Tvarita Ltd.

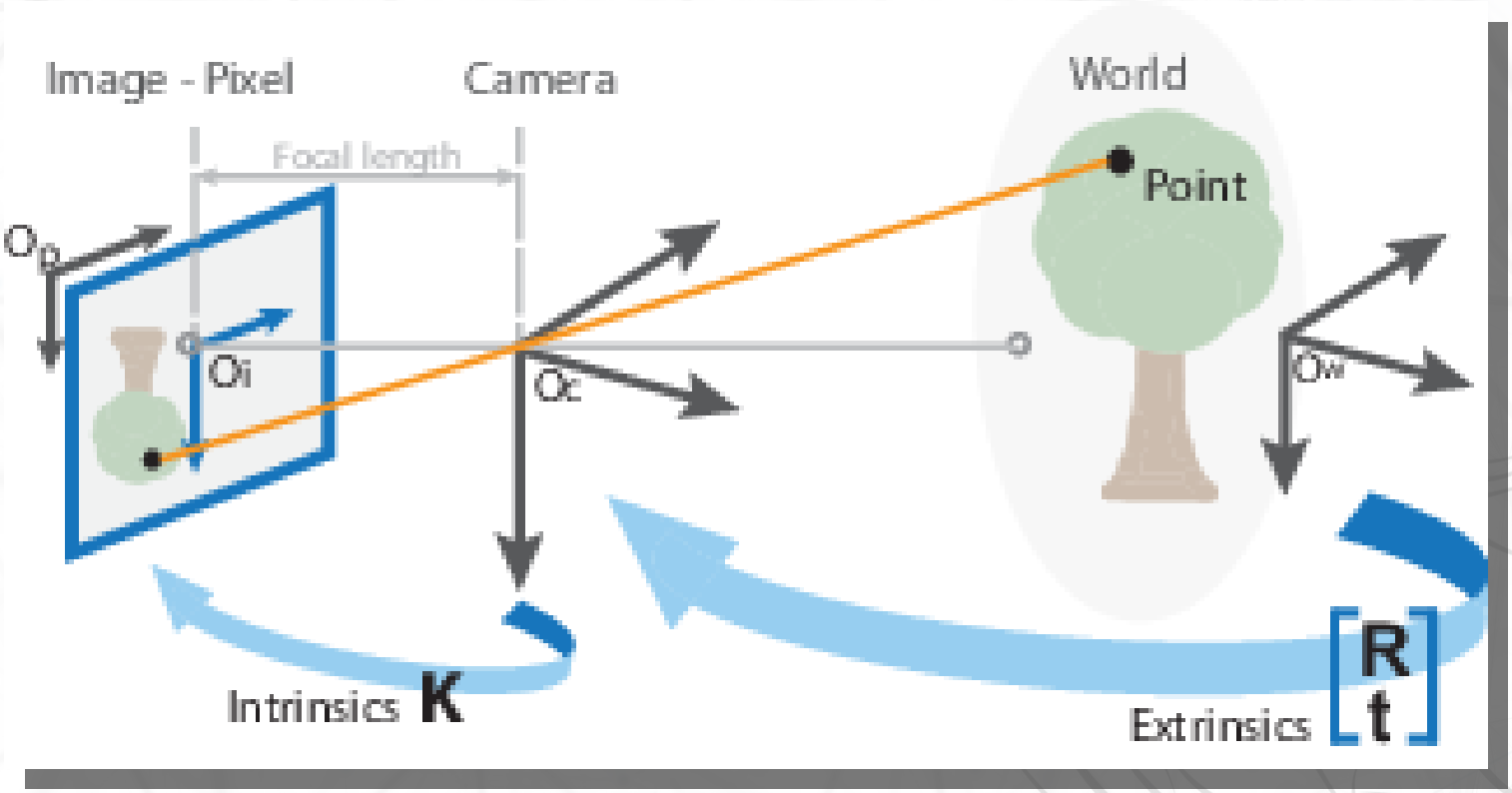
PyFace – 3D facial reconstruction from multiple 2D poses

Alexandru-Ion Marinescu
amarinescu@cs.ubbcluj.ro

Adrian-Sergiu Darabant
dadi@cs.ubbcluj.ro

Problem presentation

Intrinsic and extrinsic params



Intrinsic: *k_mat* and *d_vec*
Constant for each **camera**...

