Intent Capturing through Multimodal Inputs

Weimin Guo, Cheng Cheng, Mingkai Cheng, Yonghan Jiang, and Honglin Tang

School of Computer Science, Beijing Institute of Technology, Beijing 100081 PRC,
Beijing Laboratory of Intelligent Information Technology, Beijing Institute of Technology, Beijing 100081 PRC
cc@bit.edu.cn
http://isc.cs.bit.edu.cn/faculties/chengcheng/index.html

Abstract. Virtual manufacturing environments need complex and accurate 3D human-computer interaction. One main problem of current virtual environments (*VE*s) is the heavy overloads of the users on both cognitive and motor operational aspects. This paper investigated multimodal intent delivery and intent inferring in virtual environments. Eye gazing modality is added into virtual assembly system. Typical intents expressed by dual hands and eye gazing modalities are designed. The reliability and accuracy of eye gazing modality is examined through experiments. The experiments showed that eye gazing and hand multimodal cooperation has a great potential to enhance the naturalness and efficiency of human-computer interaction (*HCI*).

Keywords: Eye tracking, multimodal input, virtual environment, human-computer interaction, virtual assembly, intent.

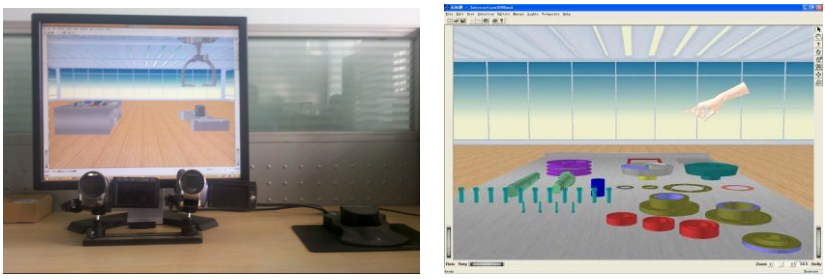
1 Introduction

Multimodal interaction (*MMI*) is an emerging *HCI* research area which has been developping rapidly in recent years. It is well accommadate the idea of Human-centered design [1]. *MMI* refers to a human-computer interaction paradiam that users utilize a variety of modalities to communicate with computer. Although some natural modalities such as visual input, natural language and gesture input, have their inherent inaccuracy, and is difficult to always meet 100% success rate of recognition, it is sure that they can make the cognitive load of human beings be reduced greatly during interactive process [8]. Employment of visual input modality in a critical *VE* system, e.g. the virtual assembly system, is a first time try to use it to eliminate the ambiguity of user's intents in single-channel input scenarioes.

As a natural modality, eye gaze tracking has been developed for human-computer interaction. A number of researchers have been working in this area [7,8,9], and one typical researches is the *Intelligent Gaze-Added Operating System (IGO)* developed by Salvucci and Anderson [10,11]. They compared the performances of the mouse and visual method to show that the visual method has higher error rate than mouse, since the sight often jumps; but the visual input is easier to learn and use, so it is more attractive for users.

M. Kurosu (Ed.): Human-Computer Interaction, Part IV, HCII 2013, LNCS 8007, pp. 243–251, 2013.

Virtual environments are a type of systems established on the base of multimodal interaction. *VE* aim at providing user with an immersive and natural interactive effect, which can hardly be achieved by traditional *WIMP* (Window, Icon,Menu and Pointing) based systems. But the *VE* ability of handling with mixed and fuzzy input data from multiple modalities is still a big problem. Despite researches and developments on hardware have already made a considerable progress and there have emerged a variety of interactive devices, direct manipulation in *VE* today is still a serious technical bottle-neck [4,5]. This point is particularly evident in critical manufacturing environments such as virtual assembly. Building a virtual assembly system with eye tracking modality has an especial merit [6], because eye interaction can free hands from liberous operations.



(a) (b)

Fig. 1. Multimodal virtual assembly system Interaction3D. (a) System paradiagm, (b) Virtual assembly scene.

For this reason we build an eye tracking augmented virtual assembly system *Interaction3D*, as figure 1 shows, where virtual parts are composed with geometric features. One 3D space mouse and a 2D mouse are two modalities to manipulate virtual parts. Two cameras based eye tracking subsystem is another modality to capture user's current focus into the virtual scene. Those input devices work together to constitute multiple inputs. The paper firstly describes eye gaze modality which use *PCCR* technology to estimate user's gaze point on screen [3]. Then introduces how to analyze the gaze point to promote the utility of eye gaze modality in virtual environments.

