



## ViVIM, Visió per computador per Vídeo Immersiu Multi-plataforma

### LL3.2.1: Programari de reconeixement d'imatge i documentació

*This deliverable reports on the first functional version for each of the listed tools / components. Future versions of this deliverable will provide a comprehensive review of the evolution of such tools, along with the different released versions for each of them.*

Project receiving funding from the Agència per la Competitivitat de l'Empresa de la Generalitat de Catalunya, ACCIÓ within the Comunitat RIS3CAT Media reference number **COMRDI18-1-0008**

**Date:** 31/12/2021  
**Version:** 1.0



## Document Information

<b>Project Number</b>	COMRDI18-1-0008	<b>Acronym</b>	ViViM
<b>Full title</b>	Visió per computador per Vídeo Immersiu Multi-plataforma		
<b>Project coordinator</b>	I2CAT Foundation		
<b>Agency's Officer</b>	Lluís Maria Tortras (ACCIÓ)		

<b>Deliverable</b>	<b>LL3.2.1</b>	<b>Title</b>	<b>Programari de reconeixement d'imatge i documentació</b>
<b>Work Package</b>	<b>PT3</b>	<b>Title</b>	Eines de Producció Immersiva

<b>Delivery</b>	<b>Data</b>	31/12/2021 (Month 15+2)	<b>Version</b>	V1.0
<b>Nature</b>	Prototype <input type="checkbox"/> Report <input type="checkbox"/> Dissemination <input type="checkbox"/> Other <input checked="" type="checkbox"/>			
<b>Dissemination Level</b>	Public <input type="checkbox"/> Consortium <input checked="" type="checkbox"/>			

<b>Responsible author/s</b>	Bogdan Raducanu, Edgar Gracia Llarosa, Coen Antens, Meri Triviño	<b>Email</b>	bogdan@cvc.uab.es
<b>Partner</b>	CVC, Visyon, i2CAT]		

<b>Abstract</b>	L'objectiu d'aquest document és la descripció tècnica de les eines de detecció i seguiment d'objectes desenvolupades fins a aquest punt del projecte i la seva relació amb el requisits tècnics especificats al projecte. Inclou una descripció tant a nivell científic, com a nivell de instal·lació i usuari final i resultats esperats.
-----------------	--

## Table of contents

<b>1. Introducció.....</b>	<b>5</b>
1.1. Objectiu d'aquest lliurable .....	5
1.2. Àmbit d'aquest lliurable.....	5
1.3. L'estatus d'aquest lliurable .....	5
1.4. Relació amb altres Activitats de VIVIM .....	5
<b>2. Introducció al Programari.....</b>	<b>6</b>
2.1. Visió i Objectius.....	6
2.2. Funcionalitats Fulla de Ruta.....	7
2.2.1. Detecció i seguiment persones basades en parts .....	7
2.2.2. Esqueletonització 3D .....	8
2.3. Llistat de requeriments.....	9
2.4. Usabilitat en el marc dels pilots de VIVIM .....	11
2.5. Arquitectura .....	11
<b>3. Detecció i seguiment de persones basades en parts .....</b>	<b>12</b>
3.1. DensePose.....	12
3.2. Refinament de silueta .....	13
3.2.1. Aproximació de la segmentació basada en parts amb contorns .....	13
3.2.2. Representació simplificada del contorn .....	14
3.2.3. Suavitització de polígons amb corbes de Bézier .....	14
3.2.4. Suavitització temporal .....	15
3.2.5. Avaluació.....	16
3.2.6. Arquitectura plug-in de Natron .....	16
3.3. Manual de d'instal·lació i configuració.....	17
3.3.1. Instal·lació .....	17
3.3.2. Funcionament .....	19
3.4. Requisits de software/hardware .....	22
3.5. Prestacions .....	23
3.6. Estatut d'integració.....	23
3.7. Model de distribució .....	24
<b>4. Esqueletonització 3D.....</b>	<b>25</b>
4.1. Introducció .....	25
4.2. Estimació de la postura en 2D .....	25
4.3. Estimació de la postura en 3D .....	26
4.4. Esquelet 'bvH' .....	28
4.5. Implementació.....	28

4.6.	Requisits de software/hardware .....	29
4.7.	Prestacions .....	29
4.8.	Exportació a un plug-in de Natron .....	29
4.9.	Manual de d'instal·lació i configuració .....	30
4.9.1.	Instal·lació .....	30
4.9.2.	Tracking de videos 360 .....	30
4.9.3.	Extracció de l'esquelet .....	30

## 1. Introducció

### 1.1. Objectiu d'aquest lliurable

Aquest lliurable persegueix els següents objectius:

- Descriure les capacitats del software desenvolupat per eines disponibles en el camp de la post-producció audiovisual.
- Especificar el requeriments complerts inicialment definits al projecte a nivell de usabilitat, capacitats, computació,..etc.
- Detallar els fonaments bàsics de la tecnologia.
- Reportar requeriments de rendiment i computació.
- Proveir d'un manual d'instal·lació i ús del software desenvolupat.

### 1.2. Àmbit d'aquest lliurable

Aquest lliurable llista i descriu els software de detecció i seguiment de persones basades en parts i esqueletonització 3D al projecte Vivim, així com la seva integració en el plugin de post-producció per entorns 360 en relació i grau de compliment pel que respecta als requeriments identificats i la seva idoneïtat pel que respecta als escenaris i casos d'ús seleccionats.

### 1.3. L'estatus d'aquest lliurable

Aquesta és la primera versió del lliurable LL3.2, planificada per al mes M15. Una versió posterior està planificada per al M33.

### 1.4. Relació amb altres Activitats de VIVIM

Es tracta d'un lliurable de l'Activitat 3, i està molt relacionat amb el LL2.2 [1] que proporciona una descripció detallada dels escenaris i casos d'ús del projecte, així com dels seus requeriments, i dels requeriments de les eines de producció, distribució i consum. Així mateix, està relacionat amb el LL2.3 [2], que detalla el pla de producció i avaluació del projecte.

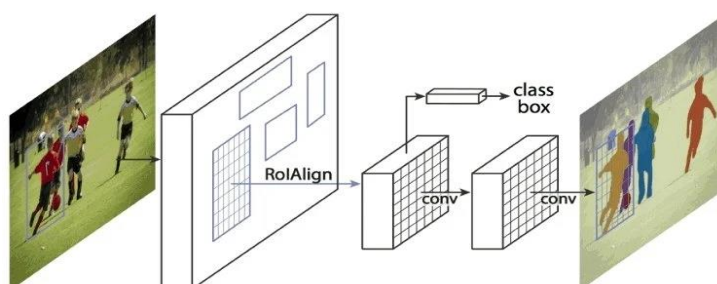
Així mateix, encara que es tracti d'un lliurable de l'Activitat 3, està molt relacionat amb l'Activitat 4, ja que reporta sobre components software a ser utilitzats en demostradors i accions pilot del projecte.

## 2. Introducció al Programari

### 2.1. Visió i Objectius

La detecció d'objectes ha despertat molt d'interès en els darrers anys tant en el món acadèmic com en la indústria, per les seves àmplies aplicacions en diversos camps: multimèdia, videovigilància, robòtica, cotxes autònoms, imatge mèdica, informàtica corporal, etc., i també en alguns avenços tecnològics. Entre molts factors que van contribuir a l'aparició d'aquest camp podem destacar els següents: el desenvolupament de tècniques avançades d'aprenentatge profund, que bàsicament van substituir quasi tots els enfocaments clàssics d'aprenentatge automàtic; la creació de grans conjunts de dades d'imatges; l'augment de la potència computacional a causa de les GPU més rendibles; i finalment, però no menys important, la distribució de codi que no només garanteix la reproductibilitat dels experiments, sinó que també accelera el desenvolupament de noves aplicacions.

La detecció d'objectes és una tecnologia de computació relacionada amb la visió per ordinador i el processament d'imatges que s'ocupa de detectar instàncies d'objectes semàntics d'una classe determinada: gossos, gats, persones, cotxes, avions, flors, etc. La detecció d'objectes es podria definir com el procés per a identificar correctament la ubicació i la categoria dels objectes presents a la imatge.



Arquitectura de Mask R-CNN per la detecció i segmentació semàntica de les persones

Mask R-CNN<sup>1</sup> (a la imatge de dalt) és una arquitectura coneguda per dur a terme la segmentació semàntica (a nivell de pixel) i d'instàncies (és a dir, distingir entre diferents objectes de la mateixa categoria). Anticipa tant les ubicacions del quadre delimitador dels diferents objectes de la imatge com una màscara que segmenta els objectes semànticament. Primer extreu mapes de característiques d'una imatge mitjançant una xarxa neuronal convolucional (CNN). Una xarxa de propostes de regió (RPN) utilitza aquests mapes de característiques per obtenir candidats de quadre delimitador per a la presència d'entitats. Els candidats al quadre delimitador seleccionen una regió del mapa de característiques. Atès que els candidats al quadre delimitador poden ser de diferents mides, la capa RoIAlign s'utilitza per reduir la mida de les característiques extretes perquè es tornin de mida uniforme. Ara, les

<sup>1</sup> K. He, G. Gkioxari, P. Dollar and R. Girshick. Mask R-CNN. *Proc. of ICCV*, pp. 2961-2969, 2017

característiques extrems es passen a les branques paral·leles de les CNN per a la predicció definitiva dels quadres delimitadors i les màscares de segmentació.

