Real-Time Eye-Interaction System Developed with Eye Tracking Glasses and Motion Capture

Haifeng Bao, Weining Fang(✉), Beiyuan Guo, and Peng Wang

State Key Laboratory of Rail Traffic Control and Safety,
Beijing Jiaotong University, No. 3 Shangyuancun Haidian District,
Beijing 100044, People's Republic of China
{12116341,wnfang,byguo,14116350}@bjtu.edu.cn

Abstract. In the industrial environment such as aircraft cockpits and train driver's cab we wished to real-timely acquire the eye-tracking position and made it synchronous with all controlled digital screens with which the machine could dynamically response to the user' s current situation awareness (SA). Wearable eye-tracking glasses could only provide the relative position to the captured video, using which we gathered the data of the eye movement data (2DOF). While the motion capture device could only provide the position and orientation data, using which we accessed the displacement and angular displacement of the head (6DOF). We combined such two devices together into a novel real-time eye-interaction system to synchronize the user's visual point on the screens. A spatial transform algorithm was proposed to calculate the visual point on the multiple digital screens. With the algorithm and the human factors analysis the machine could strengthen its dynamic service abilities.

Keywords: Eye-interaction system · Eye tracking glasses · Motion capture

1 Introduction

Nowadays the eye-tracking equipment have many types. The core function of the eye-tracking system is to judge eye rotation and make it synchronization with the image of the time. These systems are generally divided into 2 types.

One is the contact type which is usually driven by the electro-oculography (EOG) signal which need to be gathered with the wearable electrode slice in the eye. The EOG signal is rapidly conduction which also has nothing to do with the light or the operating environment. Many researchers used the EOG signal as the input interface to computer [1–3]. The developed system was real-timely well-used. But the detection ranges of the interface only included less countable directions which is the big weakness of these type.

The other type of the eye-tracking system is non-contact equipment which is the most popular and wide used on account of the non-invasion. These systems usually used an infrared sensor or several web cameras to recognize the facial features or the pupil rotation. These non-contact eye movement systems mainly include desktop application

T. Ahram and C. Falcão (eds.), *Advances in Human Factors in Wearable Technologies and Game Design*, Advances in Intelligent Systems and Computing 608,
DOI 10.1007/978-3-319-60639-2_8

for the digital screen and wearable application for the moving scene based on the gathered video captured by the micro camera on the glasses or a hat. The desktop application is used with a bar-like equipment for instance Tobii eye tracking system and SMI desktop eye tracking system which have a build-in infrared sensor. Rózanowski and Murawski went through further research which made the sensor more reliable in the harsh environment and also improved the infrared recognition algorithm [4, 5]. Besides the camera video recognition solution was also studied. Hao and Lei got the head and eye rotation data by recognizing facial features [6]. Because the eyes are very near from the front camera of the phone it is easy to recognize the pupil features. Lee and Min got the eye tracking path with the pupil data [7]. More further, Panev and Manolova used the camera and 3D depth sensor to find out the head and eye rotation [8]. In the desktop application mentioned above the infrared and the cameras must be pre-fixed in front of the eye for the algorithm which lead to a narrow field of view, just on the screen. But it can be real-timely interacted as a human-machine interface which could help the disabled even play the game [9]. In the wearable application the eye-tracking system can be used in wide field of view and the data is more accurate because the sensors are just designed on the glasses near the eyes. Also there are also several novel ways to gather the data. Using the specially designed mark points on the glasses and fixed capture cameras Lin et al. got the head gesture data and made it an eye-controlled human–machine interface [10]. In his research the movement of the eye was not considered which made the system only responsible with the head move. While the commercial wearable equipment is well designed like SMI eye-tracking glasses. But all the wearable equipment shows what you are staring at based on the moving video captured on glasses not based on a fixed screen. Such wearable equipment could only be used in the post analysis which is completely on the opposite of the desktop application.

In order to real-timely evaluate the user's SA and tune the human-machine display interface in a train or plane cab, we need to develop a new real-time eye-interaction system. The real-time of the desk application and the wide usability of the wearable application must be integrated together based on the current eye-tracking system. In this method, we could make the machine real-timely get which part you are staring at now and which part you are not noticed for a long time. This research gives the machine a real-time interface and make it possible that the machine can real-timely change the display strategy.