2 Multimodal Intent Understanding

Gaze tracking provides a potential to allow users naturally express their intents in virtual assembly system. Several main intents are defined in advance in system *Interaction3D*, then multimodal inputs are analyzed to infer users' intents. One of the challenges in gaze tracking is midas touch problem, that is, the randomness of the movement of users' sight [13]. Avoiding this problem is an important part of this research.

2.1 Eye Tracking Modality and Eye Gaze Estimation Algorithm

To accurately catch user's gaze points and from which to understand user's intent is significant. 3D gaze estimation method considers the user's head moving [19]. Relatively, 2D gaze estimation method use a calibrated sight mapping function to estimate the direction of sight [20]. Obviously combining the 2D and 3D gaze estimation methods will be an ideal way. As such we adopt a scheme to achieve this effect, i.e. adapting the map function of 2D gaze estimation to the natural movements of the head by means of head motion compensation.

PCCR (Pupil Center Corneal Reflection) based 2D eye estimation method [3,21] combining a face color and organ model is used to extract the pupil-glint (PG) vector. A sampling based *HSI* color space modeling method is adopted here. A real-time liner prediction based eye tracking algorithm is used for extracting pupil-glint vector [24]. Gaze mapping equation obtained from a calibrating process transforms the current user's PG vector to the gaze coordinates on view space. The extracted PG vector v is denoted as (x_v, y_v), and the gaze on the screen S_{gaze} is denoted as (x_g, y_g) , the mapping equation $S_g = f(x_v, y_v)$ can be expressed by the following nonlinear equation [2,23],where the parameters a_i and b_i are realized by a 9-point calibrating process :

$$x_g = a_1 + a_2 x_v + a_3 y_v + a_4 x_v y_v + a_5 x_v^2 + a_6 y_v^2 \quad (1)$$

$$y_g = b_1 + b_2 x_v + b_3 y_v + b_4 x_v y_v + b_5 x_v^2 + b_6 y_v^2 \quad (2)$$

To eliminate the discrepancy of PG vector caused by head movement, an iterative compensation algorithm is used.The main idea of head offset compensation is the conversion function $g(v_2, O_2, O_1)$ which can convert arbitrary PG vector to a calibrated position PG vector [14]. When we get the corrected PG vector,we input it into the gaze mapping function (*EQ1*, *EQ2*)to calculate a new screen gaze point S . This is an iterative process and will converge in less than 5 loops. At last we can obtain a gaze point on the screen [22].

2.2 User Intent Recognition from Multimodal Inputs

Eye gazing can reduce the number of hand modality state switches, so make the interactive process be more smooth. In our virtual assembly environment, one of the most significant intent is feature matching. The so called feature matching refers to that *VE* selecting a feature on target part which has an assembly relationship with a feature on current part. Here we just use this intent to demonstate the capability of multimodal inputs. The scenario of feature matching is like this: user selects one part and assigns it as target part, then selects another part and assigns it as current part, and move the two parts by dual hand respectively. When the current part approaches the target part, a current feature (*curFea*) on current part is determined and a feature (*tarFea*) on target part which can match the *curFea* will be searched by algorithm. For some reasons, there may be several potential candidates of *tarFea* that satisfy the requirements. Eye gaze points a feature to cancel out ambiguity.

Two kinds of eye movement states, fixations and saccades are recognized as basic eye movements [16]. A sequence of original gazing points are clustered by the gazing points clustering algorithm. The gazing points that meet both temporal and spatial constraints are collected into a cluster; Secondly, The center of cluster (COC) points are evaluated. Thirdly, determining if the COC points fall in a dwelling space threshold of the center of feature (COF) points. And at last, some smoothing filter process is done to remove the eye movement noises, as such we extract eye movement behaviors depeicted by scan lines.

The collected original signals of viewpoints on screen can be represented as a viewpoint sequence G:

$$G = \{g_i | i = 1, ..., n\}, g_i = (x_{gi}, y_{gi}, t_{gi}) \tag{3}$$

Gaze point g_i can be denoted by a 2D coordinate (x_{gi}, y_{gi}) on the plane together with the time tag t_{gi}. But the users' gaze focus cannot be expressed directly with these original gazing points G due to the midas touch problem. COC points is used to substitute the original gaze data. There are some online clustering algorithm [17]. but the known algorithms e.g. successive k-means algorithm, online hierarchical clustering algorithm and general clustering algorithms with unknown number of clustering [18], cannot meet our real-time requirements. Here we give a gaze clustering algorithm concerning both the temporal constraint and spatial constraints. This can be simplicitly explained by the distance metric formula bellow:

$$d(g_i, g_j) = \frac{k_1 * \sqrt{(x_{gi} - x_{gj})^2 + (y_{gi} - y_{gj})^2} + k_2 * |t_{gi} - t_{gj}|}{k_1 + k_2} \tag{4}$$

