



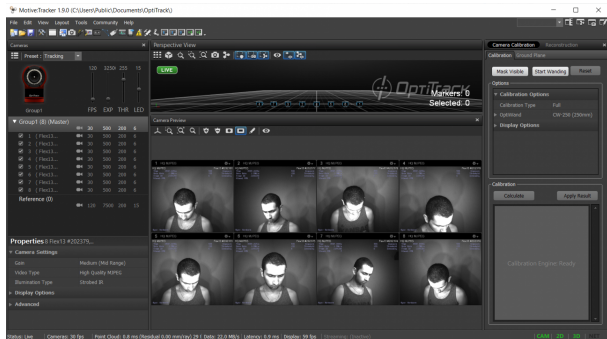
Desarrollo de un algoritmo de modelaje de plantillas faciales y etiquetado automático de marcadores de un sistema de captura de movimiento

Jose Pablo Angulo Carvajal | B50445

pab.angulo@outlook.com

1. Introducción
2. Marco Teórico
3. Desarrollo
4. Resultados
5. Conclusiones y Recomendaciones

Las máquinas y las computadoras se han vuelto ubicuas en el mundo moderno, de manera que dependemos de ellas para desarrollar nuestras vidas. La forma en la que controlamos con estas máquinas se vuelve necesariamente universal para poder facilitar la operación de estas a distintas personas, especialmente personas con discapacidades motoras. Para alcanzar esta universalidad es necesario lograr la redundancia, es decir, tener disponibles varias maneras de realizar una misma tarea.[1]



En colaboración con el PRIS-Lab, el laboratorio de reconocimiento de patrones y sistemas inteligentes de la Escuela de ingeniería eléctrica de la Universidad de Costa Rica, se realiza el proyecto en cuestión. El PRIS-Lab cuenta con equipo necesario para el rastreo de marcadores de MoCap de OptiTrack, además de un ambiente apto para la investigación y experimentación con este equipo.

El PRIS-Lab tiene varios otros proyectos tocando la temática del modelado 3D utilizando MoCap (Normalización de cuerpos tridimensionales) [2] y de telepresencia (Telepresencia robótica utilizando LeapMotion) [3] y *TERISA*, el cual consiste en el desarrollo de sistema de para la telepresencia para pacientes con esclerosis lateral amiotrófica [4] mostrando como es un ambiente ideal para desarrollar un sistema de modelado facial 3D y sistemas basados en este modelado.

El proyecto consiste en la implementación de un algoritmo apropiado para formar las plantillas faciales a partir de la posición de los marcadores faciales MoCap en las capturas; mediante la identificación de los marcadores por su posición se forma un modelo 3D de la superficie facial al armar uniones entre marcadores y se etiquetan los marcadores para mantener su identificación según la posición en los vértices 3D y acelerar el procesamiento de las capturas en términos de la etiquetación.

Además se desarrolla una interfaz gráfica para facilitar el uso del algoritmo de creación de plantillas el cual permite ingresar el número de marcadores en uso, y la posición de referencia para iniciar la etiqueta de los marcadores.

Objetivo General

- Desarrollar un algoritmo junto a una interfaz básica para la creación de plantillas faciales por capturas de marcadores faciales de MoCap los cuales sean etiquetados automáticamente por el algoritmo en cuestión.

Objetivos Específicos:

1. Realizar una investigación bibliográfica acerca de los métodos de tracking de marcadores faciales MoCap, y de algoritmos de clasificación de puntos en un espacio tridimensional.
2. Determinar el software necesario para el procesamiento y identificación de marcadores, e interfaz de uso del modelador de plantillas faciales.
3. Diseñar el algoritmo de identificación, capaz de formar las plantillas faciales etiquetando los marcadores y formando un robusto modelo facial tridimensional.
4. Validar cuantitativamente el algoritmo de creación de plantillas mediante la comparación de los vértices 3D de una captura respecto a un modelo de referencia conocido.
5. Modificar la interfaz para facilitar el uso del modelador de plantillas faciales.
6. Divulgar los resultados mediante un informe técnico, la presentación del informe, un artículo formato IEEE, un vídeo corto y la participación en el PRIS-Seminar 2021.