Extrinsic: *r_vec* and *t_vec*
Constant for each **image**...

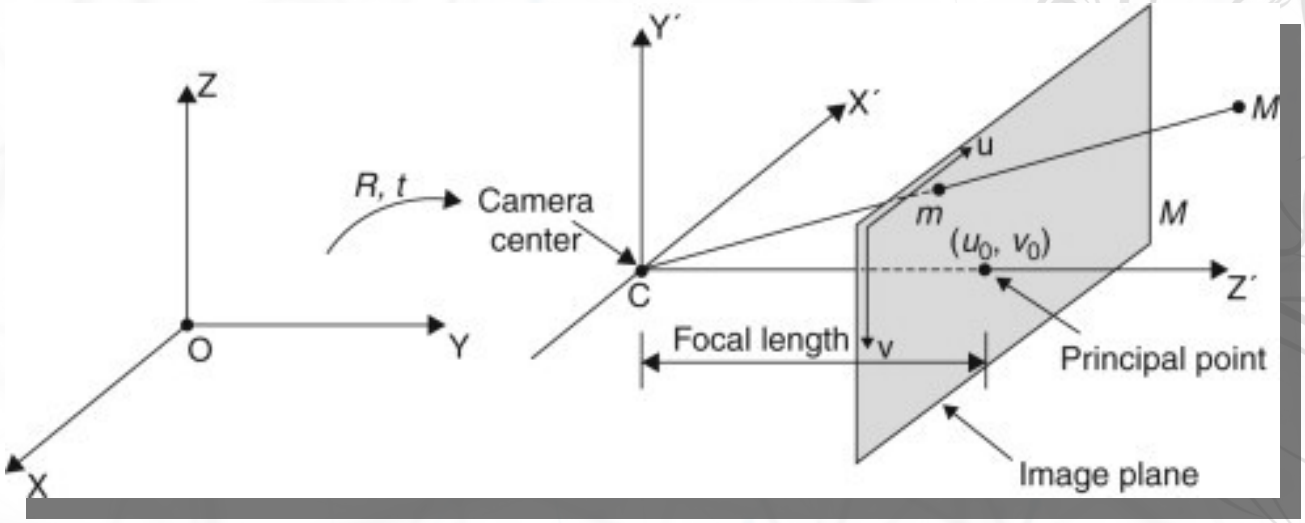
Intrinsic and extrinsic params

Either perform a camera calibration or use an approximation:

$$k_{mat} = \begin{pmatrix} W & 0 & \frac{W}{2} \\ 0 & W & \frac{H}{2} \\ 0 & 0 & 1 \end{pmatrix}$$

