Visage Technologies
FACE TRACKING & ANIMATION

Modeling Guide for AR Applications

version 1.0

Contents

1. Introduction 3

2. Size and positioning 3

3. Occlusion 4

3.1. Importing the occlusion mask 5

3.2. Adjusting the occlusion mask 5

3.2.1. Nose adjustment 5

3.2.2. Head sides 5

3.2.3. Ears 6

4. Importing the model into the final AR application 6

4.1. Three.js (used in visage|SDK for HTML5) 6

4.1.1. Editing the MTL file 6

4.2. Unity3D (used in visage|SDK for Windows, iOS, Android) 6

1. Introduction

This guide explains how to prepare 3D models to be used in augmented reality (AR) applications based on visage|SDK head tracking. As an example, the guide shows how to prepare eyeglasses for Virtual Try-on application. The same principles may be applied to other objects to be used in AR applications.

2. Size and positioning

Real-life size, in meter units.
X positioning: centre of glasses at X=0
Y positioning: visual axes at Y = 0
Z positioning: eyeball front edge at Z = 0 (therefore lenses at approx Z = 0.02 - 0.025 m depending on glasses model)

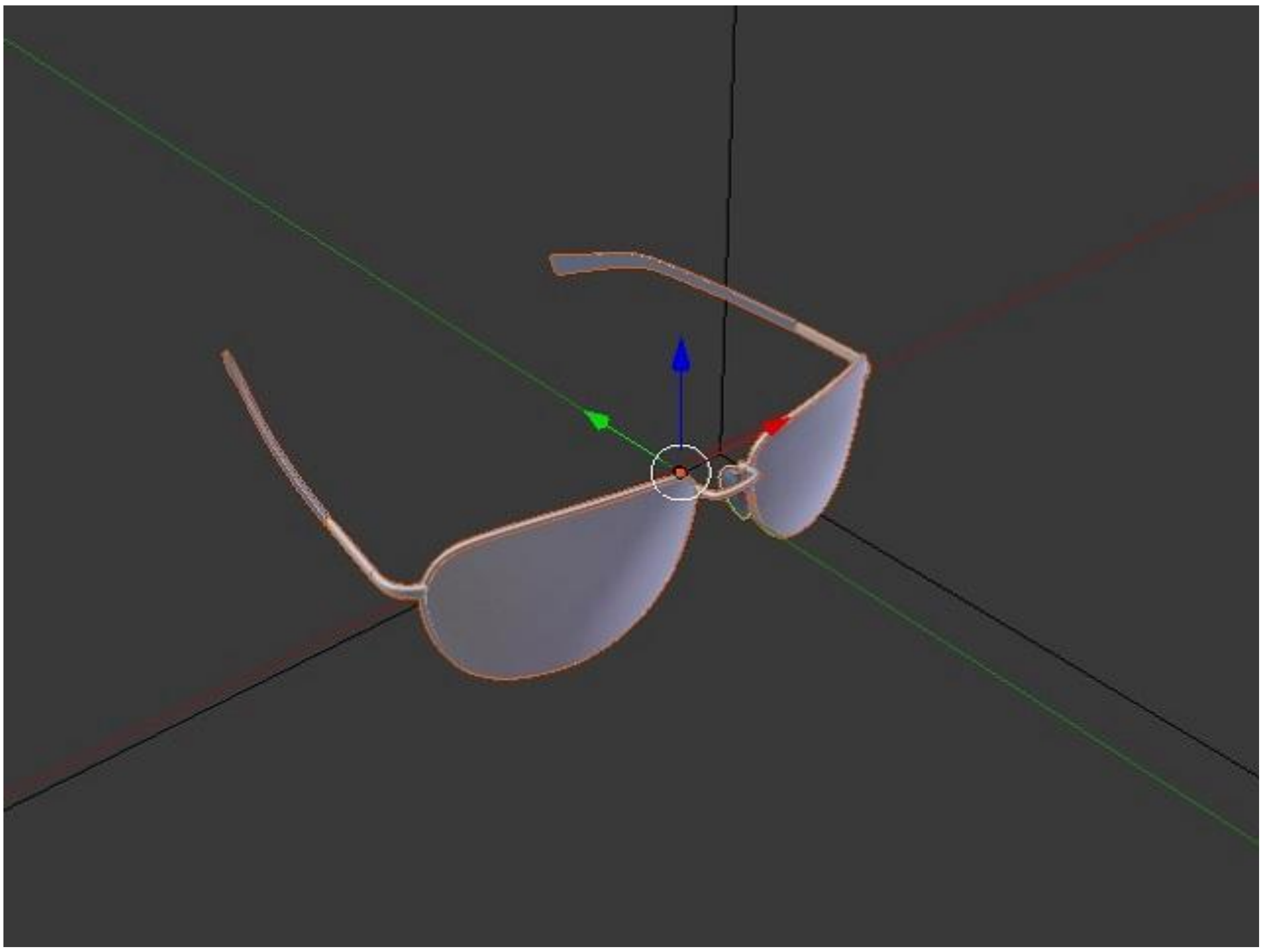

Fig. 1 Size and positioning of the glasses in 3D authoring tool.

3. Occlusion

If occlusions are not handled, glasses would always appear on top of the face like this:

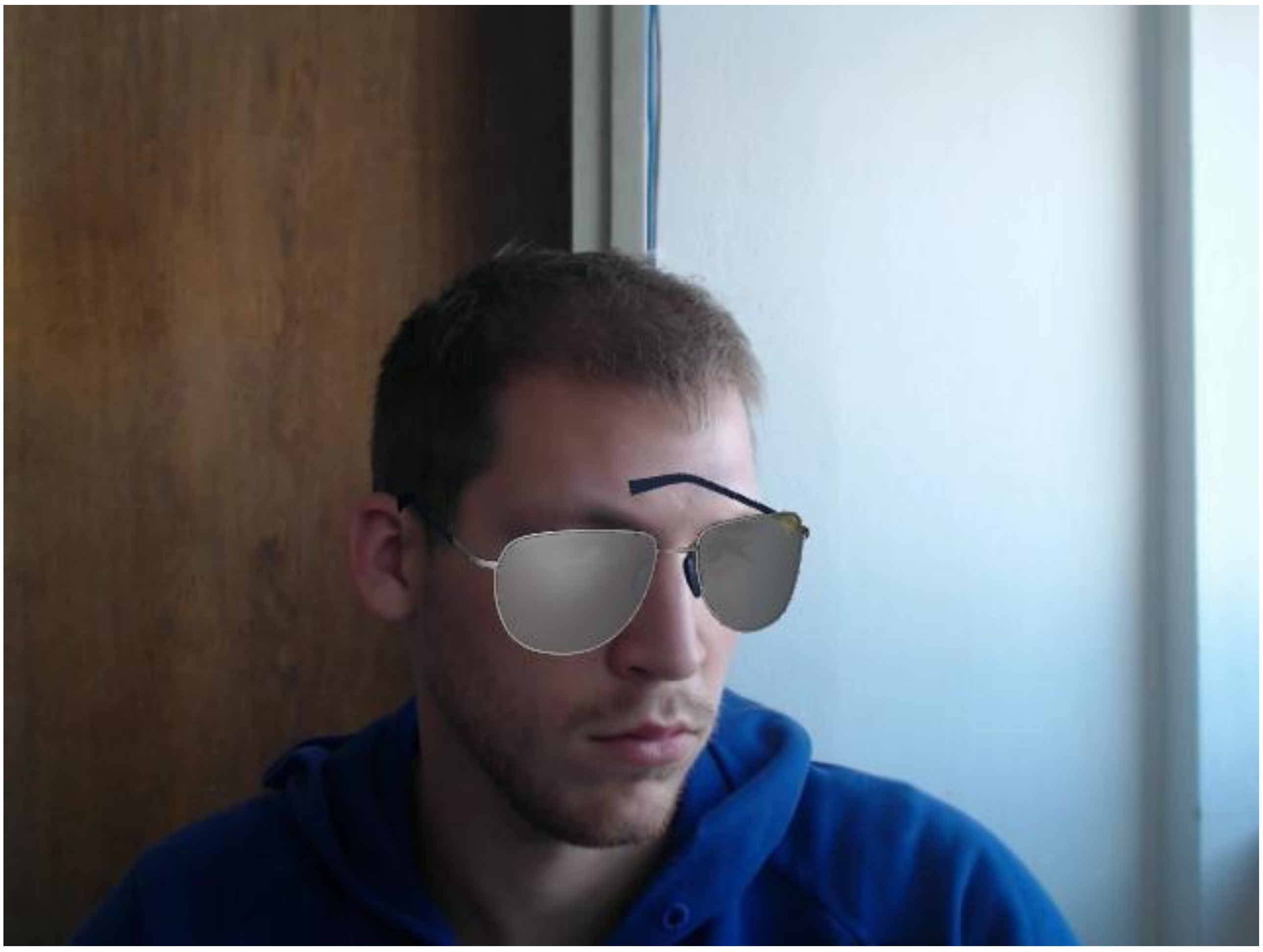

Fig. 2 No occlusion

In reality, parts of the glasses are occluded by the head, nose or ears. To achieve the same effect in augmented reality, an occlusion mask is used. This is an object shaped roughly like a human head and placed inside the glasses. The material of this object is called *occluder_mat*. At runtime, the occlusion mask covers the object behind it but is not itself visible, achieving the desired effect. The next two images show the occlusion mesh highlighted as wireframe (left), and the final effect (right).

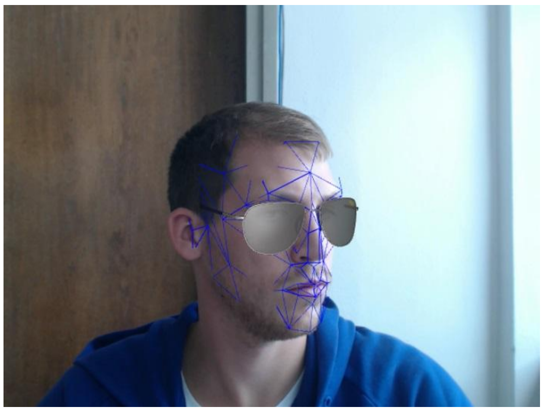