El MoCap, en inglés *Motion Capture*, se refiere al campo de aplicación de tecnología para grabar los datos de la posición y orientación de objetos en el espacio. En el MoCap se utilizan distintas tecnologías para grabar las posiciones y movimientos de los cuerpos: captura mecánica, captura óptica, captura magnética, captura con radiofrecuencia, captura infrarroja y demás.[5]





Estos sistemas utilizan la reflexión infrarroja para detectar y rastrear marcadores reflectivos; tienen la ventaja de que en condiciones ideales sufren mucha menor interferencia que sistemas ópticos.

En el PRIS-Lab se encuentra disponible un sistema de la empresa OptiTrack, en este caso 8 cámaras Flex 13 de rastreo óptico de marcadores con precisión sub-milimétrica y el software de interfaz gráfica y procesamiento Motive de la misma empresa.

Cuando se habla de la etiquetación de marcadores de MoCap se refiere a la asignación de títulos a marcadores específicos con el propósito de poder identificarlos y mantener una continuidad entre los puntos de referencia conforme se desarrollan los movimientos. El etiquetado de los marcadores permite formar el esqueleto de modelo 3D del cuerpo, aunque en veces se usa el esqueleto para etiquetar los marcadores en un principio. [6]

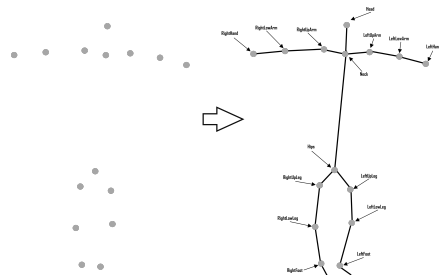


Figura: Nube de puntos a esqueleto etiquetado
[Elaboración Propia]

Un problema común que dificulta el lidiar con la etiquetación de marcadores es la obstaculización de los marcadores por objetos en la escena o las mismas partes del cuerpo del sujeto, por lo cual no se logra triangular la posición de ciertos marcadores y se dejan por fuera en partes de la captura. Por otro lado, la interferencia y/o reaparición de marcadores se vuelve un problema al tener más marcadores que etiquetas o marcadores no rastreados.

Una gran parte de las soluciones propuestas utilizan las restricciones de movimiento del cuerpo humano para crear modelos predictivos y estimar la posición de los marcadores que se pierden; tal y como se presenta al utilizar algoritmos constreñidos con un filtro de Kalman en [7] o asignando pesos a predicciones hechas por el algoritmo tomando en cuenta la rigidez de los segmentos entre articulaciones humanas en [8]. Esto no es práctico en la captura de movimientos faciales.

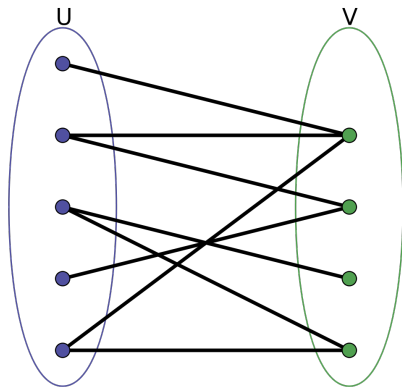


Figura: Ejemplo de Bigrafo [9]

Un grafo bipartito, también llamado bigrafo, es un tipo de modelo de representación por pareja de relaciones entre objetos en el cual los vértices (UV) se pueden separar en dos conjuntos independientes (U y V) y cada vértice en cada conjunto está conectado a un vértice en el otro conjunto. Si cada vértice en U está conectado a cada vértice en V y viceversa, se llama al grafo como grafo bipartito completo. [10]

El algoritmo húngaro, también llamado algoritmo de asociación de Munkres luego de la revisión de James Munkres en 1957, es un algoritmo de optimización combinatoria para resolver el problema de asociación con una complejidad de $O(n^4)$ y es un algoritmo muy popular para resolver la asociación de objetos con la capacidad de representarse en un grafo bipartito completo. [11]

