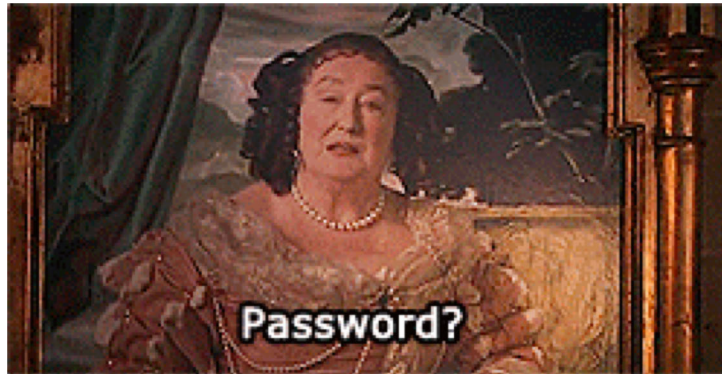


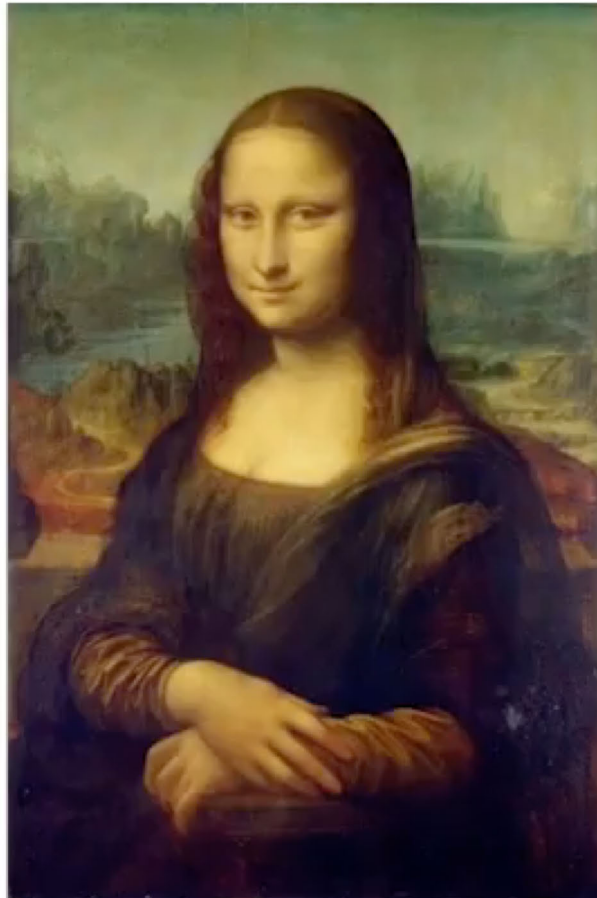
# Bringing Portraits to Life

CS448V: Lecture 13

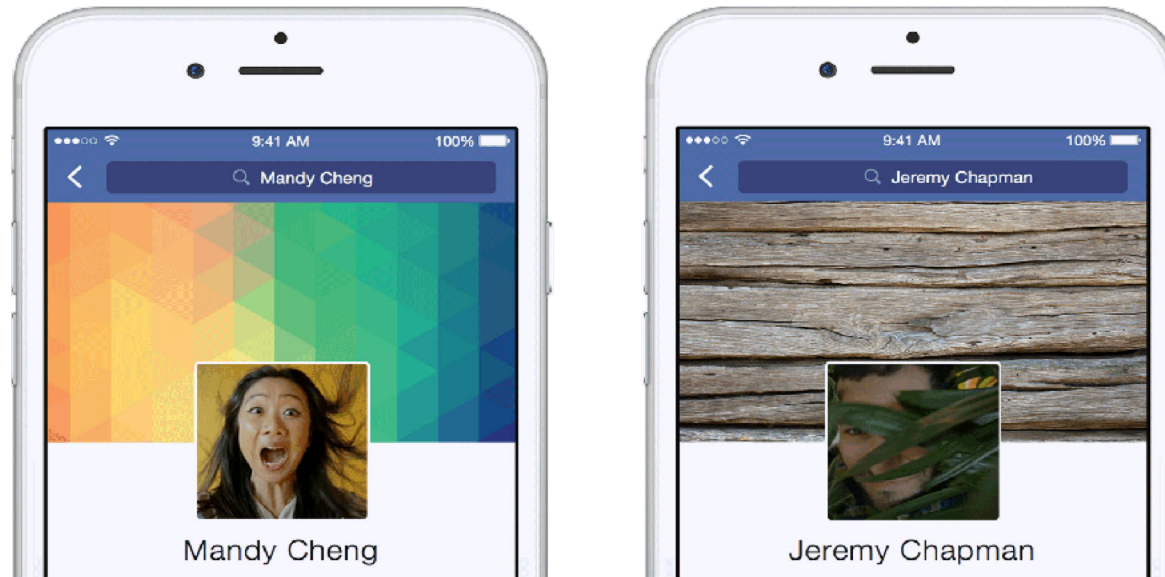
# Motivation



# Motivation



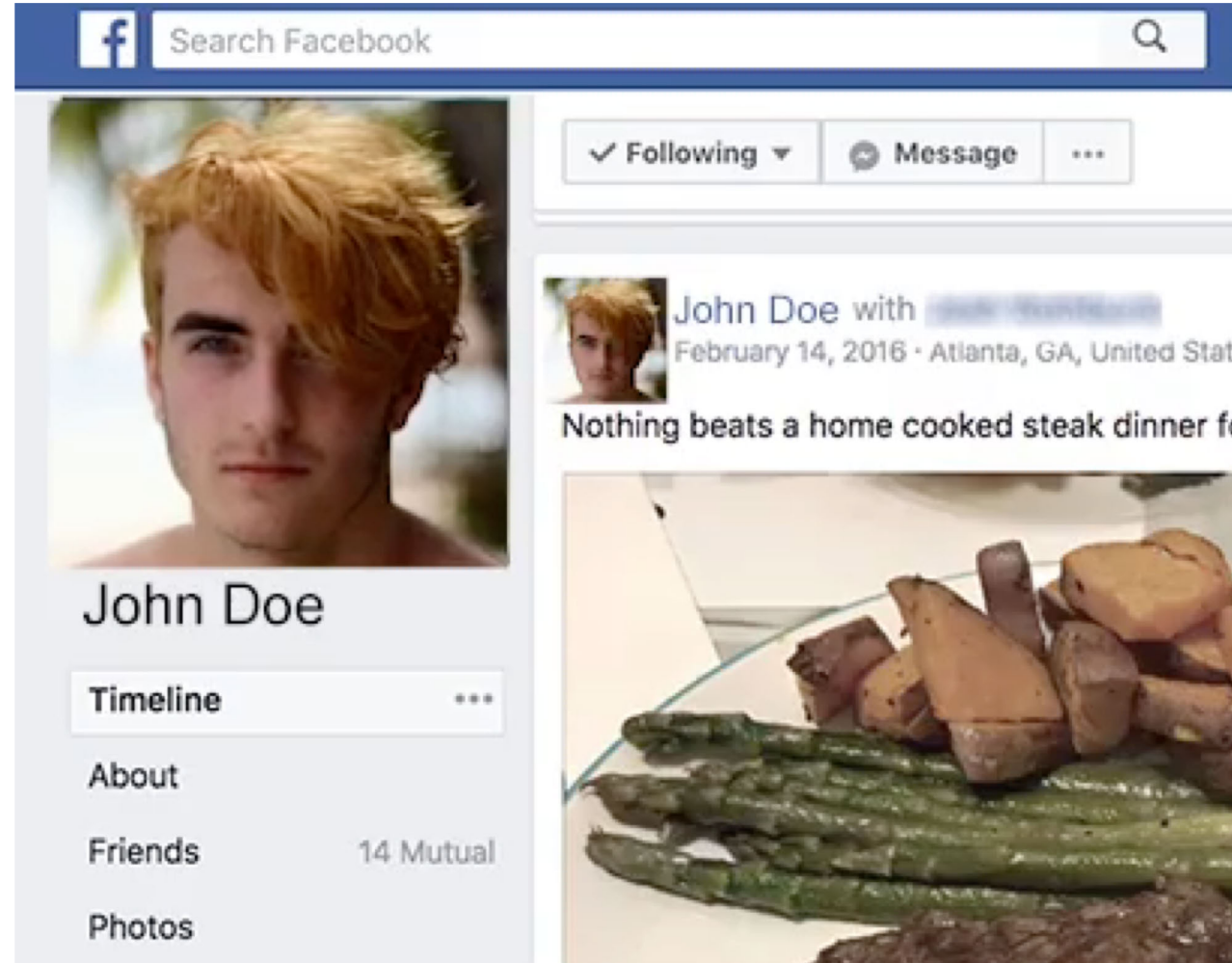
# Motivation



**Bring Your Profile to Life**  
Facebook (2015)

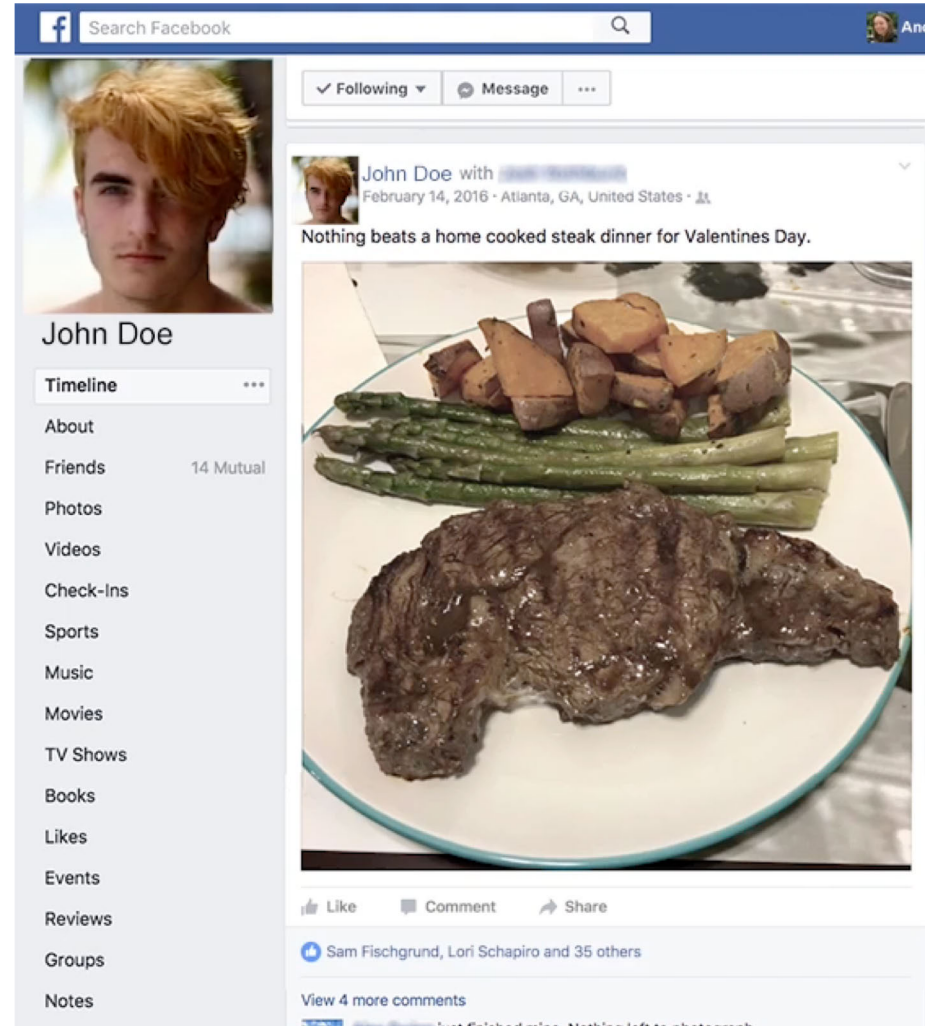


# Motivation



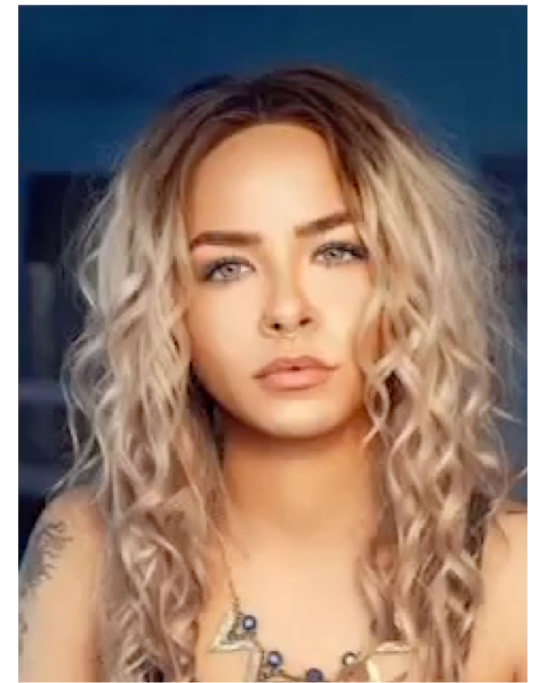
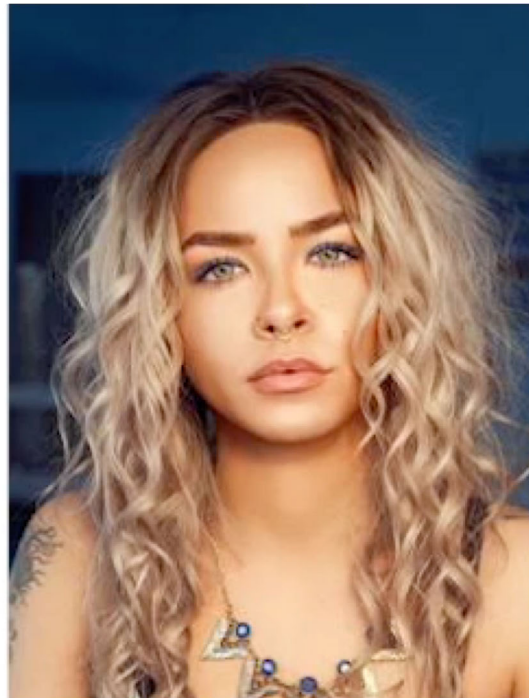
**Breathing Profile**