Here k_1 and k_2 are weighting coefficients which adjust the importance of spatio-temporal constraints. The viewpoints sequence in the scope of certain spatio-temporal thresholds are chosen to be a viewpoint cluster C_i, and update the current cluster center. The analysis of eye movement is conducted. If the viewpoint contineously falls into a COF range, that is, the distance of a COC and a COF is less than the space threshold, and the duration is more than time threshold, then it can be determined as a visual dwelling, thus it can be thought as focusing on a feature.

Feature matching intent is determined by scenario other than individual eye modality. The eye gaze on interest targets does not always means feature matching intent. Only in a specific perceptive scenario does the eye gaze indicate user current intent in *Interaction3D*. As such, we mainly infer intent with the scenario and eye modality data. Users manipulate virtual parts through dual hand modalities to direct the scenario. Another important role of scenario is *VE* . It is responsible for perceiving spatio-temporal relationships among the virtual objects and gives a representation of specific perception. The virtual objects in the virtual environment, the multimodal inputs and the perceptive representation together compose an interactive scenario. We suppose every modality has

several states. The hand modality state set is { $FreStatic$, $FreTrans$, $FreRot$, $PntGst$, $SwitchTR$, $DisableCN$, $GrspStatic$, $GrspTrans$, $GrspRot$}, etc. For example, $GrspTrans$ represents the state that user is grasping and translating a part. The scenario for feature matching intent can be described as: the user grasps and translates two parts with dual hands individually and makes the two parts approach each other. *VE* perceives matched features and highlight the features. User's eye casts a gaze ray through a $tarFea$.

3 Experiments and Discussion

In order to demonstrate the feasibility of eye modality and also explore the way to evaluate and analyze the accuracy of eye modality in *Interaction3D*, we design an eye tracking experiment. The multimodal intent understanding was evaluated in a user study in which computer students were tasked with various feature matching exercises carried out using the system *Interaction3D*.

3.1 Participants

The experiment included 20 participants who are all graduate students in university. They are all volunteers and all agree to be honest to report the experiment results. Because the experiment involves a developing eye tracking system, the participants spend several hours to adapt themselves to the eye gaze tracking subsystem. Before the experiment, the participants were also introduced about the virtual assembly system *Interaction3D*, although participants need not do any real assembling operations.

3.2 Apparatus

The system hardwares primarily are personal computer , two cameras,and a 3D space mouse. The CPU is Intel Core 2,3.0 GHz, with 2G memory. The screen is 19 inch LCD at a resolution of 1280*1024 pixels. The cameras are all *Panasonic HDC-SD60*. Dual cameras are fixed on a bracket and located just below the screen. The video card is *Nvidia Geforce 450*. The 3D mouse is *3Dconnexion Spacemouse XT*.

Software used in the experiment includes two prototype systems, eye tracking system and virtual assembly system , and a small statistical program to experiment itself. *Interaction3D* is developed on virtual environment development platform *Open Inventor 5.0*, and the eye tracking system is on the platform *OpenCV*. During the interactive process, participants eye movement are limited in the range of 10° with an average viewing distance of 50cm. Participants use 3D space mouse to pick and move the parts to a place to garrantee there is a minimal distance 70 pixels between different geomeric features. Another software is programmed specifically for the experiment which draws out the eye gaze points with colors indicating timing, gives a statistical results about the clustering of the gaze points, and calculate errors of the gaze points respect to the center of feature.

(a) (b)

Fig. 2. Experiment deployment. (a) input devices, (b) multimodal testing.

3.3 Procedure

Participants were seated in front of the computer display at a distance of about 50cm, and were allowed to move heads within a space of approximately 200*200*300 mm (width*height*depth). In the stage of experiment preparation, a process of nine-point calibration is done for every participant. Before the experiment, participants were given a brief practice session to familiarize themselves with the input devices and tasks, see figure 2. During the regular trails, the participants firstly uses 3D space mouse to pick and move a part to the middle of the screen and make it occuping about 1/3 view space, secondly to push an icon on the interface to start eye tracking procedure and gaze at the features one by one for 10 seconds totally. thirdly, the participant push an icon to stop the eye tracking procedure. and the gaze data were collected in a log automatically. Forthly, experiment designer takes a screenshot for each trial. Participants were required to repeat the experiment three times with different part each time. At last, the participants completed an experiment questionnaire to describe which features were gazed at for the different three parts respectively.

