot motions and the routines for both recording customized animations and playback. The user interface modules are also at this level as they are a middleware that helps data sharing among different communication routines at each level; at the lower level between a phone and a robot, at the middle level between a user and a phone, and at the higher level between a phone and a mobile network service. The following subsections describe those middle level software structures. The highest level structure deals with messaging protocols that enable our system to support various types of social robot services using the mobile network. In the last subsection, we briefly introduce how the messaging model can be integrated into current mobile phone native services and other wireless networks that are available with recent smart phones.

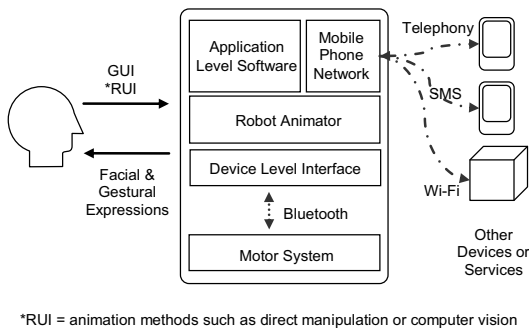


Figure 5. Software Structure of our mobile phone robot system

A. Device Level Interface

Our system was developed to integrate and to run on two different hardware platforms; a robot and a cell phone. The bottom level structure of the software, thus, manages the interface between the two devices so that higher level software on the phone can easily drive the motor system of the robot.

To allow the communication, we first modified the robot control board by adding an ACODE-300 Bluetooth embedded chip which is configured in single-link server mode (or slave mode) [23]. It runs a wait routine until another active client (or a master client) requests a connection. The server module hides itself from arbitrary clients, so that a Bluetooth compatible cell phone device can request a connection only when it knows the address and the password key of the server.

Once the Bluetooth connection is established, it works as like a RS-232C serial port. The serial communication protocol is strongly dependent on robot controllers, so a kind of device driver should be developed for each controller board. For example, the data transfer speed in our system is set 57,600bps because the Bioloid robot kit allows no other options. There are a couple of other restrictions the robot kit has, such as; a single command packet must be two bytes; and the input buffer only allows up to two command packets to be received at a time. So the current robot driver in our system is forced to pack each command and parameters in a 16bits instead of using a text based protocol. Also, to avoid the second limitation, the data communication is managed in a timely manner with contingent control.

The main routine on the microcontroller of the robot is rather simple. It interprets commands from the phone to set up a mode or to control motors, and sends responds or motor readings to the phone.

B. Data Structure for Robot Animation

One of the main structures in the middle level software is the animation management module for robot gestures. The animation module consists of four components containing information on a motor, a robot pose, a robot motion, and a list of animations. They are hierarchically abstract to robot motions. A Motor component, representing a moving part of a robot, basically has three numeric members mapped to the motor index, the motor angle and the speed. More members can be included to handle optional attributes of a motor such as acceleration, the range of movement, temperature, torque, or so forth.

A Pose object consists of multiple Motor components and has peripheral attributes in order to show a posture of a robot. The peripheral members include the index of the pose in a motion and the delay time that determines how long the pose stays still.

A Motion is a series of multiple poses that construct a complete cycle of a moving gesture of a robot. Each Motion has a repeat count, an index, and the next motion's index, so that a combination of motions generates a more complicated animation.

The top level component, called the Animation module, is a collection of robot motions. It has 23 default animation sets and enables the system to manage motion playback and recording.

C. User Interface for Customizing Robot Animations

Our system integrates robotic movements with a mobile phone in order to provide an intuitive way that helps a user interact with other users and services. Gesture playback is one

way that transfers phone information to a user. The other half is gesture animation recording. However, most phone devices have only limited screen area to support an easy GUI for gesture recording. Our system employs a Grab-and-Move style interface, so that a user can directly manipulate each part of the robot and generate his/her own robot gestures. When the system runs in the recording mode, it continuously reads motor status to construct a robot pose at the moment while a user moves a robot's limbs. The posture data can be recorded either continuously in every 50 milliseconds (say, *continuous* mode) or at a certain time point that a user selects (*discrete* mode). The Animation module then collects the postures to build a robot's gesture animation. The selection between the two recording modes, *continuous* and *discrete*, is made depending on the application context. For example, discrete motion data would be preferable to be sent via SMS, whereas continuous one would be better for local applications (e.g. customizing incoming call gestures) or with a faster wireless messaging service (e.g. using an instant messaging typed communication). Those recording modes are described more in the next section with application examples.