La mateixa arquitectura de Mask R-CNN es pot estendre simplement per a l'estimació de la postura humana. La ubicació d'un punt clau es defineix com una màscara binària (un vector *one-hot*) i s'adapta Mask R-CNN per predir  $K$  màscares, una per a cadascun dels  $K$  tipus de punts clau (p. ex., espatlla esquerra, colze dret, etc.). Combinant la informació d'ubicació de la persona i el seu conjunt de punts clau, podem obtenir l'esquelet de la postura humana per a cada persona de la imatge.



Exemple de detecció d'objectes múltiples mitjançant segmentació d'instàncies (esquerra) i l'esquelet de la postura humana (dreta).

Aquest tipus d'eina es pot fer servir per diferents tasques de post-producció que tinguin a veure les siluetes de persones. Un exemple d'ús seria per la rotoscòpia segmentada d'un personatge gravat en 360°. En experiències amb hibridació de vídeo 360 i entorn 3D, fent ús de la rotoscòpia segmentada, l'artista pot modificar el personatge a nivell artístic per integrar-lo amb l'estil gràfic de la peça en qüestió. Sense aquesta eina, l'artista hauria de fer aquesta tasca de forma manual, i augmentaria el volum d'hores de dedicació per portar a terme aquest procés.

## 2.2. Funcionalitats Fulla de Ruta

En el context del projecte VIVIM, ens hem centrat en una categoria particular d'objectes: les persones. Amb aquesta finalitat, hem desenvolupat dos components software que faciliten les tasques de manipulació en la *post-producció*, és a dir: (i) detecció i seguiment de persones basades en parts i (ii) esqueletonització 3D.

### 2.2.1. Detecció i seguiment persones basades en parts

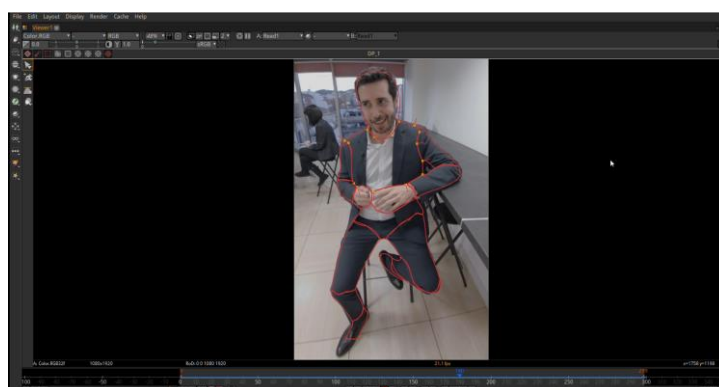
El primer component, dedicat a la detecció i seguiment de persones, va adoptar un enfocament basat en la segmentació d'instàncies que és capaç de classificar diferents parts

del cos. Més concretament, el nostre component es basa en el mètode DensePose<sup>2</sup>. El component rep com a entrada una imatge amb una o diverses persones i torna com a sortida una imatge on estan segmentades les diferents parts del cos. A partir d'allà la informació obtinguda es pot utilitzar per a post-producció o bé de manera individual (de cada part segmentada) o de manera global (de la persona sencera), tal com es mostra a la figura de sota.



Exemple de segmentació d'instància basada en parts: imatge original (esquerra), parts segmentades (sortida del mètode DensePose) (centre) i la rotoscòpia corresponent (dreta).

COMPONENT	DESCRIPCIÓ	FUNCIONALITAT(s)	DATA LLIURAMENT
DensePose	Detecció i seguiment de parts del cos	Múltiples resolucions	Sep 2021
		Detecció i seguiment	Sep 2021



Exemple de funcionament del plugin DensePose dins la plataforma Natron.

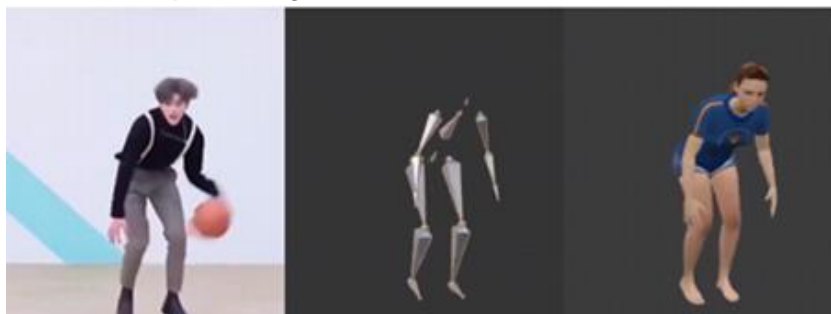
## 2.2.2. Esqueletonització 3D

El segon component que hem inclòs en aquest lliurament es basa en el mètode 'video2bvh', que estima l'esquelet 3D d'una persona. Aquest procés es basa en tres passos: (i) estimador

<sup>2</sup> R. A. Güller, N. Neverova, and I. Kokkinos. Densepose: Dense human pose estimation in the wild. *Proc. of CVPR*, pp. 7297-7306, 2018



de postura 2D basat en OpenPose<sup>3</sup>; (ii) un estimador de postures 3D, VideoPose3D<sup>4</sup>; i (iii) el mòdul que converteix la postura 3D en angles d'articulacions. El component rep com a entrada una imatge amb una o diverses persones i torna com a sortida una imatge amb el esquelet 3D del cos. A partir d'allà la informació obtinguda es pot utilitzar per a post-producció, per a transferir el moviment a un personatge o avatar diferent, tal com es mostra a la figura de sota.



Transferència de la postura 3D d'una imatge d'entrada (esquerra) a un avatar (dreta), en funció de l'esquelet 3D estimat (centre)

COMPONENT	DESCRIPCIÓ	FUNCIONALITAT(S)	DATA LLIURAMENT
video2bvh	Esqueletonització 3D	Múltiples resolucions	Sep 2021
		Extracció de l'esquelet 3D	Sep 2021

## 2.3. Llistat de requeriments

Llistat dels requeriments definits al projecte ViViM i la relació amb les dues components desenvolupades per reconeixement i seguiment de persones.

Id.	Component	Descripció	Relacionat	Detecció i Seguiment	Esquelet 3D
FR01	Plataforma	La plataforma ViViM ha d'implementar un sistema de producció i consum audiovisual centrat en una nova forma de narració audiovisual que combini formats tradicionals, omnidireccionals (360°) i de Realitat Virtual 3D, sent compatible amb els llenguatges audiovisuals clàssics (pla/contrapla, càmera lenta, etc)	Proposta (Objectiu General) + Ampliació Scope	Com eina de suport pel requeriment general	Com eina de suport pel requeriment general
FR04	Plataforma	Es requereix un nou llenguatge cinematogràfic (sistema de producció, modalitats d'interacció, i model de metadades) que permeti explotar les possibilitats de la varietat de formats audiovisuals, escenaris i dispositius de	Proposta (Objectiu General + Objectiu 1)	Permet enriquir l'experiència a noves formes de interacció amb el contingut en un	Permet enriquir l'experiència a noves formes de interacció amb el contingut en un entorn multi-

<sup>3</sup> Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *Proc. of CVPR*, pp. 7291-7299, 2017

<sup>4</sup> D. Pavlo, C. Feichtenhofer, D. Grangier, and M. Auli. 3D Human Pose Estimation in Video with Temporal Convolutions and Semi-supervised Training. *Proc. of CVPR*, pp. 7753-7762, 2019

Id.	Component	Descripció	Relacionat	Detecció i Seguiment	Esquelet 3D
		consum contemplats al projecte		entorn multi-plataforma	plataforma
FR05	Plataforma	La plataforma ViViM ha de suportar la definició de portal dinàmics i interactius que possibilitin la inserció de seqüències audiovisual o elements multimèdia addicionals, seguint una narrativa especificada en procés de producció, i que permeti experiències més interactives i personalitzades	Proposta (Objectiu General + Objectiu 1)	Com eina de suport pel requeriment general	Com eina de suport pel requeriment general
FR06	Plataforma	La plataforma ViViM ha de permetre la producció i distribució de continguts en diferit	Proposta (Objectiu General + Objectius 1 i 4)	Es possible la reproducció en diferit (offline) per defecte	Es possible la reproducció en diferit (offline) per defecte
FR07	Plataforma	La plataforma ViViM ha de permetre la producció i distribució de continguts en viu	Proposta (Objectiu General + Objectius 1 i 4)	ND	ND
FR09	Plataforma	La plataforma ViViM ha de possibilitar experiències multimèdia personalitzades, en funció de les possibilitats de selecció de continguts i dispositius de consum, així com modalitats d'interacció	Proposta	Com eina de suport pel requeriment general	Com eina de suport pel requeriment general
FR15	Producció	La producció de continguts s'ha d'adequar als diferents tipus de dispositius de consum contemplats al projecte, evitant problemes de compatibilitat	Proposta (Objectiu 2)	Inserció vídeo convencional en entorns RV	Inserció vídeo convencional en entorns RV
FR19	Producció	Els continguts a produir han de permetre unes narratives interactives adaptades als entorns multi-format i multi-dispositiu contemplats al projecte	Proposta (Derivat dels Objectius 2 i 3)	Inserció vídeo convencional en entorns RV	Inserció vídeo convencional en entorns RV
FR21	Post-Producció	La plataforma ha d'incloure un conjunt d'eines de producció (e.g. "plugins" compatibles amb eines existents, com Adobe Premier Pro) que permeten l'edició i post-producció d'experiències audiovisuals multi-plataforma.	Proposta (Objectiu 2)	Plugin NATRON	Component Software
FR24	Producció	Les eines de producció han de proporcionar mecanismes de seguiment (tracking) de persones i d'objectes en vídeos omnidireccionals amb tècniques de visió per computador.	Proposta (Objectiu 2)	SI	SI
FR25	Producció	Els mecanismes de tracking deuen possibilitar la inserció dinàmica d'elements audiovisual addicionals, així com nous mecanismes d'interacció	Proposta (Objectiu 2)	SI	SI