# Motivation



**Reactive Profile**

# Approach



**Driving Video** ( $S = \{s_0, s_1, \dots\}$ )

**Target Image** ( $t^*$ )

**Output Video** ( $T = \{t_0, t_1, \dots\}$ )

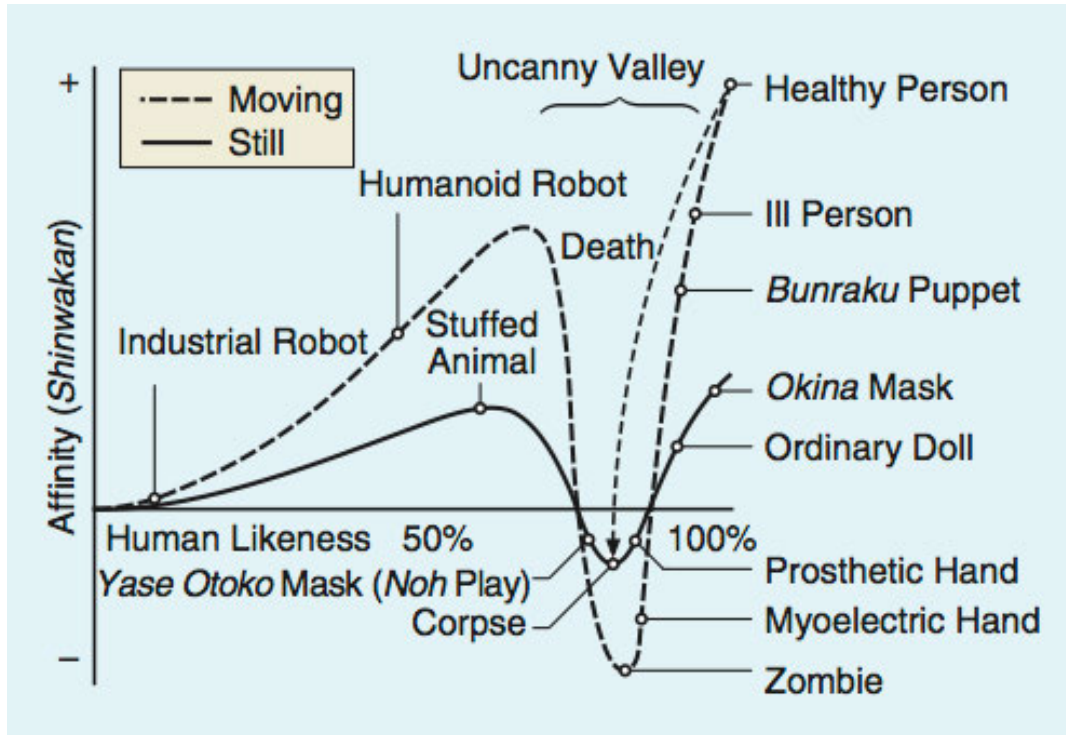


# A Challenging Problem

- Uncanny Valley



<https://www.facebook.com/pam.richardcoones/posts/10103240387205162>



Mori (1970)

ADVANCED CAPABILITIES	
<p><b>NAO</b> is a little French humanoid created by Aldebaran Robotics. It talks, tracks faces, and with 25 degrees of freedom, it can even perform Michael Jackson choreography.</p>	<p><b>NEXI</b> is the size of a 3-year-old child and was conceived at MIT's Media Lab. With a 15-degree-of-freedom face, this little social bot can show you when it's happy, sad, mad—or bored.</p>
<p><b>M3-NEONY</b> is an Osaka University offspring. The robot has 90 tactile sensors, 22 motors, 2 cameras, a compact computer, and a fancy name. M3 stands for "man-made man."</p>	<p><b>ICUB</b> has multiple parents—11 European robotics labs. The size of a 3½-year-old, it's learning to walk, talk, and handle objects. It's probably the most advanced—and expensive—artificial child ever built.</p>
<p><b>SIMON</b> is the child of Georgia Tech researchers. This social robot has an expressive face, articulated torso, dexterous hands, and supercute ears.</p>	<p><b>DIEGO-SAN</b> is the progeny of roboticists at the University of California, San Diego, and the Japanese firm Kokoro Co. They're teaching it to walk and hold objects—and they're also designing a smaller head.</p>
<p><b>ZENO</b> is a small, cartoon-like humanoid designed at Hanson Robotics, in Richardson, Texas. It talks, understands speech, and can learn names and faces. It also has big green eyes that look as if they're ready to shoot lasers.</p>	<p><b>CB2</b> was born at Osaka University, in Japan. This robot mimics how infants learn by interacting with the world. And with 51 pneumatic actuators, it's powered by air.</p>
<p><b>YOTARO</b> is a baby robot that giggles, cries, and simulates a runny nose—but not a soiled diaper. Researchers at the University of Tsukuba, in Japan, designed it to show how rewarding babies can be. Really.</p>	<p><b>REPLIEE R1</b> is a copy of a real 4-year-old girl. Built at Osaka University, it has nine DC motors in its head, prosthetic eyeballs, and silicone skin. Opinions on how cute it is are mixed.</p>
<p><b>ROBOTINHO</b> is the mechanical child of engineers at the University of Bonn, in Germany. They use the robot as a museum guide—and also to play robot soccer.</p>	<p><b>REALCARE BABY</b> is a lifelike doll by Realityworks, in Eau Claire, Wis. You have to feed, burp, rock, and diaper the cyberinfant around the clock or it will cry—a real, recorded baby cry.</p>

ROBOTIC APPEARANCE

HUMAN APPEARANCE

LIMITED CAPABILITIES

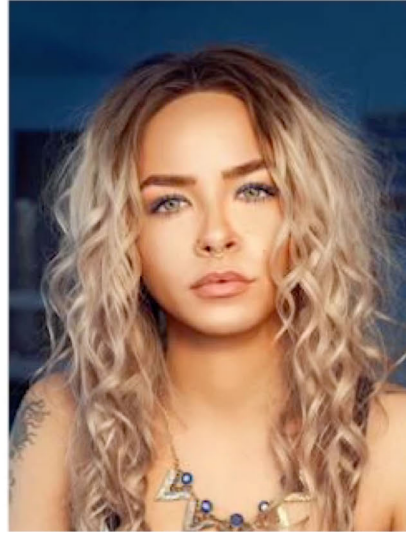
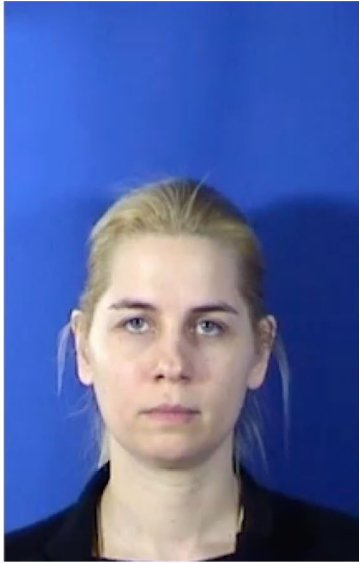
<https://spectrum.ieee.org/automaton/robotics/humanoids/invasion-of-the-robot-babies-infographic>



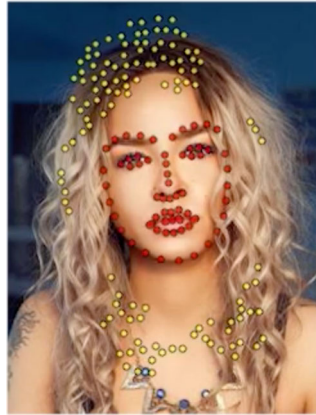
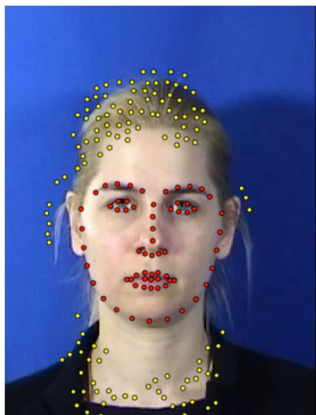
# Assumptions

- Front facing faces
- Target image is a neutral face
- Driving video includes an instance of a neutral face ( $s^*$ )

# Pipeline



**Driving Video ( $S$ )** **Target Image ( $t^*$ )**



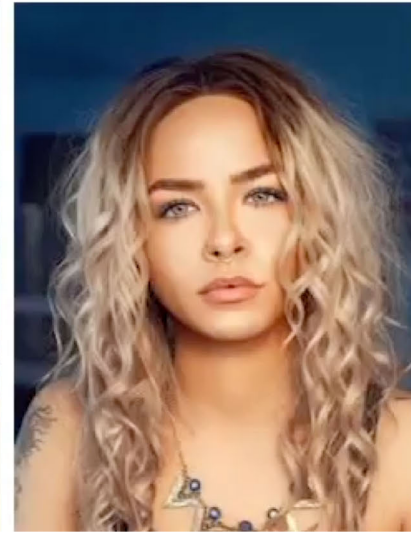
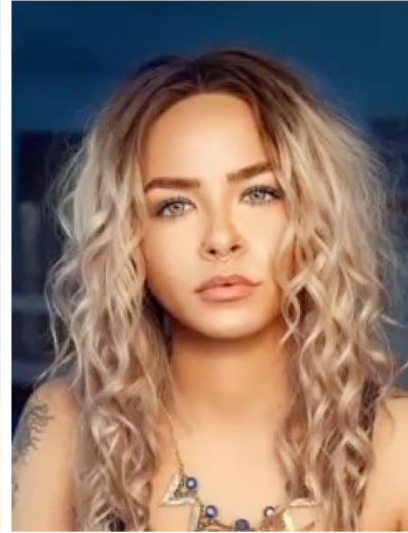
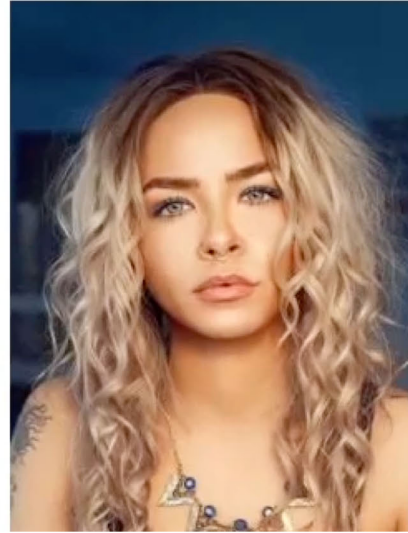
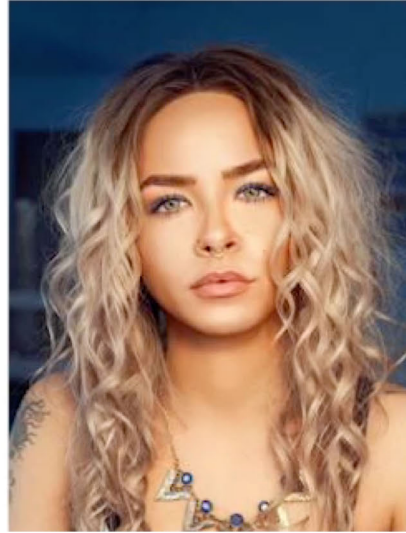
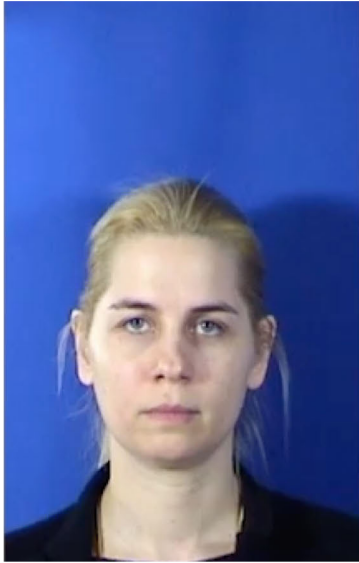
**Step 1: Feature Correspondence**

**Step 2:  
Coarse Target  
Video Synthesis**

**Step 3:  
Transferring  
Hidden Regions**

**Step 4:  
Transferring  
Fine-Scale Details**

# Pipeline



**Driving Video ( $S$ ) Target Image ( $t^*$ )**

**Step 2:  
Coarse Target  
Video Synthesis**

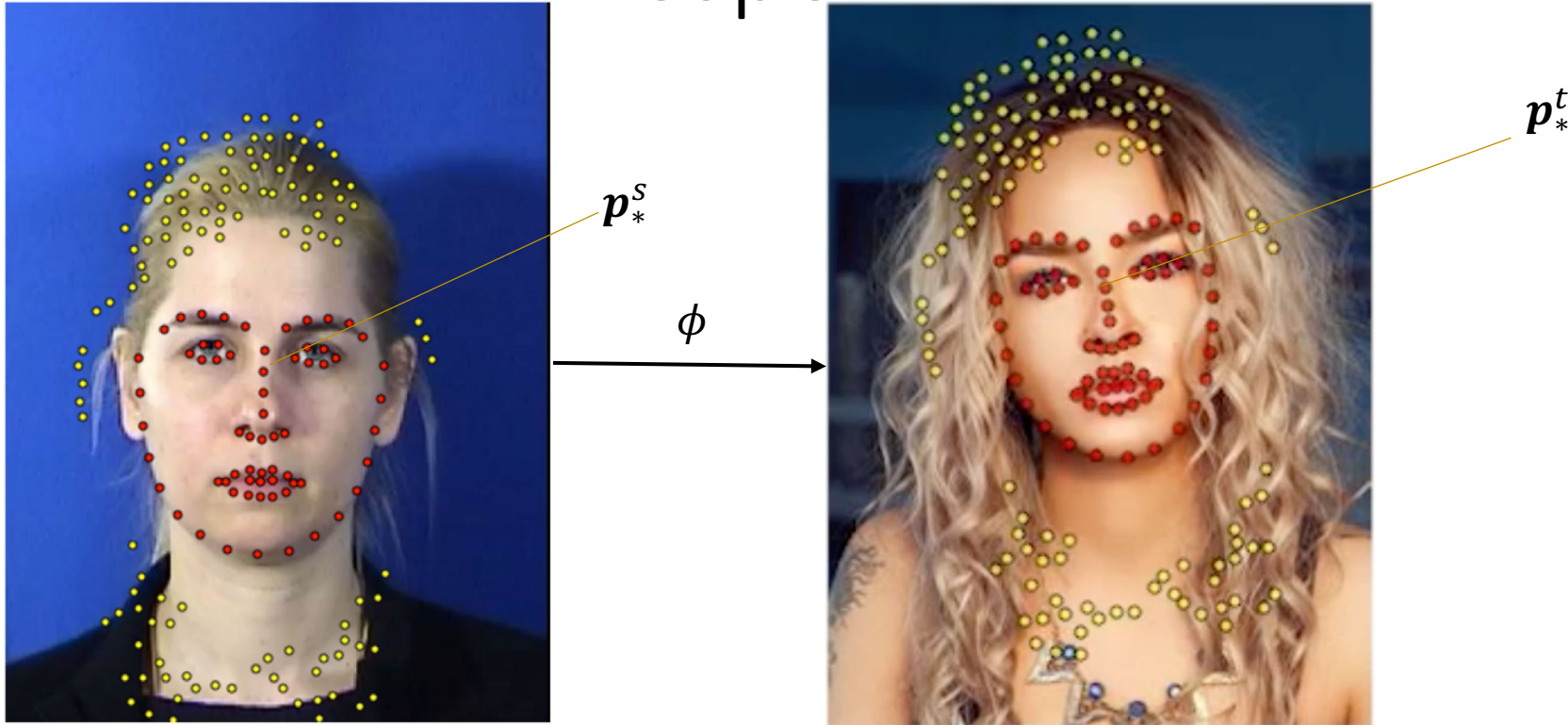
**Step 3:  
Transferring  
Hidden Regions**