Fig3. - Occlusion final(wireframe)

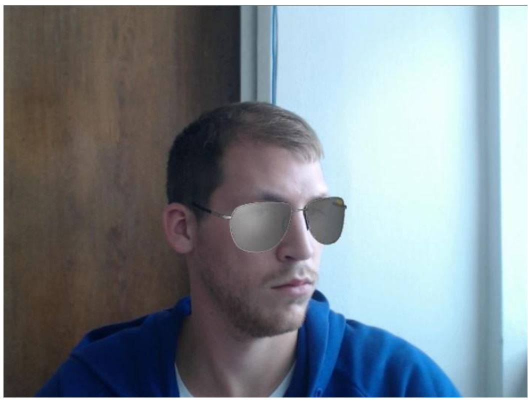

Fig4. - Occlusion final

3.1. Importing the occlusion mask

Occlusion mask is available: models/occlusion_mask.obj. When imported into the scene, the mask should automatically be correctly sized and positioned, as follows:
Real-life size, in meter units.
X positioning: point between the eye centres at X=0
Y positioning: eye centres at Y = 0
Z positioning: eyeball front edge at Z = 0
The occlusion mask must not be moved or scaled.

3.2. Adjusting the occlusion mask

Small adjustments to the mask can be made so it better fits the specific glasses model.