	A2	B2	C2			A2	B2	C2
A1	30	25	10	➡	A1	30	25	10
B1	15	10	20		B1	15	10	20
C1	25	20	15		C1	25	20	15

Figura: Ejemplo de asociación por algoritmo húngaro [Elaboración Propia]

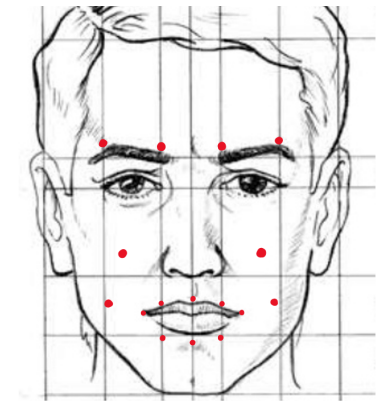


Figura: Layout/Plantilla de 14 marcadores
[Elaboración Propia]

Para la captura de datos de MoCap se sugiere utilizar una cantidad redundante de cámaras de ser posible cubriendo diferentes posibles ángulos y una escena espaciosa con poca interferencia óptica o infrarroja. En condiciones ideales el rastreo de marcadores tienen una tasa de falla mínima a nula.

Es necesario tener una plantilla preparada para colocar los marcadores, ya sea con una plantilla propia o utilizando las que se encuentran disponibles en el programa.

Se utilizan las herramientas de Tkinter en Python, ya que esta herramienta ya se encuentra integrada en las instalaciones estándar de Python por lo cual no requiere ninguna instalación extra y corre en las plataformas de Windows, Linux y Mac OS.

Se inicia seleccionando el archivo CSV con los datos de las coordenadas para rastrear. Luego se selecciona una de las imágenes de plantillas para utilizar.

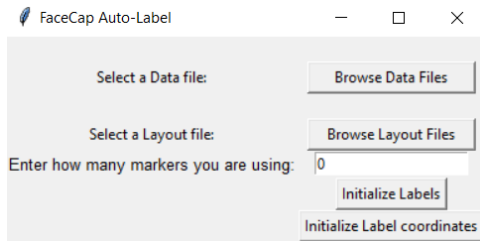


Figura: Menú principal de GUI [Elaboración Propia]

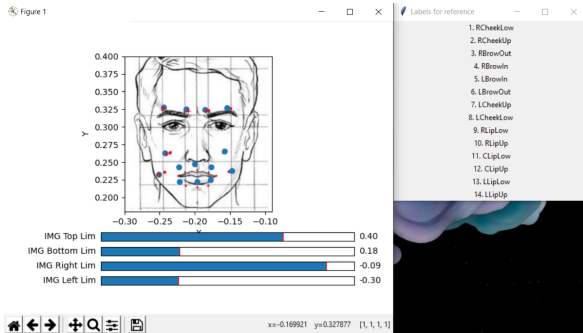


Figura: Inicialización de coordenadas [Elaboración Propia]

Luego de seleccionar los archivos base, se indica la cantidad de etiquetas a utilizar. Con la cantidad de etiquetas se inicializa los nombres de las etiquetas en una ventana que aparece al presionar *"Initialize Labels"*.

Con las etiquetas agregadas al programa, se procede a inicializar las coordenadas con sus etiquetas apropiadas. Para seleccionar los puntos se utiliza la función GInput de Matplotlib.

El procesamiento de los datos consiste en la preparación del formato de estos para ser recibidos por algoritmo de asociación, la asociación trama por trama y la unificación de etiquetas y coordenadas con los resultados de asociación en una estructura de DataFrame del paquete Pandas de Python para luego exportarse a un archivo CSV.