**Step 4:  
Transferring  
Fine-Scale Details**



**Step 1: Feature Correspondence**

# Step 1: Feature Correspondence



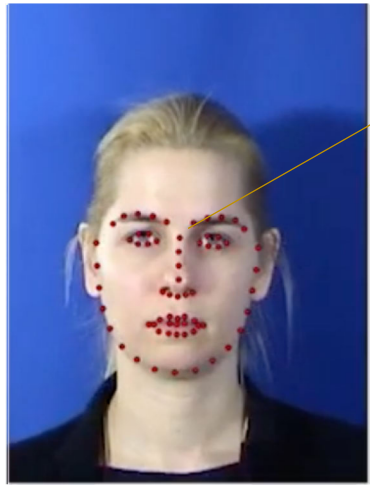
Neutral Video Frame ( $s^*$ )

Target image ( $t^*$ )

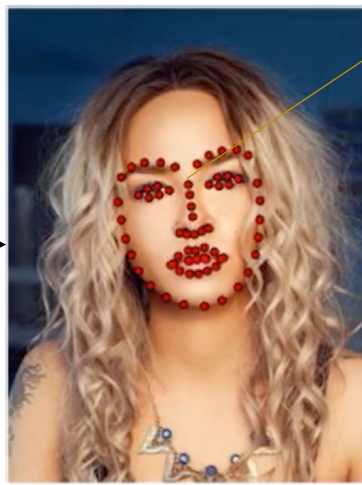
- 68 facial landmarks for facial region (red)
- Peripheral points outside facial region (yellow)



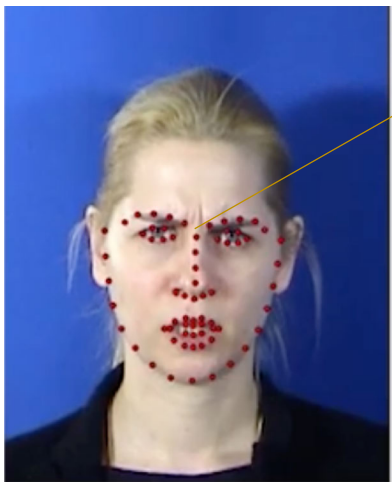
# Step 1: Feature Correspondence



Neutral Video Frame ( $s^*$ )



Target image ( $t^*$ )



Driving Video Frame  $i$  ( $s_i$ )

$p_*^s$

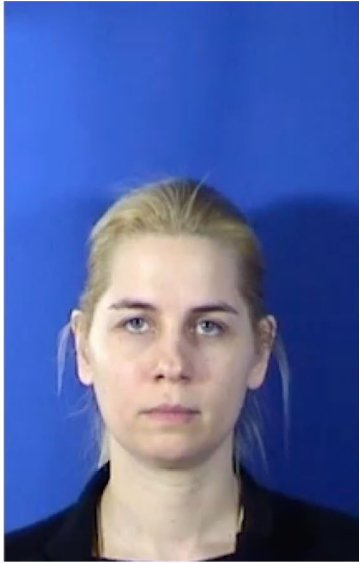
$\phi$

$p_*^t$

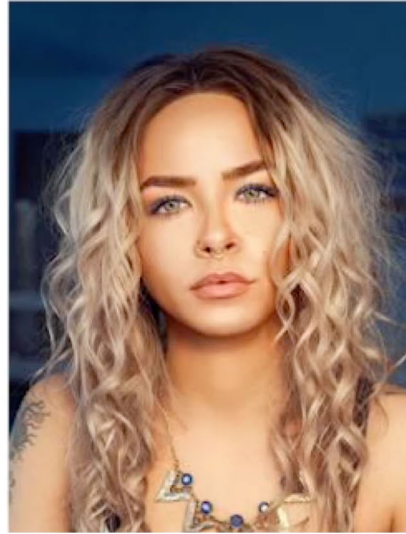
$p_i^s$

- Facial landmark detection
- $\phi$ : best *similarity transform* between control points in driving video and target image
- How do we handle regions outside the face?
  - Peripheral points!
- On neutral video frame:
  - Feature point detection
- On other frames in driving video:
  - Optical flow
- On target image:
  - Map peripheral points with  $\phi$

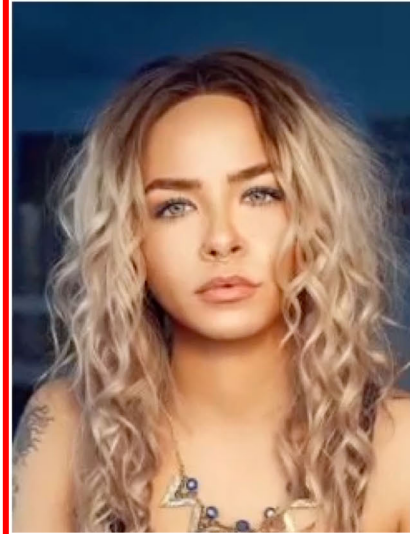
# Pipeline



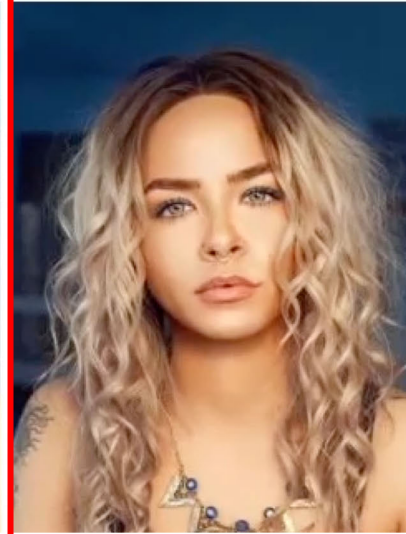
**Driving Video ( $S$ )**



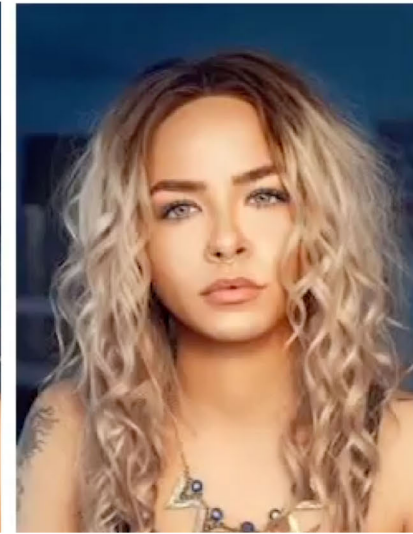
**Target Image ( $t^*$ )**



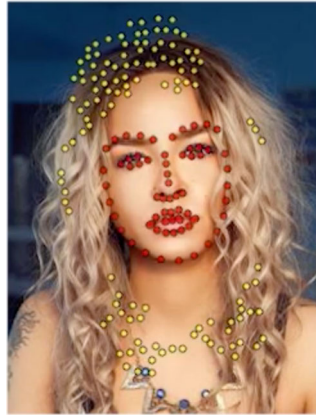
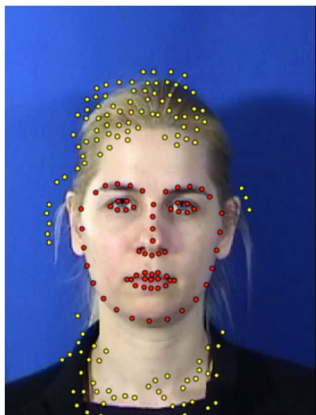
**Step 2:  
Coarse Target  
Video Synthesis**



**Step 3:  
Transferring  
Hidden Regions**

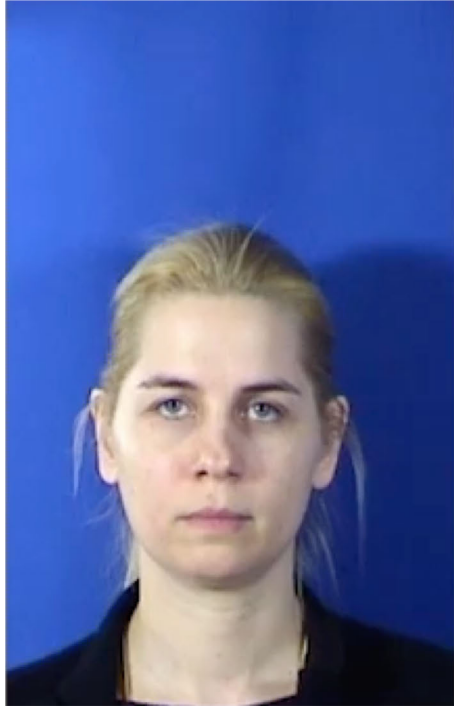


**Step 4:  
Transferring  
Fine-Scale Details**

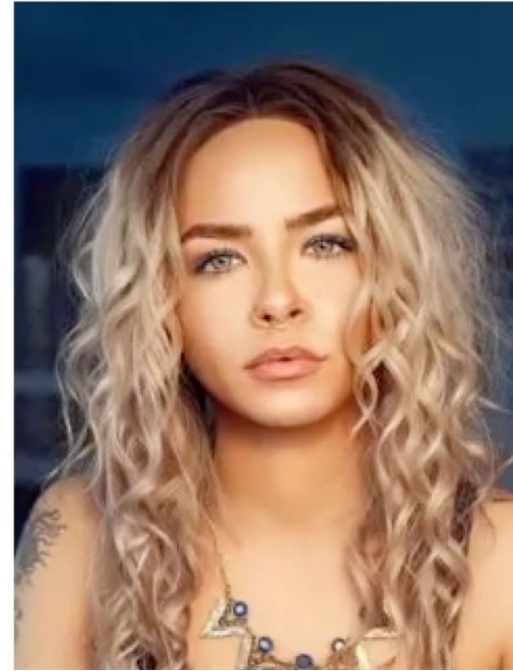


**Step 1: Feature Correspondence**

## Step 2: Coarse Target Video Synthesis



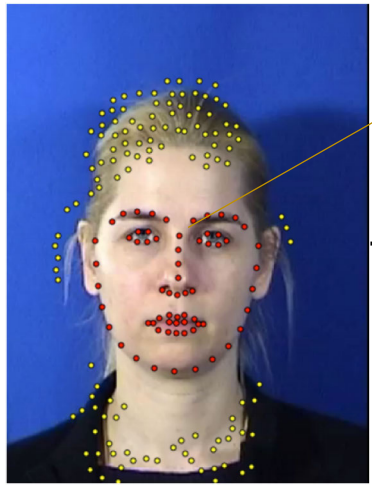
**Driving Video**



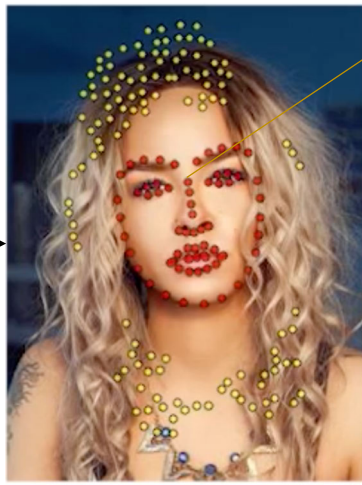
**Step 2 Result**

- “Confidence-aware warping”
  - Interpolate warp fields + smooth warping in regions outside the face

# Step 2: Coarse Target Video Synthesis



Neutral Video Frame ( $s^*$ )



Target image ( $t^*$ )

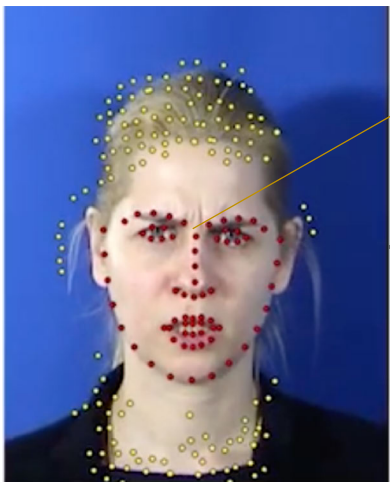
$p_*^s$

$\phi$

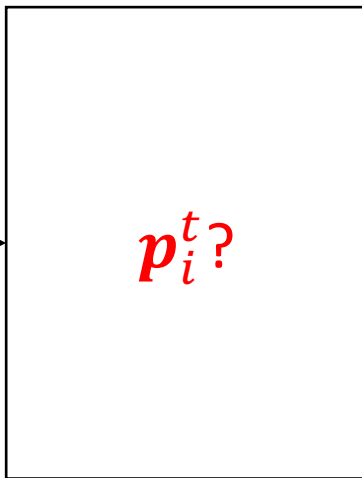
$p_*^t$

- $\phi$ : mapping between control points in driving video and target image/resulting video

- $p_*^t = \phi \cdot p_*^s, p_i^t = \phi \cdot p_i^s$



Driving Video Frame  $i$  ( $s_i$ )



Result Video Frame  $i$  ( $t_i$ )

