Método de estimación de *head pose* para la obtención del punto de atención en base a una cámara web estándar

Camila García[1, 2] and Juan P. D'Amato[1, 3]

[1]PLADEMA, UNICEN, Tandil, 7000, Argentina
[2]Comisión de Investigaciones Científicas, Provincia de Buenos Aires, Argentina
[3]Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina

Abstract. En toda interfaz de Interacción Humano-Computadora (IHC), se requiere un mecanismo específico que permita conocer el lugar donde el usuario tiene centrada su atención. Ciertos enfoques de la IHC se apoyan en la Visión Computacional para obtener esta información a partir del análisis del rostro. En los últimos años, han surgido muchos trabajos relacionados, en los cuales se propone estimar el punto de atención en una escena a partir de la orientación del rostro (o *head pose*). En los mismos, se emplea mayoritariamente hardware de captura específico (como cámaras infrarrojas, cámaras en *stereo* o cámaras 3D), que suele ser muy costoso e inacccesible. Por otro lado, existen diferentes bibliotecas de análisis facial que, tal como se encuentran actualmente, no pueden ser utilizadas para la tarea perseguida. En este trabajo se presenta una versión inicial de un método de estimación de la *head pose* para IHC, basado en reconstrucción 3D y seguimiento de características faciales. Uno de los aportes presentados en este paper es la definición de un pipeline de procesamiento unificado que pueda funcionar en tiempo real. Adicionalmente, se incluye el estudio de factibilidad del uso cámaras web estándar, incluídas en las notebooks actuales. Para validar el esquema propuesto, se comparó la precisión de la estimación en dos implementaciones basadas en diferentes bibliotecas de análisis facial. Se realizó un estudio preliminar con 3 usuarios en diferentes escenarios.

Keywords: head pose, computer vision, HCI, visual tracking

1 Introducción

Para construir interfaces inteligentes de computadoras, es necesario que el sistema conozca la intención del usuario y pueda inferir la zona o el punto en donde el mismo tiene depositada su atención. Dado que el movimiento de la posición de la cabeza de una persona (*head pose*, en inglés) y la dirección de su mirada (*gaze*) están profundamente relacionadas con su atención (e intención), la detección de dicha información puede ser utilizada para respaldar interacciones que no requieran el contacto con dispositivos periféricos [1], [2], [3].

La entrada de datos controlada por la posición del usuario o su mirada, es especialmente atractiva para la selección de elementos y tareas de seguimiento

de blancos porque utiliza los movimientos inherentes a estas tareas como forma de control. Esta clase de interfaces resulta ideal, sobre todo, en aquellos casos en los que la motricidad del usuario se halla comprometida [4], [5].

Los elementos de captura disponibles en la actualidad, han impulsado el avance de las capacidades de procesamiento computacional de las imágenes digitales. Una gran área del campo de procesamiento de imágenes digitales está relacionada con la detección facial, tarea que ha dado lugar a una amplia variedad de análisis y aplicaciones con los más diversos objetivos. Entre las posibilidades que se presentan ante el avance del análisis facial en imágenes se encuentra, precisamente, la capacidad de estudiar la posición de la cabeza de un usuario.

En visión computacional, la pose de un objeto se refiere a su orientación relativa y posición respecto a una cámara. La pose puede variar tanto moviendo el objeto con respecto a la cámara, o la cámara con respecto al objeto. Para el análisis de la posición de la cabeza se ha utilizado, en gran parte, el reconocimiento de puntos de referencia faciales como base [6], [7].

Resulta evidente, entonces, que el procesamiento digital de imágenes puede ser utilizado como cimiento hacia una Interacción Humano-Computadora diferente. Ello permite pensar en el movimiento cefálico del usuario como una variable de control para su comunicación con la tecnología.

Actualmente, el mercado cuenta con la presencia de diversos dispositivos que realizan el análisis de la mirada o de la posición de la cabeza. El desafío actual en este campo se encuentra en el empleo de algoritmos que no requieran hardware específico para la captura de imágenes, sino que puedan realizar un procesamiento robusto y confiable en base a cámaras web estándar y sin costos adicionales.