RCheekLowX	RCheekLowY	RCheekLowZ	RCheekUpX	RCheekUpY	...	LLipUpX	LLipUpY	LLipUpZ
-0.250965	0.23298	0.196249	-0.242958	0.262678	...	-0.176847	0.243076	0.176141
-0.250958	0.232998	0.196248	-0.242933	0.26271	...	-0.176834	0.243094	0.176147
-0.250955	0.233015	0.196237	-0.242936	0.262742	...	-0.176824	0.243117	0.176165
...
-0.245993	0.230349	0.189965	-0.238316	0.259832	...	-0.171921	0.239737	0.170635

Cuadro: Ejemplo de DataFrame resultante al completar el procesamiento [Elaboración Propia]

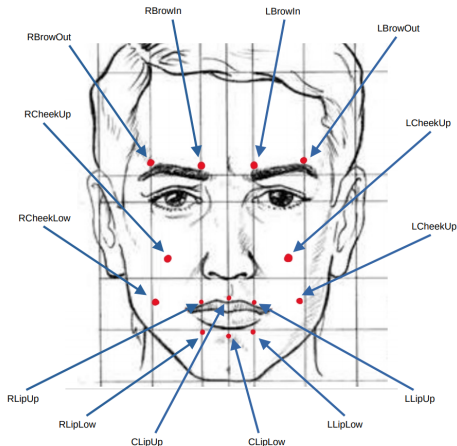


Figura: Plantilla de 14 marcadores y etiquetas [Elaboración Propia]

Se utiliza los datos del archivo “alzar cejas1.csv”, el cual corresponde a una captura en ambiente ideal del movimiento de alzar las cejas, tomada con la plantilla de 14 marcadores predeterminada del programa. Esta toma no cuenta con ninguna pérdida de marcadores, interferencia o cambios de identidad entre marcadores rastreados por lo cual es una buena referencia para comprobar la funcionalidad del algoritmo para rastrear marcadores.

Para obtener la diferencia en metros se promedia la diferencia trama a trama entre los marcadores de “alzar cejas1.csv” y los marcadores etiquetados por el algoritmo; es de notar que debido a transformaciones de Python y del programa para analizar estos resultados, en cada trama, se dan pequeñas desigualdades de coordenadas que deberían ser idénticas por redondeos. Estas desigualdades tienen una magnitud tan pequeña que se pueden omitir.

Etiqueta	Diferencia (m)	Etiqueta	Diferencia (m)	TOTAL (m)
RCheekLow	$2,78 \times 10^{-17}$	LCheekLow	$1,93 \times 10^{-17}$	
RCheekUp	$2,50 \times 10^{-17}$	RLipLow	$2,33 \times 10^{-17}$	
RBrowOut	$2,45 \times 10^{-17}$	RLipUp	$2,36 \times 10^{-17}$	
RBrowIn	$2,39 \times 10^{-17}$	CLipLow	$1,96 \times 10^{-17}$	
LBrowIn	$5,55 \times 10^{-17}$	CLipUp	$1,96 \times 10^{-17}$	
LBrowOut	$1,95 \times 10^{-17}$	LLipLow	$1,91 \times 10^{-17}$	
LCheekUp	$1,98 \times 10^{-17}$	LLipUp	$2,36 \times 10^{-17}$	
				$2,15 \times 10^{-17}$

Cuadro: Diferencia entre marcadores en “alzar cejas1.csv” y el resultado del algoritmo de asociación

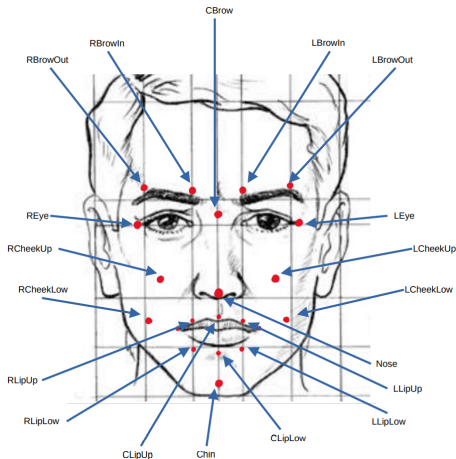


Figura: Plantilla de 19 marcadores y etiquetas [Elaboración Propia]

Se compara el desempeño del algoritmo con un set de datos deliberadamente alterado, cubriendo y revelando el marcador *LCheekLow* repetidas veces en la captura. El archivo original reconoce 22 marcadores, pero la plantilla utilizada es de 19 marcadores, esta plantilla es una de las que tiene el programa por defecto. Para mostrar una alternativa, se tiene un set con una asignación manual de las etiquetas, en las secciones donde no se reconocen suficientes marcadores, menos a 19, se procede a copiar el último valor del marcador perdido hasta que nuevas coordenadas puedan tomar su lugar.