$p_i^s$

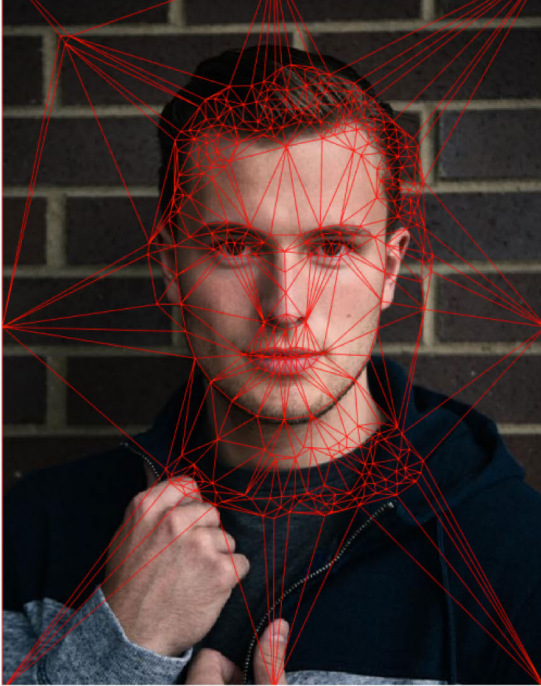
$\phi$

$p_i^t?$

$$p_i^t = p_*^t - \phi \cdot (p_i^s - p_*^s)$$

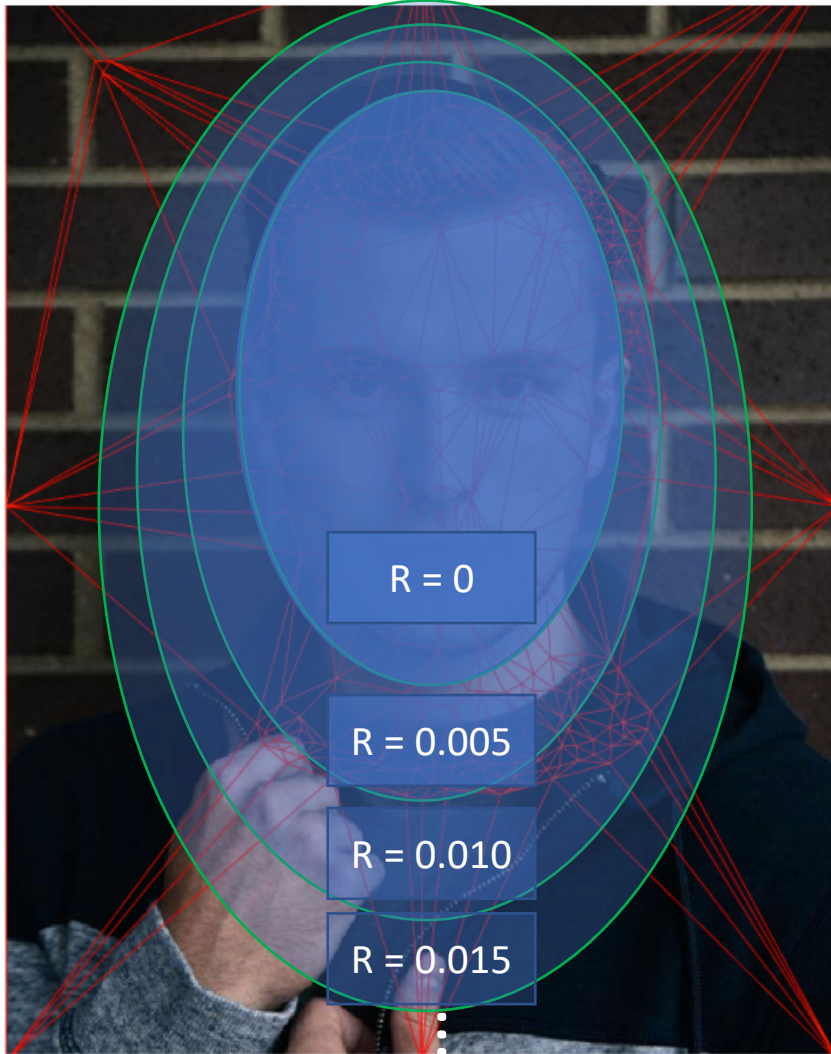


## Step 2: Coarse Target Video Synthesis



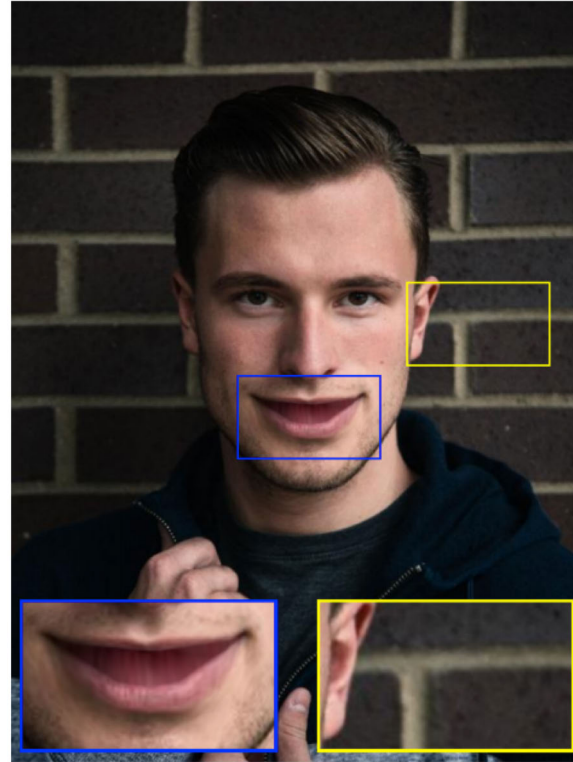
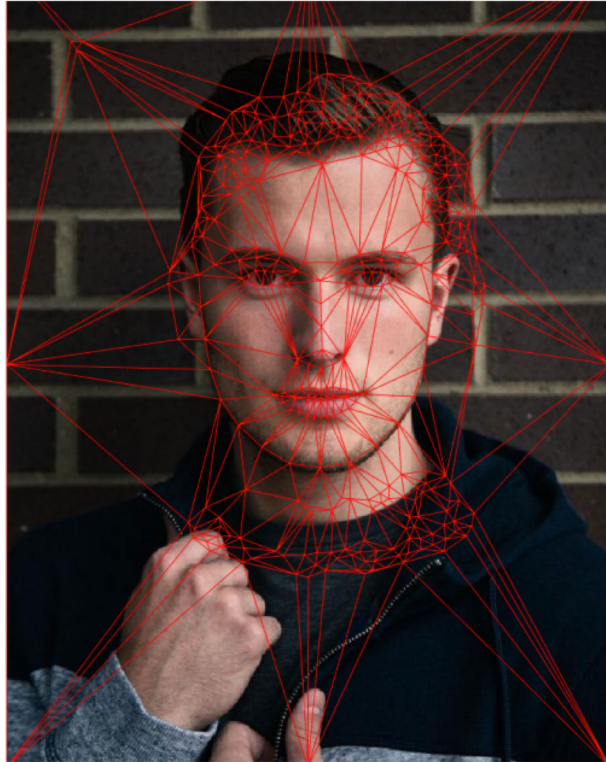
- Linear interpolation of warp field using Delauney triangulation (HW2)
- But...
  - Discontinuities outside the facial region
- We can apply uniform smoothing to the entire warp field, but...
  - Removes facial expression

# Step 2: Coarse Target Video Synthesis



- Smooth w.r.t. confidence
- No smoothing within face (high confidence)
- The farther away from face, the greater the radius of blurring kernel (lower confidence)
  - 10 discrete blur radii

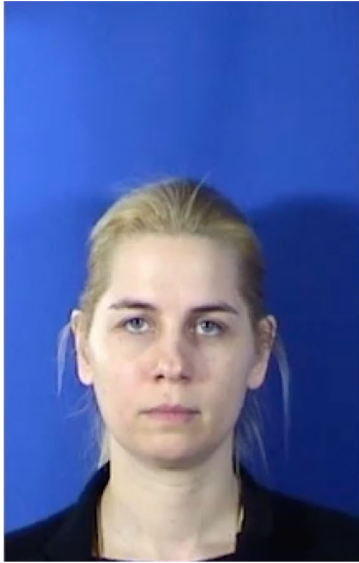
## Step 2: Coarse Target Video Synthesis



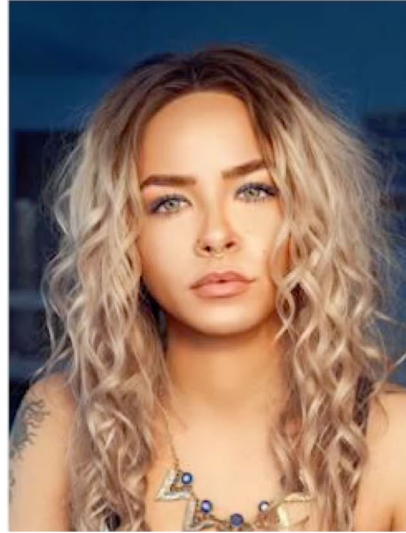
- What is missing and why?
  - Inside the mouth!



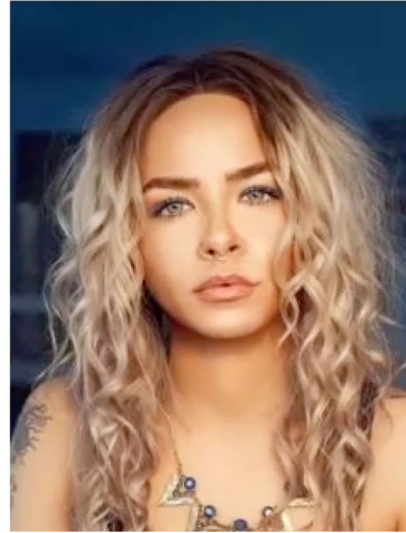
# Pipeline



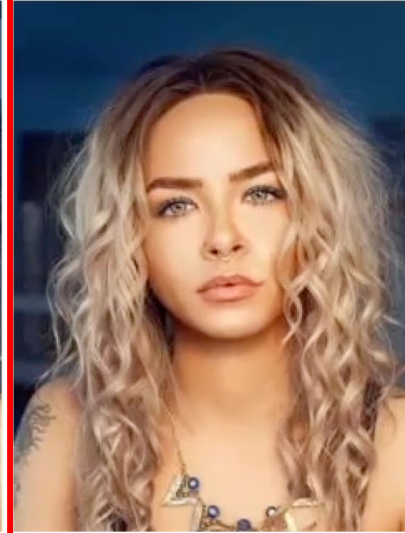
**Driving Video ( $S$ )**



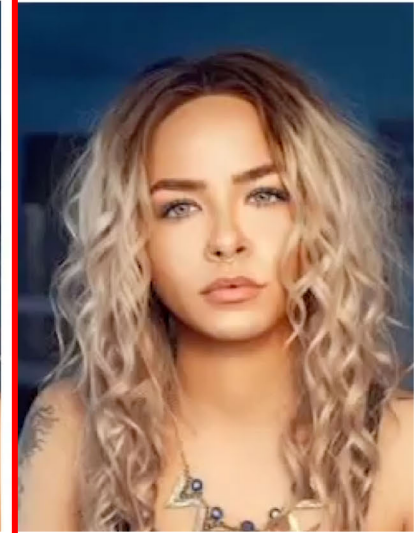
**Target Image ( $t^*$ )**



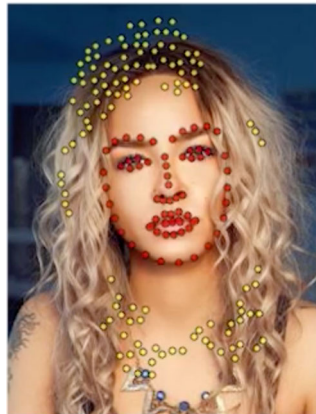
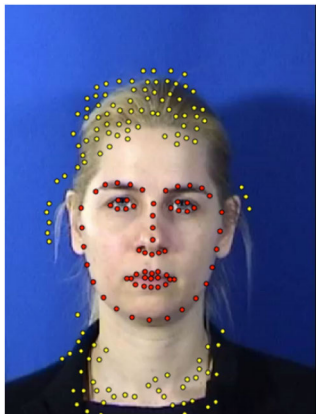
**Step 2:  
Coarse Target  
Video Synthesis**



**Step 3:  
Transferring  
Hidden Regions**



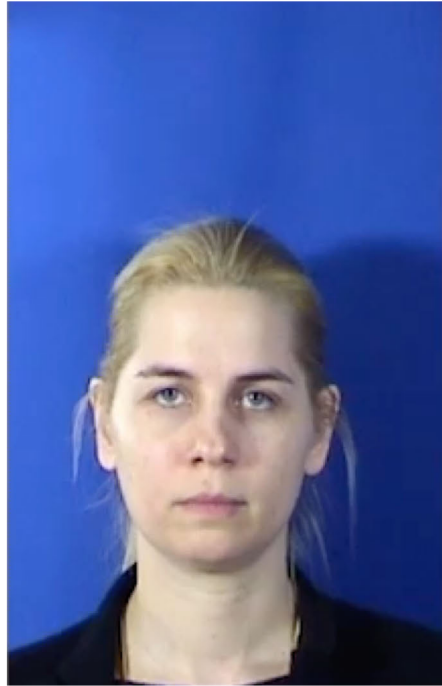
**Step 4:  
Transferring  
Fine-Scale Details**



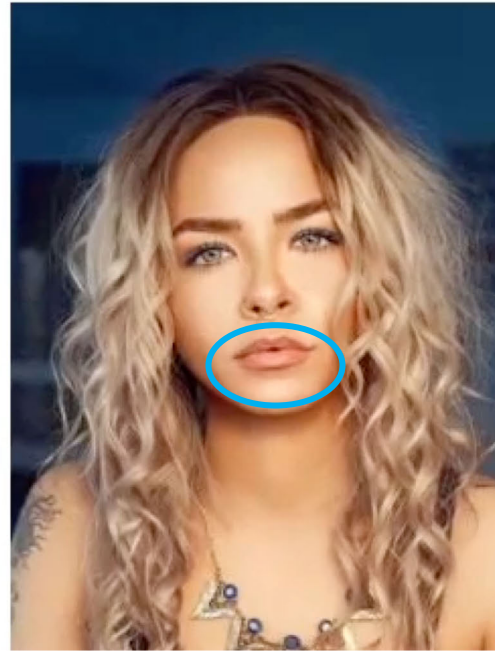
**Step 1: Feature Correspondence**



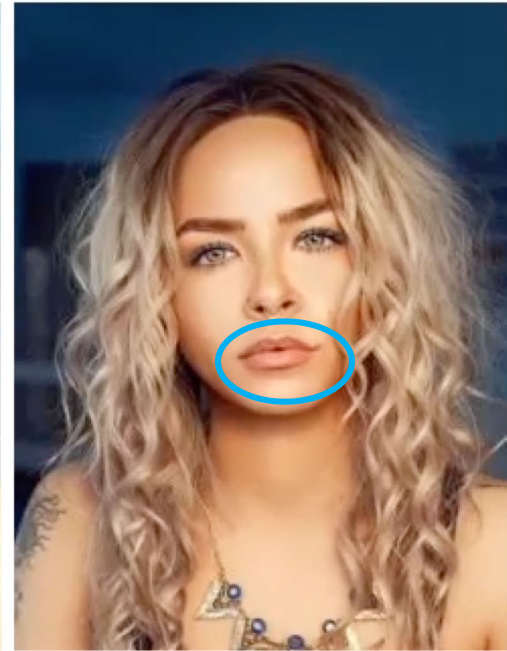
# Step 3: Transferring Hidden Regions



**Driving Video**



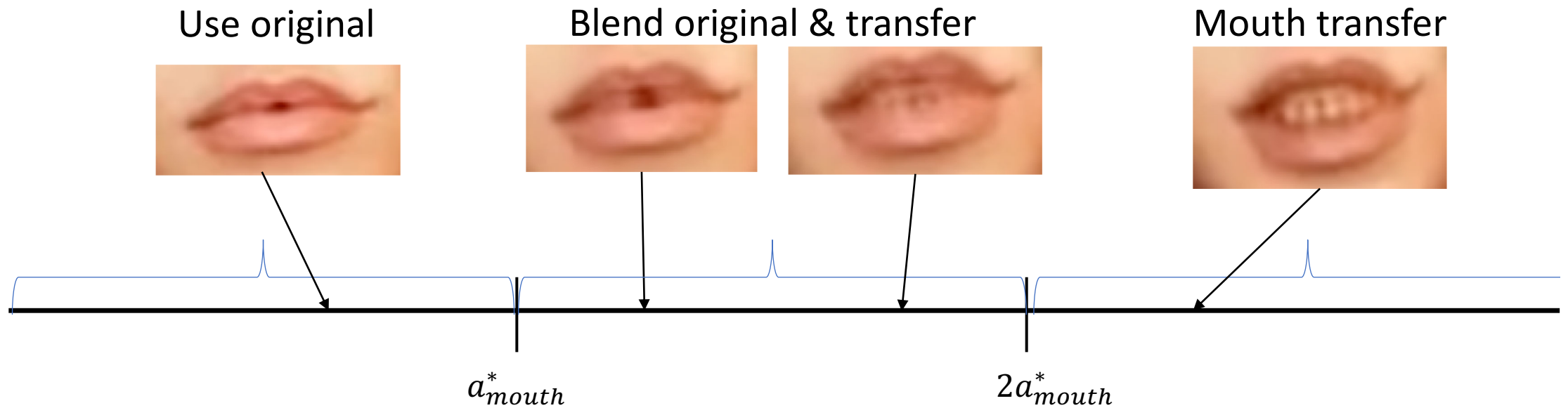
**Step 2 Result**



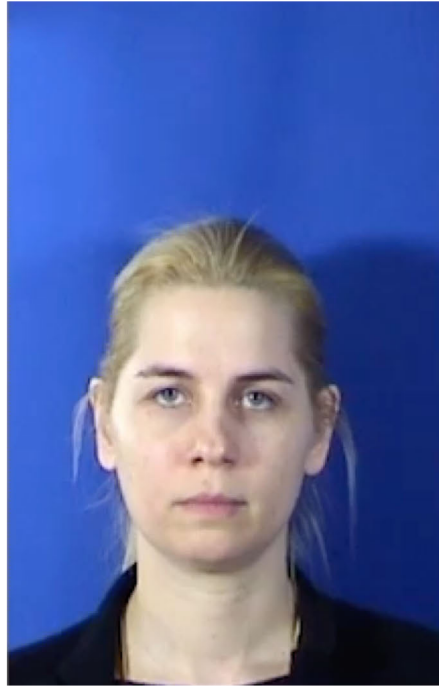
**Step 3 Result**

- Transfer mouth region from driving video to fill in the mouth

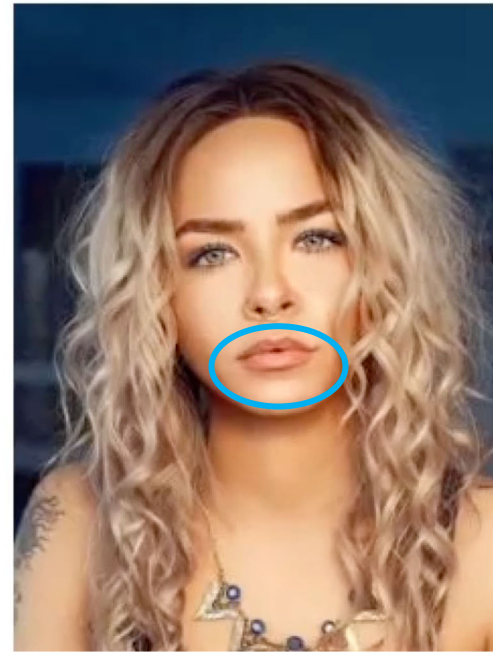
# Step 3: Transferring Hidden Regions



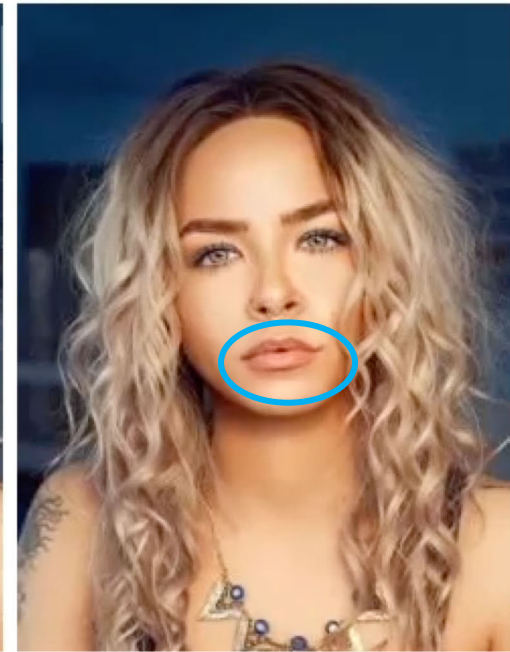
# Step 3: Transferring Hidden Regions



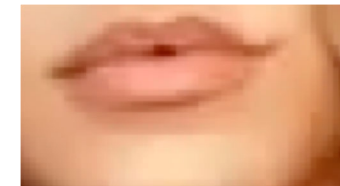
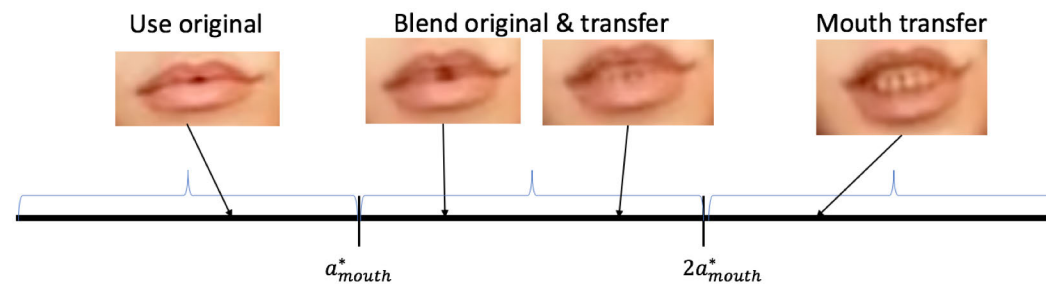
Driving Video



Step 2 Result

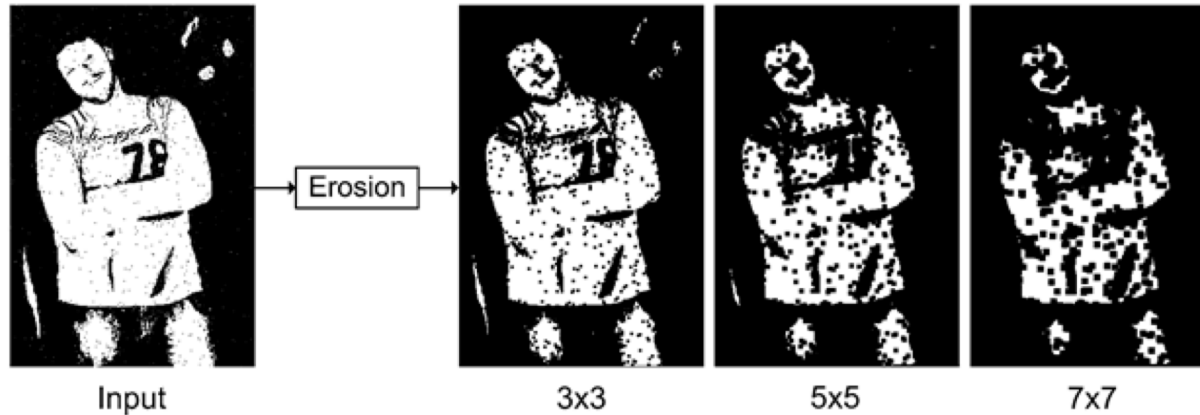


Step 3 Result



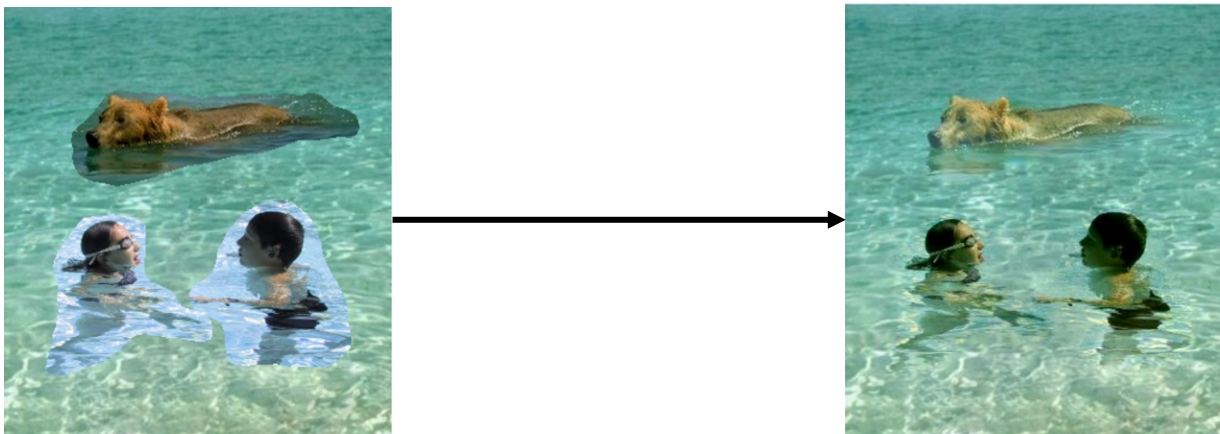
# Step 3: Transferring Hidden Regions

- **Morphological erosion**: be conservative about the pixels to replace (want to keep lips)



<http://what-when-how.com/introduction-to-video-and-image-processing/morphology-introduction-to-video-and-image-processing-part-2/>

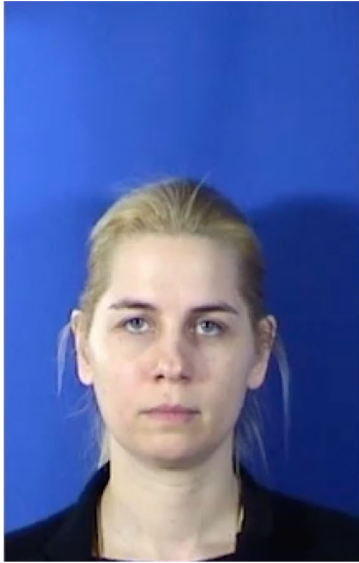
- **Poisson blending**: satisfy boundary while preserving patch structure



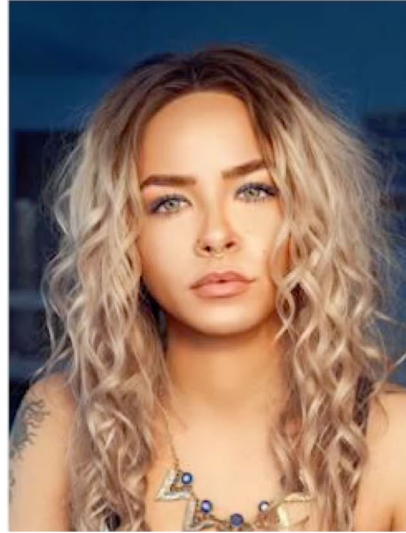
<http://eric-yuan.me/poisson-blending/>



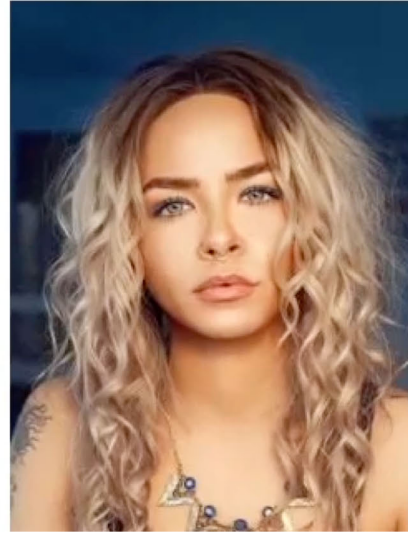
# Pipeline



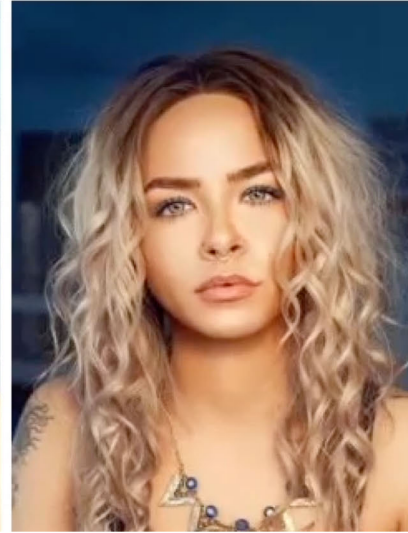
**Driving Video ( $S$ )**



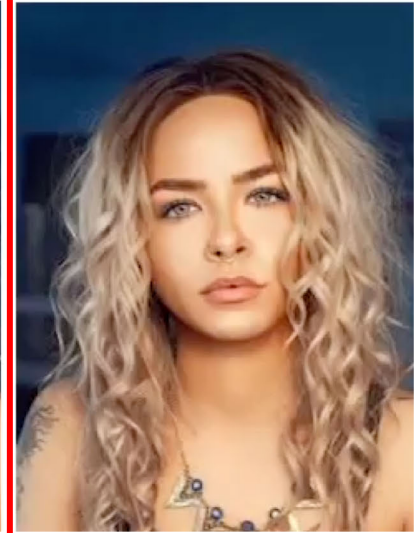
**Target Image ( $t^*$ )**



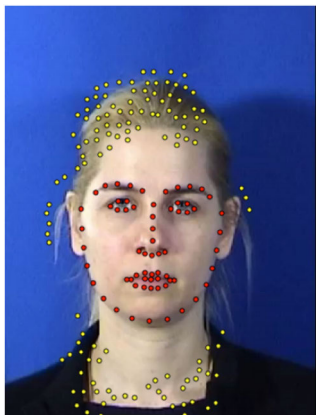
**Step 2:  
Coarse Target  
Video Synthesis**



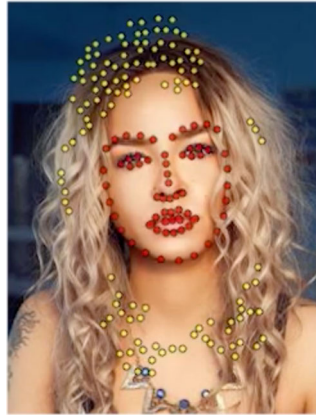
**Step 3:  
Transferring  
Hidden Regions**



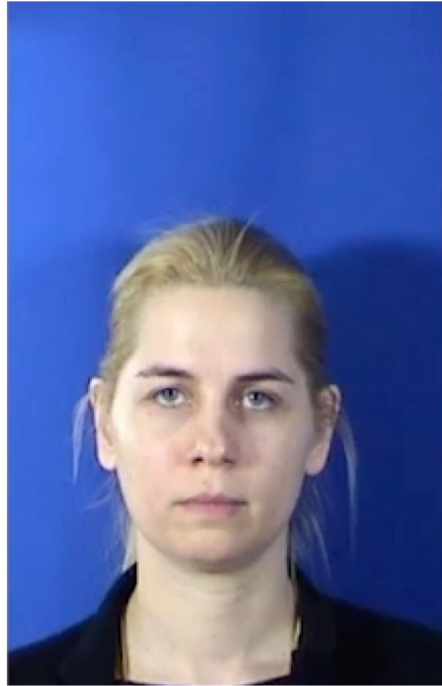
**Step 4:  
Transferring  
Fine-Scale Details**



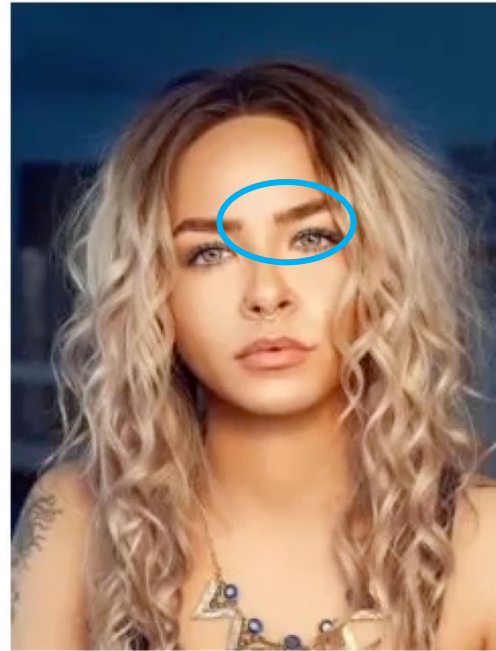
**Step 1: Feature Correspondence**



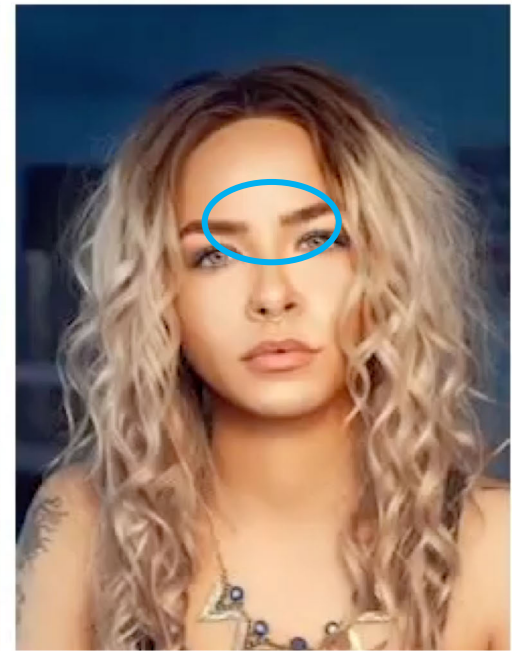
# Step 4: Transferring Fine-Scale Details



**Driving Video**



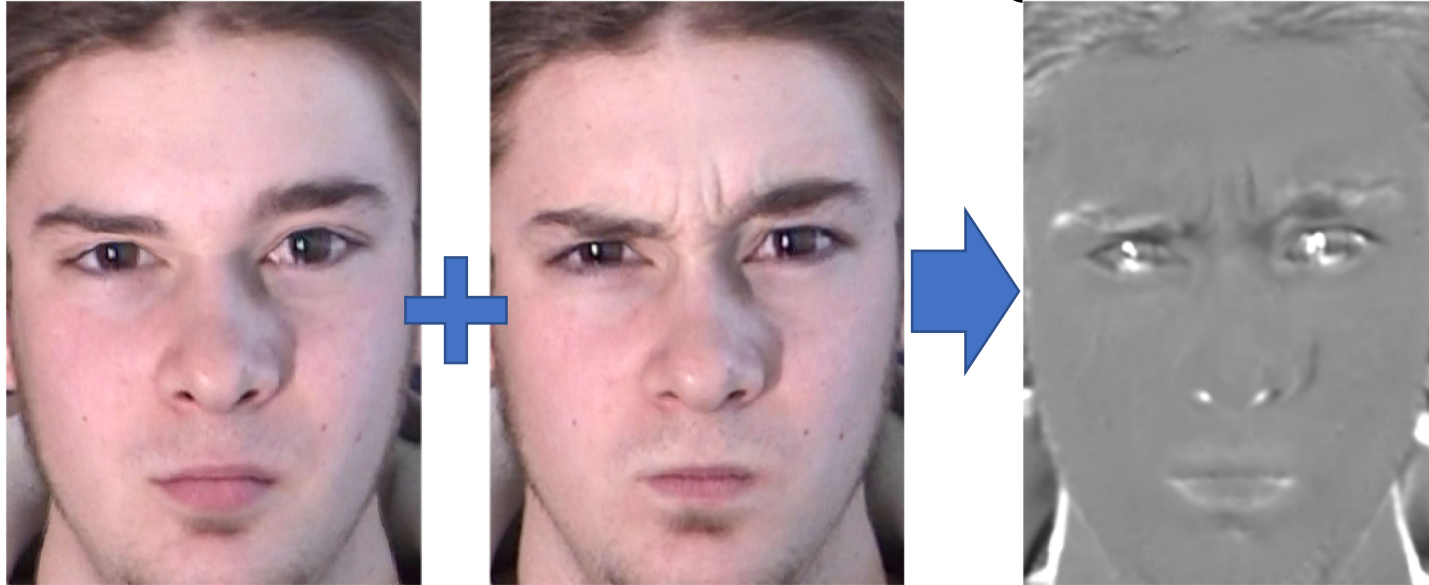
**Step 3 Result**



**Final Result**

- Add shading changes from driving video
- Correct for undesired artifacts

## Step 4: Transferring Fine-Scale Details



Warped Neutral  
Frame ( $\bar{s}^*$ )

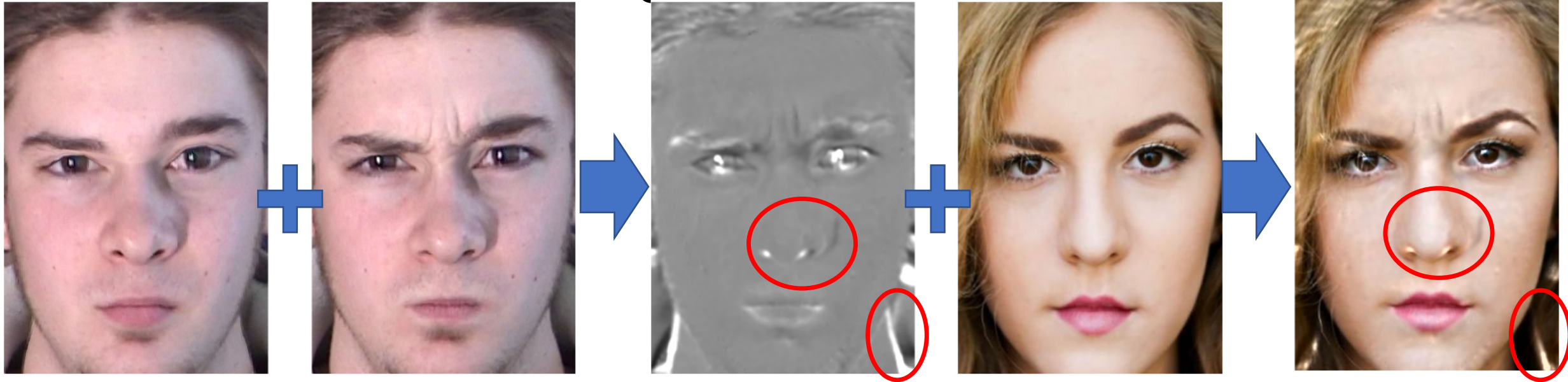
Driving Video  
Frame  $i$  ( $s_i$ )

Ratio Image ( $R_i$ )

- Use warped neutral frame  $\bar{s}^* = \phi_{s^* \rightarrow s_i} \cdot s^*$
- Compute **ratio image**  $R_i$ 
  - $R_i = \frac{f(s_i)}{f(\bar{s}^*)}$



## Step 4: Transferring Fine-Scale Details



Warped Neutral  
Frame ( $\bar{s}^*$ )

Driving Video  
Frame  $i$  ( $s_i$ )

Ratio Image ( $R_i$ )

Step 3 Video  
Frame  $i$  ( $t_i'$ )

Result Video(?)  
Frame  $i$  ( $t_i$ )

- Apply ratio image  $R_i$  to Step 3 video frame after warping into target space

- $t_i = (\phi \cdot R_i) \cdot t_i'$

- Issues here?

- Saturation

- Outlier (misaligned shadow)

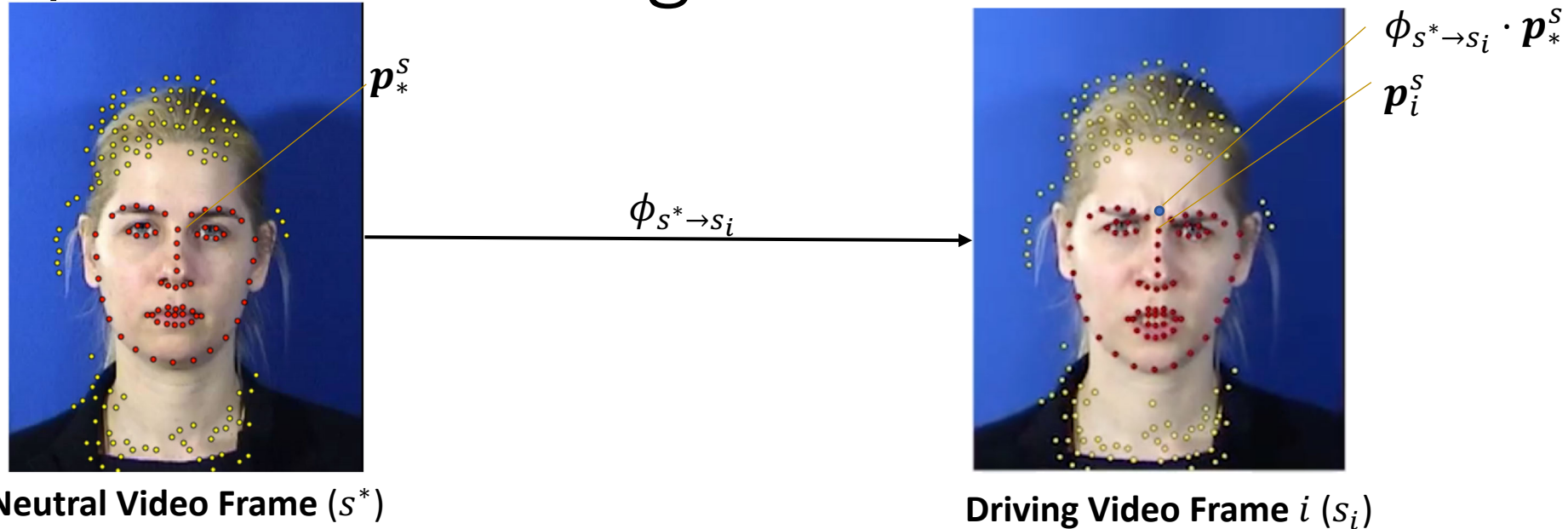


# Step 4: Transferring Fine-Scale Details

- ***Robustness to saturation***
  - Darkening inside in wrinkles is essential, but not as much for brightening
  - Reduce 'brightening effect' by a factor of 0.01
- ***Facial region estimation***: remove all effect outside the face
  - Need the forehead (not a landmark)
  - Best ellipse fit for points on chin
  - *Grab-cut*: segmentation using graph cut



# Step 4: Transferring Fine-Scale Details



- **Outlier detection & elimination**

- General idea: “Outliers (e.g., shadows) appear in similar regions across all frames, so detect them by comparing with the most distant frame and find corresponding outliers in other frames to remove them”
- *Reference frame* ( $s_{ref}$ ): the ‘most distant frame’ with the greatest ‘non-similarity’ deformations

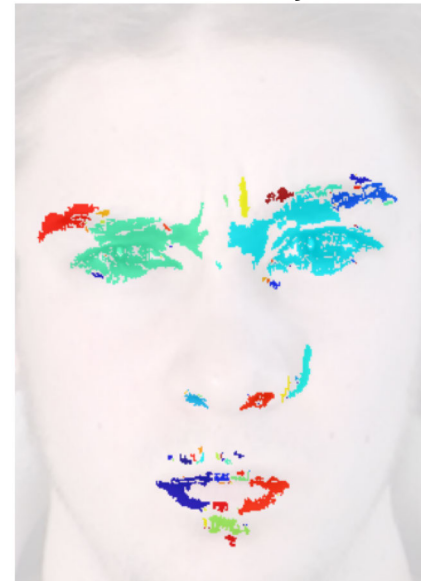
- The frame with the greatest:  $\frac{1}{n} \sum \|\phi_{s^* \rightarrow s_i} \cdot \mathbf{p}_*^s - \mathbf{p}_i^s\|_2^2$

# Step 4: Transferring Fine-Scale Details

- ***Outlier detection & elimination*** (Cont.)
  - *Significant ratio values*: pixels with values  $>1.1$  or  $<1/1.1$  in ratio image
- 1. Outlier detection in  $S_{ref}$ 
  - Find connected components of significant ratio values in  $R_{ref}$



**Ratio Image ( $R_{ref}$ )**



**Connected components  
of significant ratio values**

# Step 4: Transferring Fine-Scale Details

- **Outlier detection & elimination** (Cont.)

- *Significant ratio values*: pixels with values  $>1.1$  or  $<1/1.1$  in ratio image

1. Outlier detection in  $s_{ref}$

- Find connected components of significant ratio values in  $R_{ref}$
- Check if the a pixels in the connected component 'appear' in  $\bar{s}^*$  (patch comparison)

3	7	5	6	0	9	4
5	0	2	4	6	2	5
2	2	2	5	4	2	5
3	5	0	3	2	3	0
1	5	7	5	1	3	5
2	6	9	2	0	4	4
1	3	1	3	1	5	8

Reference Frame ( $s_{ref}$ )

5	3	1	1	2	3	4
2	4	7	2	1	1	3
2	5	3	9	9	1	4
0	4	2	8	7	8	7
5	5	1	7	5	7	6
9	2	1	6	1	5	3
0	2	0	0	3	1	4

Warped Neutral Frame ( $\bar{s}^*$ )



# Step 4: Transferring Fine-Scale Details

- **Outlier detection & elimination** (Cont.)