En los últimos veinte años se ha enfatizado el impacto que tendrían las cámaras embebidas en el avance de los métodos de determinación de la mirada [8], permitiendo una mayor disponibilidad de herramientas asistenciales. Como prueba de ello, es posible ver diversos trabajos en los que se ha abordado la problemática desde entonces, empleando mecanismos que, en su procesamiento, logran compensar las dificultades de utilizar esta clase de dispositivos [9], [10], [12].

2 Método propuesto

Idealmente, tras el procesamiento de una imagen conteniendo una cara, podría obtenerse información sobre la posición de la cabeza respecto a la pantalla y utilizar los ángulos para proyectar un punto a la pantalla. Se presenta en esta sección una descripción general del método seguido para la estimación del punto de atención de un usuario, partiendo desde la captura de la imagen del mismo frente a la cámara de una computadora, con su posterior procesamiento y análisis. La implementación del método se realizó en una aplicación en C++ y utilizando OpenCV [13].

El pipeline implementado puede exponerse de manera general como se indica en la Figura 1. Se ha pretendido desarrollar un método de procesamiento unifi-

cado de forma que, definiendo el conjunto de datos de entrada requerido para su funcionamiento, pueda ser utilizado en conjunto con cualquier herramienta de análisis facial que genere tal información.

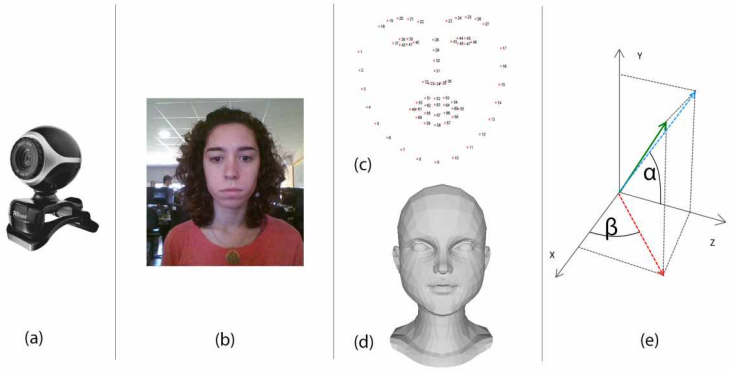

Fig. 1. Pipeline general con las fases del método seguido. (a) Dispositivo de captura, cámara web estándar; (b) Captura de *frames*; (c) Procesamiento 2D y obtención del conjunto de *landmarks* faciales; (d) Malla 3D obtenida mediante el análisis de imágenes o provista de forma estática; (e) Estimación de la transformación y cálculo de los ángulos de rotación de la cabeza.

2.1 Captura de frames

Como ha sido mencionado, para efectuar la captura de imágenes fue propuesta la utilización de un dispositivo disponible al público general por un bajo costo. En la mayoría de los casos, las computadoras portátiles modernas cuentan con una cámara web estándar embebida. Esta clase de periféricos no provee una alta definición en las imágenes capturadas, característica que afecta directamente la precisión y fiabilidad de los algoritmos de procesamiento utilizados en base a ellas. Sin embargo, teniendo en cuenta que el objetivo del análisis realizado en este trabajo se encuentra directamente influenciado por la accesibilidad, se consideró preferible la disminución de la precisión en pos de la posibilidad de brindar capacidad de acceso a un mayor número de personas.

2.2 Análisis 2D de la imagen

El proceso comienza con la extracción de información a partir de una imagen capturada con una cámara web. Sobre ésta, pueden ser aplicados diversos algoritmos de procesamiento que permiten detectar la presencia o ausencia de rostros y, ante la presencia de uno o más, realizar la extracción de puntos característicos faciales o *landmarks*. Esta etapa es determinada como la entrada al

método implementado y se espera, cualquiera sea la biblioteca empleada para su realización, la obtención de un conjunto mínimo de 9 landmarks faciales.

