Multimedia Tools and Applications
https://doi.org/10.1007/s11042-023-14556-1

Pictonaut: movie cartoonization using 3D human pose estimation and GANs

Ruben Tous[1]

Received: 26 July 2021 / Revised: 9 June 2022 / Accepted: 31 January 2023

Abstract
This article describes Pictonaut, a novel method to automatically synthetise animated shots from motion picture footage. Its results are editable (backgrounds, characters, lighting, etc.) with conventional 3D software, and they have the finish of professional 2D animation. Rather than addressing the challenge solely as an image translation problem, a hybrid approach combining multi-person 3D human pose estimation and GANs is taken. Sub-sampled video frames are processed with OpenPose and SMPLify-X to obtain the 3D parameters of the pose (body, hands and face expression) of all depicted characters. The captured parameters are retargeted into manually selected 3D models, cel shaded to mimic the style of a 2D cartoon. The results of sub-sampled frames are interpolated to generate a complete and smooth motion for all the characters. The background is cartoonized with a GAN. Qualitative evaluation shows that the approach is feasible, and a small dataset of synthetised shots obtained from real movie scenes is provided.

Keywords Digital content creation · Deep learning · Computer vision · Computer graphics · Image cartoonization · Human pose estimation · Generative adversarial networks

1 Introduction

Techniques such as rotoscoping are used in the movie industry to utilise motion picture footage in the process of creation of 2D animated films. Existing techniques still require manual intervention at frame level and are very costly. Recent advances in Artificial intelligence (AI) open the possibility to attempt automating this process. Being able to automatically synthetise animated shots from motion picture footage would open the door to many new exciting applications, not only in the movie industry but also in the context of amateur animated film making and other creative contexts.

✉ Ruben Tous
rtous@ac.upc.edu

[1] Department of Computer Architecture, Universitat Politècnica de Catalunya (UPC), Jordi Girona, 1-3. 08034 Barcelona, Spain

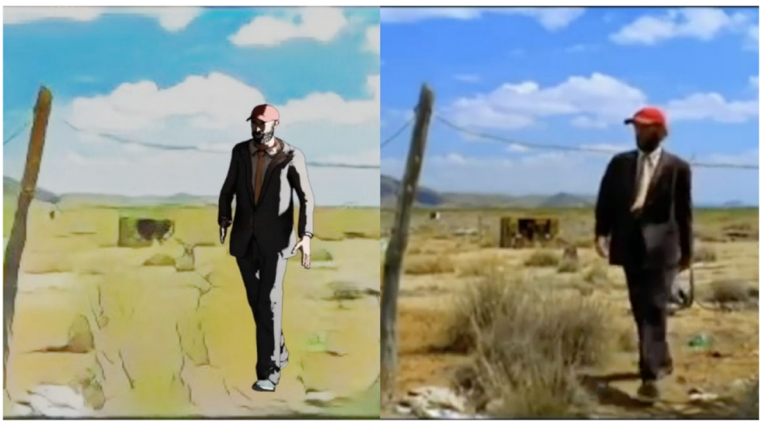

Fig. 1 Example synthetised frame (left) and its related original frame (right) from a sequence of the movie Paris, Texas

A straightforward approach to this challenge would be addressing it solely as an image translation problem, directly applying techniques such as Generative Adversarial Networks (GANs). However, a solution solely based on GANs would have some important limitations in this context. On the one hand, current GANs performance for video-to-video translation does still not achieve the quality of professional 2D animation, currently dominated by high resolution renders of interpolated and cel shaded 3D animations. On the other hand, in order to suit the professional animation process, a solution should provide a certain level of editability. Professional animation involves decouplable components (backgrounds, characters, lighting, etc.) which enable specialized professionals to work on specific aspects of the result in an independent way.

This article describes Pictonaut, a novel method, to automatically synthetise animated shots from motion picture footage. Rather than addressing the challenge solely as an image translation problem, a hybrid approach combining multi-person 3D human pose estimation and GANs is taken. A sequence of frames are extracted from an input monocular video through uniform temporal subsampling. On the one hand, the persons appearing in the frame and their body keypoints are detected with OpenPose [4]. A 3D model (SMPL-X [21]) of each person's pose (body, hands and face expression) is captured with SMPLify-X [21]. The SMPL-X parameters are retargeted into manually selected 3D models (designed with 3D character modelling software). The 3D models are cel shaded with procedural texturing in Blender. All the poses (from all the keyframes) are interpolated with cubic Bezier curves to create the final movement of the characters. On the other hand, the characters are removed from the background in the original frames with YOLO [2]. The holes in the background are inpainted with Telea [25]. The clean background is cartoonized with CartoonGAN [6]. The cartoonized background is combined with the animated 3D characters in Blender, where the final render is generated (Fig. 1).