- *Significant ratio values*: pixels with values  $>1.1$  or  $<1/1.1$  in ratio image

1. Outlier detection in  $s_{ref}$

- Find connected components of significant ratio values in  $R_{ref}$
- Check if the a pixels in the connected component 'appear' in  $\bar{s}^*$  (patch comparison)

3	7	5	6	0	9	4
5	0	2	4	6	2	5
2	2	2	5	4	2	5
3	5	0	3	2	3	0
1	5	7	5	1	3	5
2	6	9	2	0	4	4
1	3	1	3	1	5	8

Reference Frame ( $s_{ref}$ )

5	3	1	1	2	3	4
2	4	7	2	1	1	3
2	5	3	9	9	1	4
0	4	2	8	7	8	7
5	5	1	7	5	7	6
9	2	1	6	1	5	3
0	2	0	0	3	1	4

Warped Neutral Frame ( $\bar{s}^*$ )

# Step 4: Transferring Fine-Scale Details

- **Outlier detection & elimination** (Cont.)

- *Significant ratio values*: pixels with values  $>1.1$  or  $<1/1.1$  in ratio image

1. Outlier detection in  $s_{ref}$

- Find connected components of significant ratio values in  $R_{ref}$
- Check if the a pixels in the connected component 'appear' in  $\bar{s}^*$  (patch comparison)

3	7	5	6	0	9	4
5	0	2	4	6	2	5
2	2	2	5	4	2	5
3	5	0	3	2	3	0
1	5	7	5	1	3	5
2	6	9	2	0	4	4
1	3	1	3	1	5	8

Reference Frame ( $s_{ref}$ )

5	3	1	1	2	3	4
2	4	7	2	1	1	3
2	5	3	9	9	1	4
0	4	2	8	7	8	7
5	5	1	7	5	7	6
9	2	1	6	1	5	3
0	2	0	0	3	1	4

Warped Neutral Frame ( $\bar{s}^*$ )

# Step 4: Transferring Fine-Scale Details

- **Outlier detection & elimination** (Cont.)

- *Significant ratio values*: pixels with values  $>1.1$  or  $<1/1.1$  in ratio image

1. Outlier detection in  $s_{ref}$

- Find connected components of significant ratio values in  $R_{ref}$
- Check if the a pixels in the connected component 'appear' in  $\bar{s}^*$  (patch comparison)

3	7	5	6	0	9	4
5	0	2	4	6	2	5
2	2	2	5	4	2	5
3	5	0	3	2	3	0
1	5	7	5	1	3	5
2	6	9	2	0	4	4
1	3	1	3	1	5	8

Reference Frame ( $s_{ref}$ )

5	3	1	1	2	3	4
2	4	7	2	1	1	3
2	5	3	9	9	1	4
0	4	2	8	7	8	7
5	5	1	7	5	7	6
9	2	1	6	1	5	3
0	2	0	0	3	1	4

Warped Neutral Frame ( $\bar{s}^*$ )

3	1	5
0	1	3
2	3	1

Difference

Max Difference: 5

# Step 4: Transferring Fine-Scale Details

- **Outlier detection & elimination** (Cont.)

- *Significant ratio values*: pixels with values  $>1.1$  or  $<1/1.1$  in ratio image

1. Outlier detection in  $s_{ref}$

- Find connected components of significant ratio values in  $R_{ref}$
- Check if the a pixels in the connected component 'appear' in  $\bar{s}^*$  (patch comparison)

3	7	5	6	0	9	4
5	0	2	4	6	2	5
2	2	2	5	4	2	5
3	5	0	3	2	3	0
1	5	7	5	1	3	5
2	6	9	2	0	4	4
1	3	1	3	1	5	8

Reference Frame ( $s_{ref}$ )

5	3	1	1	2	3	4
2	4	7	2	1	1	3
2	5	3	9	9	1	4
0	4	2	8	7	8	7
5	5	1	7	5	7	6
9	2	1	6	1	5	3
0	2	0	0	3	1	4

Warped Neutral Frame ( $\bar{s}^*$ )

1	3	5
2	2	2
5	0	7

Difference

Max Difference: 7



# Step 4: Transferring Fine-Scale Details

- **Outlier detection & elimination** (Cont.)

- *Significant ratio values*: pixels with values  $>1.1$  or  $<1/1.1$  in ratio image

1. Outlier detection in  $s_{ref}$

- Find connected components of significant ratio values in  $R_{ref}$
- Check if the a pixels in the connected component 'appear' in  $\bar{s}^*$  (patch comparison)

3	7	5	6	0	9	4
5	0	2	4	6	2	5
2	2	2	5	4	2	5
3	5	0	3	2	3	0
1	5	7	5	1	3	5
2	6	9	2	0	4	4
1	3	1	3	1	5	8

Reference Frame ( $s_{ref}$ )

5	3	1	1	2	3	4
2	4	7	2	1	1	3
2	5	3	9	9	1	4
0	4	2	8	7	8	7
5	5	1	7	5	7	6
9	2	1	6	1	5	3
0	2	0	0	3	1	4

Warped Neutral Frame ( $\bar{s}^*$ )

0	0	1
0	0	1
0	1	0

Difference

Max Difference: 1

1
-
-

Minimum  
Max Difference

# Step 4: Transferring Fine-Scale Details

- **Outlier detection & elimination** (Cont.)

- *Significant ratio values*: pixels with values  $>1.1$  or  $<1/1.1$  in ratio image

1. Outlier detection in  $s_{ref}$

- Find connected components of significant ratio values in  $R_{ref}$
- Check if the a pixels in the connected component 'appear' in  $\bar{s}^*$  (patch comparison)

3	7	5	6	0	9	4
5	0	2	4	6	2	5
2	2	2	5	4	2	5
3	5	0	3	2	3	0
1	5	7	5	1	3	5
2	6	9	2	0	4	4
1	3	1	3	1	5	8

Reference Frame ( $s_{ref}$ )

5	3	1	1	2	3	4
2	4	7	2	1	1	3
2	5	3	9	9	1	4
0	4	2	8	7	8	7
5	5	1	7	5	7	6
9	2	1	6	1	5	3
0	2	0	0	3	1	4

Warped Neutral Frame ( $\bar{s}^*$ )

0	0	1
0	1	0
2	0	0

Difference

Max Difference: 2

1
2
-

Minimum  
Max Difference

# Step 4: Transferring Fine-Scale Details

- **Outlier detection & elimination** (Cont.)

- *Significant ratio values*: pixels with values  $>1.1$  or  $<1/1.1$  in ratio image

1. Outlier detection in  $s_{ref}$

- Find connected components of significant ratio values in  $R_{ref}$
- Check if the a pixels in the connected component 'appear' in  $\bar{s}^*$  (patch comparison)

3	7	5	6	0	9	4
5	0	2	4	6	2	5
2	2	2	5	4	2	5
3	5	0	3	2	3	0
1	5	7	5	1	3	5
2	6	9	2	0	4	4
1	3	1	3	1	5	8

Reference Frame ( $s_{ref}$ )

5	3	1	1	2	3	4
2	4	7	2	1	1	3
2	5	3	9	9	1	4
0	4	2	8	7	8	7
5	5	1	7	5	7	6
9	2	1	6	1	5	3
0	2	0	0	3	1	4

Warped Neutral Frame ( $\bar{s}^*$ )

0	1	0
2	0	0
0	0	1

Difference

Max Difference: 2

1
2
2

Minimum

Max Difference

Average:  $1.7 < 5 \rightarrow$  Outlier

# Step 4: Transferring Fine-Scale Details

- ***Outlier detection & elimination*** (Cont.)

2. Outlier elimination in  $s_i$

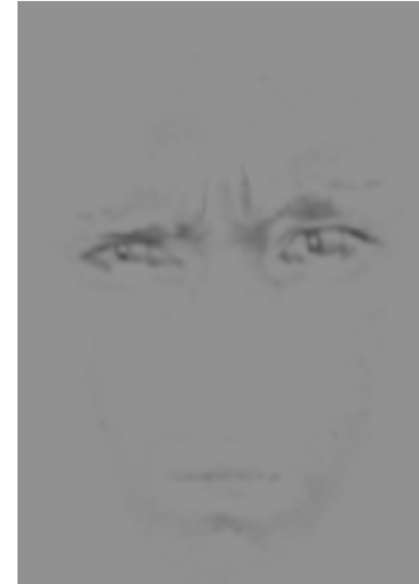
- Compute  $\phi_{s_i \rightarrow s_{ref}}$  (transform from  $s_i$  to  $s_{ref}$ )
- Exclude if outlier is close to the transformed pixel (20 px)



**Connected components  
of significant ratio values**



**Outliers (red)**



**Modified Ratio Image**

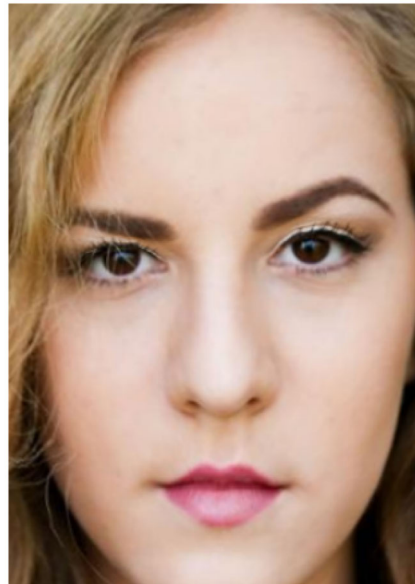


# Step 4: Transferring Fine-Scale Details

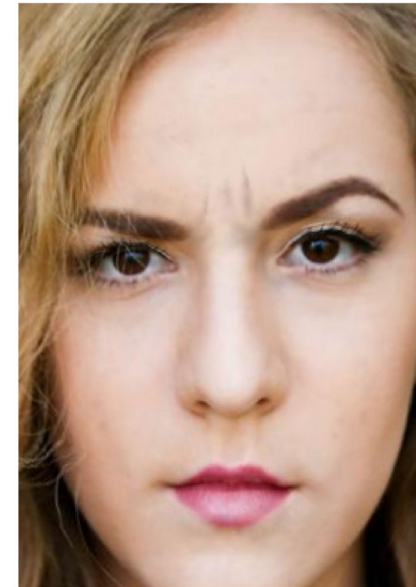
- ***Temporal stability***
  - Temporal Gaussian filter of size 21
- Apply the modified ratio image!



**Modified Ratio Image**

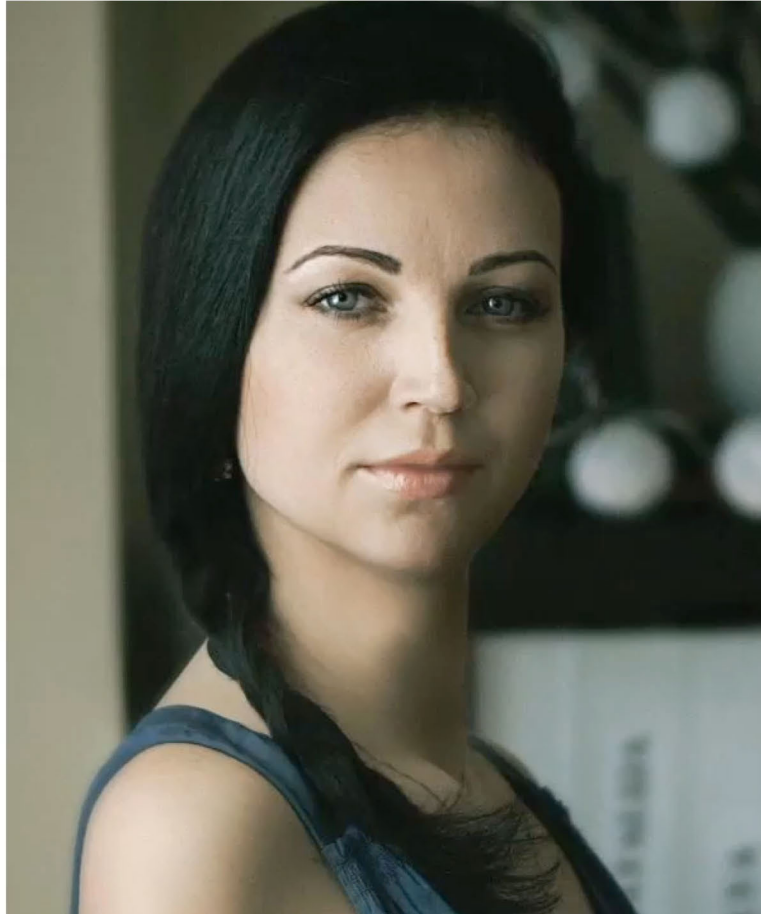


**Step 3 Video  
Frame  $i (t_i')$**



**Result Video  
Frame  $i (t_i)$**

# Results

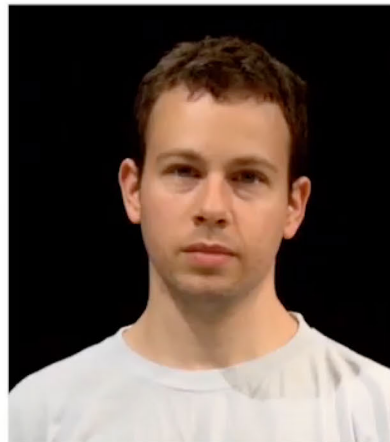


# Results

- Animated facial avatars



Driving video:



# Evaluation

- User study
  - Presented 24 videos (8 real) to each of the 30 participants
  - Asked to rate *very likely fake* (1), *likely fake* (2), *could equally be real or fake* (3), *likely real* (4), *very likely real* (5)

	<b>Real Videos</b>					<b>Animated Videos</b>				
	1	2	3	4	5	1	2	3	4	5
Anger	0.00	0.11	0.11	0.42	0.35	0.10	0.28	0.21	0.28	0.13
Fear	0.00	0.02	0.08	0.6	0.31	0.08	0.26	0.16	0.34	0.15
Happy	0.00	0.02	0.11	0.40	0.47	0.02	0.17	0.23	0.33	0.24
Surprise	0.00	0.03	0.13	0.26	0.57	0.12	0.33	0.19	0.25	0.11
Average	0.00	0.04	0.11	0.42	0.43	0.08	0.26	0.20	0.30	0.16