D. Gesture Messaging Protocol

Motion data is encoded into a text string to be sent to other devices. We standardized a gesture messaging format so that it fits well with existing text based communication services such as SMS or Instant Messaging (IM). This enables a developer to build a new system independent of hardware configuration. If one needs to implement a PC application or to use a new robot system that communicates with existing robot platforms, for example, only one or two lower level data structures need to be written.

A gesture message consists of a header and a body as shown in [Table 1]. The very first marker of the header is the Start of Gesture Message indicator for which we arbitrarily use “##”. It is followed by a text emoticon with a 2-byte checksum that determines a facial expression to be displayed. The header length and the protocol version come next at one byte each. The next four bytes are reserved to link multiple messages as an animation data may consist of one or more motions. The last two-bytes state the number of motors of the robot system and the number of poses included in the motion. The message body, which is a Motion data object, consists of multiple Poses. A Pose, again, is a series of motor information plus the period of time the pose stays. Some exceptional formats such as emoticon-only messages are allowed for ease of use. For example, a text message with one of the default emoticons triggers the corresponding gesture animation with a facial expression. The preset of the expressions includes common emoticons such as “:D”, “=P”, “:\$”, and so forth.

VI. APPLICATIONS AND MESSAGING INTERFACE

The phone software takes care of robot motions and messaging protocols that enable our system to support various types of social robot services using the mobile network. As a simple example, CALLO, our second prototype, can intercept incoming calls or recognize emoticons from a SMS message, then activates a robot animation with a corresponding facial expression displayed. As another example, motion data is

TABLE I. MESSAGING PROTOCOL FOR EXPRESSIVE ROBOT PLATFORMS OVER MOBILE PHONE SERVICE

Name	Structure and Description		
	Start of message	“##”	
	Emoticon **	e.g. “:~:”, “=P” 2-4 bytes	
	Checksum	Numeric 2 bytes	
Header	Header length		
	Protocol version		
	Number of motions		
	Index of current motion	Numeric	
	Index of next motion	1 byte each	
	Number of motors		
	Number of poses		
	Time span		
	Reserved	Numeric	
Pose #1	Motor #1	Moving speed	1 byte each
		Goal position	
Body	Motor #2		Same as
	...		Motor #1
	Motor #N		
	Pose #2		Same as
	...		Pose #1
	Pose #N		

** A text message including only an emoticon can also trigger a pre-defined robot animation with a facial expression.

encoded into a text string to be sent to other devices, so that a user can share his/her customized robot gestures with other users over SMS or Wi-Fi. However, it is a very painful process for a user to create a gesture message using a conventional SMS editor or IM dialog. In the following sections, two robot animation methods of CALLO are presented with difference application scenarios.

A. Sending Robot Gestures over SMS

Our robot animation protocol, which is basically a serialized collection of numbers, is transferred to control a remote system over standard short text messaging service.

The first and the simplest SMS application we developed with CALLO was the one that responds to a text message containing a known emoticon. For example, a text message “going out with Chris tonight? :O” comes with a “surprise” expression, whereas a message “=D coming here with Chris tonight?” shows big smiley face and gesture. CALLO currently has 23 expressions pre-defined, each consists of a simple text emoticon, a face image, and a moving robot gesture.

A more sophisticated gesture messaging example between two or more CALLO devices would involve recording tasks, as some people prefer to use customized animations in PC-based instant messages. A messaging application in our system helps a user generate a text message that is interpretable to a facial expression and a gesture animation, so that the user can easily

send it using the standard SMS user interface of the phone. [Figure 6 (top)] shows an example of this type of text message. We used “##” as a delimiter to locate the header of the message.

A standard short text message has a length limit either of 160 7-bits or 70 2-bytes characters which only allows 8 poses to be included in a message considering that the motor system of CALLO has 6 degrees of freedom. The gesture messaging application allows the *discrete* animation format to be used over SMS. As shown in [Figure 6 (bottom)], a user can create an animation sequence in a small chunk of text string by following procedures; 1) to set an emoticon, 2) to shape a posture by moving robot pieces, 3) to record current pose, 4) to add more pose by repeating two previous tasks, 5) to edit the generated text, and 6) to send the text message.