3.2.1. Nose adjustment

Nose can be widened in order to fit the glasses' nose pads as close as possible. Following images illustrate the correct fitting of the nose:

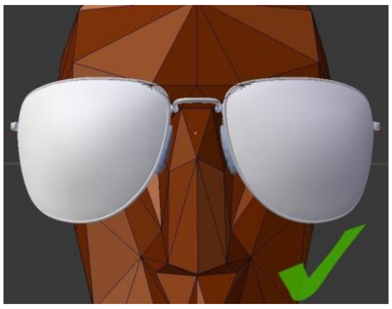
Fig5. - Correct nose adjustments

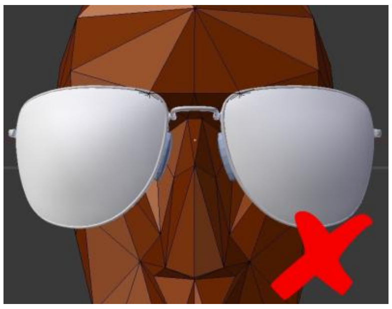
Fig6. - Nose too narrow

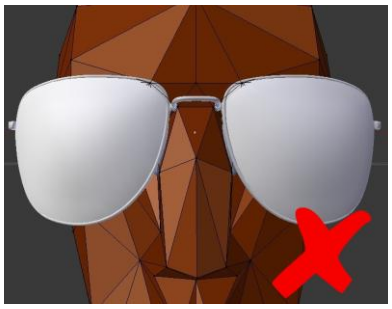
Fig7. - Nose too wide

3.2.2. Head sides

If the occlusion mask is too wide or too narrow, it can be adjusted by moving the sides of the head or adjusting them so that the temples of the glasses fit closely to the mask.

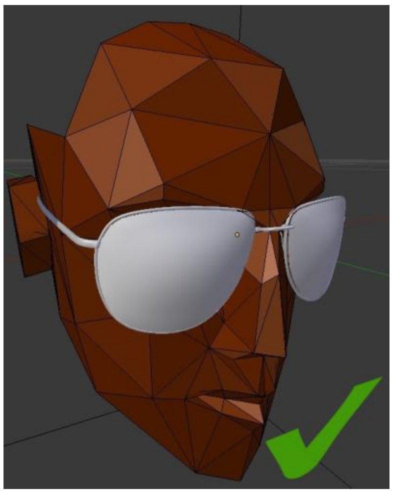
Fig8. - Correct head sides adjustments

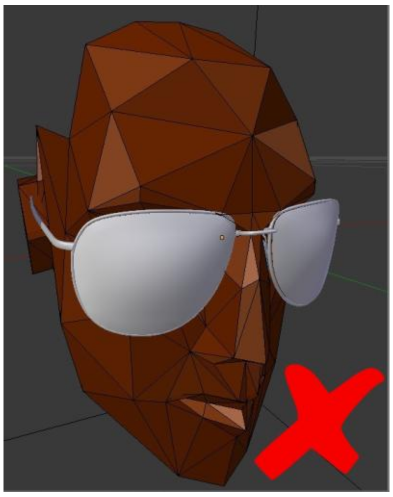
Fig9. - Head sides too narrow

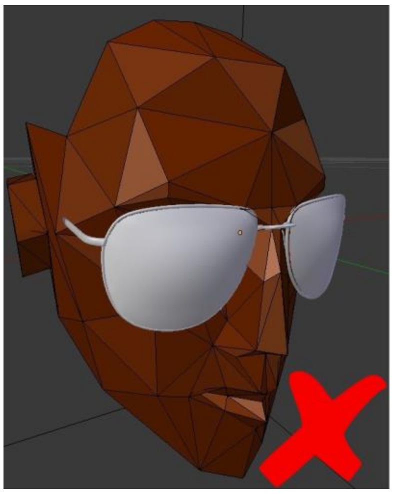
Fig10. - Head sides too wide

3.2.3. Ears

The ears of the occlusion mask should fully cover the back part of the temples.