Al comparar la diferencia en posición de cada marcador trama a trama, se obtiene la diferencia promedio de toda la captura entre las posiciones del set original, el set con asociación manual y el set asociado por el algoritmo.

	Promedio (m)
Manual vs Algoritmo en LCheekLow	1.7904x-17
Captura total Manual	0.016311
Captura total Manual sin LCheekLow	0.000000
Captura total Algoritmo	0.014450
Captura total Algoritmo sin LCheekLow	0.001964
Captura total Manual vs Algoritmo	0.001860

Cuadro: Diferencia promedio entre asignación del algoritmo y asignación manual

1. Se reconocen varios acercamientos que se han presentado en estudios pasados para resolver rastreo y etiquetación de marcadores de MoCap, como utilizar restricciones del movimiento del cuerpo humano y machine learning para lidiar con la flexibilidad del MoCap facial.
2. Se utiliza Python por su robustez como lenguaje de programación, la rapidez de implementar distintas versiones del código y su gran documentación en línea, además de contar con paquetes como Tkinter, Pandas, Matplotlib, NumPy y Hungarian algorithm.
3. Se identifica el algoritmo húngaro y la representación de datos como grafos bipartitos como una alternativa con mucho potencial para resolver el problema de asignación de etiquetas.
4. Se prueba el funcionamiento del algoritmo al comparar sus resultados con capturas ideales donde no se pierde ningún marcador, y capturas deliberadamente modificadas, de manera de que el marcador perdido se conoce y se realiza una etiquetación manual. Además se reconocen las limitaciones de la implementación realizada en este proyecto.
5. Se crea una interfaz gráfica para preparar los datos, asignar una plantilla y etiquetas a una posición inicial de referencia para el algoritmo.
6. Se preparó un informe técnico y una presentación para la promoción del proyecto, adicionalmente, se tiene planeado un artículo en formato IEEE y la participación en el PRIS-Seminar 2021.

- El algoritmo está limitado a la comparación entre la trama a asociar y una trama de referencia, por esta razón instancias donde se pierde uno o más marcadores por varias tramas consecutivas pueden sufrir en la disminución de la eficacia del algoritmo; un acercamiento predictivo o la utilización de grafos multipartitos tomando en cuentas múltiples tramas pasadas pueden aliviar este efecto.
- El algoritmo utilizado tiene ciertas limitaciones intencionales para mejorar su velocidad de desempeño, como lo es la replicación de la última trama completa al reconocer menos marcadores que el mínimo para aplicar el algoritmo; se pueden realizar procesos alternativos en estos casos para limitar la cantidad de marcadores copiados de trama a trama.
- Si bien Tkinter tiene ventajas al ser liviano y estar integrado en las instalaciones de Python en varios sistemas operativos, otras plataformas de desarrollo de interfaces gráficas pueden permitir la creación de una interfaz más moderna y completa.



Laboratorio de investigación en reconocimiento de patrones y sistemas inteligentes

Escuela de Ingeniería Eléctrica, Ciudad de la Investigación
Universidad de Costa Rica

Tel: (+506) 2511 2603 / (+506) 2511 2642

Email: pris@ucr.ac.cr

facebook: [/PRISLaboratory](https://www.facebook.com/PRISLaboratory)

youtube: [/PRISLaboratory](https://www.youtube.com/PRISLaboratory)

El algoritmo se puede ejemplificar mediante una matriz de costo $n \times n$, donde cada conjunto del grafo bipartito se representa en las filas y columnas.

En este caso se tiene $V_1 = (A_1, B_1, C_1)$ y $V_2 = (A_2, B_2, C_2)$ y una matriz G donde se representa el costo de asociar cada vértice de V_1 con cada vértice de V_2 y viceversa.