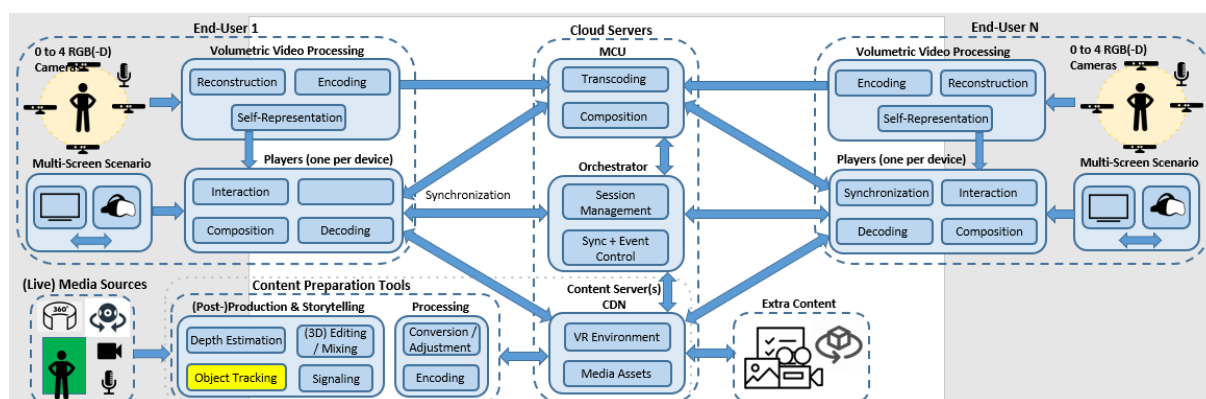
## 2.4. Usabilitat en el marc dels pilots de ViViM

Per una banda, els pilots ViViM inclouen entorns immersius realistes i interactius, que poden integrar diferents tipologies de continguts. En aquest context, l'eina DensePose s'utilitzarà per a la (post-)producció de continguts de cara als pilots finals del projecte.

Per altra banda, els pilots ViViM inclouen entorns multiusuari integrant tecnologia d'holoportació / holoconferència. En aquests entorns, les representacions dels usuaris basades en segmentació de parts o esquelet 3D poden ampliar i millorar les opcions d'ús de la plataforma ViViM. Però per aconseguir-ho, el primer pas el representa l'optimització del programari per a temps real.

La llista d'accions pilots i el seu calendari es proporciona al lliurable LL2.3 del projecte, a ser actualitzat després de l'execució del pilot 1.B. Aquesta llista inclourà una acció pilot específica per avaluar objectiva i subjectivament els beneficis aportat pel software presentat a aquest lliurable.

## 2.5. Arquitectura

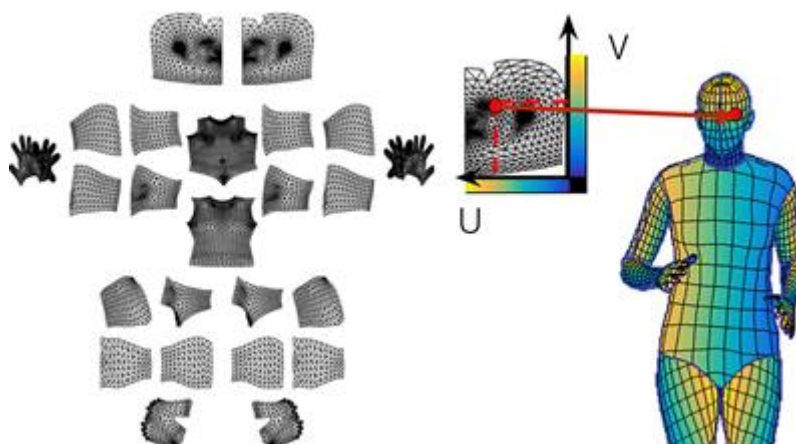


El programari desenvolupat per la detecció i seguiment d'objectes s'integra mitjançant la instal·lació de un *plugin* a la plataforma de post-producció de contingut audiovisual 360. En aquest cas, la plataforma seleccionada per facilitar de integració és Natron. Una eina open-source que igualment permet la seva integració fluida dins del fluxos de treball en plataformes d'edició comercial com pot ser Adobe Premiere, Final Cut o Nuke. Aquest programari rep el flux de un vídeo i genera un flux amb la segmentació de diferents parts del cos. El seguiment s'aconsegueix detectant els objectes a cada imatge. Aquest flux amb la detecció de parts del cos es rebut per altres *plugins* dins de la plataforma tal como inserció d'infografia.

### 3. Detecció i seguiment de persones basades en parts

#### 3.1. DensePose

La tècnica que s'ha desenvolupar es basa en el mètode DensePose, que aborda la tasca d'estimació densa de postures humanes mitjançant models entrenats discriminativament. L'estratègia seguida consisteix en establir una densa correspondència entre una imatge RGB i una representació paramètrica basada en la superfície del cos humà. Així, per a cada píxel, el mètode determina: (1) a quina superfície pertany i (2) on correspon dins la parametrització 2D d'aquesta part. Les parts que es podrien identificar són: cap, tors, braços inferiors/superiors, cames inferiors/superiors, mans i peus. Aquest enfocament es representa a la figura 1.



Particionament i parametrització UV de la superfície corporal.

DensePose és una combinació d'arquitectures existents de DenseReg<sup>5</sup> i Mask-RCNN. En un primer pas, un píxel es classifica com a pertanyent al fons o a una de les parts de la superfície de un cos humà. En un segon pas, un sistema de regressió indica les coordenades exactes del píxel dins de la part. Intuïtivament, podríem dir que el primer pas és una aproximació tosca d'on pertany el píxel, mentre que el segon l'alinea a la seva posició exacta mitjançant una correcció a petita escala.

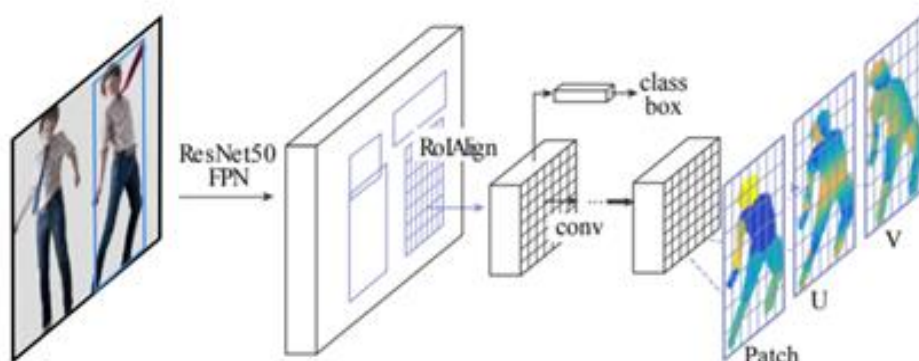
Des del punt de vista de l'arquitectura, DensePose consisteix en un mòdul que implica la construcció de característiques a partir d'una Feature Pyramid Network<sup>6</sup> i una agrupació ROI-Align, que és important per a tasques que requereixen precisió espacial. Com es mostra a la figura 2, a més de l'agrupació de ROI s'introdueix una xarxa totalment convolucional (FCN) que es dedica completament a aquestes dues tasques, generant un cap de classificació i un cap de regressió que proporcionen l'assignació de parts i les prediccions de coordenades de les parts. Com a *backbone* de DensePose es pot utilitzar Resnet-50 o Resnet 101<sup>7</sup>. La FCN

<sup>5</sup> R. A. Güler, G. Trigeorgis, E. Antonakos, P. Snape, S. Zafeiriou, and I. Kokkinos. Densereg: Fully convolutional dense shape regression in-the-wild. In *Proc. of CVPR*, pp. 6799-6808, 2017

<sup>6</sup> T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature Pyramid Networks for Object Detection. *Proc. of CVPR*, pp. 2117-2125, 2017

<sup>7</sup> K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *Proc. CVPR*, pp. 770-778, 2016

consisteix d'una pila de 8 capes alternant de  $3 \times 3$  totalment convolucionals i ReLU amb 512 canals.



L'arquitectura DensePose consisteix en una cascada de generació de propostes de regions i característiques, seguida d'una FCN que prediu densament etiquetes discretes de parts i coordenades de superfícies contínues.

## 3.2. Refinament de silueta

### 3.2.1. Aproximació de la segmentació basada en parts amb contorns

Un cop obtinguda la segmentació basada en parts (màscares) del cos humà, el següent pas és obtenir una aproximació del contorn per a cada regió en cada part. Aquest primer pas de refinament s'aconsegueix mitjançant la funció OpenCV 'findContours'. Aquest procés es representa a la figura següent.

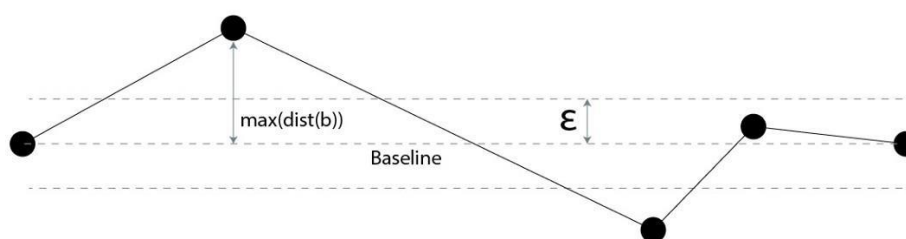


Representació basada en contorns. Imatge d'entrada original (esquerra), representació de màscares de parts DensePose (mig) i representació basada en el contorn de cada part (dreta).

### 3.2.2. Representació simplificada del contorn

Per poder ser manipulats i ajustats manualment, simplifiquem la representació del contorn amb un nombre reduït de punts. Per tal de trobar l'aproximació més adequada, fem servir l'algorisme RDP (Ramer - Douglas - Peucker). Tanmateix, no apliquem directament l'algorisme RDP a la representació del contorn original (figura 3 dreta), sinó que primer es redueix el nombre de punts que defineixen el contorn (de milers a dos centenars, per optimitzar el cost computacional).

L'algorisme RDP defineix una mètrica de "dissimilació" entre la corba original i la corba simplificada, de manera que la corba simplificada consisteix en un subconjunt dels punts que van definir l'original. Es basa en un procés recurrent, començant pel primer i els darrers punts. A continuació, troba el punt més llunyà de la línia que connecta el primer i l'últim punt del contorn. Si aquesta distància és superior a un llindar determinat ( $\epsilon$ ), aquest punt es marca com a "mantingut", en cas contrari es descarta. I el procés s'aplica de manera recursiva tenint en compte el parell de (primer punt; punt conservat) i (punt conservat i últim punt). En la figura d'abaix s'il·lustra el funcionament de l'algorisme.



Funcionament de l'algorisme RDP

Per tal de facilitar encara més l'ajustament manual de cada forma, s'estableix el nombre de punts retornats pel RDP en base a proves heurístiques en el rang 10-40, en funció del perímetre de la forma: per tant, per a àrees més grans (per exemple, tors) aquest nombre de punts és més proper a 40, mentre que, per a àrees més petites (per exemple, peus o mans), aquest nombre es troba al voltant de 10.

### 3.2.3. Suavització de polígons amb corbes de Bézier

Els punts obtinguts com a resultat de l'algorisme RDP defineixen un polígon que s'aproxima a la part del cos corresponent. Com que representaria una aproximació molt simplificada del contorn, no utilitzem directament aquesta representació de polígons, sinó que en fem una versió més suavitzada. Aquesta aproximació suau s'obté amb les corbes de Bézier<sup>8</sup>, una eina ben coneguda utilitzada per a la interpolació, àmpliament usada en gràfics per ordinador, per exemple. Les corbes de Bézier es representen com a polinomis de grau 'n' que es ponderen amb un conjunt de punts de control. En el nostre cas, el conjunt de punts de control és la sortida de l'algorisme RDP i hem utilitzat polinomis de 3r grau per tal d'obtenir una aproximació suavitzada del polígon definit pels punts de control. Aquest procés es representa a la figura 4.

<sup>8</sup> [https://en.wikipedia.org/wiki/B%C3%A9zier\\_curve](https://en.wikipedia.org/wiki/B%C3%A9zier_curve)





La corba de Bézier (grogà) representa una aproximació suavitzada del polígon (negre).

### 3.2.4. Suavitzió temporal

Com que estem utilitzant un nombre variable de punts RDP a cada fotograma per representar la mateixa part del cos, pot donar lloc a transicions abruptes d'un fotograma a un altre. Com que no disposem d'informació a priori per construir un model dinàmic, per tal de realitzar el seguiment d'objectes en la seqüència de vídeo, adoptem l'estratègia de "seguiment per detecció", és a dir, detectem l'objecte d'interès a cada quadre. Per tal de suavitzar aquestes transicions, hem introduït una estratègia adaptativa basada en la història acumulativa del model, tal com es mostra a l'eq.1:

$$X^*(t) = (1-a)X^*(t-1) + aX(t) \quad (1)$$

El valor real del model en el moment  $t$  es calcula com una suma ponderada del model anterior en el temps anterior  $t-1$  i del model estimat actual. El paràmetre  $a$  és un pes que indica si el model passat o l'estimació actual té més rellevància. Es recomana un valor petit del paràmetre  $a$  per a transicions entre fotogrames lentes, mentre que per a transicions ràpides, on el model s'hauria d'adaptar ràpidament, es recomana un valor gran. Als nostres experiments vam establir el valor entre 0,5 i 0,9.

A la següent figura es representa el diagrama complet del procés descrit anteriorment.

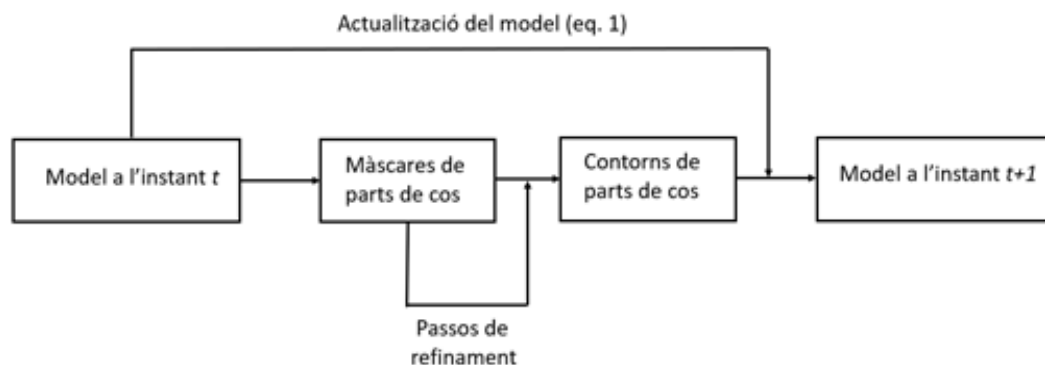


Diagrama del flux de refinament de silueta

### 3.2.5. Avaluació

Per avaluar el resultat del nostre enfocament, hem utilitzat dues mètriques: IoU i DICE. L'IoU (també conegut com a índex Jaccard) és una mètrica popular per estimar la semblança entre dos conjunts de punts (dues deteccions) i es defineix com a la relació entre la intersecció de l'objecte detectat i l'objecte real i la mida de la seva unió. DICE és una mètrica similar, definida com la proporció entre dues vegades la intersecció dividida pel nombre total de píxels de l'objecte detectat i el real.

Presentem els resultats de la detecció de màscares DensePose (cos sencer) aplicades a 6 fotogrames per als quals tenim el ground-truth. Els resultats es presenten a la taula del costat (com més alt, millor). Un valor de coeficient 1 representaria una superposició perfecta entre la forma detectada i la real. Al contrari, un valor de coeficient 0 representaria un desajust total.

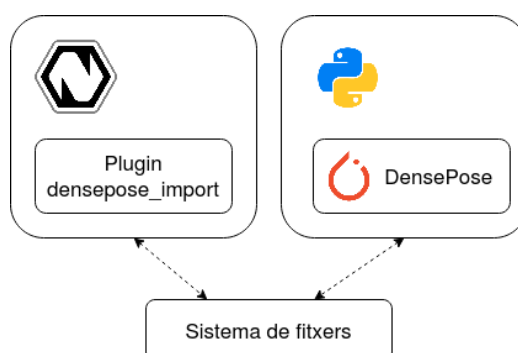
	IoU	DICE
0 (001)	0.895	0.944
1 (031)	0.882	0.937
2 (061)	0.890	0.941
3 (091)	0.878	0.935
4 (121)	0.879	0.935
5 (151)	0.882	0.937

IoU vs DICE per a 6 fotogrames (s'indica el número de fotograma de la seqüència de vídeo).

### 3.2.6. Arquitectura plug-in de Natron

L'eina de detecció i seguiment de persones (DensePose) s'ha desenvolupat integralment en Python, presentant la sortida del software com a un plugin de la plataforma Natron per post-producció.

El modul DensePose rep com a entrada una imatge o un vídeo d'una o més persones i s'executa amb diferents paràmetres que l'usuari pot modificar mitjançant el plugin de Natron, en funció de les seves necessitats. La sortida del programari, que representa les corbes de Bézier per a cada part del cos, s'importa a la plataforma Natron per a un millor perfeccionament manual i integració en altres mòduls (per exemple, per a rotoscòpia). L'arquitectura del plug-in i la seva integració a la plataforma de Natron es mostra a la imatge de sota.



Arquitectura plug-in DensePose i la seva integració a Natron



### 3.3. Manual de d'instal·lació i configuració

#### 3.3.1. Instal·lació

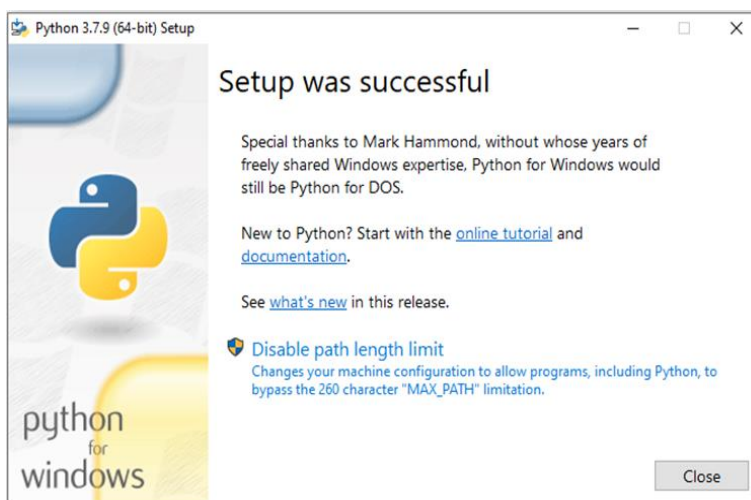
El plugin ha estat provat amb la versió de Natron 2.3.15. Per a realitzar la instal·lació es necessita l'arxiu *DensePose\_plugin.zip* i connexió a Internet. Primerament, s'extreu tot el contingut de *DensePose\_plugin.zip* en una carpeta. A continuació s'instal·len les dependències del projecte:

##### Python3.7:

- S'executa l'arxiu *python-3.7.9-amd64.exe*, se selecciona la opció *Add Python 3.7 to PATH* i es fa clic a *Install Now*

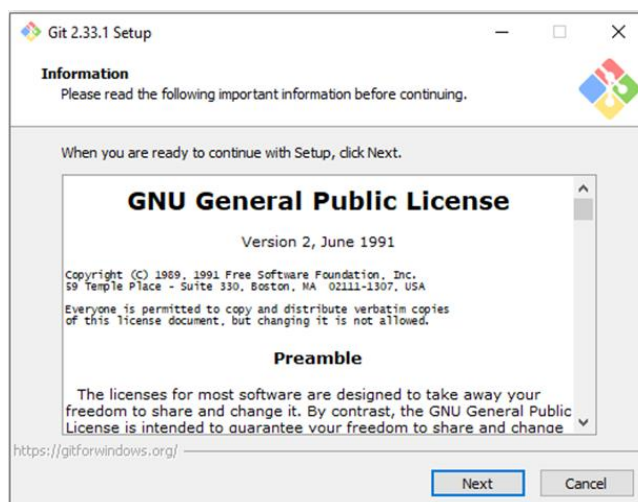


- Per últim, s'ha de fer clic a *Disable path length limit* i a *close*



##### Git:

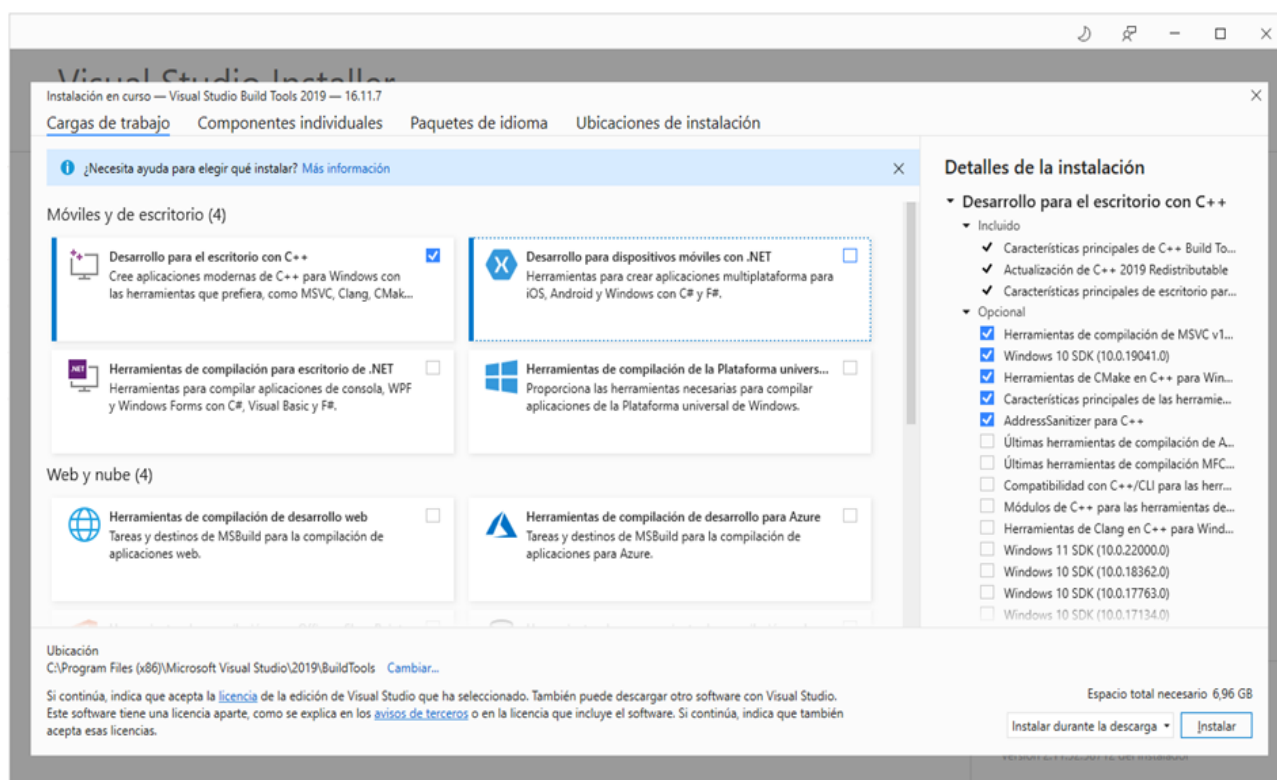
- S'executa l'arxiu *Git-2.33.1-64-bit.exe*



- Es fa clic en *Next* sense modificar cap opció fins que finalitzi la instal·lació.

### Visual c++:

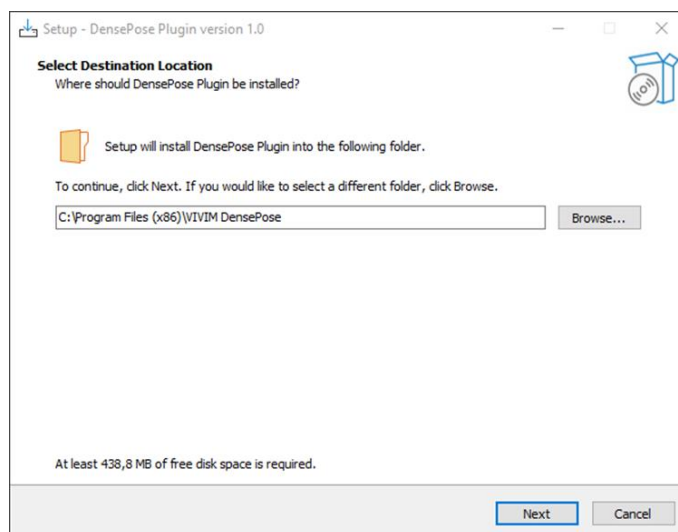
- S'executa l'arxiu vs\_BuildTools.exe. L'instal·lador descarregarà les dades necessàries i ens mostrarà la següent finestra:



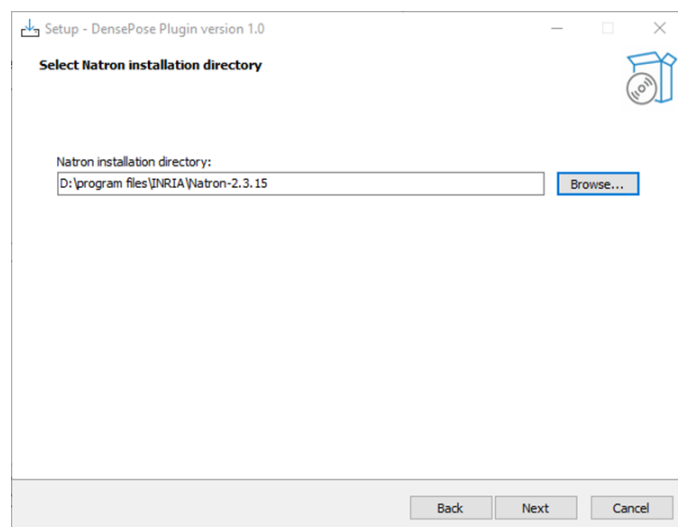
- Se selecciona la opció *Desarrollo para el escritorio con c++* i es fa clic a *Instalar*
- Quan finalitzi la instal·lació podem tancar la finestra.

Un cop instal·lades les dependències instal·lem DensePose i el plugin de Natron.

- S'executa l'arxiu *vivim\_densepose\_cpu.exe* i s'indica on es desitja instal·lar la DensePose:



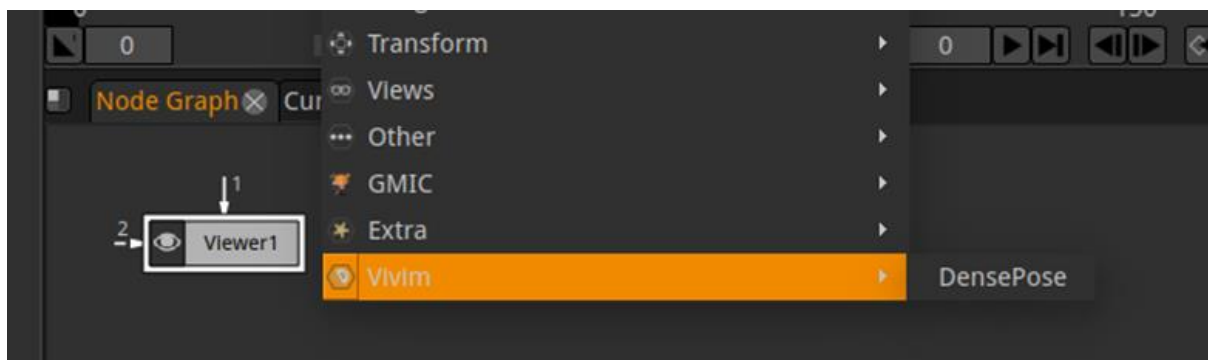
- A continuació s'indica el directori on es troba instal·lat Natron. Habitualment es pot trobar en *C:\Program Files\INRIA\Natron-2.3.15*



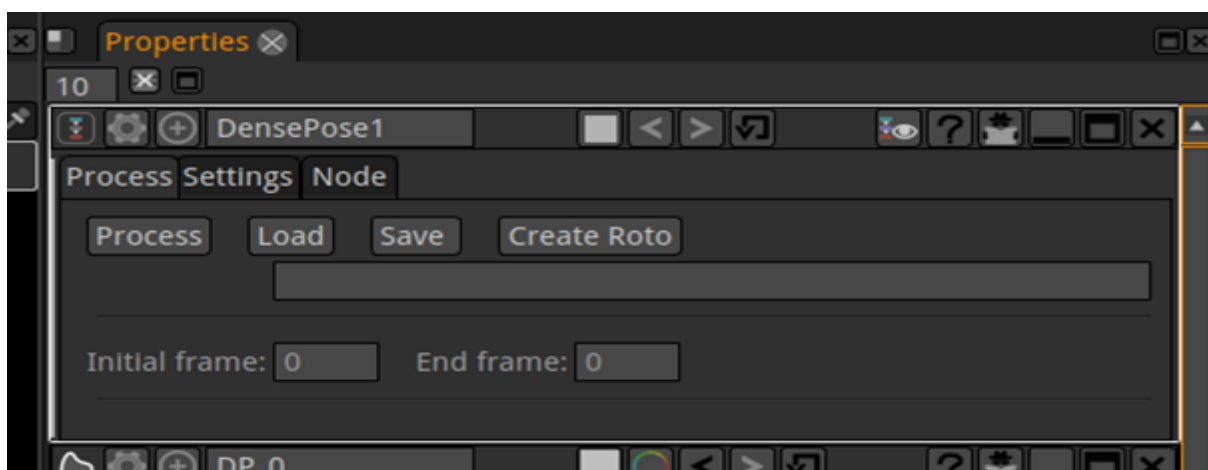
- Finalment es fa clic a *Next* i *Install*. Quan acabi aquest procés serà necessari reiniciar el sistema i ja es podrà utilitzar el plugin en Natron

### 3.3.2. Funcionament

Un cop instal·lat el plugin es podrà crear un node DensePose en Natron. Per això es fa clic dret dins del gràfic de nodes i se selecciona DensePose dins del menú VIVIM.



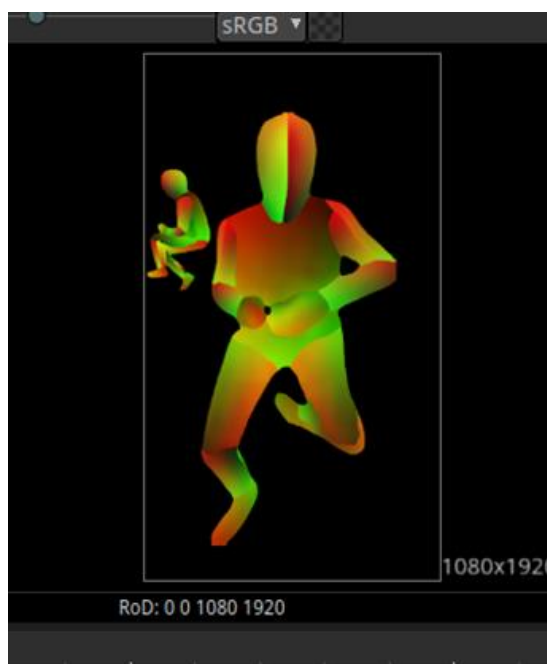
Aquest node accepta una entrada i una sortida. Les imatges d'entrada seran utilitzades per fer la segmentació amb DensePose i la sortida mostrarà el mapa IUV generat per la xarxa.



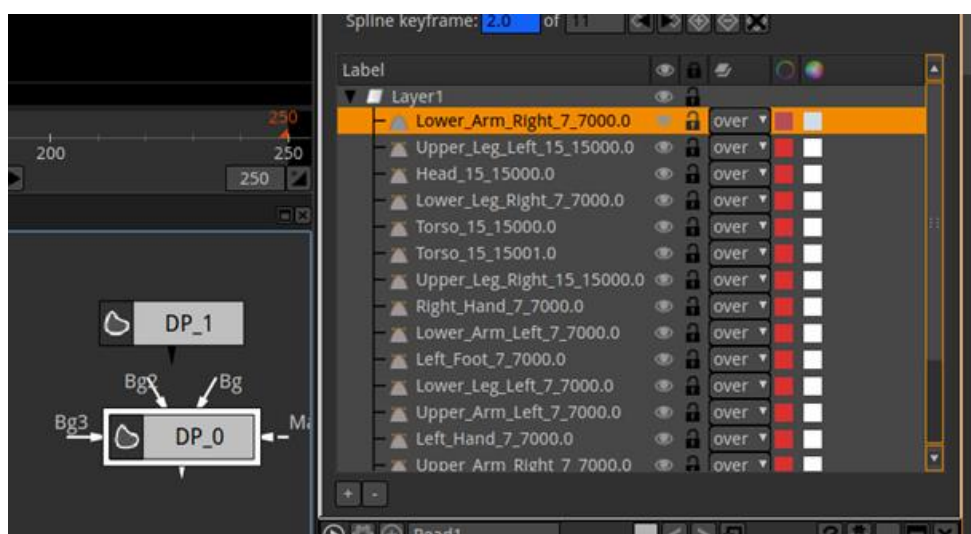
Per processar les imatges cal especificar el rang de frames en els camps "Initial frame" i "End frame" i fer clic en "Process".

A continuació Natron realitzarà un renderitzat de les imatges d'entrada en un directori temporal i executarà de forma paral·lela la xarxa DensePose. El quadre de text situat a sota del botó "Process" mostrarà l'estat del procés en cada moment.

Un cop acabat el processat de les imatges es mostrarà el mapa IUV en la sortida del node. Aquest mapa codifica un mapa UV en els canals vermell i verd, individual per a cada part del cos, i estableix un valor diferent en el canal blau per cada part.



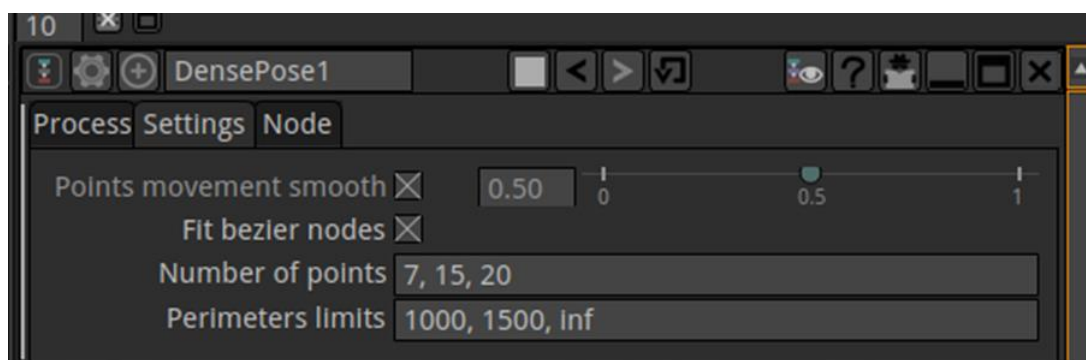
Per carregar la rotoscòpia automàtica s'ha de fer clic al botó “Create Roto”. Aquest procés pot trigar varis minuts en funció de la complexitat y longitud de la seqüència d'imatges, durant aquest període Natron quedarà bloquejat. Un cop finalitzat s'obtindrà un node de rotoscòpia per a cada persona detectada.



Dins la pagina “settings” es trobaran diferents paràmetres relacionats amb la creació de la rotoscòpia automàtica que podem ajustar:

- “Points movement smooth”: Permet activar i establir el valor de suavitzat temporal que s'aplica als punts de les corbes Bézier. Un valor més gran crea un suavitzat major, prioritant els valors de les corbes de frames anteriors.
- “Fit Bézier nodes”: Permet crear corbes Bézier que s'ajustin el màxim possible al mapa de IUV.
- “Number points”: Permet establir el nombre de punts de les corbes de Bézier de les rotoscòpia. Es pot definir diferents nombres de punts objectius que s'utilitzaran en funció de la mida de cada màscara.

- “Perimeter limits”: Determina els valors límits de perímetres de les màscares que s'utilitzaran per a determinar el nombre de punts.



Si es vol modificar algun d'aquests paràmetres s'haurà de tornar a processar la seqüència i crear les rotoscopies per a que els canvis tinguin lloc.

El processat de les imatges es realitza en dos passos. Primerament una detecció amb la xarxa neuronal DensePose i a continuació un post-processat d'aquestes dades per a generar les rotoscòpia. Si es fa clic en “process” múltiples vegades amb les mateixes imatges d'entrada i amb paràmetres diferents, DensePose reutilitzarà les deteccions prèvies per tal d'executar únicament el segon pas i reduir el temps de processat.

Quan es tanca Natron, les imatges i dades generades per DensePose seran eliminades. Si es desitja guardar-les es pot fer clic en “Save” i seleccionar un directori on guardar totes les dades. També es pot carregar aquestes dades fent clic en el botó “Load” i seleccionant la carpeta que les conte.

### Com establir “Number points” i “Perimeter limits”:

El nombre de punts de les màscares de les rotoscopies ha de ser establert de forma experimental per a obtenir els resultats esperats. Si s'estableixen els següents paràmetres:

“Number points”: 7, 15, 20

“Perimeter limits”: 1000, 1500, inf

DensePose calcularà el perímetre de cada màscara de cada part del cos. Els perímetres inferiors a 1000 utilitzaran 7 punts, els situats entre 1000 i 1500 utilitzaran 15 i els situats entre 1500 i infinit, 20. Es pot definir tants grups com es volen.

## 3.4. Requisits de software/hardware

Per poder executar-lo es necessita una màquina Windows que disposi d'una GPU (de gamma mitjana o alta), amb CUDA, Python 3.0 i Natron instal·lats.