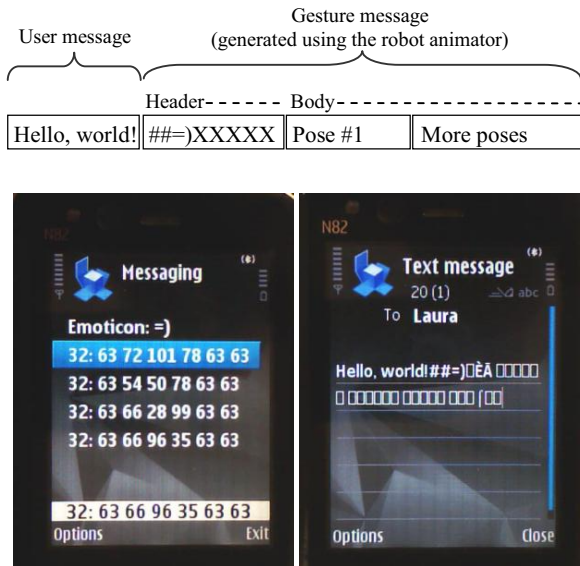


Figure 6. An example of gesture message; an ‘X’ indicates a 2-bytes Unicode letter as our system uses 140bytes SMS standard (top); a screen capture of the animation editor where each line represents a pose (left); and the text message generated from the gesture data. The letters are usually not readable. The first characters of the message, “Hello, world!”, was added using the phone’s native text messaging application (right).

B. Synchronizing Gestures in IM

The *continuous* motion data format is preferable in terms of user interface when the message length is not an issue, for examples, in an instant messaging, in a multi-user chat, or in an application that stores the data in a local memory (e.g. creating a gesture as an incoming call indicator). The animation data format is similar to the discrete one’s except the message header does not specify the length of gesture data (Figure 7). CALLO’s instant messaging application demonstrates the use of continuous data format and the easiness of its user interface. Here is how a user manipulates his/her robot to synchronize the movement to the counterparty’s; 1) to request a permission to control the remote device and to set each robot in recording and playback mode if the request is guaranteed, 2) to send a message header with an emoticon, 3) to move his/her robot, and 4) to send emoticons to change facial expressions.

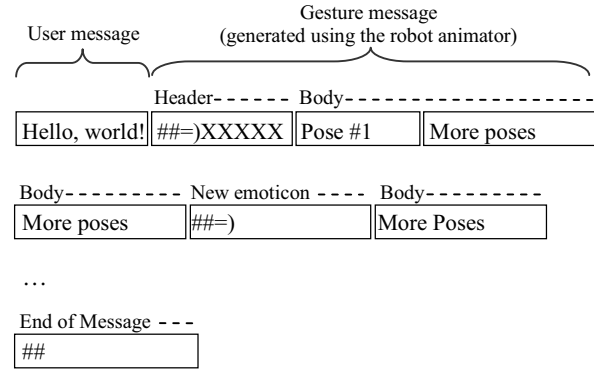


Figure 7. Creating a gesture message in the continuous format for Instant Messaging applications

C. User’s Feedback for Direct Manipulation Interface

We conducted a pilot user study to find potential benefits and limitations of the developed message interface of CALLO. We were also interested in how exactly people can generate robot animations as they want. Six participants in age from 23 to 31 were recruited for the test. The subject group consisted of 3 male and 3 female graduate students who had no experience in robot study or in playing with robot applications.

The study consisted of three stages and a questionnaire session. First, in order to decide the maximum speed of motor manipulation, participants were asked to move a robot arm as fast as they can, from the bottom to top (180 degrees up), and in opposite direction (180 degrees down), five times each. In the second phase, they were asked to create four robot gestures expressing four different emotional states namely; happy, sad, surprised, and angry. Two-arm movements, one degree of freedom each, were recorded for each expression. Then the participants were asked to re-produce each gesture in five times.