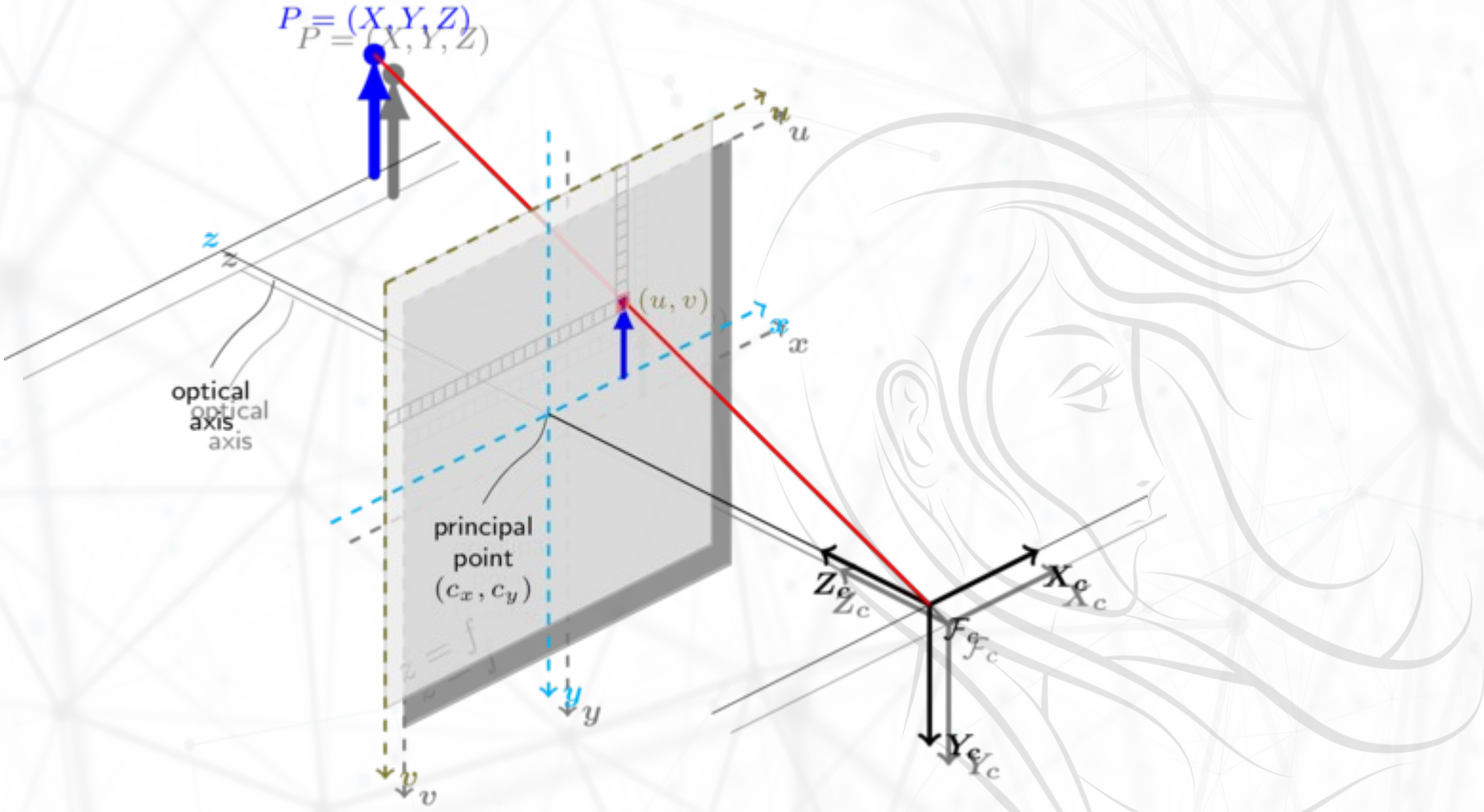
$$d_{vec} = [0, 0, 0, 0, 0]$$

$$\boldsymbol{r_{vec}, t_{vec} = solvePnPRansac(pts_{obj}, pts_{img}, k_{mat}, d_{vec})}$$

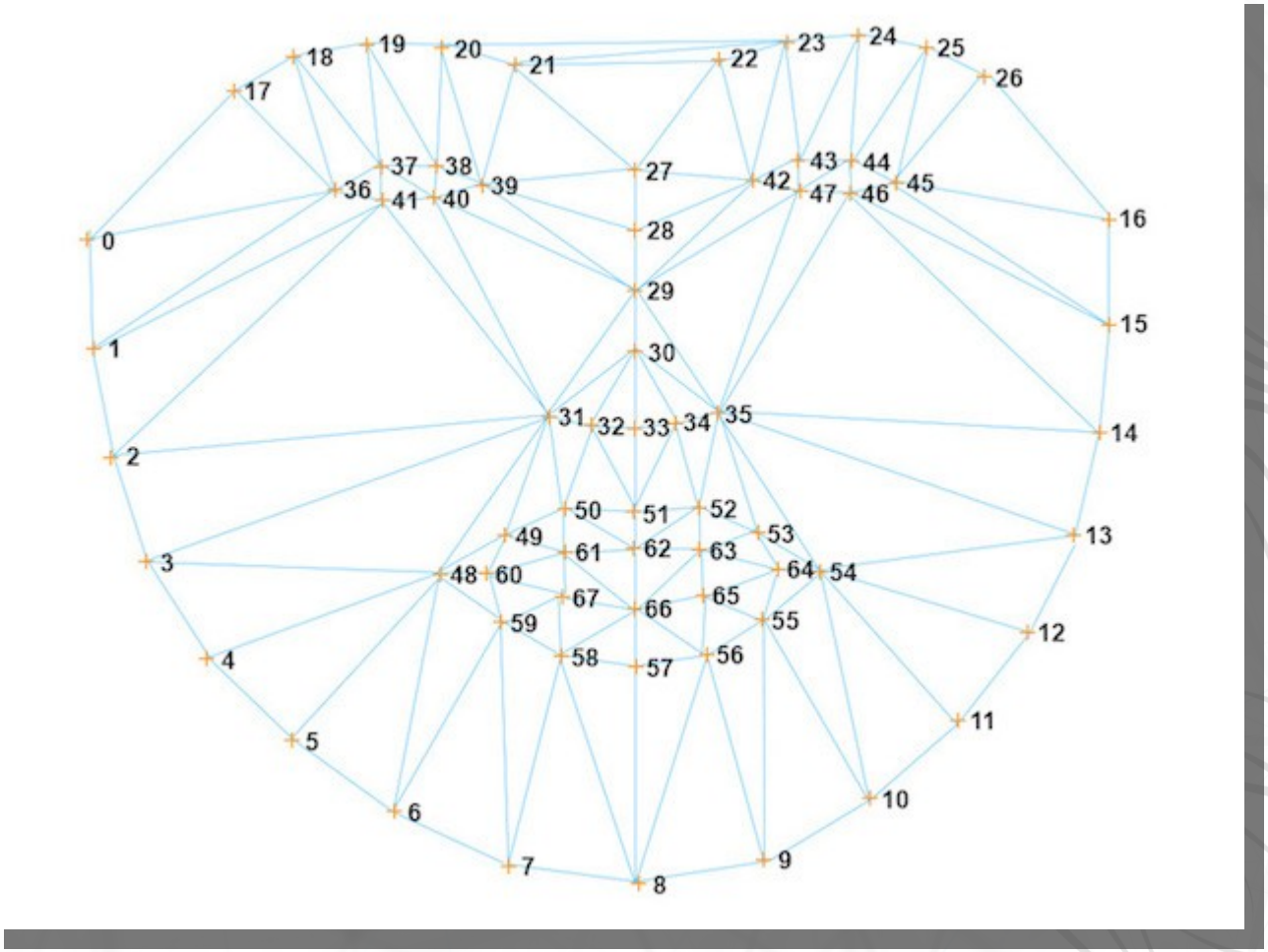


Pinhole camera model

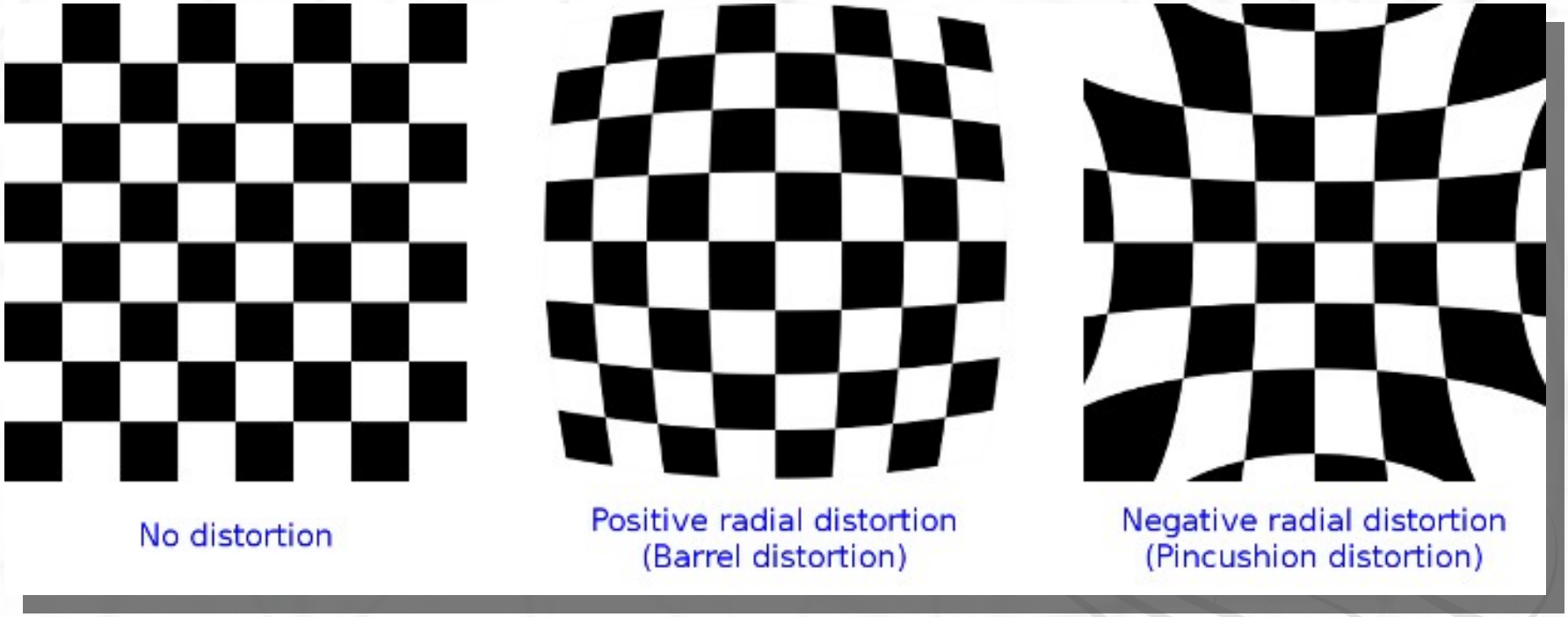
Assumes **zero** distortion...



Landmark detection



Examples of lens distortion



solvePnPRansac(...)

$$s\,p_c = K\,[\,R|\,T\,]\,p_w$$