As the concept of awareness situation was defined by Endsley in 1995 [11], the researchers have proposed some method to assess it. In particular, Endsley came up with the situation awareness global assessment technique (SAGAT) which was an objective measure of SA [12]. With our real-time interface the machine could automatically answer the queries to replace the ones needed in the SAGAT during freezes in a simulation. When SA is decreasing to the waring edge, the display of the machine could change their interface to highlight the overlooked information based on the Itti's attention map [13]. According to the SEEV model [14], the machine will improve the non-noticed part's saliency until the user see it.

We wish this unique wearable equipment could be an input interface of the machine to enhance the machine's perception of the user's situation aware and strengthen the machine's dynamic service abilities.

2 System Description

As mentioned above, the core function of the eye-tracking system is to define the head and eye rotation. With the eye tracking glasses, we could get the eye movement and with the motion capture, we could get the movement of the head. Combined with these two devices data with the software SDK, our designed eye-interaction system could get the head line (head gesture) and visual line (considering the eye movement). Meanwhile the system used physical parameters of the screen and device such as the length, height and field of view to calculate the visual point on the digital screen. The algorithm runes processes from the root node of body (the waist joint), the eye point, the head line, the visual line, finally to the visual point on the digital display screen (Fig. 1 shows the algorithm processes 1–5). Every screen need to be calibrated with 3 points to be defined its plane position in the space before used. Our eye-interface system is real-time and wide field of view which could be used as a novel human-machine interface.

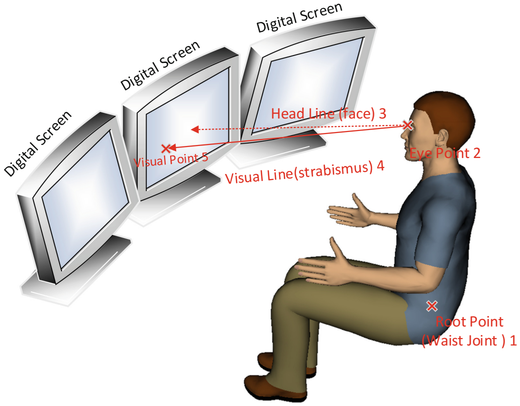

Fig. 1. The spatial relationships between the key control points and the lines. Calculation sequence is from the root point, the eye point, the head line, the visual line, finally to the visual point.

2.1 Root Point and Coordinate Definition

Usually the user of this system is not moving who is standing mostly sitting on the chair such as drivers, pilots and system controller. So the screens location is an invariable data relatively to the user during the use period. In order to construct the algorithm, we set the waist joint of the user as the root point of the coordinate which is not moving during use. Based on the root point we set up the right-hand coordinate system in which the X axis is positive in the forward direction while the Y axis is positive in the left direction and the Z axis is positive in the upward direction (Fig. 2 shows the coordinate).

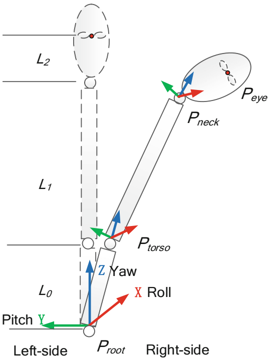

Fig. 2. The define of the key parts of the human segments, coordinate and rotation angles.

2.2 Eye Point and Head Line Calculation

In order to calculate the visual line of the user which need the eye point (the root point of the visual line) and the direction vector of the line. Wearable eye-tracking glasses and motion capture device are the common devices in human analysis. Using the motion caption APIs, we could real-timely get the postures of the key joints. Combined with the joint segment parameters the eye point position relatively to the root point could be calculated.

The size data of the human segments must be pre-given in the process. It contains 3 length size parts between the joints or key points, marking L_0, L_1, L_2. Unusually the motion capture device could provide the postures, marking *Roll(α)*, *Pitch(β)*, *Yaw(γ)*. Specially the rotation data which is gathering from the motion capture device must be defined whether it is a Euler angle in a dynamic coordinate system. It is a big different about the angles definition whether the coordinate rotates every time. Based on the test result the FAB system which is used in this research as the motion capture device provides the Euler angles which are in the dynamic coordinate system by the transform sequence (Z-Y-X Yaw-Pitch-Roll). The rotation angels are positive defined by the right-hand rotation rule.

