May 06, 2019

# Live head avatar using a single camera

Per Niklas Enbom

Dillon Cower

Guangyu Zhou

Tarek Hefny

# Live head avatar using a single camera

## ABSTRACT

This disclosure describes generation of a photo-realistic, three-dimensional video of the head of a user using a single mobile device camera. The three-dimensional head video, referred to as an avatar, is live. The video closely resembles the user's skin texture, and mimics the user's facial gestures and expressions in real time. The live head avatar is generated and utilized with the user's permission.

## KEYWORDS

- Virtual reality
- Augmented reality
- Head model
- Photo-realistic model
- Three-dimensional model
- Live avatar
- Face mesh
- Face wireframe
- Head mesh
- Face texture
- Live texturing
- Occlusion detection

## BACKGROUND

Social media experiences increasingly occur within virtual reality (VR) or augmented reality (AR) settings. For example, in a VR call, the far end receives a binocular or three-dimensional video of the user at the near end. For such VR or AR experiences, a photo-realistic, three-dimensional video of the user's head is to be produced from the available video source, e.g., video received from a single mobile-device camera that is directed at the user. In the context of a call, the three-dimensional head video, referred to as an avatar, is live (provided in real time) and closely resembles the user's skin texture, and mimics the user's facial gestures and expressions.

The view angle at the far end may be different from the pose of the near-end user. For example, the near-end user may face the camera straight on, while the far-end may choose to view the user's avatar from the left side. A single, static camera cannot capture the entire head due to limited field of view. Further, some parts of the head may be occluded by other parts, e.g., the back of the head is occluded by the front of the head.

## DESCRIPTION

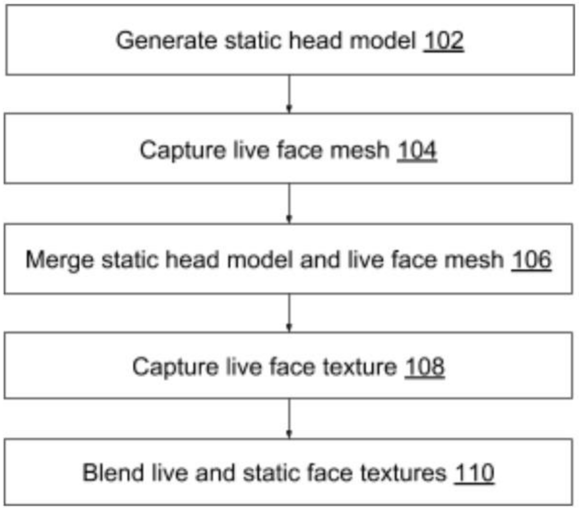
Generate static head model 102
Capture live face mesh 104
Merge static head model and live face mesh 106
Capture live face texture 108
Blend live and static face textures 110

**Fig. 1: Generating a live head avatar**

Fig. 1 illustrates an example process to generate a live head avatar, per techniques of this disclosure. The techniques are implemented with user permission. A static head model is generated (102) by requesting the user to turn their head while in front of the mobile-device camera, with a static, neutral expression. Augmented reality APIs are utilized to combine the camera feed and depth-sensor information to generate a static, three-dimensional head model with face-tracking information. The static head model is created in real-time and on-device, and is based on the color and depth video stream of the camera. The static head model includes a head mesh, e.g., a wireframe model of the head, and an associated high-resolution texture. The static head model includes the entire head, e.g., the face including facial landmarks; the back, the top, the sides of the head; the chin, etc. The static model is generated once, or as needed.

During a call or other video-sharing contexts, the far-end may view the avatar from a pose or angle different from the pose or angle-of-capture of the near-end user. If only the current video frame is used, parts of the head that ought to be visible at the far end are not captured at the near end, and hence go missing at the far end. Similarly, if only the current video frame is used, other parts of the head that ought to be visible at the far end are occluded at the near end, thus decreasing realism. Such problems are addressed by filling in the missing or occluded parts with the corresponding part of static head model.

During a VR call, the face is dynamic, e.g., the mouth opens and closes; the eyeballs move; eyebrows rise; etc. To animate the face part of the head synchronous with the actual face expressions of the user, a live face mesh, which includes facial landmarks, is captured (104). An augmented reality API can be used to capture the live face mesh in real time. The live face mesh includes a live wireframe and a live texture. The live wireframe morphs during the VR call to match the facial expressions of the user.

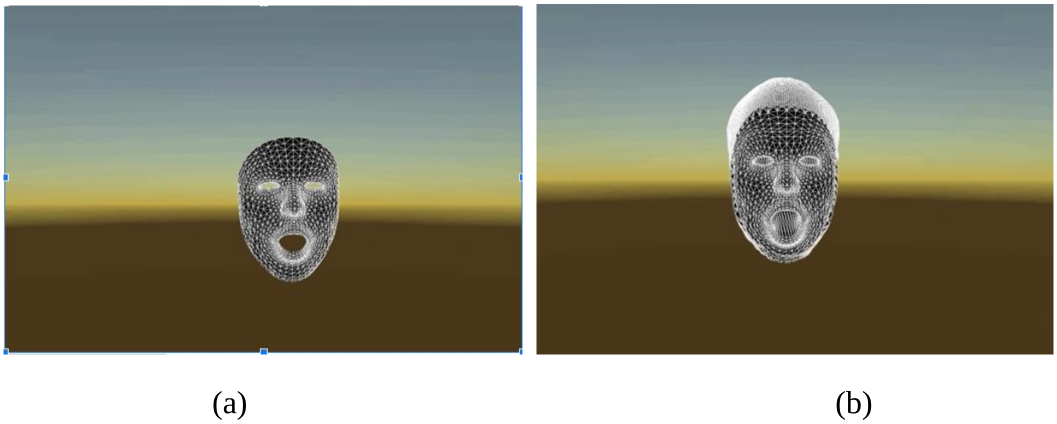

**Fig. 2: Live face mesh (a) before facial hole-filling (b) after facial hole-filling**

Fig. 2(a) illustrates an example of a snapshot of a live face mesh. Typically, a live face mesh does not capture the eyes or the mouth; corresponding shapes (e.g., triangles) are added to the mesh at regions of the face where holes are present, e.g., at the eyes or at the mouth, as illustrated in Fig. 2(b).

The static head model and the live face mesh are merged (106). The face being part of the head, the face mesh area is covered by head mesh area. The live face mesh has a structure that is updated from the static face mesh. The static face mesh replaces the static head mesh in the face area of the static head mesh. Alternatively, the shape of the head mesh can be adjusted in the face area to match the live face mesh.

During head model creation, an anchor frame, is used to align all other frames. The face mesh of the anchor frame and the head mesh are merged during head model creation. Further, the chin area of head mesh is adjusted to match that of the live face mesh. The associated adjustment is pre-calculated during head model creation, and is used in the live avatar. With this, the wire frame moves synchronously with the user’s actual facial movements, while the texture is static.

The live face texture, including finer details such as wrinkles, smile lines, frown lines, lip lines, etc., are captured and updated in real time (108). However, as mentioned before, the far-end user may view the avatar from a pose or angle different from the pose or angle-of-capture of the near-end user, leading to occlusions in the view at the far end. A determination is made of which facial areas are visible at the far end using occlusion detection techniques, e.g., using the angles of the near-end camera and of the far-end view. The non-occluded pixels are rendered with live texture, e.g., updated with color and depth feeds. The vertices of the head mesh are assigned texture-space (*u*,*v*) coordinates that map points on the head mesh to points on the two-dimensional live texture. In this manner, a two-dimensional texture is created that spans the whole head.

The static texture spans the whole mesh, while the live texture captures the finer face details. The static and the live textures of the face pixels are blended (110) such that sudden jumps in color do not occur, e.g., due to differing lighting of the static and the live textures. The static texture is used as a base texture, and the live texture is blended on top of it as pieces become available.

A pixel-specific blending weight, which is a number between 0 and 1 that represents the weight of the live texture relative to that of the static texture, is calculated. The blending weight is based on factors such as the depth of the pixel, the angle between the normal to the surface and the camera angle, whether the pixel is occluded or visible at the far end, etc. The higher the confidence in the live texture, the closer the blending weight is to unity. To calculate the blending weight, a Gaussian blur filter is applied on the alpha component of the live texture, thereby smoothing the weight variance for adjacent pixels. If the pixel happens to have no previously seen data, the blending weight is set to zero.

The live value is calculated as follows. For high confidence regions, the pixel carries directly the live texture value. For lower confidence regions, the live texture is blurred (smoothed) by a Gaussian blur filter to account for previously seen pixels in the neighborhood. The lower the confidence, the wider the blur filter, e.g., the blur filter width is defined as (1 − blending weight). The blended texture is calculated as:

```
Blended texture = (Live Value) × (Blending Weight) + (Static Texture Value) ×
(1 - Blending Weight).
```

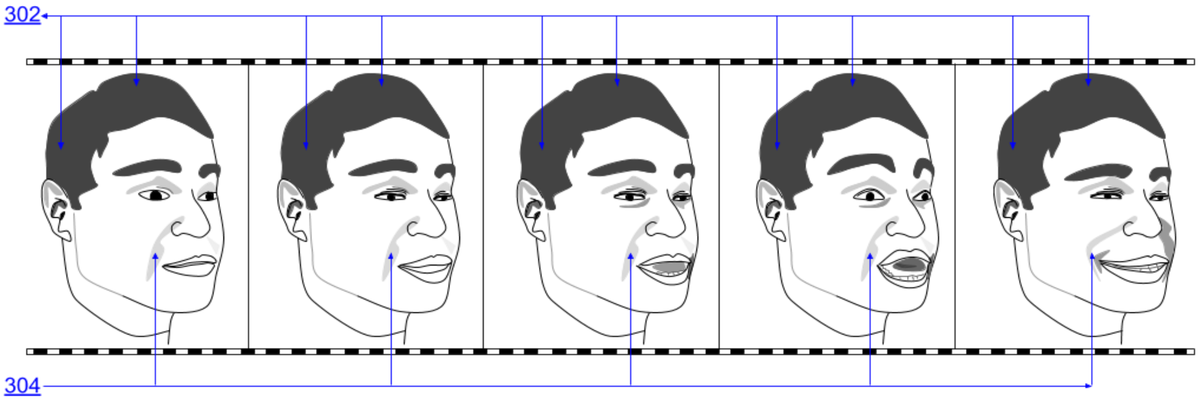

**Fig. 3: Live head avatar**

Fig. 3 illustrates example snapshots of a video of a live head avatar as seen at the far end, generated per the techniques of this disclosure. In the example of Fig. 3, the near-end camera is directed straight on at the user (not shown). The far end user views the right side of the near end user’s head. The static portions of the head (302), e.g., the top, back, and sides of the head, are modeled by the static head mesh and the static head texture. The dynamic portions of the head (304), e.g., the face, are modeled by a blend of static and live meshes, and by a blend of static and live textures.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein may enable collection of user information (e.g., information about a user’s social network, social actions or activities, profession, a user’s preferences, or a user’s current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

## CONCLUSION

This disclosure describes generation of a photo-realistic, three-dimensional video of the head of a user using a single mobile device camera. The three-dimensional head video, referred to as an avatar, is live. The video closely resembles the user's skin texture, and mimics the user's facial gestures and expressions in real time. The live head avatar is generated and utilized with the user's permission.

## REFERENCES

[1] Ichim, Alexandru Eugen, Sofien Bouaziz, and Mark Pauly. "Dynamic 3D avatar creation from hand-held video input." *ACM Transactions on Graphics (ToG)* 34, no. 4 (2015): 45.

[2] Jiang, Luo, Juyong Zhang, Bailin Deng, Hao Li, and Ligang Liu. "3D face reconstruction with geometry details from a single image." *IEEE Transactions on Image Processing* 27, no. 10 (2018): 4756-4770.