



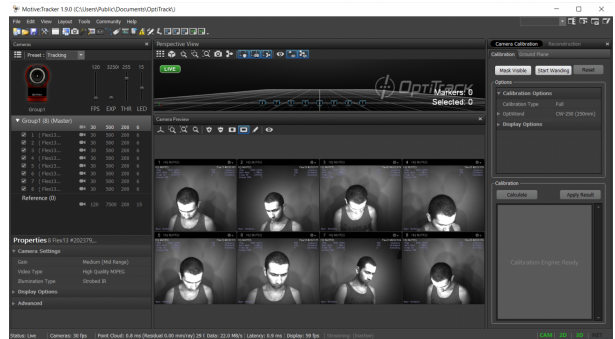
Desarrollo de un algoritmo de modelaje de plantillas faciales y etiquetado automático de marcadores de un sistema de captura de movimiento

Jose Pablo Angulo Carvajal | B50445

pab.angulo@outlook.com

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

- Vida moderna con computadores y máquinas
- Redundancia y universalidad [1]



Desarrollado en colaboración con el PRIS-Lab:

- Herramientas: Cámaras OptiTrack, ambiente de investigación
- Experiencia pasada: Normalización de cuerpos 3D [2], Telepresencia robótica con LeapMotion [3], *TERISA* (para pacientes de esclerosis lateral amiotrófica [4])

- Implementación de algoritmo apropiado para identificación y etiquetación de marcadores por posición, acelerando un proceso usualmente manual.
- Desarrollar una interfaz gráfica facilitando uso de algoritmo y creación de nuevas plantillas.

Objetivo General

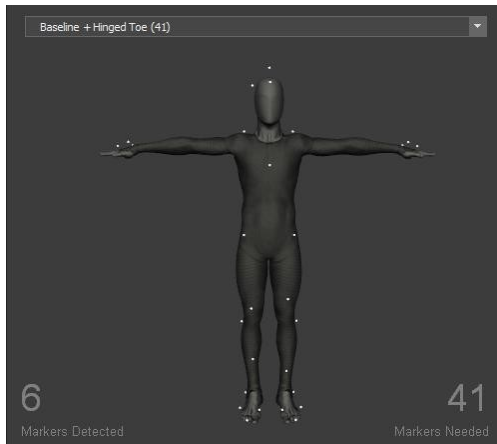
- Desarrollar un algoritmo junto a una interfaz básica para etiquetación automática de marcadores MoCap faciales.

Objetivos Específicos:

1. Realizar una investigación bibliográfica acerca de los métodos de rastreo de marcadores faciales MoCap.
2. Determinar el software necesario para el procesamiento y creación de interfaz.
3. Diseñar el algoritmo de identificación y etiquetación de marcadores.
4. Validar cuantitativamente el algoritmo al compararlo 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 video corto y la participación en el PRIS-Seminar 2021.

El MoCap (en inglés *Motion Capture*):
posición y orientación de objetos en
el espacio.

Se utilizan distintas tecnologías:
captura mecánica, captura óptica,
captura magnética, captura con
radiofrecuencia, captura infrarroja y
demás.[5]





- Ventajas de MoCap Infrarrojo
- Equipo: 8 Cámaras Flex 13 y software Motive

-

g

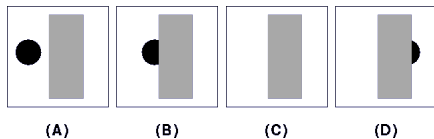
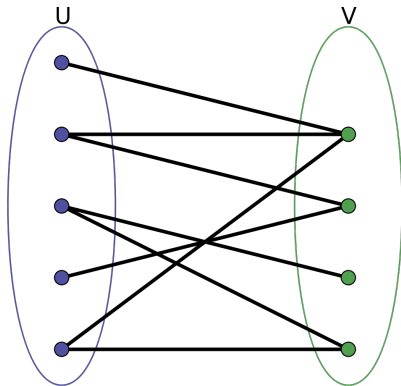


Figura: Ejemplo de oclusión [7]