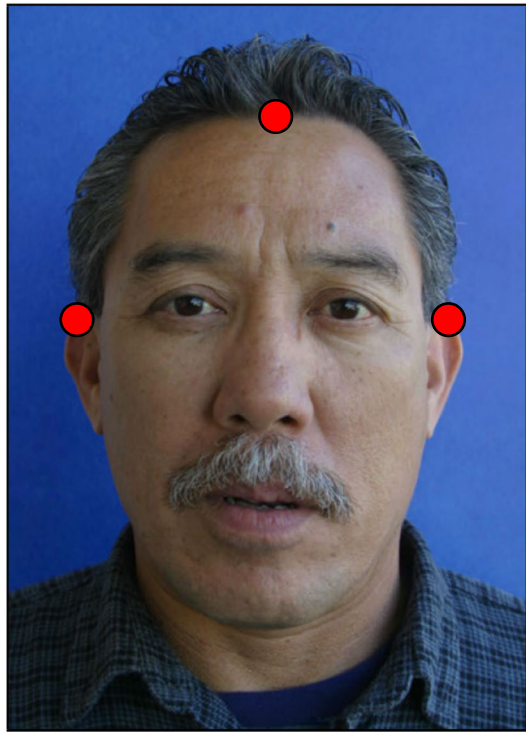
0.85

0.46

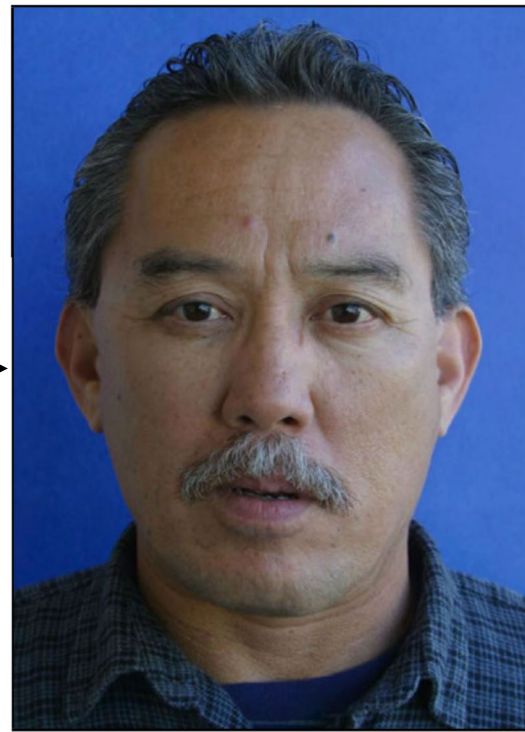


# Comparisons

- Warping comparison (Fried et al.)
  - No manual step



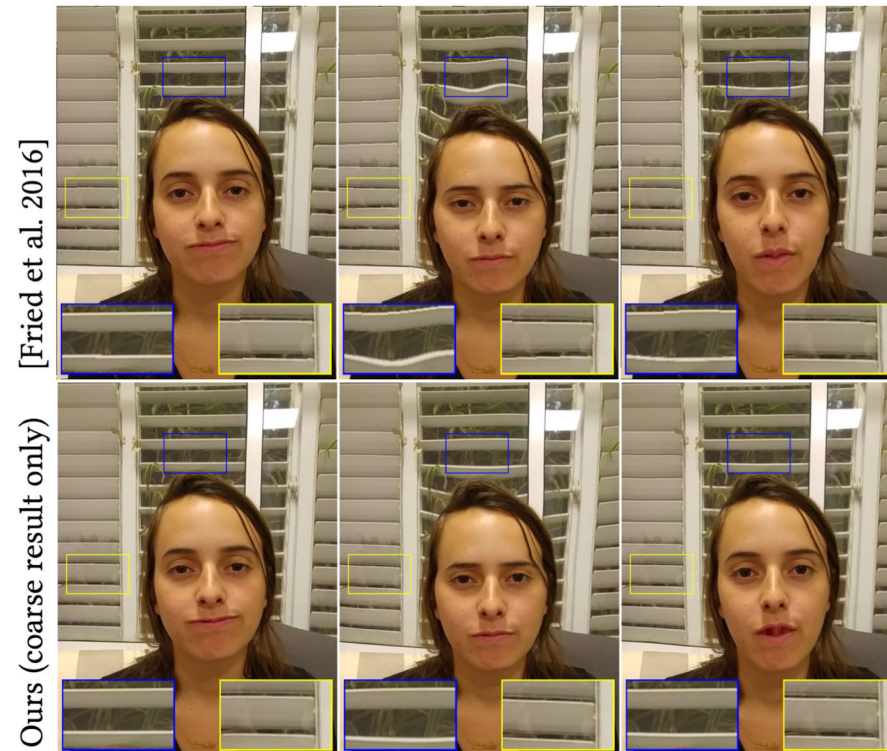
**Close-up photo**



**Generated far photo**

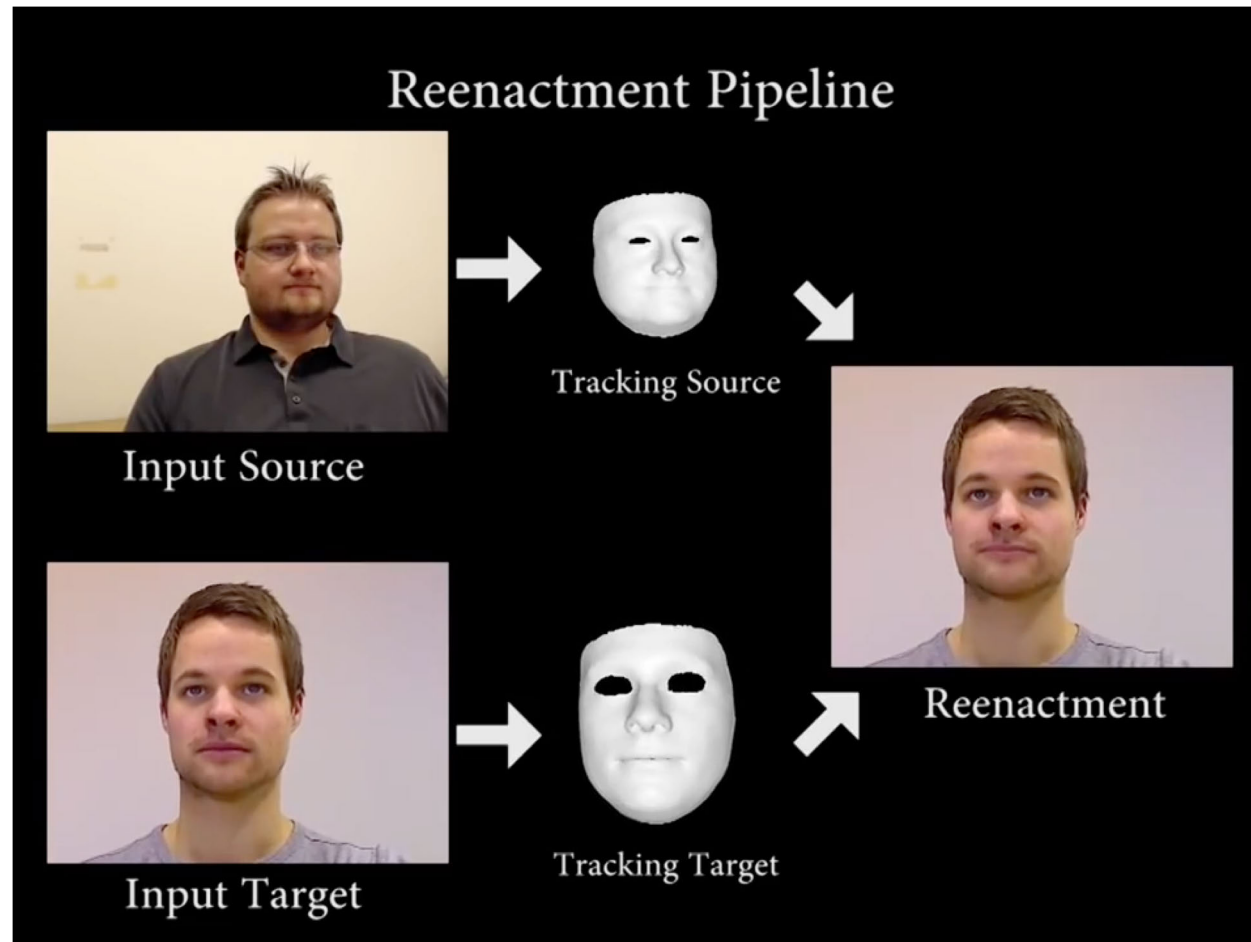
# Comparisons

- Warping comparison (Fried et al.)
  - Hidden region transfer + fine-scale details
  - Smoother results



# Comparisons

- Reenactment comparison (Thies et al.)



# Comparisons

- Reenactment comparison (Thies et al.)
  - Single picture of target as input
  - Transfers head motion
  - Fine-scale details



**Target Photo/Video**

**Driving Video**

**Thies et al.**

**Results**

# Limitations

- Frontal head pose assumption





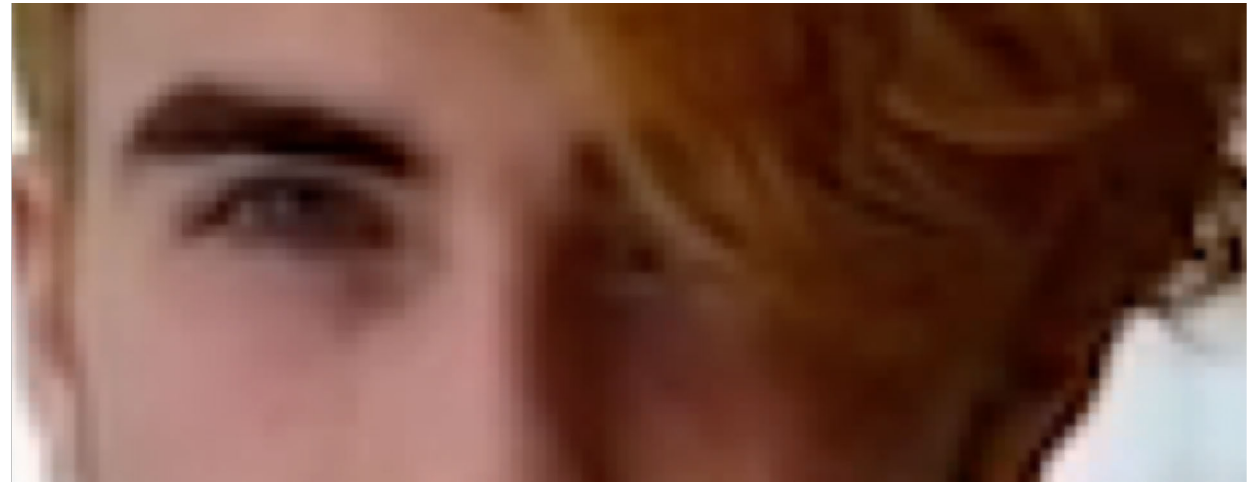
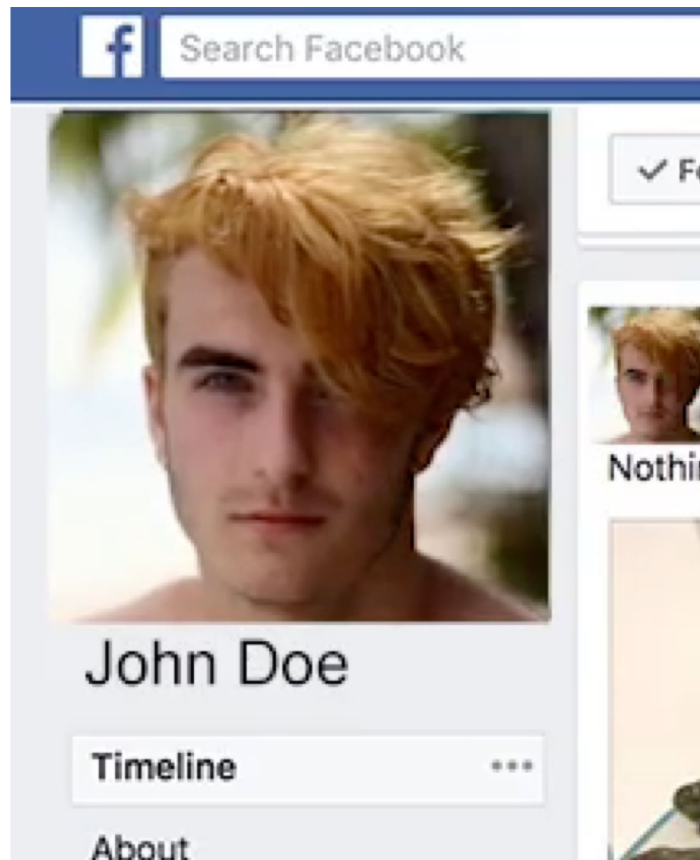
# Limitations

- Neutral target face assumption



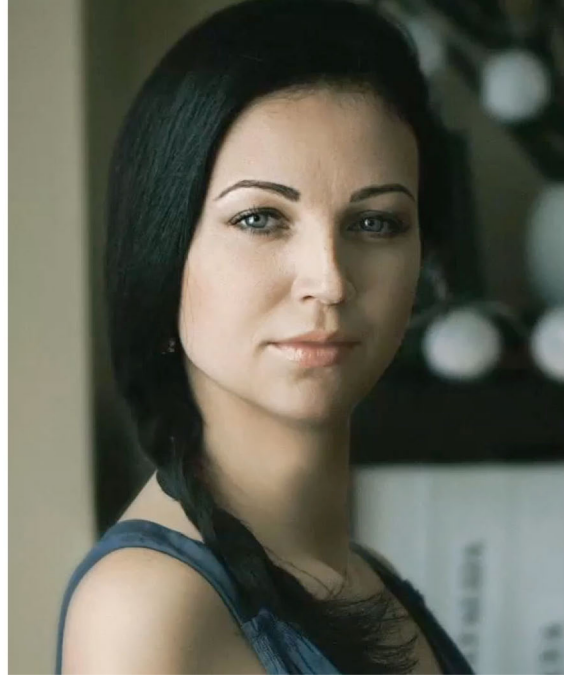
# Limitations

- Dependency on face tracker accuracy
- Eye blinking

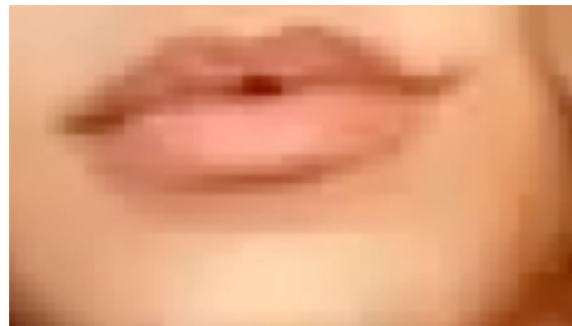


# Limitations

- Warping of background



- Mouth region



# Recent Work

- Geng et al.