Based on 3D transform method in the advanced mechanism we could build the rotation transform in the system as matrix-1.

$$\begin{aligned} R[x_0, y_0, z_0]^T &= [x_1, y_1, z_1]^T \\ R &= R_Z(\gamma)R_Y(\beta)R_X(\alpha) \end{aligned} \tag{1}$$

Furthermore, we could get the eye point position ($P_{eye\text{-}root}$) by addition of the key control points' position and the head posture rotation matrix ($R_{eye\text{-}root}$) by multiple product of the rotation matrix (Eq. 2).

$$\begin{aligned}P_{eye-root} &= P_{eye-neck}(\gamma_{head}, \beta_{head}, \alpha_{head}, L_2) + P_{neck-torso}(\gamma_{torso}, \beta_{torso}, \alpha_{torso}, L_1)\\ &\quad + P_{torso-root}(\gamma_{root}, \beta_{root}, \alpha_{root}, L_0)\\ &= [x_{eye-root}, y_{eye-root}, z_{eye-root}]^T\\ R_{eye-root} &= R_{eye-neck}(\gamma_{head}, \beta_{head}, \alpha_{head}) R_{neck-torso}(\gamma_{torso}, \beta_{torso}, \alpha_{torso}) R_{torso-root}(\gamma_{root}, \beta_{root}, \alpha_{root})\end{aligned} \tag{2}$$

Some researchers used the infrared sensors or the cameras to get the head gesture [5–8]. But such indirect accessing method need to recognize the feature of the user which may cause the error of the use because of the difference between the users also leading to the narrow usable viewing field. Nowadays, the motion capture device is tiny, wearable and wireless which provides the direct data of the head gesture. So it is a better choice than the other sensors.

2.3 Visual Line Calculation

In order to confirm the visual line, we need not only the head posture but also the eye movement. So the wearable eye tracking glasses are the best chosen device for the eye movement data. With the help of the glasses' API we could get the relative visual point in the moving video captured from a micro camera on the glasses (Fig. 3).

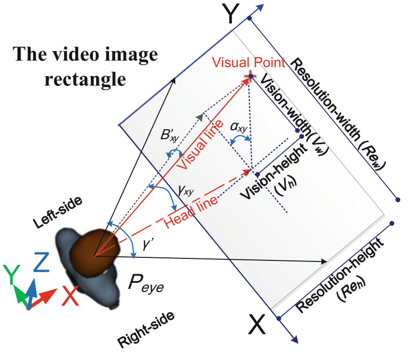

Fig. 3. The spatial relationships between the head line and the visual line. The visual line is obtained by two rotation transformations ($R_Z(\gamma), R_X(\alpha)$).

Based on the proportional relationship between the video resolution and the relative visual point we could get the elevation angle α_{xy}. Using the pre-test data or the glasses' argument list, the video view angle γ' can be defined. Then we could get the polar angle γ_{xy} by the scale calculation. Thus the direction vector of the visual line (Eq. 3) can be defined and the parameter equation of the visual line (Eq. 4) can be calculated from the eye point. After such work the position and gesture of the eyeball have been recorded with 6-DOF data by definition of the visual line.

$$\begin{aligned} R_{vision} &= R_X\left(\alpha_{xy} = \frac{V_h}{V_w}\right)R_Y(0)R_Z\left(\gamma_{xy} = \frac{V_w}{Re_w}\gamma'\right)R_{eye-root}[1,0,0]^T \\ &= [m_{vision}, n_{vision}, p_{vision}]^T \end{aligned} \tag{3}$$

$$\begin{cases} x = \lambda m_{vision} + x_{eye-root} \\ y = \lambda n_{vision} + y_{eye-root} \\ z = \lambda p_{vision} + z_{eye-root} \end{cases} \tag{4}$$

In addition, as shown in the paragraph the β'_{xy} is the exact elevation angle which is not used. It is not easy to be measured because of the difficulty of the distance measurement between the eyes and the screen. Some researchers used the 3D depth sensor to get the data [8]. But this solution will decrease the user's usable viewing field also bring the new device to the environment. Others used the physical relationship to calculate the distance [10]. This method need the user not to break the distance by moving the head during the test.

In our research we held the ideas that it was no more device to be involved in except the motion capture device and the eye tracking glasses which made us could not reach the distance data. But we could still succeed in calculating the screen distance with the math method on the next step.

2.4 Screen Plane Calculation

The digital screen is a space plane which is unknown to the base coordinate of the human. There is a commonsense that we need 3 points to define a space plane. In order to locate the screen plane, the designed system will guide the user to focus on the 3 calibrated points on the vertex of the screen rectangle during the calibration process while the 3 visual lines will be recorded in the system as lines equations set (Eq. 5) (Fig. 4).

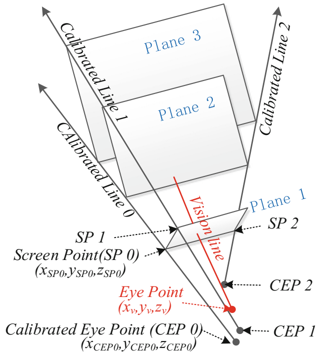

Fig. 4. The key points on the screen SP_0, SP_1, SP_2 are on the 3 calibrated visual lines which were recorded on the guidance of the system when starting. The screen rectangles on the plane 1–3 are the possible solution.

$$\text{CLine*0,1,2:}\left\{\underline{\lambda_*}\middle|m_{CL^*}, n_{CL^*}, p_{CL^*}, x_{CEP^*}, y_{CEP^*}, z_{CEP^*}\right\} \tag{5}$$

As mentioned above, we haven't any distance sensor in the system which makes us cannot define the points which the user saw on the screen that is we cannot get the λ_0, λ_1, λ_2 (in the Eq. 5) which leads to various possible planes shown in the Fig. 4. It pushed us to find a new way to achieve the plane equation.

We must import some physical parameters of the screen which are the width and the height of the digital screen to help to confirm the correct plane. Meanwhile, the Line ($\overrightarrow{SP_0SP_1}$) is perpendicular to Line ($\overrightarrow{SP_0SP_2}$). Through these 3 restrictions we could build the equation set (Eq. 6).

$$\begin{cases} \text{Width: } \left|\overrightarrow{SP_0SP_2}\right| = Screen_{width} \\ \text{Height: } \left|\overrightarrow{SP_0SP_1}\right| = Screen_{height} \\ \text{Perpendicularity: } \overrightarrow{SP_0SP_1} \bullet \overrightarrow{SP_0SP_2} = 0 \end{cases}$$
$$\Rightarrow \begin{cases} f_1(\underline{\lambda_0}, \underline{\lambda_1}, \underline{\lambda_2}) = Screen_{width} \\ f_2(\underline{\lambda_0}, \underline{\lambda_1}, \underline{\lambda_2}) = Screen_{height} \\ f_3(\underline{\lambda_0}, \underline{\lambda_1}, \underline{\lambda_2}) = 0 \end{cases} \tag{6}$$

In the equation set we can find there are 3 unknown variables (λ_0, λ_1, λ_2) and 3 equations. With geometrical relationship analysis there is only one solution of the equations. It is a little difficulty that we find solution by manual steps. But fortunately with the help of the computer we could use the existing math software library to find the solution of the equation set. Once we find the λ_0, λ_1, λ_2, we get the 3 key points SP_0, SP_1, SP_2 on the screen and the plane equation (Eq. 7).

$$\begin{aligned} &\text{Normal vector: } \overrightarrow{SP_0SP_1} \times \overrightarrow{SP_0SP_2} = (A,B,C) \\ &\text{Plane: } \mathrm{P}\left(x_{SP0}, y_{SP0}, z_{SP0} \middle| A,B,C\right) = 0 \end{aligned} \tag{7}$$

As the algorithm described, the plane equation is based on the root point and the base coordinate. As long as the user don't move the waist position, the head and eyes can move in a wide viewing field during the use which is an improvement than the current equipment.

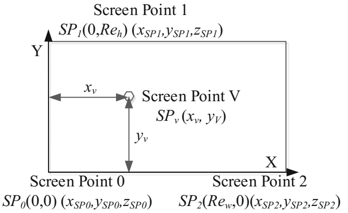
Screen Point 1
$SP_1(0,Re_h)$ $(x_{SP1},y_{SP1},z_{SP1})$
Y
x_v
Screen Point V
$SP_v(x_v, y_V)$
y_v
X
Screen Point 0
$SP_0(0,0)$ $(x_{SP0},y_{SP0},z_{SP0})$
Screen Point 2
$SP_2(Re_w,0)(x_{SP2},y_{SP2},z_{SP2})$

Fig. 5. The screen coordinate is based on the SP_0 point. The SP_1 and SP_2 are the vertices of the screen monitor. Screen point V is the virtual point which the user is starting at.