- Oclusión de marcadores
- Marcadores fantasma e interferencia
- Soluciones existentes: Filtros de Kalman constreñidos [8], Peso de predicción por rigidez de segmentos [9]
- El problema facial



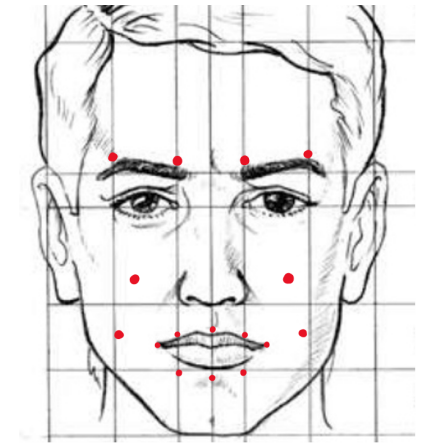
- Grafo bipartito o bigrafo
- Grafo bipartito completo[11]

Figura: Ejemplo de Bigrafo [10]

- Algoritmo de asociación húngaro o algoritmo de asignación de Munkres.
- Optimización combinatoria.
- Complejidad $O(n^4)$. [12]

	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]



- Preparación: Plantilla lista, opcionalmente utilizar las que tiene el programa por defecto
- Toma de datos: Escena y sujeto
- Minimizar interferencia y abundancia de cámaras

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

- Tkinter: ligero, compatibilidad
- Selección de datos y selección de plantilla

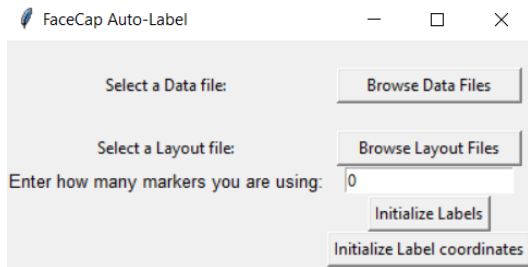
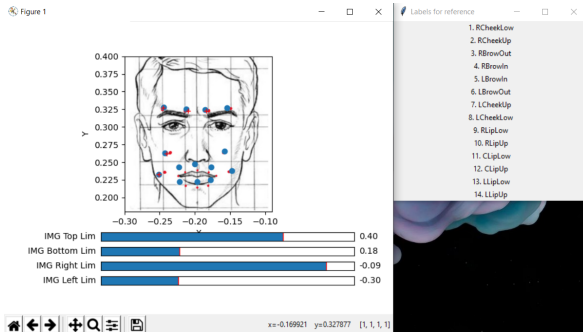


Figura: Menú principal de GUI [Elaboración Propia]



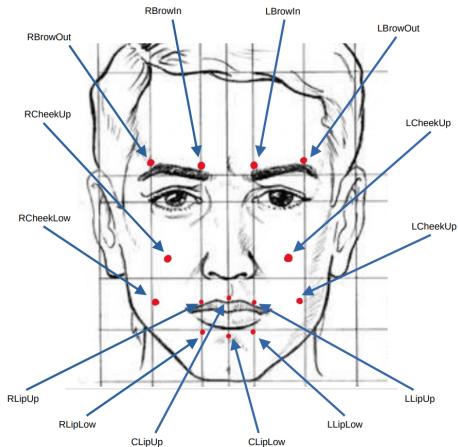
- Indicar cantidad de etiquetas a usar
- Ingresar las etiquetas en orden deseado
- Inicializar las etiquetas con coordenadas iniciales, mediante Glinput

Figura: Inicialización de coordenadas [Elaboración Propia]

- Preparación de formato de datos
- Separación trama a trama, y de cada marcador en la trama
- Asignación respecto a última trama
- Unificación de asignación
- Exportación de resultado

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]



- Se utiliza los datos del archivo “alzar cejas1.csv”
- Plantilla de 14 marcadores
- Captura ideal

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

Para obtener la diferencia en metros se promedia la diferencia en posiciones trama a trama entre los marcadores de “alzar cejas1.csv” y los marcadores etiquetados por el algoritmo