2 Related work

Using still or moving images to generate animated cartoons have been used in the movie industry since Max Fleischer invented the rotoscope in 1915. Variants of this technique have

been used to this day, with outstanding results such as the animated science-fiction film "A Scanner Darkly" (2006) or the television animated series "Undone" (2019). Latest productions incorporate image processing algorithms to facilitate the task, but they are mainly based on costly manual work. Automating the process with a professional finish is currently not possible, but recent advances in Artificial intelligence (AI), especially in the field of deep learning, opens the possibility to attempt advancing to fully automation. A candidate technique are Generative Adversarial Networks (GANs), prevalent in image translation tasks nowadays. Methods such as CartoonGAN [6], AnimeGAN [5] or the recent work in [18] are able to transform photos of real-world scenes into cartoon style images with very good results. One limitation of these methods is that they provide little control to meet artists' requirements. In order to alleviate this problem, [28] proposes a method, also based on GANs, that operates over three different representations (surface, structure and texture). However, beyond the performance limitations of these methods for video-to-video translation in the wild, a pure image translation solution would not enable professional animators to edit the final result with the tools they are currently using, mainly 3D modeling software. Unlike these methods, the work presented in this paper takes a hybrid approach combining multi-person 3D human pose estimation and GANs and, as far as we know, it is the first work to address the problem this way. Some previous works combined computer vision techniques with 3D human pose reconstruction to synthetise animated videos. In [17] and [20], methods to generate 3D cartoons from broadcast soccer videos were proposed. A similar approach, but with a different goal, is taken in [29], in which 3D animations are generated from single photos. The work uses SMPL [19] as the underlying 3D model (we employ SMPL-X [21], which includes hands and face expressions). Like what happens in that work, the performance of the proposed method strongly relies on the performance of the underlying human pose estimation algorithms. A large amount of work has been devoted to this field, but 3D human pose estimation from monocular images remains a challenging problem because of its inherent difficulties (occlusions, background clutters, depth ambiguities, etc.). Nevertheless, remarkable progress has been facilitated by the availability of large-scale 3D pose datasets like Human3.6M [11] or CMUPanoptic [13] and the evolution of deep neural networks. Existing methods can be classified into three categories [12]: 3D pose tracking (most of the early works), 2D-3D pose lifting (e.g. [21]) and pose regression from images (e.g. [14] and [22]). In this work, a 2D-3D pose lifting method is used, SMPLify-X [21]. There are more recent 2D-3D pose lifting methods such as MixSTE [31], but some features of SMPLify-X (in-the-wild performance, estimation of camera parameters, hands pose and facial expressions) makes it specailly suited for this work. 2D-3D pose lifting methods consist of two processing steps, 2D multi-person pose estimation and 3D pose lifting. Popular 2D multi-person pose estimation methods are AlphaPose [8], OpenPose [4], HRNet [24] and its sequels [7, 33]. Lately, transformer-based methods such as TransPose [30] and TokenPose [16] are attracting the attention of the community due ther improved performance in occlusion scenarios. In this work, the employed 2D multi-person pose estimation method is OpenPose [4], because it is the default method used by SMPLify-X [21]. SMPLify-X fits SMPL-X to single RGB images through an optimization process consisting on minimizing the distance between the image's 2D body keypoints (which are estimated with OpenPose) and the 2D projection of the corresponding posed SMPL-X model. SMPL-X (SMPL eXpressive) is a realistic 3D model of the human body, learned from thousands of 3D body scans, with shape parameters trained jointly for the face, hands and body. SMPL-X combines SMPL with the FLAME head model [15] and the MANO [23] hand model (Fig. 2).

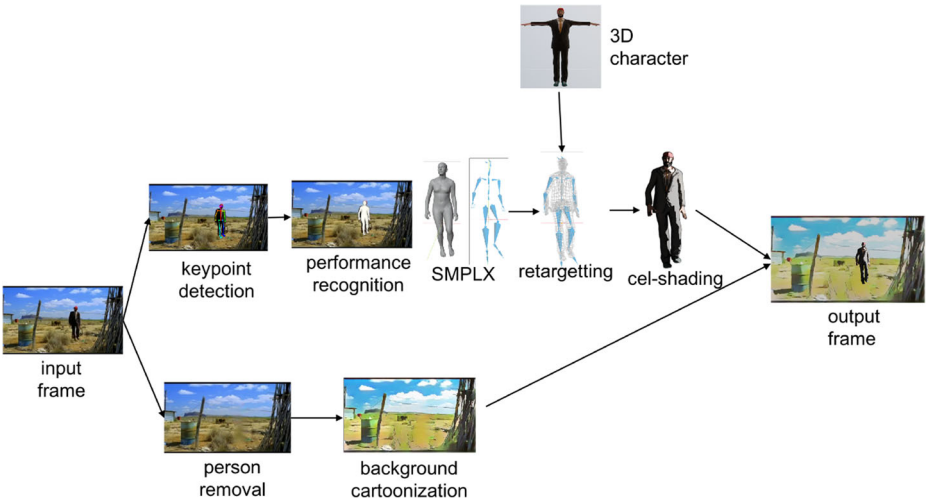

Fig. 2 Overview of the system's workflow for a single frame. Only a fraction of the frames of the input video undergo this process. The remaining frames are generated through interpolation

3 Methodology

The method translates a video to its cartoon version. It consists of three independent processing paths. The main one consists of estimating the 3D poses of the people that appear in the video frames. Only some of the frames go through this step, the rest are interpolated. A second processing path consists of cartoonizing the background. The third processing path consists of stylizing the input 3D models (cel shading, lighting, etc.). The results of the different processing paths are combined to obtain the final cartoonized video. Figure 3 shows the overall workflow of the method. Figure 2 visually shows the workflow for a single frame. In the following, each component is described in detail.