Participants moved a motor at speed of 144.57 degrees per second in average. Minimum and maximum were 92.40 degs/s and 212.52 degs/s respectively. There were no difference found for each user, but the speeds of moving up and down were significantly different ($M=137.65$, $SD=25.00$ when moving up, $M=151.48$, $SD=32.16$ for moving down, $t(29)=3.16$, $p=.004$). In the questionnaire, the participant reported that they found a better grip when they moved the robot arm to down direction. We collected a series of {time, motor position} datasets from each task of the second and the third stages. The data sets from the third tasks were then compared to the original records from the second stage. Fréchet distance [24] and user ratings were used as a test metric, yet we didn’t find any statistically significant result due to the small number of sample size.

In the questionnaire, participants reported that the robot interface provided an exciting new experience and that the animation method was intuitive. Motor sound was revealed another good feedback for expressions. Most limitations were found from the motor system. The servo motor we used has a relatively heavy resistant torque loaded even when no signal is on, so participants felt afraid to break the robot. Some subjects could not find a good grip during earlier trials and sometimes

had their thumbs caught in the robot parts. As the degrees of freedom of CALLO's motor system were felt limited to fully describe emotions, participants desired facial expressions too (we didn't provide the robot's facial expressions during the test). The pilot study result suggested improvements for the robot's gesture animation interface that may be accomplished by either better, at least more easily movable, motors or other manipulation methods, for example, a vision-based method in the following subsection.

D. Vision Based Gesture Animation

Our recent robot animation module utilizes a fast computer vision engine that extracts user's hands and face positions. When a streaming video is captured from the front camera of the phone, the animation module first runs a Gaussian mixture of skin color to distinguish hands and face regions from background. We used two classification models, for skin and non-skin colors, which are composed of 16 Gaussian kernels (N=16) respectively based on Bayesian rule as seen in the following formula. Parameters of mean, covariance matrix, and weight values are additionally used as suggested in [25] to improve the detection rate.

$$f(x) = \begin{cases} \text{skin}, & \text{if } \frac{P(x|\text{skin})}{P(x|\text{non skin})} > T, T = \frac{c_n P(\text{non skin})}{c_s P(\text{skin})}, \\ \text{non skin}, & \text{otherwise} \end{cases}$$

where, x = RGB color vector of an input pixel, T = threshold from prior probabilities, $P(\text{skin})$, $P(\text{non skin})$, and the costs of false positive and false negative C_n and C_s .

The strictness level of skin color classifier is dependent on the costs, so if C_n increases and C_s decreases, skin classifier determines whether an input pixel is skin color or not more critically. We reduced the computational time by transforming the input images into 40×30 resolutions, since we aim for a quick and adequate detection result, not a precise estimation, as pointed in [12][18].

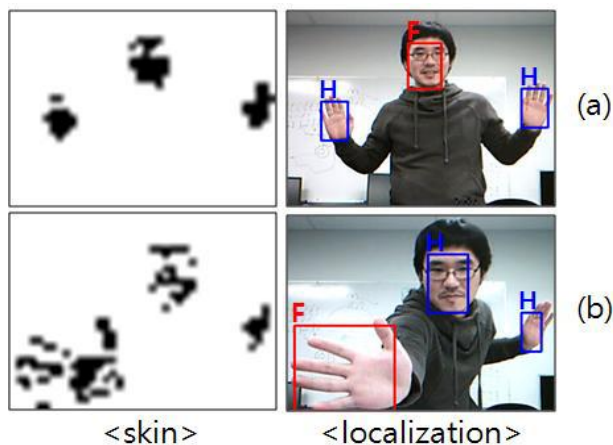


Figure 8. Hands and face regions before face detection is applied; correct result (top); false result (bottom).

Then the system runs two routines, K-means clustering and face-detection, to locate face and hand regions from the extracted skin color data. Once three groups of skin areas are classified through K-means algorithm (Figure 8), the face detection routine [26] determines the face region and two others as hands.

After the localization process, the gesture module shapes a robot posture according to the coordinates of the user's face and hands. The computer vision based animation method is fast and easy, yet not applicable for SMS or other applications that use the discrete gesture messaging format.