2.5 Visual Point Calculation

On the premise of getting the screen plane equation, we could calculate the intersection between visual line and the plane which is a point position in the three-dimensional space data ($\underline{x}$, $\underline{y}$, $\underline{z}$) in equation (Eq. 8). As long as we get the parameter $\underline{\underline{\lambda}}$ we get the visual point in space based on the root point.

$$\begin{cases} \text{Plane: } \mathrm{P}(x_{SP0}, y_{SP0}, z_{SP0} | A, B, C) = 0 \\ \text{Visual Line: } \mathrm{L}(\underline{\underline{\lambda}} | m_{vision}, n_{vision}, p_{vision}, x_{eye-root}, y_{eye-root}, z_{eye-root}) = 0 \end{cases} \tag{8}$$

However, we need print the point on the digital screen. That is, we need transform the point from the three-dimensional space based on the root of the human to the 2-dimensional plane based on the screen basic point. Choosing the starting point of the screen image resolution as the base control point we could build a new 2-dimensional coordinate (shows in Fig. 5).

By counting the cross product of the vector and some geometric methods, we could get the physical position of the screen point V (the visual point) which is described by the parameters (unit of length). Then based on the proportional relationship between the physical length and height data of the screen and the resolution of the screen, we could find the exact virtual point (x_v, y_v) (unit of resolution) on the screen and mark it with the software in real time. Pantograph equation shows in (Eq. 9).

$$\begin{cases} x_v = \dfrac{\left|\overrightarrow{MP_0MP_v} \times \overrightarrow{MP_0MP_1}\right|}{\left|\overrightarrow{MP_0MP_1}\right|\left|\overrightarrow{MP_0MP_2}\right|} \times Re_w \\ y_v = \dfrac{\left|\overrightarrow{MP_0MP_v} \times \overrightarrow{MP_0MP_2}\right|}{\left|\overrightarrow{MP_0MP_2}\right|\left|\overrightarrow{MP_0MP_1}\right|} \times Re_h \end{cases} \tag{9}$$

2.6 Data Synchronization

A wide viewing field eye-interaction system require synchronizing all the controlled digital screens. A data synchronization module was developed to estimate whether the computational visual point was on one of the screens. It is a center design which processes all data and records the eye-tracking path.

In order to indicate the visual point, the system will send commands to the sub-module in the screen and mark the point on the interface. Also with the enough restricts or importing in advance the system could synchronize that point on the control desk. For example, the button or the indicator on the driver's desk could light up in specific color to indicate you are staring at it.

3 Application

SA is a word for the human which includes 3 levels (perception, comprehension and projection) [11]. The system is designed to real-timely get the visual point which the user is staring at. With the help of the designed eye-interaction system we could calculate the user's perception. The information was divided into different important levels(*IL*). The sum of all the information importance score is 100. Once one information is changed in the display interface, the score will reduce the corresponding value (*ILc*). Once the user notices the information the score will recover the corresponding value(*ILn*). We set the starting score 100 as the best SA perception (SA_{P}-). And the current SA perception could be defined in Eq. 10. When the SA_P is under the warning level (60 scores). The system will highlight the information which have lost focus for a long time and whose important level is high.

$$SA_P^{N+1} = SA_P^N - \sum\nolimits_i ILc_i^{N+1} + \sum\nolimits_i ILn_i^{N+1} \Big| SA_P^0 = 100 \tag{10}$$

The attention map could be calculated by the Itti's model [13] when the user is starting at a point on the screen. We could cumulate saliency-based visual attention rate with the elapsed time in Eq. 11. The $\bar{S}$ is the average number of the attention saliency rate during a task with the time t_e. The information area of interest is A_0 to A_i. The IL_A is the importance level of the area. The function $\delta(A, t)$ is the attention saliency intensity of the area A at the time t. Considering the SEEV model [14] the user will take less effort to see the salient area which means the less $\bar{S}$ is the better.

$$\bar{S} = \frac{1}{t_e} \sum\nolimits_{A=0}^{A_i} IL_A \int_{t=0}^{t_e} \delta(A, t) dt \tag{11}$$

With the dynamic saliency readjustment based on the SA_P and $\bar{S}$, the machine will enhance the perception of the human and offer the better service.