Input The system receives two inputs, a movie video scene clip (monocular) and C 3D models $\{\boldsymbol{m}_j\}_{j=1}^{C}$, one for each different character appearing in the scene $\{\boldsymbol{c}_j\}_{j=1}^{C}$. Given a N-frame video sequence, $M = N/R$ frames $\{\boldsymbol{f}_i\}_{i=1}^{M}$ are extracted from it through uniform temporal subsampling with ratio R : 1. Regarding the 3D models, although, theoretically, they may be automatically generated, it is expected that in this context (professional or DIY animation) they will be designed with 3D character creation software to model their appearance with artistic criteria. To emulate this situation, 3D models for the test dataset were created with Mixamo [9]. However, it is envisaged that a professional tool such as Reallusion's Character Creator [10] would be used in a production scenario.

3D human pose estimation First, given an i-th extracted frame, 2D body keypoints (body, hands, feet, and face features) $\{\boldsymbol{k}_{ij}\}_{j=1}^{C}$ are extracted for each character $\boldsymbol{c}_j$ with OpenPose [4]. OpenPose employs a multi-stage deep convolutional network (CNN) to detect body parts and Part Affinity Fields (PAFs) to associate body parts with individuals (as it is multi-person).

Second, a 3D model (SMPL-X [21]) of each person's pose (body, hands and face expression) is fitted into to the 2D features with SMPLify-X [21]. SMPL-X is parameterized by

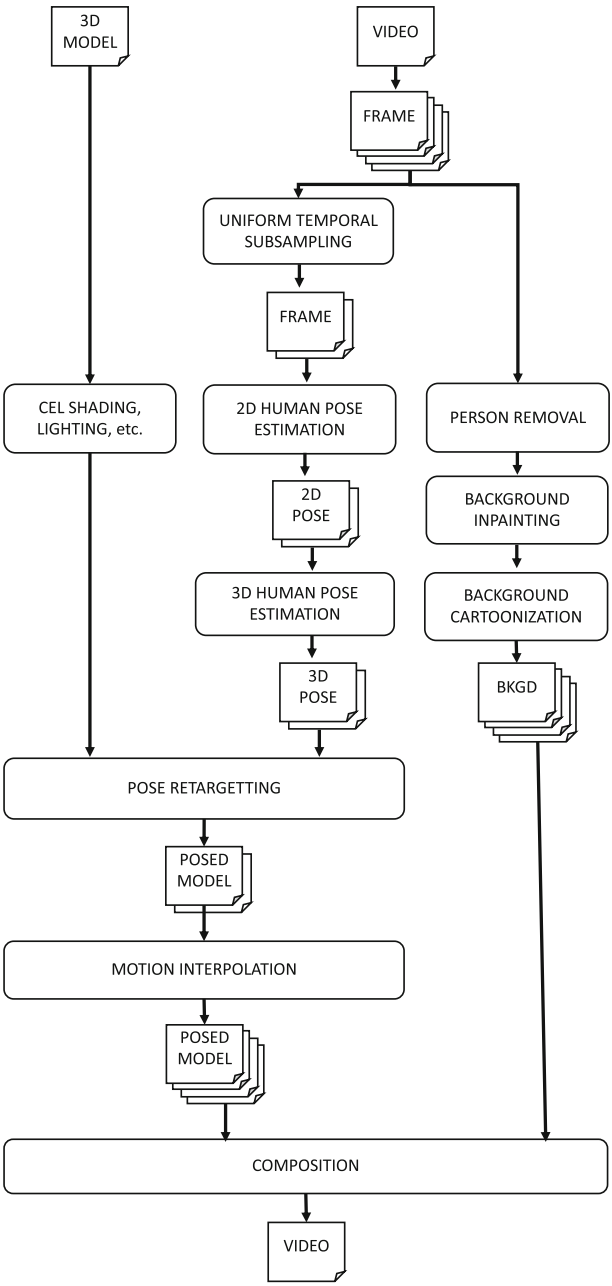

Fig. 3 Flowchart of the method. Squared rectangles represent processing steps. Document shapes represent inputs and outputs

$(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\psi})$. $\boldsymbol{\theta}$ is the pose $\boldsymbol{\theta} \in \mathbb{R}^{3(K+1)}$ where K is the number of body joints in addition to a joint for global orientation. There are 54 body joints (K = 54): 21 general joints, one joint the jaw, 2 joints for the eyes and 30 joints for the fingers. The compact axis-angle

format (three values) is used to represent the rotation of each joint. $\boldsymbol{\beta}$ are 10 linear subject shape coefficients (capturing shape variations due to different person identity) $\boldsymbol{\beta} \in \mathbb{R}^{|\beta|}$ and $\boldsymbol{\psi}$ are 10 facial expression parameters $\boldsymbol{\psi} \in \mathbb{R}^{|\psi|}$. SMPL-X is defined by a function $M(\theta, \beta, \psi)$: $\mathbb{R}^{|\theta| \times |\beta| \times |\psi|} \rightarrow \mathbb{R}^{3N}$ that maps the parameters into N mesh vertices (N = 10,475) through standard vertex-based linear blend skinning.

Through an optimization process, SMPLify-X minimizes the distance between the input image's 2D body keypoints (the ones estimated with OpenPose) and the 2D projection (the camera parameters $\boldsymbol{\phi}$ are also inferred) of the corresponding posed SMPL-X model. A pose-prior (VPoser [21]), implemented with a variational autoencoder, is used to overcome the intrinsic ambiguity of inferring a 3D pose from 2D features.

Each extracted frame $\boldsymbol{f}_i$ is processed first with OpenPose and second with SMPLify-X to obtain the motion, expression and camera parameters for each person appearing in the frame. We denote the set of all parameters $(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\beta}, \boldsymbol{\phi})$ produced at this stage for the character j in the i-th extracted frame by $\boldsymbol{\Omega}_{ij}$. Figure 4 shows an example of the pose estimation step.

Pose retargeting The next step is to apply the SMPL-X parameters $\boldsymbol{\Omega}_{ij}$ from each frame $\boldsymbol{f}_i$ and involved character $\boldsymbol{c}_j$ to each 3D character model $\boldsymbol{m}_j$ with the desired aspect (hair, clothes, etc.). Ideally, the 3D character would be based on SMPL-X, which would make this step straightforward. As there're still no proper tools that enable that option, the inferred SMPL-X pose and expression need to be retargeted into a 3D model generated by an existing 3D character creation software. For this work, a tool to retarget Mixamo [9] characters has been created. For rigged characters, the retargeting just aligns the rest pose of the Mixamo character to the SMPL-X rest pose. For not-rigged characters, the tool rigs the character with a re-scaled SMPL-X skeleton. The SMPL-X skeletal rig has been reverse engineered from the SMPLify-X code. Figure 5 shows an example of the pose retargeting step.

Motion interpolation All the captured poses (from all the keyframes) are applied to the retargeted character to create the animation. Poses from omitted keyframes are interpolated with cubic Bézier curves. As interpolation is applied over each rotation expressed with a quaternion q, an additional correction is necessary to avoid rotating the joints 360 degrees, as q and $-q$ represent the same rotation (see Algorithm 1).

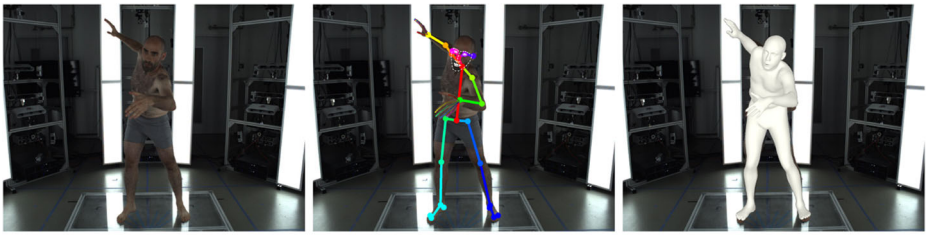

Fig. 4 3D human pose estimation. (a) Input frame, (b) Detected body keypoints, (c) Inferred SMPL-X 3D model (body, hands and face expression)

Require: $Q_w[\,]$, $Q_x[\,]$, $Q_y[\,]$, $Q_z[\,]$ Bezier curve points arrays for the rotation quaternion components w, x, y, z

function BEZIERQUATERNIONCORRECTION($Q_w[\,]$, $Q_x[\,]$, $Q_y[\,]$, $Q_z[\,]$)
  $N \leftarrow length(Q_w)$
  for $i \leftarrow 0$ to $N - 1$ **do**
    startQuat $\leftarrow \left[Q_w[i], Q_x[i], Q_y[i], Q_z[i]\right]$
    endQuat $\leftarrow \left[Q_w[i+1], Q_x[i+1], Q_y[i+1], Q_z[i+1]\right]$
    if $startQuat \bullet endQuat < 0$ **then**
      $Q_w[i+1] = -Q_w[i]$
      $Q_x[i+1] = -Q_x[i]$
      $Q_y[i+1] = -Q_y[i]$
      $Q_z[i+1] = -Q_z[i]$
    end if
  end for
end function

Algorithm 1 Interpolation correction.

Although the obtained interpolation results are good for the test dataset, component-wise interpolation could cause some artefacts and the evaluation of alternative solutions such as spherical interpolation would be convenient.

Editing final aspect, Cel shading, lighting, etc. In order to make the obtained 3D animation mimic the style of a traditional 2D animation (or a comic book or cartoon), a cel shading non-photorealistic rendering is applied. Three shaders have been developed for testing: a flat color shader, a black and white shader and a black and white with textured shadows shader (Charles Burns style). All of them were developed with procedural texturing for the Blender's realtime (rasterization) render engine. Figure 6 shows an example result of each shader. Another important postprocessing step is lighting. The system doesn't

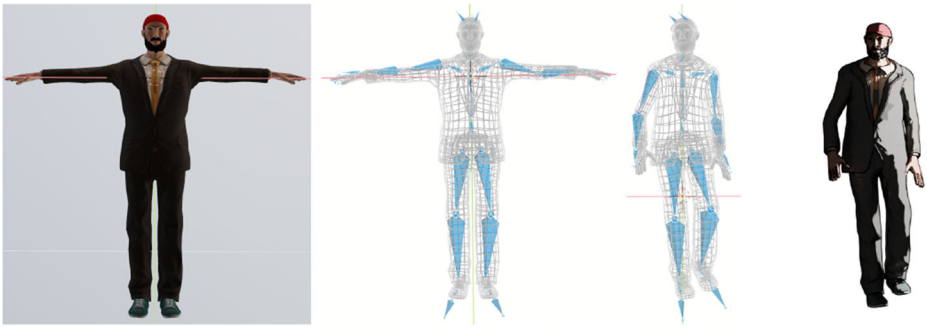

Fig. 5 Pose retargeting. (a) 3D Mixamo character, (b) Rigged with the SMPL-X skeleton, (c) posed, (d) cel shaded

Fig. 6 Different procedural cel shaders: (a) flat color, (b) black and white shader and (c) a black and white with textured shadows (Charles Burns style)

currently attempt to infer the location and intensity of light sources in the input image. It just automatically places a sun light object over the 3D scene. Adding more light sources and adjusting lights location, type (sun, point, spot or area) and intensity needs to be done manually.

Background cartoonization In order to cartoonize the background, the characters are first removed from the background in the original frames with YOLO [2], a deep convolutional neural network for real-time object detection. Then, the holes in the background are inpainted with Telea [25], that uses the Fast Marching Method [1] to propagate an image smoothness estimator along the image gradient. Finally, the clean background is cartoonized with CartoonGAN [6], a method for transforming real-world photos into cartoon style images. CartoonGAN consists of a generative adversarial network (GAN) trained over unpaired real-world photos and cartoon images extracted from several short animation videos. It combines an adversarial loss with a content loss. The content loss is obtained by processing both, the input photo and the generated cartoon image, through a pre-trained VGG network and comparing the high-level feature maps. CartoonGAN overcomes some of the problems that state-of-the-art image-to-image translation frameworks (e.g. [32]) have with cartoon styles. One of this problems is the difficulty of reproducing the necessary clear edges and smooth shading while retaining the content of the input photo. CartoonGAN achieves this with an edge-promoting adversarial loss for clear edges, and the usage of l_1 sparse regularization of the high-level VGG feature maps in the content loss for reproducing smooth shading.

The cartoonized background is combined with the animated 3D characters in Blender, where the final render is generated. A more precise person removal, that would make inpainting unnecessary, could be achieved, using the SMPL-X mesh as a mask or with a specialized algorithm. However, the applied strategy provided good results in combination with CartoonGAN, and it's very fast. In scenes with dynamic backgrounds all the original frames (and not only the ones subsambled) are processed.

4 Experiments and results

Two kind of input videos were used for testing. On the one hand, videos from the CMU Motion Capture Database (CMU Mocap) [27] and the Human3.6M dataset [11] were used to test the performance of the 3D human pose estimation and pose retargeting components, and the resulting interpolated animation. On the other hand, real movie video scene clips (from Top Gun, Terminator, Paris Texas, Alien, Night of the Living Dead and Charade) were used to test the overall system in a real scenario. 3D characters were created, with Mixamo [9]. The input videos, their related 3D characters and the resulting animated shots are publicly accessible at [26].

Results in controlled conditions The results obtained over videos from CMU Mocap and Human3.6M show that the pose estimation and pose retargeting components work reasonably well under controlled conditions. In general, Human3.6M results are better because the videos have a greater resolution and the scenes are more constrained (centered, no truncations, etc.). The limitations of the 2D and 3D pose estimation components are the main source of errors. For instance, subject 35 - motion 17 of CMU Mocap cannot be properly processed because the 2D pose estimation method cannot deal with the truncations of this scene. One aspect that needs to be significantly improved is the estimation of the location and angle of the camera, currently performed by SMPLify-X. In some videos (e.g. subject 9 - motion 1 of CMU Mocap and subject 7 - motion 5 of Human3.6M) the estimated variations in the camera parameters introduce some unrealistic transitions. Another limitation is related to the pose retargetting stage. The hands pose, despite of being computed, is not currently mapped to the 3D Mixamo character. This has a higher impact on Human3.6M (e.g. subject 5 - motion 1) because Human3.6M scenes involve a greater variety of hand positions.

Results on real film shots A small dataset of real movie video scene clips was built to test the method in a real context. Different shot sizes were included (see Fig. 7). According to [3], except for extreme scale shots (ELS and ECU), a relevant proportion of all shot types can be expected in a film. The specific proportion is determined by the specific demands of the narrative of each film, rather than being indicative of the style of a particular director, era or national cinema [3]. Both scenes with a fixed background and scenes with a changing

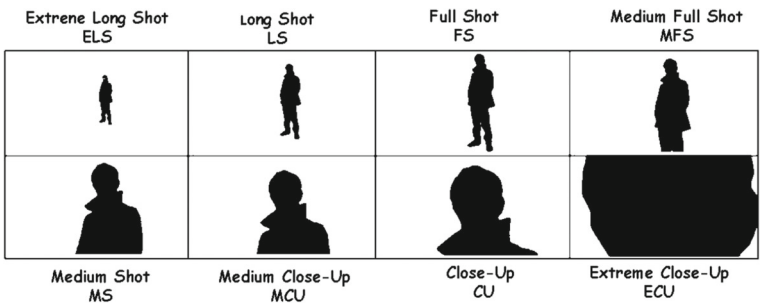

Fig. 7 Examples of eight different shot sizes: Extreme Long Shot (ELS), Long Shot (LS), Full Shot (MS), Medium Full Shot (MFS), Medium Shot (MS), Medium Close-Up (MCU), Close-Up (CU), Extreme Close-Up (ECU)

background were included. One 3D character was created, with Mixamo [9], for each scene. All the videos last between 3 and 10 seconds and were subsampled with a 5:1 ratio. The scenes in the dataset are numbered. Scene numbers will be used in this section to refer to specific challenges or results. Figure 8 shows screenshots of the output videos. Qualitative evaluation shows that the method accomplishes its goal, synthetizing an editable animated

Fig. 8 Example synthetised frames (left) and its related original frame (right) from a sequence of the movies (a) Top Gun, (b) Terminator, (c) Alien, (d) Paris, Texas, (e) Charade, (f) Night of the Living Dead

shot, for some scenes. However, several technical challenges remain to be solved before the method can be used in production.

Limitations On the one hand, there are problems related to the 3D pose estimation stage. The involved algorithms (OpenPose and SMPLify-X) show several limitations when applied to arbitrary movie scenes. The usage of a pose-prior enables processing partial body shots and the algorithms perform well on most LS (e.g. scene 12) and FS shots (e.g. scene 13), and some ELS (e.g. scene 11) and MFS shots (e.g. scene 1). However, they are not able to deal with MS, MCU, CU, ECU shots. Applying specialized algorithms to different groups of shot types could be a way to overcome this problem. Sometimes OpenPose detects non-existent characters (in the shadows, the vegetation, etc.), especially in low resolution videos (e.g. scenes 4, 5 and 7). On the other hand, there are problems related to the pose retargeting stage. Retargeting between different 3D models is lossy, and sometimes the retargeted Mixamo characters show non-realistic poses or motion trajectories. Besides, Hand poses and face expressions are captured but they are not yet retargeted to the Mixamo characters. While a better retargeting approach could be investigated, using a common parametrizable 3D model through all the stages would be the proper solution to these problems. Another challenge partially satisfied are multi-person scenes. The system currently processes scenes with multiple characters but it's still not able to correctly render them. The completion of this feature has been left for future work. Finally, another challenge are characters interacting with objects (e.g. scenes 11, 12 and 13). Sometimes this can be solved making the object part of the target 3D model (e.g. a gun), but sometimes manually editing the final scene is necessary.

Strengths Complicated poses (including back poses) are correctly processed on most LS and FS shots (e.g. scenes 2 and 13). Dynamic shots (either by the movement of the camera or by the movement of the character) are correctly processed (always by moving the 3D camera). Scenes from old black and white films, with very low resolution, can be processed too (e.g. scenes 5-10). Frames with incorrect 3D pose estimation results can be manually suppressed and are automatically interpolated by the system. The system is able to process scenes with a changing background (e.g. scenes 12 and 13).

Comparison with other methods As far as we know, this is the first work that specifically addresses the automation of movie 3D-based cartoonization. The closest methods would be those oriented to generic image cartoonization. Figure 9 shows some examples comparing the results of CartoonGAN [6] and White-box-Cartoonization [28] with the method presented here (Pictonaut) for a single frame. Comparative videos have been also generated and are publicly accessible at [26].

These examples are only provided to help understanding the differences between the approaches, not to evaluate them. CartoonGAN and White-box-Cartoonization are pure end-to-end image translation approaches, they are much easier to apply but provide very little margin for editing the result. When applied to videos, all elements in the scene (static or dynamic) show constant changes in their aspect, as each frame is processed independently. This results in blurry and shaky videos. This problem also affects to the backgrounds of the method presented here, as they are also processed with a GAN. However, this only affects to the background and only when it changes.

Computational cost The experiments were run on a high-end server with a quadcore Intel i7-3820 at 3.6 GHz with 64 GB of DDR3 RAM memory, and 4 NVIDIA Tesla K40 GPU

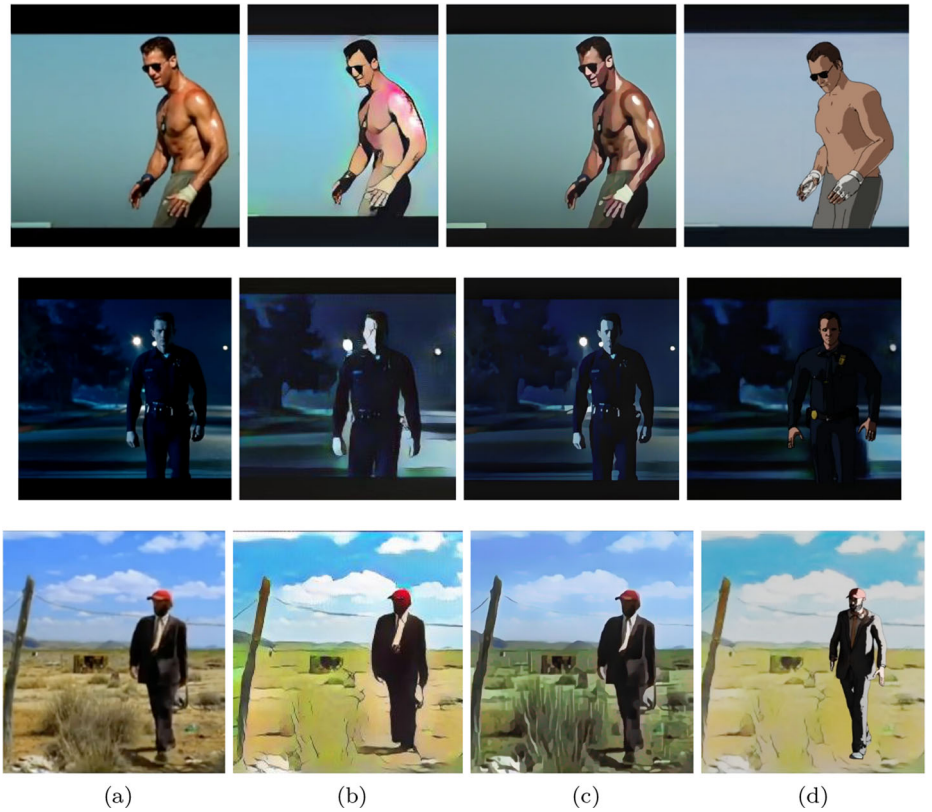

Fig. 9 Single frame results comparison: (a) Original footage, (b) CartoonGAN [6], (c) White-box-Cartoonization [28] and (d) Pictonaut

cards with 12 GB of GDDR5 each. The system was also tested on a commodity computer without the support of the GPU (a MacBook Pro with a Intel Core i5 at 2,7 GHz and 8GB of RAM memory). The method is compute-intensive and runs around 0.1 FPS on the high-end server and around 0.01 FPS on the commodity computer. Fine-tuning Open Pose and SMPLify-X may drastically increase the computational performance and will be addressed in future work.

5 Conclusions

In this work we present Pictonaut, a method for automatically synthetise animated shots from motion picture footage. Rather than addressing the challenge solely as an image translation problem, a hybrid approach combining multi-person 3D human pose estimation and GANs is taken. As far as we know, this is the first work that specifically addresses the automation of movie 3D-based cartoonization. The advantage of the method with respect to GAN-based generic image cartoonization methods is that its results are editable (backgrounds, characters, lighting, etc.) with conventional 3D software and that they have the finish of professional 2D animation.

Sub-sampled video frames are processed with OpenPose and SMPLify-X to obtain the 3D parameters of the pose (body, hands and face expression) of all depicted characters. The captured parameters are retargeted into manually selected 3D models, cel shaded to mimic the style of a 2D cartoon. The results of sub-sampled frames are interpolated with cubic Bezier curves to generate a complete and smooth motion for all the characters. The background (after removing the characters with YOLO and Telea) is cartoonized with CartoonGAN.

Qualitative evaluation shows that the approach is feasible, and a small dataset of synthetised shots obtained from real movie scenes is provided. The method can handle complicated human poses, a changing subject-to-camera distance, a changing background, black and white and low resolution films, and can interpolate problematic frames. However, many technical problems remain to be solved before the method can be used in production (mainly related to the limitations of the employed 3D pose estimation algorithms and to the pose retargeting between different 3D models). Further research is needed to address these problems and to advance towards fully automation.

Acknowledgements This work is partially supported by the Spanish Ministry of Science and Innovation under contract PID2019-107255GB, and by the SGR programme 2017-SGR-1414 of the Catalan Government.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work is partially supported by the Spanish Ministry of Science and Innovation under contract PID2019-107255GB, and by the SGR programme of the Catalan Government under contract 2021 SGR 00478.

Data Availability Data is publicly available at https://github.com/rtous/pictonaut.

Declarations

Ethics approval The authors declare that all procedures performed in the study were in accordance with the legislation of The Spanish Data Protection Agency[1] (Agencia Española de Protección de Datos, AEPD - Organic Law 3/2018 of 5 December on the Protection of Personal Data), the ethical standards of the European Commission's European Textbook on Ethics in Research: Chapter 4[2]: Privacy and confidentiality, the Treaty on the Functioning of the European Union, article 16,[3] the Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995[4] and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Conflict of Interests The authors declare that they have no conflict of interest.

[1] http://www.agpd.es/

[2] EUROPEAN COMMISSION, European Textbook on Ethics in Research (2010), pp. 75–93. http://ec.europa.eu/research/science-society/index.cfm?fuseaction=public.topic&id=1362

[3] Consolidated Version of the Treaty on the Functioning of the European Union, Official Journal of the European Union (30.3.2010) No. 83/55, article 16.

[4] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.