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Perspective-n-Point is the problem of estimating the pose of a calibrated camera given a set of n *3D* points in the world and their corresponding *2D* projections in the image.

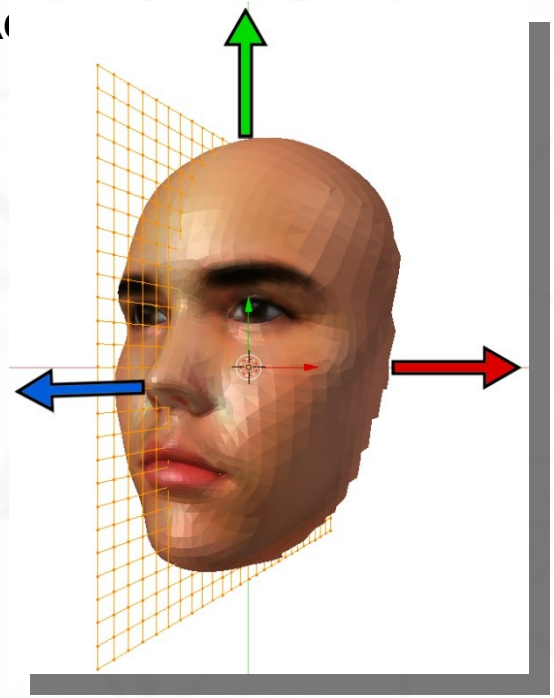
The camera pose consists of *6 degrees-of-freedom*...