2.3 Modelo 3D facial

Para poder estimar la posición de un objeto respecto a una cámara, se requiere un modelo en tres dimensiones del mismo, que establezca el punto de partida desde el cual analizar sus movimientos. En el caso de estudio, el escenario ideal implica una malla 3D que replique el rostro de cada usuario particular, reduciendo las posibilidades de introducción de errores en los cálculos al proveer las proporciones particulares a cada muestra facial. Sin embargo, es posible reducir la exigencia de los requerimientos utilizando únicamente un conjunto reducido de puntos 3D, correspondientes a una selección de *landmarks* específicos para la detección en el análisis 2D. En los casos en los que no resulta factible obtener un modelo facial del usuario específico que utilizará el sistema, es una posibilidad emplear uno genérico, aunque la precisión de los resultados obtenidos en base al mismo podría verse comprometida.

2.4 Obtención de una matriz de transformación

Contando con los puntos fiduciales 2D y el modelo 3D de la geometría facial, es posible llevar a cabo la identificación de la transformación sufrida por el rostro en un frame determinado respecto a su posición neutral. Dicha transformación es representada por la traslación y las rotaciones en los tres ejes del modelo. La realización de tal tarea fue basada en el algoritmo *solvePnP* provisto por *OpenCV*, que se apoya en el modelo de cámara estenopeica para definir la vista de una escena determinada como la proyección de puntos 3D en el plano de imagen usando una transformación de perspectiva:

$$m' = A[R|t]M', \tag{1}$$

donde: A es la matriz de parámetros intrínsecos de la cámara; [R|t] la matriz de rotación-traslación; t el vector de traslación y M la matriz de los puntos 3D del modelo. La matriz A es independiente de la escena, por lo que puede ser establecida al comienzo del procesamiento y re-usada, utilizando aproximaciones relativas a las características de las imágenes capturadas cuando no se cuenta con información específica sobre las distancias focales o el punto principal de la cámara en uso. Considerando este modelo y la información disponible, el algoritmo previamente mencionado *solvePnP* es capaz de encontrar [R|t], la matriz de rotación-traslación. La misma, expresa los ángulos de rotación percibidos en el rostro con tres grados de libertad, además del desplazamiento del mismo respecto a su posición original.

2.5 Proyección del puntero a la pantalla

Obtenida la matriz de rotaciones, resulta posible extraer los ángulos de rotación correspondientes a los ejes de interés para la estimación de un punto de atención

en pantalla. Las rotaciones relevantes a esta tarea son *yaw* (β) (o guiñada, el movimiento con centro en el eje vertical, que permite desplazar la mirada a lo largo del eje horizontal) y *pitch* (α) (o cabeceo, el movimiento con centro en el eje horizontal, con el cual se puede variar el punto de atención a través del eje vertical). La combinación de estos dos grados de libertad son los que permiten recorrer el plano de la pantalla, pudiendo descartar el movimiento alrededor del eje z (*roll* o alabeo).

$$m_x = \left(\frac{\beta}{\beta_{max}}\right) \times \left(\frac{w}{2}\right) + \left(\frac{w}{2}\right),\ m_y = \left(\frac{\alpha}{\alpha_{max}}\right) \times \left(\frac{h}{2}\right) + \left(\frac{h}{2}\right). \quad (2)$$

El puntero proyectado a la pantalla es obtenido con las ecuaciones en (2), donde m_x y m_y son las coordenadas de desplazamiento, w y h los datos de ancho (*width*) y alto (*height*) de la pantalla, y α_{max} y β_{max} los ángulos máximos considerados para pitch y yaw respectivamente, correspondientes a movimientos máximos establecidos para alcanzar los extremos de la pantalla [14].