References

1. Andrew AM (2000) Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. Robotica 18(1):89–92
2. Bochkovskiy A, Wang CY, Liao HYM (2020) Yolov4: optimal speed and accuracy of object detection
3. Canini L, Benini S, Leonardi R (2011) Affective analysis on patterns of shot types in movies, pp 253–258
4. Cao Z, Hidalgo Martinez G, Simon T, Wei S, Sheikh YA (2019) Openpose: realtime multi-person 2D pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence
5. Chen J, Liu G, Chen X (2020) AnimeGAN: a novel lightweight GAN for photo animation, pp. 242–256. Artificial Intelligence Algorithms and Applications. https://doi.org/10.1007/978-981-15-5577-0_18
6. Chen Y, Lai YK, Liu YJ (2018) Cartoongan: generative adversarial networks for photo cartoonization. In: CVPR. IEEE Computer Society, pp 9465–9474
7. Cheng B, Xiao B, Wang J, Shi H, Huang TS, Zhang L (2020) HigherHRNet: scale-aware representation learning for bottom-up human pose estimation. In: CVPR
8. Fang HS, Xie S, Tai YW, Lu C (2017) RMPE: Regional multi-person pose estimation. In: ICCV
9. Inc A (2021) Animate 3D characters with mixamo. https://www.mixamo.com/. Accessed 21 Mar 2021
10. Inc R (2021) Character creator 3. 3D character creation for animation, game, AR, and VR. https://www.reallusion.com/character-creator/. Accessed 21 Mar 2021
11. Ionescu C, Papava D, Olaru V, Sminchisescu C (2014) Human3.6m: large scale datasets and predictive methods for 3D human sensing in natural environments. IEEE Trans Pattern Anal Mach Intell 36(7):1325–1339
12. Ji X, Fang Q, Dong J, Shuai Q, Jiang W, Zhou X (2020) A survey on monocular 3D human pose estimation. Virtual Real Intell Hardw 2(6):471–500. https://doi.org/10.1016/j.vrih.2020.04.005. https://www.sciencedirect.com/science/article/pii/S2096579620300887
13. Joo H, Liu H, Tan L, Gui L, Nabbe BC, Matthews IA, Kanade T, Nobuhara S, Sheikh Y (2015) Panoptic studio: a massively multiview system for social motion capture. In: ICCV. IEEE Computer Society, pp 3334–3342
14. Kanazawa A, Black MJ, Jacobs DW, Malik J (2018) End-to-end recovery of human shape and pose. In: Computer vision and pattern regognition (CVPR)
15. Li T, Bolkart T, Black M, Li H, Romero J (2017) Learning a model of facial shape and expression from 4D scans. ACM Trans Graph 36:1–17. https://doi.org/10.1145/3130800.3130813
16. Li Y, Zhang S, Wang Z, Yang S, Yang W, Xia ST, Zhou E (2021) TokenPose: learning keypoint tokens for human pose estimation. In: IEEE/CVF international conference on computer vision (ICCV)
17. Liang D, Liu Y, Huang Q, Zhu G, Jiang S, Zhang Z, Gao W (2005) Video2Cartoon: generating 3D cartoon from broadcast soccer video. In: MULTIMEDIA '05
18. Liu Z, Li L, Jiang H, Jin X, Tu D, Wang S, Zha Z (2022) Unsupervised coherent video cartoonization with perceptual motion consistency. arXiv:2204.00795
19. Loper M, Mahmood N, Romero J, Pons-Moll G, Black MJ (2015) SMPL: a skinned multi-person linear model. ACM Trans Graphics (Proc SIGGRAPH Asia) 34(6):248:1–248:16
20. Ngo V, Cai J (2008) Converting 2D soccer video to 3D cartoon. In: 2008 10th international conference on control, automation, robotics and vision, pp 103–107, https://doi.org/10.1109/ICARCV.2008.4795500
21. Pavlakos G, Choutas V, Ghorbani N, Bolkart T, Osman AAA, Tzionas D, Black MJ (2019) Expressive body capture: 3D hands, face, and body from a single image. In: Proceedings IEEE conf. on computer vision and pattern recognition (CVPR), pp 10975–10985
22. Rogez G, Weinzaepfel P, Schmid C (2019) LCR-Net++: multi-person 2D and 3D pose detection in natural images. IEEE Transactions on Pattern Analysis and Machine Intelligence. https://doi.org/10.1109/TPAMI.2019.2892985
23. Sridhar S, Oulasvirta A, Theobalt C (2013) Interactive markerless articulated hand motion tracking using RGB and depth data. 2013 IEEE international conference on computer vision, pp 2456–2463
24. Sun K, Xiao B, Liu D, Wang J (2019) Deep high-resolution representation learning for human pose estimation. In: CVPR
25. Telea A (2004) An image inpainting technique based on the fast marching method. J Graphics, GPU, Game Tools 9(1):23–34
26. Tous R (2021) Pictonaut, editable movie cartoonization through 3D performance capture. https://github.com/rtous/pictonaut. Accessed 05 May 2021
27. University CM (2001) Animate 3D characters with mixamo. http://mocap.cs.cmu.edu/. Accessed 21 Mar 2021

28. Wang X, Yu J (2020) Learning to cartoonize using white-box cartoon representations. pp 8087–8096. https://doi.org/10.1109/CVPR42600.2020.00811
29. Weng CY, Curless B, Kemelmacher-Shlizerman I (2019) Photo wake-up: 3D character animation from a single photo. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)
30. Yang S, Quan Z, Nie M, Yang W (2021) Transpose: keypoint localization via transformer. In: IEEE/CVF international conference on computer vision (ICCV)
31. Zhang J, Tu Z, Yang J, Chen Y, Yuan J (2022) MixSTE: Seq2seq mixed spatio-temporal encoder for 3D human pose estimation in video. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 13232–13242
32. Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: 2017 IEEE international conference on computer vision (ICCV)
33. Zigang G, Sun K, Xiao B, Zhang Z, Wang J (2021) Bottom-up human pose estimation via disentangled keypoint regression. In: CVPR