PnP is prone to errors if there are **outliers** in the set of point correspondences. **RANSAC** can be used with existing approaches to make the solution for the camera pose more *robust to outliers*...

Image pre-processing

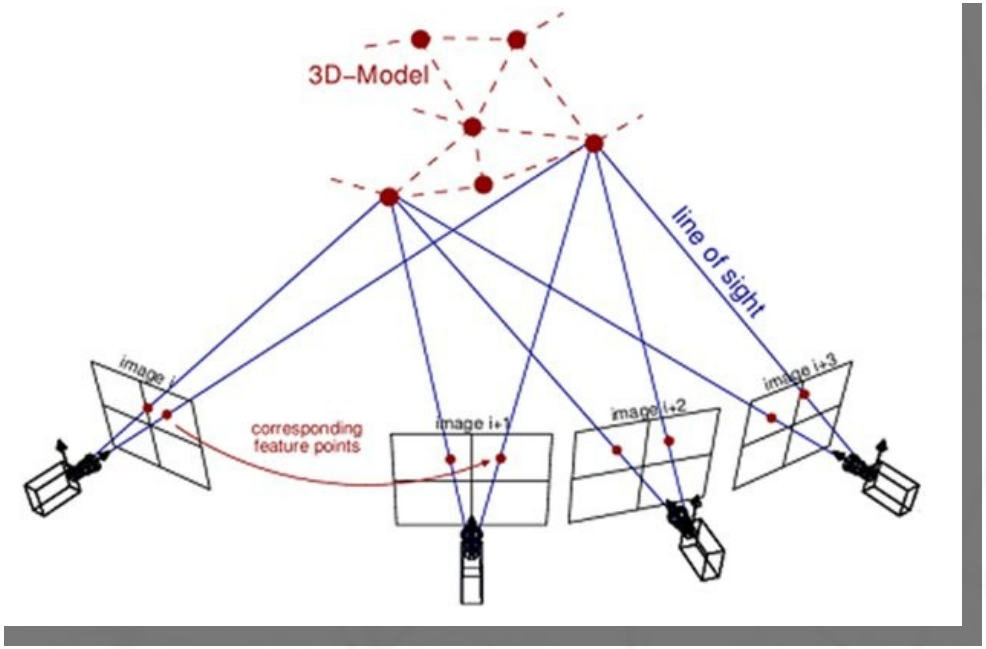
Crop square **center** to face...