2.6 Estabilización de puntero

La salida del pipeline implementado está condicionada por los landmarks 2D resultantes de la utilización de una biblioteca externa de análisis facial y recibidos como entrada para el procesamiento. Considerando esto, la proyección del punto de atención a la pantalla entre un frame y el siguiente puede sufrir fluctuaciones que deberán ser corregidas. Landmarks erróneamente detectados o una detección poco estable generan un efecto de arrastre del error que debe ser corregido de acuerdo al contexto. Para ello, se añadió a la etapa final una implementación del filtro de Kalman [15], un estimador que puede inferir parámetros de interés a partir de observaciones poco precisas.

La técnica fue aplicada sobre los puntos de atención obtenidos, buscando lograr la reducción del ruido percibido. Adicionalmente, permitió eliminar un gran porcentaje de los casos de error que, introducidos en alguna de las etapas del método, repercutían en el resultado final con el puntero proyectado.

3 Resultados

Para todos los experimentos, se utilizó el mismo equipamiento: una notebook con un procesador Intel i7-6500 2.50GHz, 8GB de RAM, cámara web integrada con resolución máxima de 1280x720px, pantalla de 15.6" con una resolución de 1366x768 y bajo el sistema operativo Windows 10 de 64bits.

Las características faciales fueron detectadas con dos herramientas gratuitas que realizan análisis facial 2D a partir de imágenes capturadas con cámaras web estándar: *OpenFace* [10] y *RealSense* de Intel [11]. En el caso de *OpenFace*, el procesamiento efectuado provee, de forma adicional a los landmarks faciales en 2D, un modelo 3D ajustado a la geometría del usuario. Por el contrario, *RealSense* sólo realiza el reconocimiento de puntos característicos en 2D. Para

el empleo del método implementado con tal herramienta, se obtuvo una malla 3D de cada uno de los usuarios que participaron del experimento mediante la funcionalidad provista por la demo de Jackson et al. [16].

3.1 Experimentos realizados

Los experimentos realizados responden a dos características consideradas relevantes a la hora de evaluar la factibilidad del método a ser usado como base para la Interacción Humano-Computadora.

Pruebas de seguimiento con una trayectoria rectangular. Resulta necesario, para considerar la posibilidad de utilizar este método como la base de una interfaz de IHC, asegurar la capacidad de realizar movimientos fluidos con el cursor en pantalla. Para ilustrar tal requerimiento, puede analizarse el empleo de un teclado virtual en pantalla que utilice escritura basada en gestos [17]. Esta clase de entrada se centra en el trazado de diversos patrones sobre los caracteres. En el trabajo publicado por [18] se plantean los conceptos de *pose tracking* y *gaze tracking*. Allí, se propone la exposición de una trayectoria rectangular mediante el desplazamiento a velocidad constante de un punto en pantalla. Para este experimento, dicha prueba fue replicada y realizada por tres usuarios, estableciendo el objetivo de seguir de cerca el recorrido trazado por el punto con el puntero proyectado por la orientación de la cabeza.

Las Figuras 2 y 3 muestran la comparación de los resultados de esta trayectoria para los tres usuarios. Se representa en violeta al Usuario A, en naranja al Usuario B y en verde al Usuario C. El trazado de los puntos proyectados incrementa la intensidad con el transcurso de la exploración.

Tras la realización de estas exploraciones, se pudo plantear un análisis respecto al uso de las rotaciones cefálicas como modo de controlar un puntero en la pantalla. En primer lugar, de forma opuesta a lo planteado por [4] para el movimiento ocular, la cabeza permite la realización de desplazamientos fluídos. Sin embargo, se requiere de un período de adaptación para lograr movimientos precisos. En las Figuras 2 y 3 esto se hace notorio, principalmente, en las esquinas de la trayectoria trazada, donde debe pausarse la guiñada (*yaw*) de la cabeza y comenzarse el cabeceo (*pitch*). Además, no resulta una tarea sencilla mantener una velocidad constante en las rotaciones realizadas. Esto resulta visible en las figuras, ya que las secciones trazadas de forma punteada representan un movimiento veloz o repentino, y aquellas con mayor aglomeración de puntos se corresponden con movimientos lentos.

