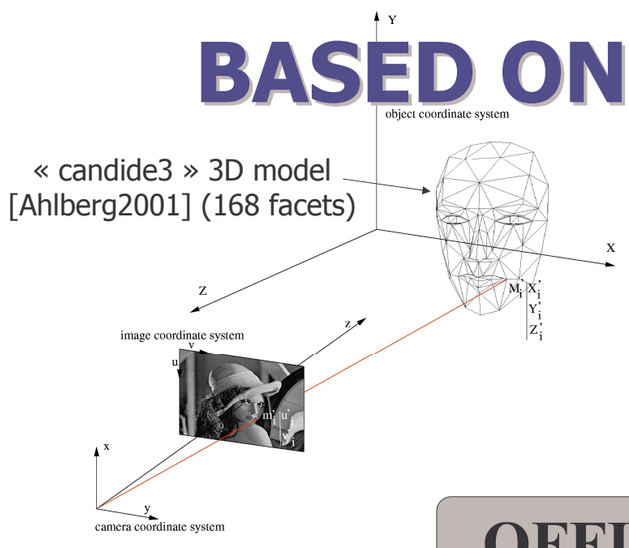
3D-FACE MODEL TRACKING BASED ON A MULTI-RESOLUTION ACTIVE SEARCH

Marc CHAUMONT and William PUECH
marc.chaumont@lirmm.fr and william.puech@lirmm.fr
LIRMM, UMR CNRS 5506, Université de Montpellier II



GOAL:
Robust and real-time tracking of a face (and its animations) in a un-calibrated video sequence.
PRINCIPLE:
An offline learning (3D model shape, texture, update matrix) and a online multi-resolution active search.

OFFLINE LEARNING STEP

1- Shape learning (σ learning):

Least square minimization of:

$$\{T_{2\times4}, \sigma, \alpha\} = arg \min_{\{T_{2\times4},\sigma,\alpha\}} \sum_i (u_i - u'_i)^2 + (v_i - v'_i)^2,$$

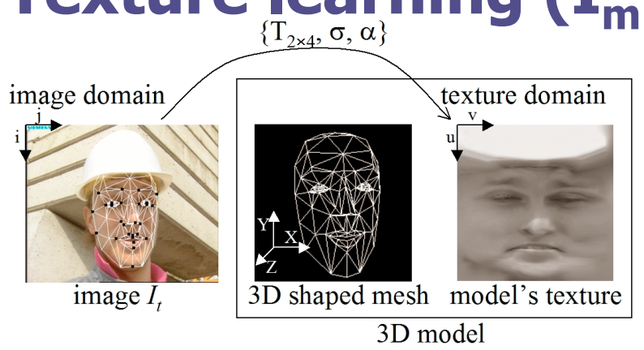
with : $\{(u_i, v_i)^t\}$: a set of 2D image points (manually set)

$\{(u'_i, v'_i)^t\}$: a set of 3D points displaced and then projected

$$(u'_i, v'_i)^t = T_{2\times4}.[\underbrace{A_i.\alpha + S_i\sigma + M_i}_{M'_i}].$$

T_{2x4} : affine projection
$S_i\ \sigma$: shape displacement (σ: intensity vector of shape units)
$A_i\ \alpha$: animation displacement (α: intensity vector of animation units)

2- Texture learning (I_m learning):



3- Update Matrix Learning (U learning):

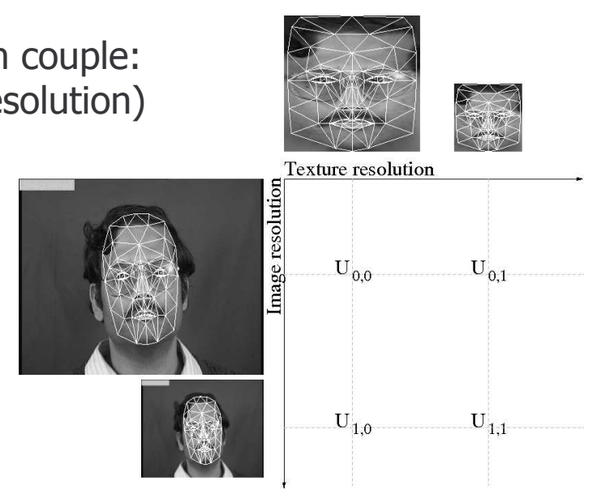
Given an image resolution and a texture resolution compute U:

Update matrix U (Negative pseudo-inverse of the gradient matrix G)

$$U = -(G^T G)^{-1} G^T,$$
$$\text{with : } G = \frac{\delta r(p)}{\delta p},$$
$$r(p) = W(I_t) - I_m.$$

$W(I_t)$: image I_t warped
I_m : model's texture
$r(p)$: residue
p : animation vector α and pose matrix T_{2x4}

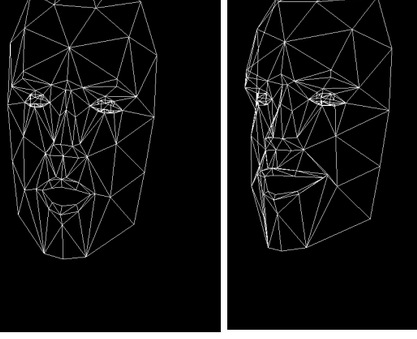
Matrix U are computed for each couple:
(image resolution × texture resolution)



INLINE TRACKING STEP

1- Face localization (for the 1st image):

- Use of a **face (frontal view) detector** algorithm; (grey bounding box on the Figure below).
- Deduce a **rough initial pose** T_{2x4} by guessing few 2D feature points and minimizing E; (white mesh on the Figure below).
- Run the Multi-resolution Active Model Search.



2- Multi-Resolution Active Model Search:

For each *valid** couple ($r_i \times r_t$) = (image resolution × texture resolution) run the algorithm

ALGORITHM
Repeat until convergence
- **Warped the current image** i.e compute $W(I_t)$;
  I_t is at resolution r_i and W warp to r_t resolution.
- Compute the **current residue** $r(p) = W(I_t) - I_m$.
- Compute the **update parameter** vector $\Delta p = U_{r_i,r_t} \times r(p)$.
- Modify the 3D model and its pose $p = p + \Delta p$.

* *valid* couples used here are the diagonal couples

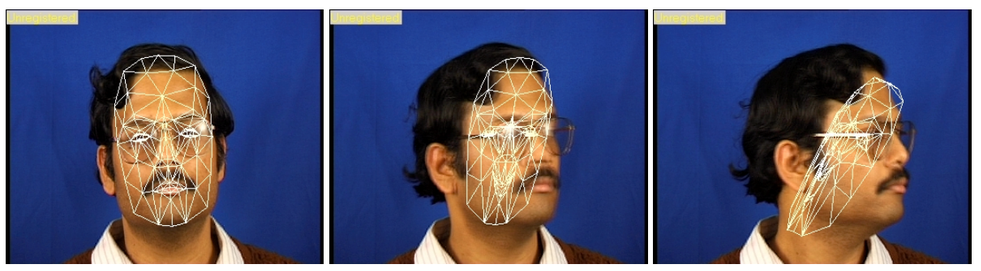
RESULTS & CONCLUSION

RESULTS:

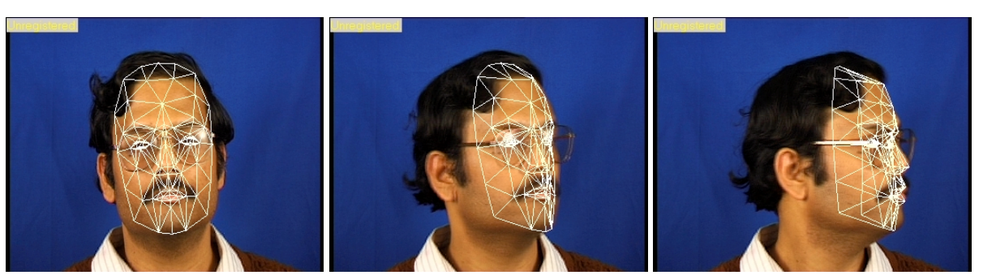
Tracking with multi-resolution: Images 3, 13, 25; Erik sequence

3D Mesh: Images 3, 13, 25; Erik sequence

Execution time: 8Hz on a Pentium 2.4Ghz, language C++



Tracking **without multi-resolution**: Images 3, 8, 13; rotation sequence



Tracking **with multi-resolution**: Images 3, 8, 13; rotation sequence

CONCLUSION:

A face **real-time** tracker (catching face animations) **robust to strong motions**.
Future works will deal with matrix dimension reduction and particle filtering.