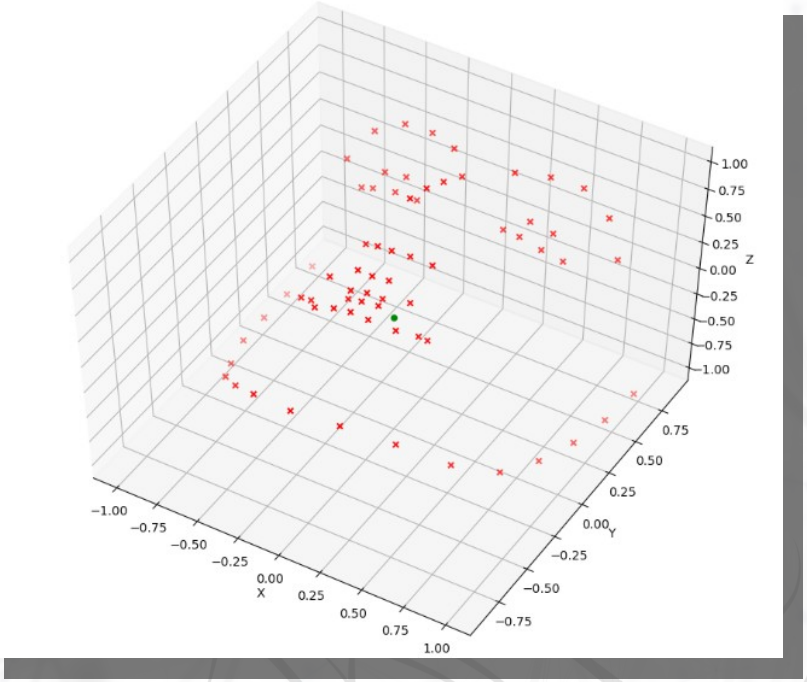
R



Bundle adjustment

Optimize **3D** point cloud with **minimal** *reprojection error*…





Bundle adjustment

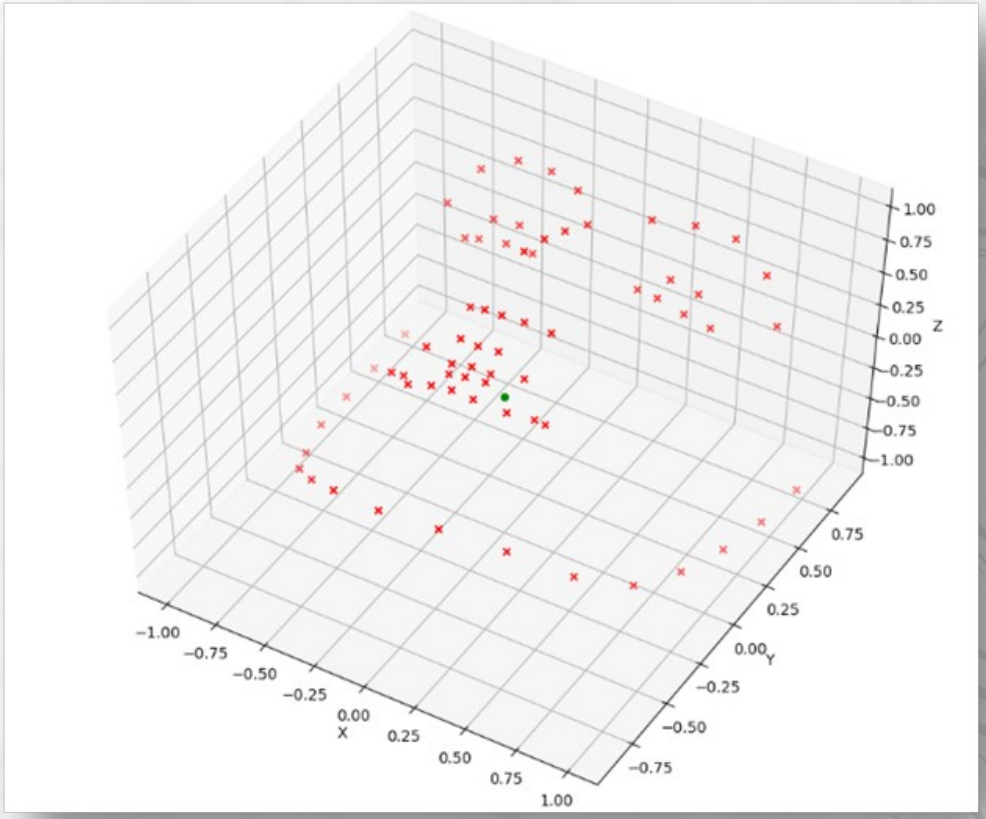
Least Squares and algorithm is *Levenberg–Marquardt*...