Pruebas de estabilidad basadas en cuadrícula. Al contemplar la interacción, por ejemplo, con el escritorio de una computadora, se hace evidente la necesidad de poder seleccionar diversos íconos para abrir programas o archivos. Alternativamente, en el caso de un teclado virtual en pantalla, debe ser posible elegir entre los distintos símbolos presentados. Para realizar tal interacción, es

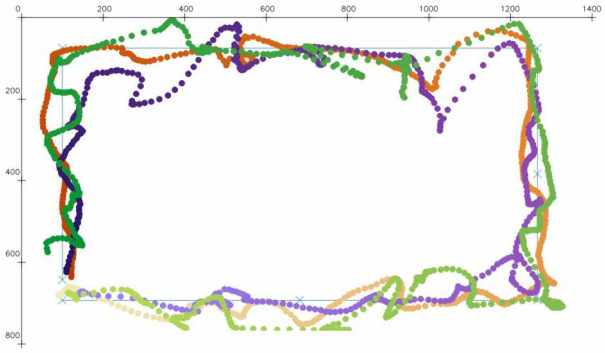

Fig. 2. Experimento de seguimiento de trayectoria con tres usuarios realizado de forma individual con *OpenFace*.

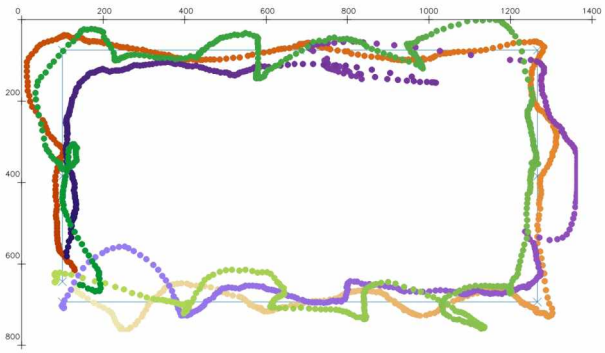

Fig. 3. Experimento de seguimiento de trayectoria con tres usuarios realizado de forma individual con *RealSense*.

requerimiento medir el nivel de precisión con el que se cuenta cuando el usuario pretende mantener su atención en un punto de manera estable.

Para llevar a cabo este estudio se desarrolló un prototipo de aplicación de estilo lúdico. En la dinámica definida, la pantalla es seccionada visualmente con una grilla de cuadrados del mismo tamaño, como muestra la Figura 4. De forma secuencial y aleatoria, son expuestas en la pantalla pistas visuales en distintas regiones de la pantalla, con el objetivo de indicar al usuario hacia dónde debe orientar su cabeza.

La exploración implementada busca estimar el tamaño mínimo que deben tener los elementos expuestos en pantalla para permitir su selección de manera cómoda. Así, la pantalla es inicialmente dividida en regiones de 96x96 pixeles, posteriormente de 48x48 y, finalmente, de 36x36 pixeles.

En el Cuadro 1 se exponen los resultados numéricos correspondientes a las pruebas realizadas por tres individuos con ambas herramientas. Los Usuarios E y D experimentaron la dinámica de interacción por primera vez inmediatamente

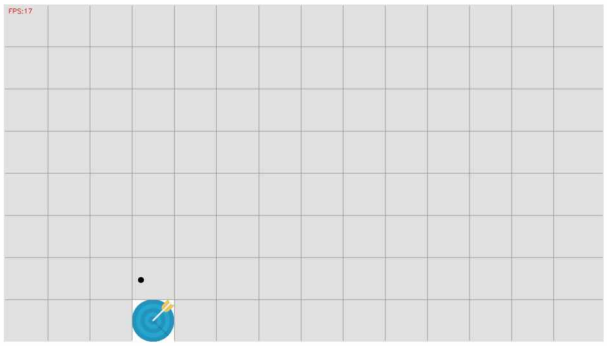

Fig. 4. Captura de pantalla de un experimento basado en cuadrícula.