Tot el codi desenvolupat és 100% Python que permet portar més fàcilment el software en diferents sistemes operatius. Les dependències de llibreries estàndard de Python durant el desenvolupament del component són les següents:

numpy = 1.19.1

```

matplotlib = 3.3.1
seaborn = 0.11.1
ruamel.yaml = 0.17.4
Bézier = 2020.5.19
scipy = 1.5.2
opencv_python = 4.5.1.48
torch = 1.7.1+cu110
torchvision = 0.8.2+cu110
torchaudio = 0.7.2
Shapely = 1.7.1
Pillow = 7.2.0
CUDA = 11.0

```

### 3.5. Prestacions

El component ha sigut testat en una màquina Windows 10 amb CPU Intel(R) Xeon(R) Silver 4116 CPU @ 2.10GHz amb 196 GB de RAM i en una GPU NVIDIA QUADRO RTX8000 amb 48 GB de RAM amb un video de 300 imatges. La complexitat computacional de cadascun dels subcomponents del flux es presenten a la taula següent.

Mida de les imatges (pixels)	1920 x 1080	
Nombre d'imatges	300	
Predicció DensePose	19:59 m	4.00 s/frame
Post-processat (generar polígons a partir de les mascarees)	00:48 m	6.16 frame/s
Post-processat (RDP y generar corbes de Bézier)	13:55 m	2.79 s/frame
Post-processat (exportar dades a Natron)	00:12 m	24.19 frame/s
Temps total	34:54 m	6.98 s/frame

Complexitat de càlcul estimada pel temps necessari per a cada subcomponent.

### 3.6. Estatus d'integració

Aquesta es una primera versió del programari desenvolupat per al us en *post-producció*. Per tant, no ha estat optimitzada per a us a temps real. En la propera versió del programari, ens preocuparem que funcioni en temps real. En temps d'enviament d'aquesta versió del lliurable, el software ha sigut provat satisfactòriament per Visyon i portat a Windows per a fer possible una integració amb les eines que ells utilitzen.

### 3.7. Model de distribució

El software desenvolupat fa servir llibreries estàndard de Python. El codi de DensePose es va alliberar sota una llicència “*Non Commercial Creative Commons*”. Tot i que encara està pendent el millor model de comercialització del component software desenvolupat a ViVIM, i sota quina llicència es farà, habitualment CVC arriba a acords d’exploració del software entre parts de durada determinada amb exclusivitat per a un sector concret.

El codi actualment es pot consultar en aquesta URL privada:

[https://cvcuab-my.sharepoint.com/:f/g/personal/egracia\\_cvc\\_uab\\_cat/EtDMxa7E6ChCiM34Ug9zrfoBcupPHIULp2jAgUCX1T0-6Q?e=Dfaz5j](https://cvcuab-my.sharepoint.com/:f/g/personal/egracia_cvc_uab_cat/EtDMxa7E6ChCiM34Ug9zrfoBcupPHIULp2jAgUCX1T0-6Q?e=Dfaz5j)



## 4. Esqueletonització 3D

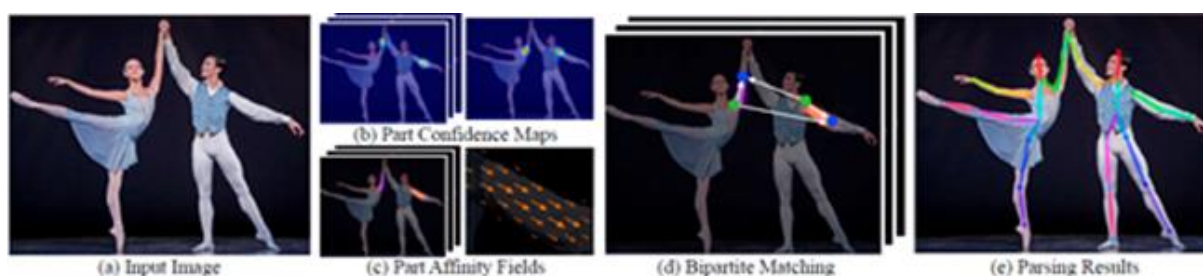
### 4.1. Introducció

El mètode que s'ha utilitzat s'anomena 'video2bvh'<sup>9</sup> i consta de tres mòduls:

- un estimador de postura en 2D basat en OpenPose que s'utilitza per estimar la postura en 2D del cos humà
- un estimador de postura 3D, que pren com a entrada el model anterior i estima la postura 3D del cos humà. Aquest estimador es basa en l'enfocament VideoPose3D
- un mòdul que estima la informació de l'esquelet a partir de la postura 3D: converteix la postura 3D en angles d'articulacions i escriu les dades de moviment en un fitxer 'bvh'

### 4.2. Estimació de la postura en 2D

L'enfocament utilitza una representació no paramètrica, que es coneix com a Camps d'Afinitat de Parts (CAP), per aprendre a associar parts del cos d'individus en la imatge. L'arquitectura codifica un context global que permet implementar un algorisme greedy bottom-up que manté una alta precisió alhora que aconsegueix un rendiment en temps real, independentment del nombre de persones a la imatge. L'arquitectura està dissenyada per aprendre conjuntament ubicacions de parts i la seva associació a través de dues branques del mateix procés de predicció seqüencial. El flux del procés es representa a la figura de sota.



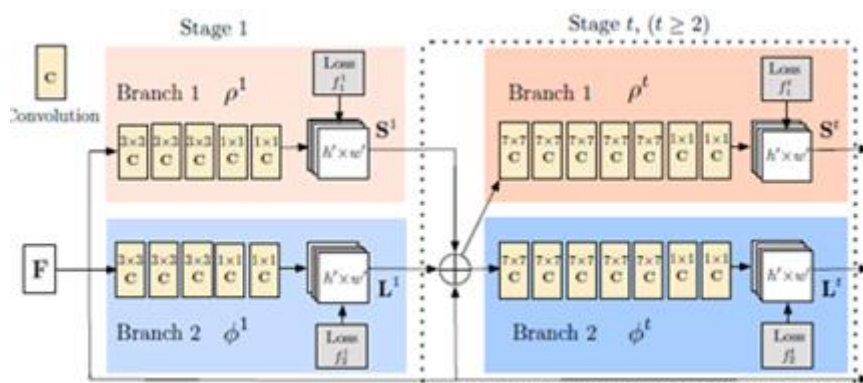
El mètode pren tota la imatge com a entrada d'una CNN de dues branques per a predir conjuntament mapes de confiança per a la detecció de parts del cos, que es mostren a (b), i camps d'afinitat per a l'associació de parts, que es mostren a (c). El pas d'anàlisi realitza un conjunt de coincidències bipartides per associar candidats a parts del cos (d), que finalment generen postures de cos sencer per a totes les persones de la imatge (e).

A un nivell més tècnic, l'arquitectura (vegeu la figura següent) prediu simultàniament mapes de confiança de detecció i camps d'afinitat que codifiquen l'associació part-a-part. La xarxa es divideix en dues branques: la branca superior, que es mostra en taronja, prediu els mapes de confiança, i la branca inferior, que es mostra en blau, prediu els camps d'afinitat. Cada branca és una arquitectura de predicció iterativa (basada en el treball de Wei<sup>10</sup>) que perfecciona les prediccions en etapes successives amb supervisió intermèdia en cada etapa. L'entrada comuna (designada per F) representa els mapes de característiques extrets amb una

<sup>9</sup> video2bvh. Available online at: <https://github.com/KevinLTT/video2bvh>

<sup>10</sup> S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. *Proc. of CVPR*, pp.4724-4732, 2016

arquitectura VGG-19<sup>11</sup>. Cada etapa (de cada branca) representa un classificador format per una successió de diverses capes convolucionals i pooling.

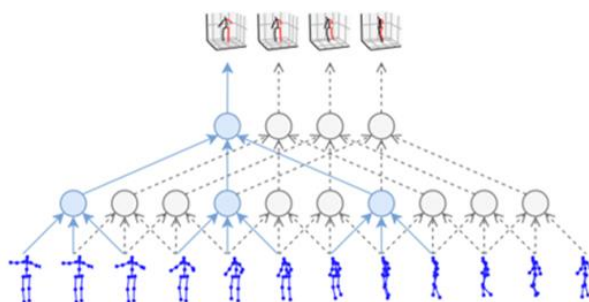


Arquitectura de la CNN multi-etapa de dues branques

Els camps d'afinitat per part són un mètode de representació de característiques que conserva la informació d'ubicació i orientació a tota la regió de suport de l'extremitat. Estimem l'alineació dels camps d'afinitat de les parts predites amb l'extremitat que es formaria connectant les parts del cos detectades. Amb tots els candidats de connexions d' extremitats, les connexions que comparteixen els mateixos candidats de detecció són agregats en una única postura de cos complet.

#### 4.3. Estimació de la postura en 3D

Aquest enfocament es basa en convolucions dilatades temporals (que s'utilitzen per modelar dependències a llarg termini) que pren com a entrada una seqüència de punts clau 2D i estima les corresponents postures 3D (vegeu la figura següent).