Mathematically, the function to minimize is:

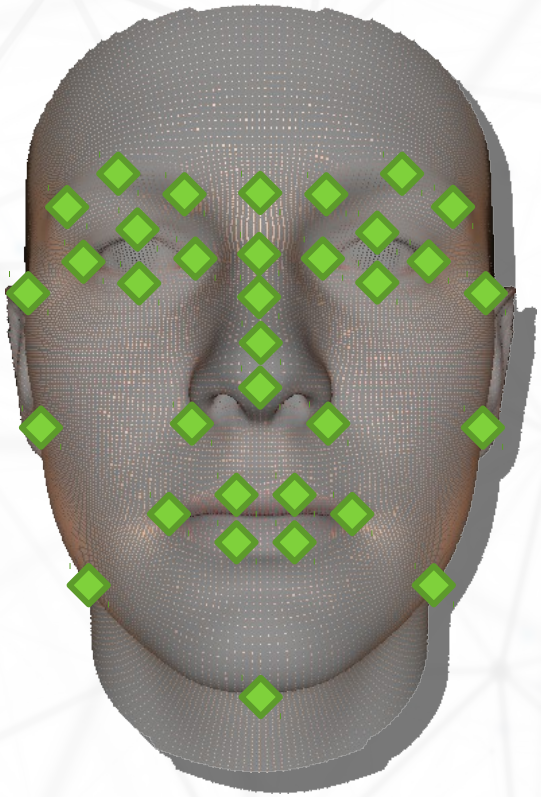
$$min_{a_j b_i} \sum_{i=1}^{n} \sum_{j=1}^{m} \left\| Q(a_j, b_i) - x_{ij} \right\|^2$$

- n 3D points to optimize
- m cameras/images
- x_{ij} is the projection of the i^{th} point on the j^{th} image
- Camera j is parameterized by a vector a_j (or (a_1, a_2, a_3)) and point i by a vector b_i (b_1, b_2, b_3)
- $Q(a_j, b_i)$ is the predicted projection of point i on image j

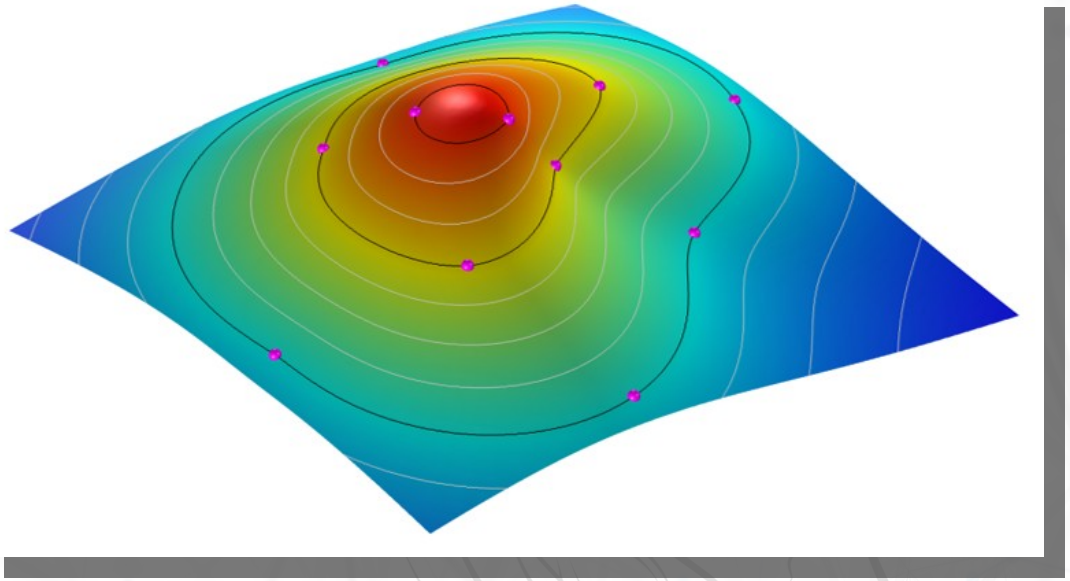
Face specific - 3D point cloud



Generic deformable models



Basel 2009
53,490 vertices
106,466
triangles



RBF interpolation using multiquadric

multiquadric $\phi(r) = \sqrt{1 + (\varepsilon r)^2}$

Forehead extraction

Is needed for complete facial reconstruction and performed by a separate pipeline...

A CNN takes 5 landmarks along the hair line