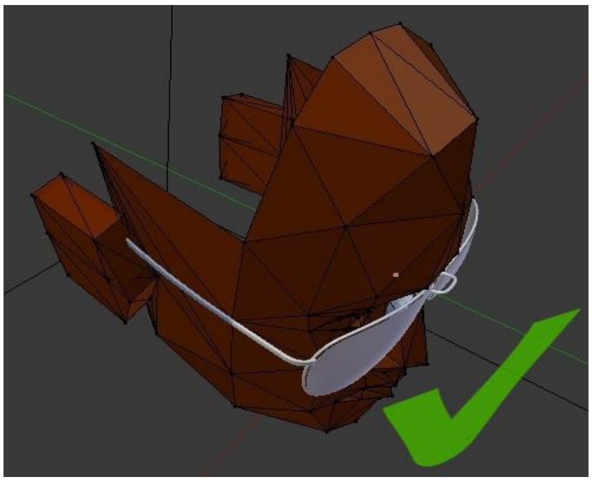
Fig11. - Correct ears adjustments

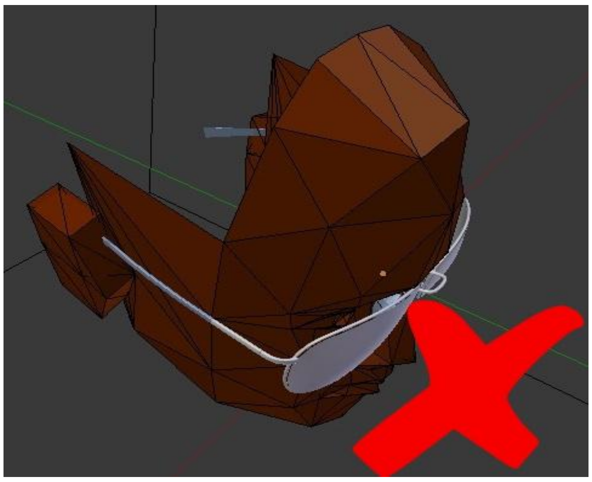
Fig12. - Temples not completely occluded

4. Importing the model into the final AR application

This section covers issues of file formats and conversions. The two subsections cover the two 3D platforms typically used to build final AR applications using visage|SDK - Three.js and Unity3D.

4.1. Three.js (used in visage|SDK for HTML5)

Models need to be in OBJ format with a corresponding MTL material file. Most 3D authoring tools provide a way to export an object to OBJ and MTL.

Resulting OBJ file must contain both the glasses and the occlusion mask. Occlusion mask mesh should be grouped under *occluder_mat* name. For example, in Blender this is done by naming the material of the mesh *occluder_mat* and exporting to OBJ with setting *Material Groups* checked.

4.1.1. Editing the MTL file

Materials are defined in the MTL file, usually generated from the materials in the authoring tool. After exporting advanced users can do fine tuning by editing the file. For syntax reference see here. Textures used in the MTL file should be placed in the path relative to the MTL folder. Supported texture formats include JPG, TGA and PNG.

4.2. Unity3D (used in visage|SDK for Windows, iOS, Android)

Models are typically built in a 3D authoring tool such as 3ds Max or Blender, then imported into into Unity3D. See a tutorial here for a list of supported formats and details regarding the import process. After importing process is complete **check that the size and positioning of the model and the occlusion mask are consistent with the instructions in sections 2 and 3.1.**

Unity3D package contains an occlusion shader that needs to be assigned to the *occluder_mask* object. Occlusion shader is available in the unity scene in: Materials/. Unity3D package for following versions can be found:

- Windows - Samples\OpenGL\data\VisageTrackerUnity\VisageTrackerUnity.unitypackage
- Android - Samples\Android\VisageTrackerUnityDemo\VisageTrackerUnityDemo.unitypackage
- iOS - Samples\iOS\VisageTrackerUnityDemo\VisageTrackerUnityDemo.unitypackage

Note that most materials will need to be recreated because Unity uses its own shaders.