VII. CONCLUSION AND FUTURE WORK

We presented an interactive social robot platform that could help users communicate with each other or to autonomous services via mobile phone networks. The developed system, CALLY and CALLO, successfully integrates anthropomorphic facial and body expressions to mobile phones. As an example, the gesture messaging application is implemented based on the paradigm of *cell phone as a social avatar*. The prototype shows how our system fits into current telecommunication services such as Telephony, SMS and instant messaging. From the gesture messaging system, it is suggested that a tangibly expressive avatar system also requires new robot animation methods to overcome the lack of the input modalities of small devices. Two gesture customization methods were thus realized by adopting direct manipulation and computer vision based techniques. The direct manipulation method was applicable for both *continuous* and *discrete* messaging formats. The vision-based method provided a quick and easy interface but was limited to *continuous* gesture messaging protocols.

A future work will include an improved messaging protocol. Data compression allows a longer and smoother robot animation within the length limit of a SMS message. Data conversion between the continuous and the discrete messaging formats is another room for improvement in terms of user interface, as it would enable a user to record a complicated gesture without selecting key-frames, then to send the animation data through a short message.

The vision based robot animation module we introduced in this paper shows a fast and robust performance in the PC environment. Currently it receives real time streaming images from a mobile phone camera via Bluetooth and extracts hand positions. We are adopting the implementation to a Symbian device and hope to include the result in the conference.

Future applications of CALLY and CALLO will examine personalization issues of socially interactive robotic products and other technologies such as GPS, facial animation, voice recognition and voice synthesis. Updates will be posted at our project blog, <http://cally.iat.sfu.ca/>.

ACKNOWLEDGMENT

This research was supported in part by Nokia under a Nokia University Program, by KIDP (the Korea Institute of Design Promotion) under a Government Grant for Future Designers, and by NSERC Discovery Grant. The authors marked with † were supported by the MKE (The Ministry of Knowledge

Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency #NIPA-2009-(C1090-0902-0007)).