3.4 Design

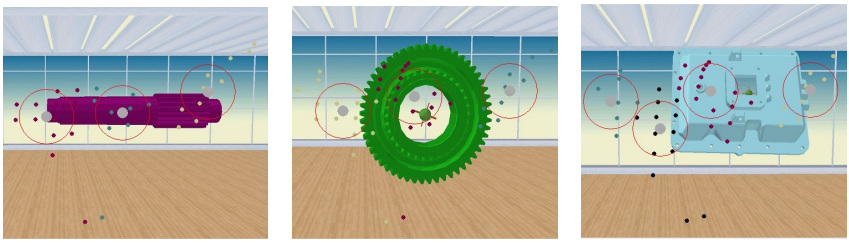
There were five typical parts and totally fifteen features were under consideration. We had named every part and have listed the features of every part in a questionnaire . The participant would just click the parts and number the features to complete the questionnaire.

The gaze point and gaze timing on view plane are drawn with colored dots. Then the gaze points are clustered with red circles, and COCs were evaluated and represented with the grey dots, see figure 3.

Evaluation use three metrics, the success rate of feature selection, the error between COC and COF, and the variance of clusters. When we have got COC, we calculate the distance between the points COCs and COFs, where the latter data comes from the quetionaires. When a distance between a COC and a COF is less than a threshold d_0, we say that the feature is successfully selected by the participant.

3.5 Results and Discussion

The experiment results are given in table 1. Three types of feature deployments are seperately tested. The correspongding explicit demonstrations of the cases are shown in figure 3. We can discover from the table that the feature selection success rates are very high for all the three types. The average errors between COCs and COFs were less than 10 mm. If the error threshold was 15mm, the features that had reported by the participants would nearly all be successfully selected. We can see from the data that the error for linear feature deployment is a little bit less than these for the other two deployments. The average error variances of the three types are all big values, and the values for array deployment and radial deployment are bigger than that of the linear deployment.



(a) (b) (c)

Fig. 3. Experiment examples. (a) shaft features, (b) gear features, (c) casecap features.

The errors between COCs and COFs come from the eye tracking algorithms used in the eye modality. It makes the user's sight have certain uniform deviation and it is less accurate in vertical direction. This measure can be considered as system error that should be minimized. The variances of errors represent the source gaze points distribution. The results illustate that the estimated eye gaze points themselves can not be directly used in intent inferring, while COC substitutes the source data makes eye modality more practical. This measure can be considered as a human factor that can be reduced. We can discover from the experiment result source data that the variances vary between the different participants. This confirms the statement that human's eyes are contineously moving when they look at something [25].

Table 1. The experiment result

FeatureDeploy	SelSuccessRate	AveFocusError	ErrorVariance
Linear	98.4%	7.26(mm)	79.21
Array	96.7%	9.86(mm)	93.36
Radial	97.1%	9.53(mm)	81.29

The experiment design considered only the eye gaze feature selection in static scene. This is because analysis of eye modality accuracy is very critical for multimodal intent inferring. From the static scene gaze evaluation, we really discovered the factors that make effects to the eye modality and have found the way to promote the efficiency of eye modality. Because user direct manipulations in virtual environments are very slow compared with eye tracking frame rate, the gaze points cluster analysis in dynamic assembly operations has not much difference from that of static situation. The experiment results give us confidence to using eye modality to infer user's intents.

4 Conclusion

In this paper we presented a novel research on intent capturing in virtual assembly environments. We built a prototype of multimodal virtual assembly system and use eye gaze to help system infer feature matching intent. Although only one intent is analyzed in this paper, the work are foundamental for future intent driven virtual environment system construction. With respect to virtual assembly, eye modality can be used to substantially reduce the cognitive overload needed for designers by inferring their interactive intents. An experiment was designed to validate the feasibility of eye modality and the potential of using multimodal input to infer user's intents in virtual environments. Results of the experiments were favorable. The online eye gaze cluster algorithm can provide an stable and accurate feature selection in virtual assembly scenario. This means that it can well support *VE* system to choose a feature from a set of candidates and as such infer the feature matching intent in the scenario.

The work introduced in this paper is still preliminary. We have just finished a specific intent capturing with eye gaze modality. Next we will explore more intents with eye gaze modality. In the following research we need to continue to study how to make eye gaze modality more applicable to designers. We will design an experiment which can validate the eye gaze modality in a dynamic multimodal situation.

References

1. Bolt, R.A.: The Human Interface. Lifetime Learning Press, California (1984)
2. Argue, R., Boardman, M., Doyle, J., Hickey, G.: Building a Low-Cost to Track Eye Movement. Faculty of Computer Science, Dalhousie University (December 9, 2004)
3. Guestrin, D., Eizenman, M.: General Theory of Remote Gaze Estimation Using the Pupil Center and Corneal Reflections. IEEE Transaction on Biomedical Enginering 53(6), 1124–1133 (2006)
4. Chen, M., Luo, J., Dong, S.: Task-Oriented Synergistic Multimodality. In: Proceedings of the First Interactional Conference on Multimodal Interface(ICMI 1996), pp. 30–33. Tsinghua University Press, Beijing (1996)
5. Lin, Y., Chen, M., Luo, J., et al.: An Architecture for Multimodal Multi-Agent Interactive System. In: Proceedings for Interactional Conference on CAD/CG 1997. Interactional Academic Press, Beijing (1997)

6. Ford, N.: Cognitive Styles and Virtual Environments. Journal of the American Society for Information Science (April 2000)
7. Dong, S.H., Chen, M., Luo, J.: The Model, Method and Instance of Multimodal User Interface. Universitatis Pekinences 34(2-3) (April 1998)
8. Fang, Z.G.: Visual Line Tracking Technology and Its Application in Multimodal User Interface. Systems Engineering and Electronics
9. Jacob, R.J.K.: What You Look at is What You Get: Eye Movement-Based Interaction Techniques. In: CHI 1990 Poceeding. ACM (1990)
10. Salvucci, D.D., Anderson, J.R.: Intelligent gaze-added ingerface. In: CHI 2000 Conference Proceedings. ACM Press, Netherlands (2000)
11. Kumar, M., Paepche, A., Winograd, T.: EyePoint Practical Pointing and Selection Using Gaze and Keyboard. In: CHI 2007, San Jose, CA, USA, pp. 421–430. ACM Press (2007)
12. Tanriverdi, V., Jacob, J.K.: Interaction with eye movements in virtual environments. In: CHI 2000 Conference Proceedings. ACM Press, Netherlands (2000)
13. Barfield, W., Furness, T.A.: Virtual Environments and Advanced Interface Design. Oxford University Press, Oxford (1995)
14. Zhu, Z., Ji, Q.: Eye Gaze Tracking Under Natural Head Movements. In: Proceeding of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR (2005)
15. Kumar, M., Klingner, J., Puranik, R., Winograd, T., Paepcke, A.: Improving the Accuracy of Gaze Input for Interaction. In: ETRA 2008 - Proceedings of the Eye Tracking Research and Application Symposium, vol. 26-28, pp. 65–68 (March 2008)
16. Carpenter, R.H.S.: Movements of the Eyes. Pion Ltd. (1988)
17. Omran, M.G.H., Engelbrecht, A.P., Salman, A.: An Overview of Clustering Methods. Intelligent Data Analysis 11(6), 538–605 (2007)
18. Guha, S., Meyerson, A., Mishra, N., Motwani, R., O' Callaghan, L.: Clustering data streams: Theory and Practice. IEEE Transaction on Knowledge and Data Enginerring 15(3), 515–528 (2003)
19. Beymer, D., Flickner, M.: Eye gaze tracking using an active stereo head. In: Proceedings of the Computer Vision and Pattern Recognition, Madison, pp. 451–458 (2003)
20. Morimoto, H., Mimica, M.: Eye gaze tracking techniques for interactive applications. Computer Vision Image Understand, Special Issue on Eye Detection and Tracking 98, 4–24 (2005)
21. Matsumoto, Y., Zelinsky, A.: An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement. In: Proc. 4th IEEE Int. Conf. Automatic Face & Gesture Recognition, pp. 499–504 (2000)
22. Shih, S.W., Liu, J.: A Novel approach to 3-d Gaze Tracking using Stereo Cameras. IEEE Trans. Syst., Man and Cybern., Part B, No 1, 234–245 (2004)
23. Duchowski, A.T.: Eye Tracking Methodology: Theory and Practice. Springer (2002)
24. Cheng, C., Jingjing, D.: Research and Realization on Real-time Linear Prediction Algorithm for Desktop Eye-Tracking. Acta Electronic Sinica 37(4A) (April 2009)
25. Slater, M., Steed, A., Chrysanthou, Y.: Computer Graphics and Virtual Environments: From Realism to Real-time. Addison Wesley (2002)