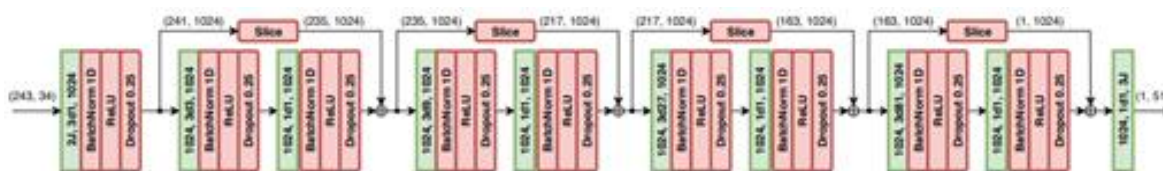


Visió general de l'enfocament: el model temporal convolucional pren seqüències de punts clau 2D (inferior) com a entrada i genera estimacions de postures en 3D com a sortida (superior)

A un nivell més tècnic, la capa d'entrada pren les coordenades concatenades  $(x, y)$  de les articulacions  $J$  per a cada imatge i aplica una convolució temporal amb una mida de *kernel*  $W$

<sup>11</sup> K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. Proc. ICLR 2015

i  $C$  canals de sortida. Això és seguit per  $B$  blocs residuals d'estil ResNet que estan envoltats per una connexió de salt. Les convolucions (excepte l'última capa) se segueixen per normalització de lots, unitats lineals rectificades i *dropout*. Una instanciació d'aquesta arquitectura es mostra a la figura de sota.

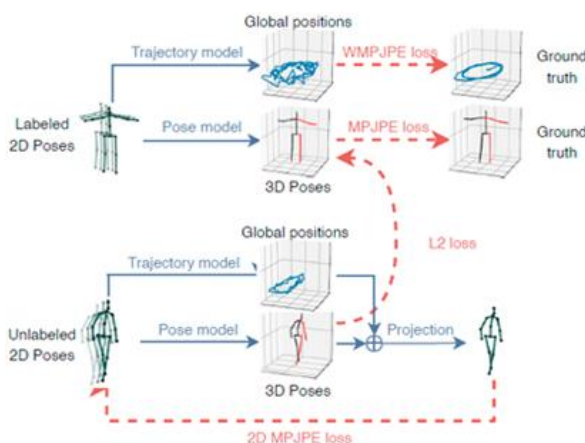


Una instanciació de l'arquitectura totalment convolucional d'estimació de poses 3D

Un dels principals avantatges de l'enfocament proposat és que no requereix dades anotades per als punts clau 2D. Amb la introducció d'una estratègia de “retroprojecció”, un mètode d'entrenament semi-supervisat senzill i eficaç, és possible aprofitar al màxim les dades de vídeo sense etiquetar. Imposant una restricció de consistència de cicle, el procés comença amb la predicció dels punts clau 2D per al vídeo sense etiquetes, a continuació, estima les posicions 3D i, finalment, torna a projectar als punts clau 2D d'entrada.

A un nivell més tècnic, l'enfocament d'entrenament semi-supervisat es podria formular com un codificador automàtic de dades de vídeo sense etiquetes: el codificador (estimador de postura) realitza una estimació de postura 3D a partir de coordenades articulars 2D i el descodificador (capa de projecció) projecta la posició 3D a coordenades articulars 2D. L'entrenament penalitza quan les coordenades d'articulacions 2D del descodificador estan lluny de l'entrada original. Aquest procés es mostra a la figura següent. Els dos objectius s'optimitzen conjuntament. Per a les dades etiquetades, fem servir el ground-truth de postures 3D com a objectiu i entrenem una loss supervisada.

A causa de la projecció en perspectiva, la postura 2D a la pantalla depèn tant de la trajectòria (es a dir la posició global de la persona) i la posició 3D (la posició de totes les articulacions respecte a l'articulació arrel). Sense la posició global, el model sempre es reproduiria al centre de la pantalla amb una escala fixa. Per tant, l'arquitectura també realitza una regressió de la trajectòria 3D de la persona, de manera que la re-projecció a 2D es pogués realitzar correctament.



Arquitectura que representa l'estratègia d'entrenament semi-supervisada, amb un model de postura 3D que pren com a entrada una seqüència de postures 2D possiblement predites. Tot s'entrena conjuntament. WMPJPE significa "MPJPE ponderat"

#### 4.4. Esquelet 'bvh'

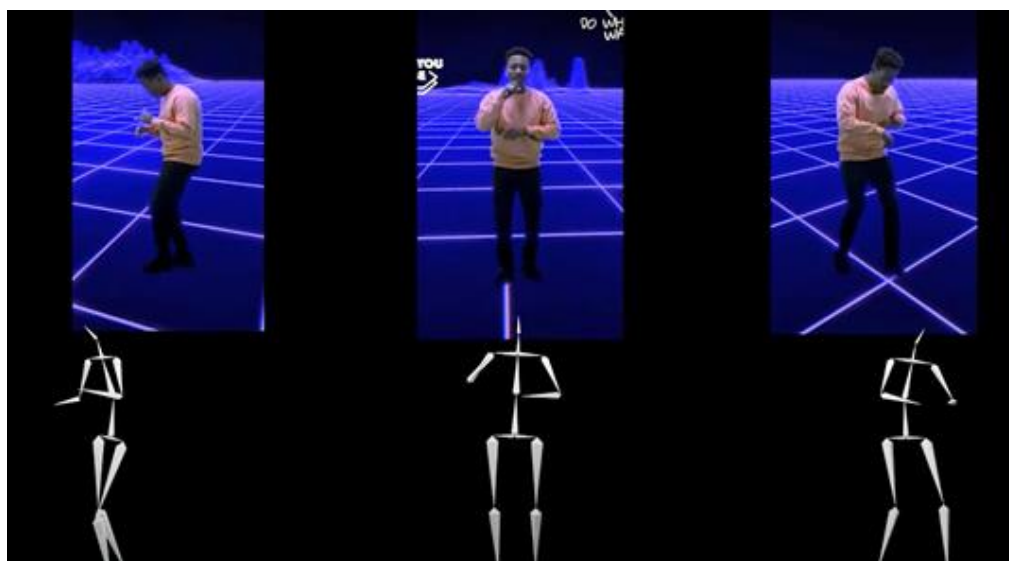
Aquest mòdul pren com a entrada la posició 3D estimada, la converteix en angle d'articulacions i escriu les dades de moviment en un fitxer 'bvh'.

#### 4.5. Implementació

En el nostre cas, hem utilitzat el mètode 'video2bvh' de la següent manera. En primer lloc, per a cada fotograma del vídeo, detectem la persona del vídeo 360, amb OpenPose. La caixa delimitadora estreta es transforma mitjançant una projecció en perspectiva en una imatge rectangular. Aquesta imatge es proporciona com a entrada al mètode 'video2bvh' que genera l'esquelet 3D. Alguns exemples de l'esquelet 3D extret es mostren a les següent dues figures.



Extracció d'esquelets 3D sota diferents punts de vista



Extracció d'esquelets 3D sota diferents punts de vista

## 4.6. Requisits de software/hardware

Per poder executar-lo es necessita una màquina Windows que disposi d'una GPU (de gamma mitjana o alta), amb CUDA, Python 3.0 instal·lats.

Tot el codi desenvolupat es 100% Python que permet portar més fàcilment el software en diferents sistemes operatius. Les dependències de llibreries estàndard de Python durant el desenvolupament del component:

```

openpose
numpy = 1.19.1
matplotlib = 3.3.1
torch = 1.7.1+cu110
torchvision = 0.8.2+cu110
opencv_python = 4.5.1.48
CUDA = 11.0

```

## 4.7. Prestacions

El component ha sigut testat en una màquina amb CPU Intel(R) Xeon(R) Silver 4116 CPU @ 2.10GHz amb 196 GB de RAM i en una GPU NVIDIA QUADRO RTX8000 amb 48 GB de RAM amb un video de 2180 imatges. La complexitat computacional d'aquest mètode es detalla a la taula següent.

Mida de les imatges (pixels)	3840 x 2160	
Nombre d'imatges	2180	
Predicció OpenPose	3:14 m	0.09 s/frame
Projecció perspectiva	17:25 m	0.5 frame/s
video2bvh	1:29 m	0.04 s/frame
Temps total	22:08 m	1.32 s/frame

Complexitat de càlcul estimada pel temps necessari per a cada subcomponent

## 4.8. Exportació a un plug-in de Natron

Aquesta és una primera versió del programari desenvolupat per a l'ús en *post-producció*. Per tant, no ha estat optimitzada per a us a temps real.. En temps d'enviament d'aquesta versió del lliurable, el software no ha estat exportat a un plug-in i per tant no ha pogut ser avaluat per

Visyon. En la propera versió del programari, ens preocuparem que estigui empaquetat en un plug-in i que funcioni en temps real.

## 4.9. Manual de d'instal·lació i configuració

### 4.9.1. Instal·lació

La instal·lació d'aquest mòdul cal realitzar-la en un entorn Linux. El contingut de `video2bvh_360.zip` s'extreu en el directori on es vol realitzar la instal·lació. A continuació dins de la carpeta `video2bvh_360` s'executa la següent comanda:

```
./install.sh
```

Aquest script instal·larà totes les dependències i mòduls necessaris.

### 4.9.2. Tracking de videos 360

Per tal de processar vídeos 360 primerament es necessari extreure una projecció perspectiva de les persones que apareixen en el vídeo. El mòdul `360_person_tracking.py` realitzarà un seguiment i crearà un vídeo de totes les persones presents en una seqüència d'imatges.

Podem executar-lo amb la següent comanda:

```
python3 360_person_tracking.py -i /ruta/a/les/imatges/ -o /directori/de/sortida/
```

Un cop processades les imatges obtindrem un o varis vídeos en el directori de sortida especificat, que podran ser utilitzats en el següent pas per a extreure el seu esquelet.

### 4.9.3. Extracció de l'esquelet

L'estimació de l'esquelet es realitza amb el mòdul `video2bvh.py`. Per a executar-lo cal introduir la següent comanda:

```
python3 video2bvh.py -v /ruta/al/vídeo/ -o /arxiu/de/sortida.bvh
```

Un cop processat el vídeo obtindrem la informació del esquelet en format `bvh` en la ruta especificada.