REFERENCES

- [1] S. King, and J. Forlizzi, "Slow messaging: intimate communication for couples living at a distance," Proc. of the 2007 Conference on Designing Pleasurable Products and interfaces (Helsinki, Finland, August 22 - 25, 2007). DPPI '07. ACM, New York, NY, pp. 451-454. 2007.
- [2] A. Ankolekar, G. Szabo, Y. Luon, B. A. Huberman, D. Wilkinson, and F. Wu, "Friendlee: a mobile application for your social life," Proc. of the 11th international Conference on Human-Computer interaction with Mobile Devices and Services (Bonn, Germany, September 15 - 18, 2009). MobileHCI '09. ACM, New York, NY, 1-4. 2009.
- [3] C. Licoppe, and J. Morel, "The collaborative work of producing meaningful shots in mobile video telephony," Proc. of the 11th international Conference on Human-Computer interaction with Mobile Devices and Services (Bonn, Germany, September 15 - 18, 2009). MobileHCI '09. ACM, New York, NY, 1-10. 2009.
- [4] A. S. Shirazi, F. Alt, A. Schmidt, A. Sarjanoja, L. Hynninen, J. Häkkinilä, and P. Holleis, "Emotion sharing via self-composed melodies on mobile phones," Proc. of the 11th international Conference on Human-Computer interaction with Mobile Devices and Services (Bonn, Germany, September 15 - 18, 2009). MobileHCI '09. ACM, New York, NY, pp. 1-4. 2009.
- [5] G. Lotan, and C. Croft, "imPulse," CHI '07 Extended Abstracts on Human Factors in Computing Systems (San Jose, CA, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, pp. 1983-1988. 2007.
- [6] J. Werner, R. Wettach, and E. Hornecker, "United-pulse: feeling your partner's pulse," Proc. of the 10th international Conference on Human Computer interaction with Mobile Devices and Services (Amsterdam, The Netherlands, September 02 - 05, 2008). MobileHCI '08. ACM, New York, NY, pp. 535-538. 2008.
- [7] F. F. Mueller, F. Vetere, M. R. Gibbs, J. Kjeldskov, S. Pedell, and S. Howard, "Hug over a distance," In CHI '05 Extended Abstracts on Human Factors in Computing Systems (Portland, OR, USA, April 02 - 07, 2005). CHI '05. ACM, New York, NY, pp. 1673-1676. 2005.
- [8] S. Brave, and A. Dahley, "inTouch: a medium for haptic interpersonal communication," CHI '97 Extended Abstracts on Human Factors in Computing Systems: Looking To the Future (Atlanta, Georgia, March 22 - 27, 1997). CHI '97. ACM, New York, NY, pp. 363-364, 1997.
- [9] R. Gutierrez, and J. Craighead, "A native iPhone packbot OCU," Proc. of the 4th ACM/IEEE international Conference on Human Robot interaction (La Jolla, California, USA, March 09 - 13, 2009). HRI '09. ACM, New York, NY, pp. 193-194. 2009.
- [10] F. Hemmert, "Ambient Life: Permanent Tactile Life-like Actuation as a Status Display in Mobile Phones," Adjunct Proc. of the 21st annual ACM symposium on User Interface Software and Technology (UIST) (Monterey, California, USA, October 20 - 22, 2008).
- [11] M. Mori, "The Uncanny Valley. Energy," 7(4), pp. 33-35, 1970.
- [12] H. Ogawa and T. Watanabe, "InterRobot: a speech-driven embodied interaction robot," Advanced Robotics, 15, pp. 371-377, 2001.
- [13] M. P. Michalowski, S. Sabanovic, and H. Kozima, "A dancing robot for rhythmic social interaction," Proc. of the ACM/IEEE international Conference on Human-Robot interaction (Arlington, Virginia, USA, March 10 - 12, 2007). HRI '07. ACM, New York, NY, pp. 89-96, 2007.
- [14] D. Sakamoto, T. Kanda, T. Ono, H. Ishiguro, and N. Hagita, "Android as a telecommunication medium with a human-like presence," Proc. of the ACM/IEEE international Conference on Human-Robot interaction (Arlington, Virginia, USA, March 10 - 12, 2007). HRI '07. ACM, New York, NY, pp. 193-200, 2007.
- [15] D. Sekiguchi, M. Inami, and S. Tachi, "RobotPHONE: RUI for interpersonal communication," CHI '01 Extended Abstracts on Human Factors in Computing Systems (Seattle, Washington, March 31 - April 05, 2001). CHI '01. ACM, New York, NY, pp. 277-278, 2001.
- [16] P. Frei, V. Su, B. Mikhak, and H. Ishii, "Curlybot: designing a new class of computational toys," Proc. of the SIGCHI Conference on Human Factors in Computing Systems (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM, New York, NY, pp. 129-136, 2000.
- [17] H. S. Raffle, A. J. Parkes, and H. Ishii, "Topobo: a constructive assembly system with kinetic memory," Proc. of the SIGCHI Conference on Human Factors in Computing Systems (Vienna, Austria, April 24 - 29, 2004). CHI '04. ACM, New York, NY, pp. 647-654, 2004.
- [18] R. Li, C. Taskiran, and M. Danielsen, "Head pose tracking and gesture detection using block motion vectors on mobile devices," Proc. of the 4th international Conference on Mobile Technology, Applications, and Systems and the 1st international Symposium on Computer Human interaction in Mobile Technology (Singapore, September 10 - 12, 2007). Mobility '07. ACM, New York, NY, pp. 572-575. 2007.
- [19] T. Fong, "A survey of socially interactive robots," Robotics and Autonomous Systems, vol. 42, 2003, pp. 143-166.
- [20] C. Breazeal, "Social Interactions in HRI: The Robot View," IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), vol. 34, 2004, pp. 181-186.
- [21] Y. Perek, FreeCaller, <http://sourceforge.net/projects/freecaller/>
- [22] Robotis Inc., Bioloid. <http://www.robotis.com/>
- [23] Firmtech co. Ltd., ACODE-300, <http://www.firmtech.co.kr/>
- [24] H. Alt and M. Godau, "Computing the Fréchet distance between two polygonal curves," International Journal of Computational Geometry and Applications (IJCGA), 5: pp. 75-91, 1995.
- [25] M. J. Jones, and J. M. Rehg, "Statistical Color Models with Application to Skin Detection," International Journal of Computer Vision, 46, Springer, pp. 81-96, 2002
- [26] P. Viola, and M. Jones, "Robust real-time face detection," International Journal of Computer Vision, 57, Springer, pp. 137-154, 2004.