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

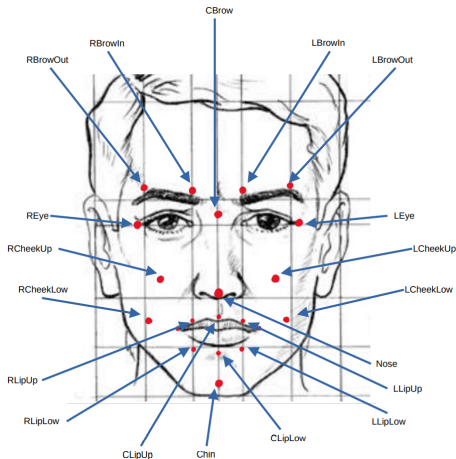


Figura: Plantilla de 19 marcadores y etiquetas [Elaboración Propia]

- Se compara el desempeño del algoritmo con un set de datos deliberadamente alterado, cubriendo y revelando el marcador *LCheekLow* repetidas veces en la captura
- Para mostrar una alternativa, se tiene un set con una asignación manual de las etiquetas
- Momentos de oclusión

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

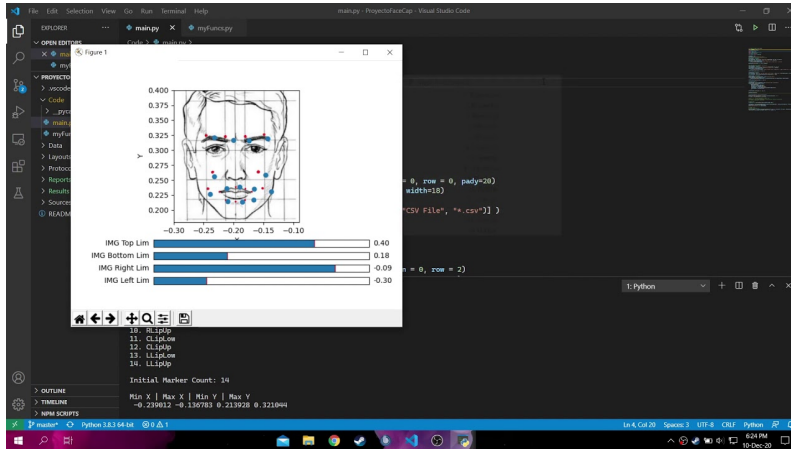


Figura: Demostración de utilización del programa

1. Se reconocen varios acercamientos pasados para resolver rastreo y etiquetación como utilizar restricciones del movimiento del cuerpo humano y machine learning.
2. Se utiliza Python por su robustez como lenguaje de programación, la rapidez de implementar distintas versiones y su gran documentación en línea.
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, y capturas deliberadamente modificadas.
5. Se reconocen las limitaciones de la implementación realizada en este proyecto.
6. Se crea una interfaz gráfica para facilitar el uso del algoritmo.
7. Se preparó un informe técnico y una presentación, adicionalmente, se tiene planeado un artículo en formato IEEE y la participación en el PRIS-Seminar 2021.

- Pérdida de marcadores en tramas consecutivas y las alternativas de grafos multipartitos o modelos predictivos.
 - Limitación al lidiar con tramas incompletas y como solventar.
 - Desventajas de Tkinter y rediseño de GUI con otras herramientas



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 de Costa Rica y Centroamérica*, vol. 67(591), pp. 89–92, 2010.
-  M. Furniss. Motion capture. MIT Communications Forum. [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.
-  Jonathan A. Marshall and Richard Alley. A self-organizing neural network that learns to detect and represent visual depth from occlusion events. The University of North Carolina. [Online]. Available: <http://cogprints.org/442/1/node1.html>

- 
- 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. 762–771.
- 
- X. Deng, S. Xia, W. Wang, Z. Wang, L. Chang, and H. Wang, "Automatic gait 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 Mathematics*, vol. 5, no. 1, pp. 32–38, 1957. [Online]. Available: <http://www.jstor.org/stable/2098689>