4 Further Work

The algorithm is an exact solution of the virtual point in mathematics. The core difficulty of the algorithm is to find the root of the ternary quadratic equation set in (Eq. 6). Through the complex spatial analysis, we could know it is only one reasonable solution. But there is also an unreasonable solution in the negative direction of the axis. Also we could only use the computer math tools to find a numerical solution, which leads to some uncertain solving process.

Furthermore, as we known applying the computer math library to solving the root is a kind of numerical calculation. So the numerical stability of equation must be considered. Also it will take time to real-timely solve the root which brings the delay to the system. If it had some way to reach the exact solution of the equation set with a mathematical expression, the problem will be solved satisfactorily.

In addition, the algorithm includes a lot of steps which may bring the error added up. The error analysis is a lot of work. Maybe it is possible to simplify the algorithm by assuming the calibrated eye points are in the same position (CEP_0, CEP_1, CEP_2) (shows in Fig. 4). With such work the Eq. 6 will be simplified and easily be solved.

For further simplification, we can assume the eye point also in the same position which will simply the Eq. 8. It will be a great save of calculated amount of the algorithm. But the error must be calculated to show whether it is in an acceptable range with such simplified calculation model.

Acknowledgments. This paper is supported by the National Natural Science Foundation of China (Grant No. 51575037).

References

1. Lv, Z., Wu, X.P., Li, M., Zhang, D.X.: A novel eye movement detection algorithm for EOG driven human computer interface. Pattern Recogn. Lett. **31**, 1041–1047 (2010)
2. Aungsakun, S., Phinyomark, A., Phukpattaranont, P., Limsakul, C.: Robust eye movement recognition using EOG signal for human-computer interface. In: Zain, J.M., Wan Mohd, W.M.B., El-Qawasmeh, E. (eds.) Software Engineering and Computer Systems, pp. 714–723. Springer, Heidelberg (2011)
3. Ma, J., Zhang, Y., Cichocki, A., Matsuno, F.: A novel EOG/EEG hybrid human-machine interface adopting eye movements and ERPs: application to robot control. IEEE Trans. Biomed. Eng. **62**, 876–889 (2015)
4. Rózanowski, K., Murawski, K.: An infrared sensor for eye tracking in a harsh car environment. Acta. Phys. Pol. A. **122**, 874–879 (2012)
5. Murawski, K., Różanowski, K., Krej, M.: Research and parameter optimization of the pattern recognition algorithm for the eye tracking infrared sensor. Acta Phys. Pol. A **124**, 513–516 (2013)
6. Hao, Z., Lei, Q.: Vision-based interface: using face and eye blinking tracking with camera. In: International Symposium on Intelligent Information Technology Application, vol. 1, pp. 306–310. IEEE (2008)
7. Lee, E.C., Min, W.P.: A new eye tracking method as a smartphone interface. Ksii Trans. Internet Inf. Syst. **7**, 834–848 (2013)
8. Panev, S., Manolova, A.: Improved multi-camera 3D eye tracking for human-computer interface. In: International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, vol. 1, pp. 276–281. IEEE (2015)
9. Soltani, S., Mahnam, A.: A practical efficient human computer interface based on saccadic eye movements for people with disabilities. Comput. Biol. Med. **70**, 163 (2016)
10. Lin, C.S., Ho, C.W., Chang, K.C., Hung, S.S., Shei, H.J., Yeh, M.S.: A novel device for head gesture measurement system in combination with eye-controlled human–machine interface. Opt. Laser. Eng. **44**, 597–614 (2006)
11. Endsley, M.R.: Toward a theory of situation awareness in dynamic systems. Hum. Factors **37**, 32–64 (1995)
12. Endsley, M.R., Selcon, S.J., Hardiman, T.D., Croft, D.G.: A comparative analysis of SAGAT and SART for evaluations of situation awareness. J. Surg. Res. **172**, 231–232 (1998)
13. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. **20**, 1254–1259 (1998)
14. Wickens, C.D., Helleberg, J., Goh, J., Xu, X., Horrey, W.J.: Pilot task management: testing an attentional expected value model of visual scanning. Technical report, UIUC Institute of Aviation (2001)