antes de participar de la actividad explorativa, a diferencia del Usuario B que ya había realizado diversas pruebas con anterioridad. Puede verse en los valores expuestos que, debido a la experiencia adquirida por el Usuario B respecto al modo de funcionamiento y la respuesta ofrecida por *OpenFace* y *RealSense* a sus movimientos, logró un desempeño casi perfecto. Debido al modelo 3D adaptativo con el que cuenta *OpenFace*, los dos sujetos inexperimentados pudieron llevar a cabo la actividad sin mayores problemas. En cambio, dado el modelo 3D estático utilizado para las pruebas con *RealSense*, se pudo observar una alta susceptibilidad a cambios de iluminación y posición del individuo respecto a la cámara, provocando un desempeño pobre, de ambos usuarios, al incrementar las restricciones.

Table 1. Resultados de los experimentos realizados con tres usuarios de las pruebas basadas en regiones.

Usuario B	96x96	48x48	36x36
Tiempo Promedio con *OpenFace*	5.576s	6.697s	6.931s
Tiempo Promedio con *RealSense*	6.434s	7.063s	7.171s
Regiones no logradas con *OpenFace*	0	0	0
Regiones no logradas con *RealSense*	0	0	1
Usuario D	96x96	48x48	36x36
Tiempo Promedio con *OpenFace*	5.841s	5.879s	6.834s
Tiempo Promedio con *RealSense*	7.172s	6.679s	7.104s
Regiones no logradas con *OpenFace*	0	0	2
Regiones no logradas con *RealSense*	1	1	6
Usuario E	96x96	48x48	36x36
Tiempo Promedio con *OpenFace*	5.814s	6.119s	7.551s
Tiempo Promedio con *RealSense*	5.396s	7.437s	7.623s
Regiones no logradas con *OpenFace*	0	0	0
Regiones no logradas con *RealSense*	0	2	6

Respecto a los tiempos promedio calculados para la selección de regiones, en la mayoría de los casos se pudo observar que la estabilidad brindada por *OpenFace* permitía alcanzarla con mayor facilidad y, como se pudo prever, la disminución del tamaño del área restrictiva introdujo demoras en el tiempo requerido para seleccionar una región en todos los casos.

4 Conclusiones

Se estudió la factibilidad de la realización de una herramienta asistencial para la interacción humano-computadora basada en los movimientos de la cabeza de los usuarios. El enfoque se basó en el modelo de *camera mouse*, presentado por Nabati y Behrad [9], para la implementación de un elemento de control que no requiera contacto físico y sea soportado por hardware ampliamente disponible y de bajo costo. Para llevarlo a cabo, se analizaron y utilizaron dos herramientas gratuitas particulares: una de ellas de código abierto, y otra comercial.

Al desarrollar el estudio planteado, se utilizó un *pipeline* unificado como estructura inicial para conectar las diversas tareas implicadas en la proyección de un puntero en la pantalla, a partir de la captura de imágenes a través de una cámara web estándar.

La tasa de procesamiento de imágenes del sistema implementado es altamente dependiente de la biblioteca base empleada para el análisis facial en 2D. Al utilizar *RealSense* se llegó a procesar a 30fps, dado que la misma cuenta con optimizaciones en su implementación para procesadores Intel. Con *OpenFace* la tasa máxima alcanzada fue de 20fps; no obstante, en el tiempo de cálculo se incluye la obtención de la malla 3D de la geometría facial del usuario.

Los resultados cuantitativos y cualitativos de los experimentos realizados, si bien han sido obtenidos con una cantidad de participantes reducida, permiten hipotetizar que el método resultaría prometedor para condiciones normales de captura con cámaras de bajo costo, aunque no se equipararía con el desempeño publicitado por los dispositivos comerciales existentes.

El trabajo futuro incluye la prueba del mismo con una muestra mayor de usuarios y la adición al sistema de un mecanismo de interacción basado en acciones. Gestos como el parpadeo, guiño o movimiento de las comisuras de la boca, podrían ser detectados en forma de acciones faciales para proveer un mecanismo de activación. Es posible, también, continuar hacia una implementación con mayor robustez, contemplando la variabilidad de iluminación y efectuando adaptaciones respecto a los ángulos máximos de movimiento utilizados para la proyección del puntero.

References

1. Böhme, M., Meyer, A., Martinetz, T., Barth, E.: Remote eye tracking: State of the art and directions for future development. In: Proc. of the 2006 Conference on Communication by Gaze Interaction (COGAIN), pp. 12–17 (2006)

2. Valenti, R., Sebe, N., Gevers, T.: Combining head pose and eye location information for gaze estimation. IEEE Transactions on Image Processing, 21(2):802–815 (2012)
3. Chandra, S., Sharma, G., Malhotra, S., Jha, D., Mittal, A. P.: Eye tracking based human computer interaction: Applications and their uses. In: Man and Machine Interfacing (MAMI), 2015 International Conference on, pp. 1–5. IEEE (2015)
4. Jacob, R. J.: What you look at is what you get: eye movement based interaction techniques. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 11–18. ACM (1990)
5. Mange, A. A., Choudhari, A. V., Prasad, S.: Gaze and blinking base human machine interaction system. In: Computational Intelligence and Computing Research (ICCIC), 2015 IEEE International Conference on, pp. 1–4. IEEE (2015)
6. Chauhan, V., Morris, T.: Face and feature tracking for cursor control. In: Proceedings of the Scandinavian Conference on Image Analysis, pp. 356–362 (2001)
7. Murphy-Chutorian, E., Trivedi, M. M.: Head pose estimation in computer vision: A survey. IEEE transactions on pattern analysis and machine intelligence, 31(4):607–626 (2009)
8. Betke, M., Gips, J., Fleming, P.: The camera mouse: visual tracking of body features to provide computer access for people with severe disabilities. IEEE Transactions on neural systems and Rehabilitation Engineering, 10(1):1–10. (2002)
9. Nabati, M., Behrad, A.: Camera mouse implementation using 3d head pose estimation by monocular video camera and 2d to 3d point and line correspondences. In Telecommunications (IST), 2010 5th International Symposium on, pp. 825–830. IEEE (2010)
10. Baltrušaitis, T., Robinson, P., Morency, L.-P.: OpenFace: an open source facial behavior analysis toolkit. En Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on, pp. 1–10. IEEE (2016)
11. Intel Corporation. Intel® RealSense™ SDK 2016 R2 Documentation. `https://software.intel.com/sites/landingpage/realsense/camera-sdk/v1.1/documentation/html/index.html?doc_devguide_introduction.html`. (Online; accedido el 12 de Marzo del 2018)
12. Cheung, Y. M., Peng, Q.: Eye gaze tracking with a web camera in a desktop environment. IEEE Transactions on Human-Machine Systems, 45(4), 419-430 (2015)
13. Bradski, G., Kaehler, A.: Learning OpenCV: Computer vision withthe OpenCV library, O'Reilly Media, Inc. (2008)
14. Nabati, M., Behrad, A.: 3D head pose estimation and camera mouse implementation using a monocular video camera. Signal, Image and Video Processing, 9(1):39–44 (2015)
15. Kalman, R. E.: A new approach to linear filtering and prediction problems. Journal of basic Engineering, 82(1):35–45 (1960)
16. Jackson, A. S., Bulat, A., Argyriou, V., Tzimiropoulos, G.: Large pose 3D face reconstruction from a single image via direct volumetric CNN regression. International Conference on Computer Vision (2017)
17. Anson, D., Brandon, C., Davis, A., Hill, M., Michalik, B., Sennett, C.: Swype vrs. conventional on-screen keyboards: Efficacy compared. In RESNA Annual Conference (2012)
18. Cazzato, D., Dominio, F., Manduchi, R., Castro, S. M.: Real-time gaze estimation via pupil center tracking. Paladyn, Journal of Behavioral Robotics, 9(1):6–18 (2018)