	A2	B2	C2
A1	30	25	10
B1	15	10	20
C1	25	20	15

Figura: Matriz de ejemplo

30	25	10
15	10	20
25	20	15

→

20	15	0
5	0	10
10	5	0

Figura: Paso 1: Reducir Filas

Paso 1: Se inicia identificando el mínimo valor de cada fila y restándolo de manera que $G'_{i \times j} = G_{i \times j} - \min G_i$.

Paso 2: Se identifica el mínimo valor de cada columna y se resta de manera que $G''_{i \times j} = G'_{i \times j} - \min G_j$.

20	15	0
5	0	10
10	5	0

→

15	15	0
0	0	10
5	5	0

Figura: Paso 2: Reducir Columnas

15	15	0
0	0	10
5	5	0

Figura: Paso 3: Etiquetación y revisión de asociación

Paso 3: Se etiquetan las filas y columnas de manera que se cubran todos los ceros utilizando la menor cantidad de etiquetas posibles. Si se tiene n filas y columnas etiquetadas, existe una asociación de costo mínimo; si se tiene menos de n filas y columnas etiquetadas se continúa el algoritmo.

Paso 4: De no tener una asociación de costo mínimo luego del paso 3, se identifica el valor mínimo no etiquetado y se resta a todas las filas no etiquetadas y se suma a todas las columnas etiquetadas de manera que no quedan valores negativos.

15	15	0
0	0	10
5	5	0

→

10	10	0
0	0	15*
0	0	0

Figura: Paso 4: Crear nuevos ceros

10	10	0
0	0	15*
0	0	0

Figura: Paso 5: Re-etiquetación y revisión de asociaciones

Paso 5: Se repite el paso 3, de contar con n filas y columnas etiquetadas se consideran asociaciones de costo mínimo posibles; de no contar con suficientes filas y columnas etiquetadas se repite el paso 4.

Paso 6: Se escoge una de las asociaciones posibles, el algoritmo asegura que todas las posibles asociaciones tienen un costo total equivalente. El costo total se obtiene sumando los costos de cada asociación individual seleccionada. En este ejemplo se tiene un costo total de 45.

	A2	B2	C2
A1	10	10	0
B1	0	0	15*
C1	0	0	0

Figura: Paso 6: Selección de asociación

	A2	B2	C2
A1	30	25	10
B1	15	10	20
C1	25	20	15

Figura: Paso 6: Costo de asociación total



Maura Casadio, Rajiv Ranganathan, and Ferdinando A. Mussa-Ivaldi, "The body-machine interface: A new perspective on an old theme," *Journal of motor behavior*, vol. 44, 2012.



N. A. Jiménez, "Desarrollo de un algoritmo para la normalización de un cuerpo tridimensional utilizando archivos bvh," *Technical report*, 2019.



J. P. Ávila López, "Desarrollo de un sistema de telepresencia robótica integrando osvr, leap motion y nao para personas con motora reducida," *Technical report*, 2017.



Melissa Fallas Sanabria, "Esclerosis lateral amiotrófica," *Revista Médica*



[Online]. Available: <http://web.mit.edu/comm-forum/legacy/papers/furniss.html#5>

J. Meyer, M. Kuderer, J. Müller, and W. Burgard, "Online marker labeling for fully automatic skeleton tracking in optical motion capture," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5652–5657.



V. Joukov, J. F. S. Lin, K. Westermann, and D. Kulić, "Real-time unlabeled marker pose estimation via constrained extended kalman filter," in *Proceedings of the 2018 International Symposium on Experimental Robotics*, J. Xiao, T. Kröger, and O. Khatib, Eds. Cham: Springer International Publishing, 2020, pp.



motion capture with missing marker fillings," in *2014 22nd International Conference on Pattern Recognition*, 2014, pp. 2507–2512.



MistWiz. (2007) Example of bipartite graph. [Online]. Available: <https://commons.wikimedia.org/w/index.php?curid=1814874>



M. L. Zepeda-Mendoza and O. Resendis-Antonio, *Bipartite Graph*. New York, NY: Springer New York, 2013, pp. 147–148. [Online]. Available: https://doi.org/10.1007/978-1-4419-9863-7_1370

J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied*