Camera Self-Calibration Using Human Faces

Masa Hu[1], Garrick Brazil[2], Nanxiang Li[3], Liu Ren[3], Xiaoming Liu[1]
[1] Michigan State University, [2] Facebook, [3] BOSCH Research North America
huynshen@msu.edu, brazilga@fb.com, nanxiang.li@us.bosch.com, liu.ren@us.bosch.com, liuxm@cse.msu.edu

***Abstract*— Despite recent advancements in depth estimation and face alignment, it remains difficult to predict the distance to a human face in arbitrary videos due to the lack of camera calibration. A typical pipeline is to perform calibration with a checkerboard before the video capture, but this is inconvenient to users or impossible for unknown cameras. This work proposes to use the human face as the calibration object to estimate metric depth information and camera intrinsics. Our novel approach alternates between optimizing the 3D face and the camera intrinsics parameterized by a neural network. Compared to prior work, our method performs camera calibration on a larger variety of videos captured by unknown cameras. Further, due to the face prior, our method is more robust to noise in 2D observations compared to previous self-calibration methods. We show that our method improves calibration and depth prediction accuracy over prior works on both synthetic and real data. Code will be available at `https://github.com/yhu9/FaceCalibration`.**

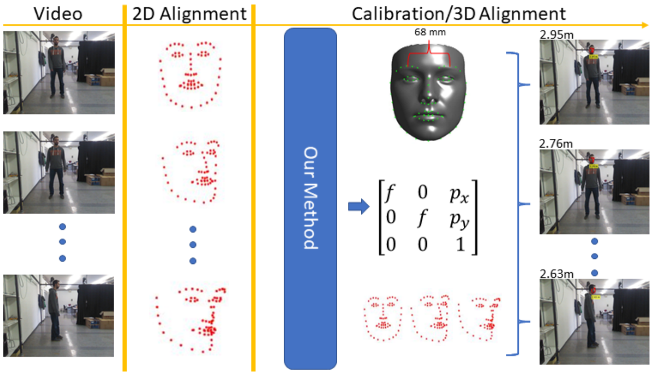
Video
2D Alignment
Calibration/3D Alignment
Our Method
68 mm
$\begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}$
2.95m
2.76m
2.63m

Fig. 1: Our self-calibration algorithm takes a face video as input and estimates the 3D face, focal length, principal point, and head pose. With calibration we can determine the distance to the face in metric units.

I. Introduction

Camera calibration is a classic computer vision problem which involves solving the internal parameters of a pinhole camera to define the projection of 3D objects to the 2D imaging plane [1]. Such calibration is crucial to any application which needs the projection of a captured 2D image to the more practically useful real-world 3D metric space.

There are two algorithm categories for camera calibration: object-based and self-calibration. The former can provide accurate calibration results, but requires active user participation and the presence of a calibration object with known 3D dimensions such as a checkerboard [38]. The latter is sensitive to noise in 2D correspondences, but can operate without any 3D information [13], [14], [24]. In this work, we propose a method in between the two which leverages prior 3D information of human faces, *i.e.* a 3D Morphable Model (3DMM), and uses 2D correspondences of the face across a video to perform self-calibration. In doing so, our method gains higher camera calibration performance compared to existing self-calibration methods while having more accessible use cases compared to object-based calibration.

Object-based calibration methods [6], [13], [26], [38], [40] can acquire highly accurate intrinsic parameters, and only require several images of a checkerboard. However, this has made camera calibration an *offline* procedure where the camera intrinsic is estimated within laboratory setting. This causes two issues: i) the camera must be recalibrated manually anytime the intrinsics are changed (zooming in/out), and ii) the camera cannot be calibrated without imaging the known calibration object. Although object-based methods are the best for accurate calibration, this is often not practical for consumer use where camera intrinsics change often. Further, if no access is available to the camera, *e.g.*, existing YouTube videos, it is impossible to use object-based calibration on that camera. Our motivation is to therefore use a common 3D object like a human face, which is more likely to appear in images or videos. By being able to perform camera calibration via a human face, it becomes more likely a in-the-wild video can be calibrated.

Self-calibration methods [5], [8], [10], [14], [15], [24] can theoretically be applied to any video since only 2D correspondences are needed. However, in reality self-calibration algorithms are extremely sensitive to noise in 2D observations to the point that they are rarely used in practice [29]. Small errors in image correspondences can significantly change optimization results on camera intrinsics using existing self-calibration methods [9], [14], [22], [24]. Further, the initial solution provided to self-calibration must be fairly accurate for proper optimization [5], [10].

As a method in between object-based and self-calibration, we propose to utilize the *human face* as a calibration object without knowing its exact 3D shape. Faces are commonly found in videos in the wild, unlike other calibration objects like checkerboards [38] or household items [10], allowing our method to be applied in a wider range of scenarios. Further, our method has higher calibration performance compared to existing self-calibration algorithms which makes no assumption on the existing 3D geometry. Finally, we are motivated to utilize faces because there is a rich history of research in face alignment and 3D face reconstruction [3], [20], [31], [32]. Many algorithms [3], [17], [31] have excelled in 2D landmark localization on numerous datasets. Moreover, 3DMMs of faces [2] provides prior knowledge of the 3D shape in metric space while fitting to diverse individuals.

Problem	Method	Input			Output			# imgs	# pts
		2D	3D	**K**	**K**	*pose*	3D		
PnP	P3P [7]	✓	✓	✓	✗	✓	✗	1	3
	EPnP [19]	✓	✓	✓	✗	✓	✗	1	$\geq$4
Object Calibrate	DLT [13]	✓	✓	✗	✓	✓	✗	1	≥ 6
	UPnP [26]	✓	✓	✗	✓	✓	✗	1	≥ 6
	DLS [40]	✓	✓	✗	✓	✓	✗	1	≥ 6
	Zhang [38]	✓	✓	✗	✓	✓	✗	≥ 2	≥ 4
Self Calibrate	Hartley [14]	✓	✗	✗	✓	✗	✗	≥ 2	≥ 8
	Louraki [22]	✓	✗	✗	✓	✗	✗	≥ 2	≥ 8
	Fetzer [9]	✓	✗	✗	✓	✗	✗	≥ 2	≥ 8
	Pedestrian [15]	✓	✗	✗	✓	✓	✗	≥ 3	≥ 2
	GP2C [10]	✓	✗	✗	✓	✓	✓	1	≥ 4
	BPnP [5]	✓	✗	✗	✓	✓	✓	≥ 1	≥ 3
	NN+AO (ours)	✓	✗	✗	✓	✓	✓	≥ 1	≥ 4

TABLE I: The 3D shape and intrinsic parameter $\mathbf{K}$ can either be estimated or required as input, depending on the problem. [Keys: 2D/3D=2D/3D points as input]. The number of images and points indicate the problem size each method handles.

We propose to jointly solve for the 3D face shape and camera intrinsic in an alternating fashion. Recent work BPnP [5] has shown that PnP solutions can provide meaningful direction for stochastic gradient descent optimization on structure from motion and camera calibration problems. Base on this work, we optimize the 3D face shape and camera intrsinsic through the EPnP algorithm [19]. We are the first work to perform optimization of the 3D shape and camera intrinsic using alternating optimization. Compared to object-based calibration work [5], [10], [38], we can perform camera calibration on any videos containing a human face. Compared to self-calibration methods [5], [8], [9], [14], [22], [33], we demonstrate that our method is significantly more robust to noise. Additionally, using faces as a calibration object allows us to utilize SOTA face alignment methods which have been refined over the years [2], [3], [17].

In summary, the contribution of this work is three fold:

1) This is the first camera calibration work that utilizes a prior 3DMM of human faces as a calibration object to determine camera intrinsics.
2) We propose a novel alternating optimization strategy between the 3D shape and camera intrinsics by optimizing through the EPnP solution.
3) We demonstrate our superior performance on focal length and depth estimation on both synthetic and real data over SOTA methods. Additionally, our method is more robust to errors in 2D observations.

II. Related Works

Tab. I shows a list of camera calibration algorithms, separated into object-based methods and self-calibration based on whether 3D information is required. Our method falls into the self-calibration category, and is most similar to recently proposed BPnP [5] and GP2C [10].

PnP. Given a 3D object and its corresponding observations in 2D, the problem of localizing the object pose into the scene is known as the PnP problem. The minimal solution requires 3 points and is known as the P3P problem. Unfortunately, the minimal solution degrades drastically in the presence of noisy correspondences, and it is better to use the EPnP [19] algorithm which utilizes all observed points.

Recently, Chen *et al.* [5] show that it is possible to train neural networks through the P3P problem. We take this knowledge a step further and show that camera calibration can be performed using the EPnP [19] algorithm. We find that EPnP is a better alternative to the P3P algorithm since it can utilize all points to solve for pose while still being a differentiable function. Our usage of the EPnP is significantly faster since we do not need to use the RANSAC procedure with P3P, and is more robust to noise thanks to the EPnP algorithm's formulation to utilize all points.

Object-based Calibration. The gold standard for object-based calibration is Zhang's method [38] which uses a planar checkerboard with known 3D dimensions. Its efficacy comes from the ability to use multiple views of a known planar object in determining camera intrinsics and extrinsics while easily acquiring accurate 2D-3D correspondences. Given the known geometry of a checkerboard beforehand allows users to match 3D geometry to 2D, allowing algorithms to know the exact lengths of lines and features on the object. Methods exist for other special objects: 1D objects [39], line segments [18], [37], and spheres [35]. In contrast, DLT [13], DLS [40], and UPnP [26] are single-image methods with lower accuracy, but are more flexible as they can calibrate with arbitrary 3D objects. Our goal is to bridge the gap between object-based methods which performs exceptionally well when given known 3D geometry, to self-calibration which only requires images but has traditionally worse performance.

Self-calibration. Unlike object-based calibration, self-calibration requires 2D observations to perform calibration. A typical pipeline is to generate corresponding points across frames in order to solve $\binom{n}{2}$ fundamental matrices, and solve for the Dual Image of the Absolute Conic (DIAC) which best minimizes Kruppa's constraints [7], [9], [13], [14], [22], [24], [33]. The differing flavors of self-calibration comes from derivation of alternative constraints from the original Kruppa equation. However, we empirically demonstrate that these approaches deteriorates severely with noise in 2D observations and are often unusable in real scenarios where the 2D correspondence information is not always reliable.

We are not the first self-calibration work to use people in order to perform camera calibration. Earlier pedestrian based camera-calibration have been proposed [15], [23], [30], but these methods perform calibration by estimating the vanishing point via the ground plane and detecting upright humans. Although showing considerable promise, these methods do not estimate the 3D shape and will fail if not in view of a flat ground plane while tracking people.

Recently a few deep learning methods are proposed for camera calibration. BPnP [5] creates a differentiable PnP [13] solver. However, we find that BPnP is highly susceptible to noisy data, and cannot provide reliable results for in-the-wild videos. GP2C [10] solves for the fPnP problem without a 3D object, by supervising a network to recognize common household objects and their 3D shapes. GP2C requires training a network with hard-to-acquire ground truth (GT) pose, known 3D objects, and clean testing images with centered and easily visible objects. We show that our

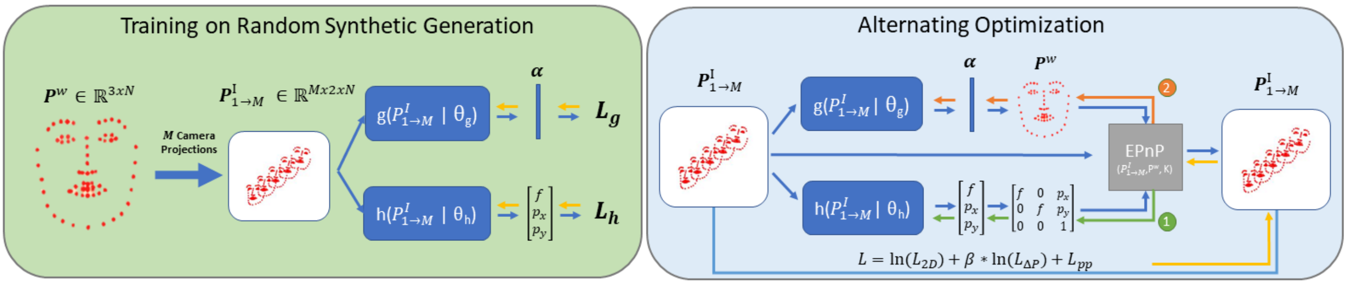
Training on Random Synthetic Generation
$P^w \in \mathbb{R}^{3xN}$
$P^{\mathrm{I}}_{1\to M} \in \mathbb{R}^{Mx2xN}$
M Camera Projections
$g(P^I_{1\to M} \mid \theta_g)$
$h(P^I_{1\to M} \mid \theta_h)$
α
L_g
$\begin{bmatrix} f \\ p_x \\ p_y \end{bmatrix}$
L_h
Alternating Optimization
$P^{\mathrm{I}}_{1\to M}$
P^w
EPnP
$(P^I_{1\to M}, P^w, K)$
$\begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}$
1
2
$L = \ln(L_{2D}) + \beta * \ln(L_{\Delta P}) + L_{pp}$

Fig. 2: Overview of our method for self-calibration using human faces. We use networks $g(\cdot)$ and $h(\cdot)$ to estimate the 3D shape and camera intrinsics. Training is done entirely offline on 3DMM-generated synthetic face landmarks under varying pose and camera intrinsics. Forward processes are represented by blue arrows and colored arrows indicate gradient computation. During testing, our Alternating Optimization alternates between 1) camera intrinsic estimation, and 2) 3DMM face shape estimation.

method is more robust to noise compared to BPnP [5], and outperforms it on both real and synthetic data.

Face Shapes. Recent work has shown success in acquiring wrinkle-level details for the 3D face shape reconstructed from 2D images [3], [11], [20], [31], [32], [41]. Blanz *et al.* [2] have also shown that one can represent faces with metric information by a set of linear bases, forming a 3DMM. 3DMM offers a strong prior to the 3D geometry of faces in videos, while face alignment [3], [17], [20], [20] can acquire the 2D correspondences in a variety of pose and lighting. By utilizing the prior 3D information captured by 3DMM and the 2D face alignment and pose detection, our algorithm is able to robustly acquire camera intrinsic information by observing faces within videos. We take advantage of prior work in faces for our camera calibration method.

III. Methodology

Our algorithm can be summarized into training on synthetic data, and applying optimization during testing, as shown in Fig. 2. We elect to utilize a network based model to overparameterize the unknown 3D shape and camera intrinsics through a neural network. We train two separate networks for the camera intrinsic and 3D shape estimation using 2D facial landmark correspondences across synthetic videos as input. During testing, we optimize the results of this solution by finetuning the weights of both the 3D shape and camera intrinsic estimation networks.

Previous methods typically do not rely on prior data and attempt to directly solve the self-calibration problem of estimating both 3D shape, camera intrinsics, and pose simultaneously. However, camera self-calibration is known to be highly sensitive to noise in 2D correspondences [10], [14], [24], and current formulations which directly solve the self-calibration problem rely on highly accurate correspondences for its estimations. Unlike prior self-calibration work, our method takes advantage of a prior 3DMM representation of a human face to estimate an accurate 3D shape to perform self-calibration. By utilizing a neural network as a model, we can pre-train on synthetic data to predict to predict 3D shape and camera intrinsics from 2D information. Moreover, by finetuning a trained network during testing, our method is more robust and can still perform well under severe noise in 2D correspondences.

A. *Existing Formulas*

Camera Projection. We adopt the standard pinhole camera model. Given a 3D shape and a set of corresponding 2D points, the projection of the 3D points in the world coordinates onto the image coordinates can be explained through the intrinsic and extrinsic matrices up to a scale,

$$\lambda \begin{bmatrix} \mathbf{P}^I_j \\ \mathbf{1} \end{bmatrix} = \mathbf{A}_j \begin{bmatrix} \mathbf{P}^w \\ \mathbf{1} \end{bmatrix}, \tag{1}$$

$$\mathbf{A}_j = \mathbf{K} \left[\mathbf{R} | \mathbf{t} \right]_j, \tag{2}$$

where $\mathbf{P}^I_j$ are the 2D landmarks of the j-th frame in homogeneous coordinates, $j = [1, M]$, M is the frame number, $\mathbf{P}^w$ are the 3D points in homogeneous coordinates, $\mathbf{K}$ is the intrinsic matrix, $\mathbf{R}$ is the 3×3 orthonormal rotation matrix, $\mathbf{t}$ is the translation vector, and λ is the depth in z-direction.

Intrinsic Matrix. Assuming zero skew s and square pixels, we parameterize the intrinsic matrix with 3 degrees of freedom (DoF): the focal length f and principal point p_x, p_y.

$$\mathbf{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

3DMM. We represent the 3D facial landmarks using the formulation defined by Blanz and Vetter [2],

$$\mathbf{P}^w = \bar{\mathbf{S}} + \mathbf{V} * \mathrm{diag}(\boldsymbol{\sigma}) * \boldsymbol{\alpha}, \tag{4}$$

where $\bar{\mathbf{S}}$ is the mean shape, $\mathbf{V}$ is the 3DMM basis vectors, $\boldsymbol{\sigma}$ is the eigenvalues diagonalized in a descending order, and $\boldsymbol{\alpha}$ is the 3DMM coefficients to be estimated. We use the public 3DMM fitter [42] which has 199 basis vectors to define the 3DMM shape.

3DMM offers several advantages. Thanks to the acquisition procedure of gathering 3D face scans, the predicted 3D shape is naturally in metric units, which carries over to enable the pose prediction on the same metric scale. Another is how the usage of 3DMM leverages face priors to predict realistic 3D facial geometry captured during its creation, avoiding solutions with severe perspective ambiguity and unrealistic shape. Finally, using 3DMM reduces the number of unknowns for determining the 3D shape from $3N$ where N is the number of points to the number of basis vectors which is 199 in our case and is typically less than N.

We note that it is possible to add expression basis such as [4], but due to the minimal changes of facial expression in our video data we elect to exclude this extension. We leave the handling of facial expressions to future work.

B. Training

Unlike previous methods, we integrate Deep Neural Networks (DNNs) to predict the 3D shape and the focal length based on the input 2D points. We propose to optimize the DNNs in alternating fashion instead of directly optimizing shape and focal length directly due to the sensitivity of the focal length parameter. Previous methods which directly optimize both shape and focal length together can quickly degenerate to wrong solutions when in the presence of noisy 2D and 3D correspondences. Our method attempts to decrease this sensitivity by introducing a DNNs pretrained on ground truth synthetic data to predict 3D shape and camera intrinsics from 2D points, and finetune the DNNs instead of direct optimization of 3D shape and focal length. By finetuning the DNNs during optimization, we find that we are able to avoid the sensitivity to noise previous self-calibration methods face while also leveraging the pre-trained solution as a reasonable starting point of the optimization problem.

Model. Given 2D points $\mathbf{P}^I_{1\to M} = \mathbf{P}^I_1, ..., \mathbf{P}^I_M$, we define networks $g(\cdot)$ and $h(\cdot)$ parameterized by θ_g and θ_h to predict $\boldsymbol{\alpha}$, the coefficients of the 3DMM [2], and the 3 DoF of the intrinsic matrix: f, p_x, p_y. Since the videos are assumed to have constant intrinsics and 3D shape, we average the $\boldsymbol{\alpha}$ and intrinsic matrix estimation across all frames. We denote the 3D shape prediction network $g(\cdot)$ and intrinsic prediction network $h(\cdot)$ as

$$g(\mathbf{P}^I_{1\to M}; \theta_g) = \boldsymbol{\alpha}, \quad (5) \qquad h(\mathbf{P}^I_{1\to M}; \theta_h) = \boldsymbol{K}. \quad (6)$$

We use a variant of PointNet [28] for both $g(\cdot)$ and $h(\cdot)$. Note that PointNet takes an arbitrary number of points as input. As result, this allows us to define $\mathbf{P}^I \in R^{M\times 2\times N}$. Our method is able to make predictions with a dynamic length on M and N.

Since the original PointNet [28] takes 3D points as input, we make several modification to handle 2D points instead. We first change the 3×3 transformation layer to a 2×2 transformation. Next we adjust the input of the first convolutional layer to take 2D points instead of 3D ones. Finally, the last layer of the network is adjusted to output a 3-dim vector for the intrinsic matrix estimation, and a 199-dim vector representing the 3DMM coefficients.

Synthetic Training. We pre-train both $g(\cdot)$ and $h(\cdot)$ on synthetic data and use them to predict an initial solution at inference. Our synthetic data is generated with known intrinsics and 3D shape. Thus, we directly supervise the networks via a shape loss L_g and focal length loss L_h,

$$L_g = ||\alpha - \tilde{\alpha}||, \tag{7}$$

$$L_h = \left|\left| \left[f, p_x, p_y\right] - \left[\tilde{f}, \tilde{p_x}, \tilde{p_y}\right] \right|\right|, \tag{8}$$

where $\tilde{\alpha}$ and $\tilde{f}$ denote the respective ground truths.

Algorithm 1: Alternating Optimization.

```
Input: $\mathbf{P^I_{1\to M}}$
Output: $\mathbf{P}^w$, $f$, $[\mathbf{R}|\mathbf{t}]_{1\to M}$
$\mathbf{P}^w = g(\mathbf{P}^I_{1\to M}|\theta_g)$;
$\boldsymbol{K} = h(\mathbf{P}^I_{1\to M}|\theta_h)$;
for 1 to q do
    for 1 to 5 do
        $\boldsymbol{K} = h(\mathbf{P}^I_{1\to M}|\theta_h)$;
        $[\mathbf{R}|\mathbf{t}]_{1\to M} = \mathrm{EPnP}(\mathbf{P}^I_{1\to M}, \mathbf{P}^w, \boldsymbol{K})$;
        $L = ln(L_{2D}) + \beta * ln(L_{\Delta\mathbf{P}}) + L_{pp}$;
        $\theta_h = \theta_h - \frac{\partial L}{\partial \theta_h}$;
    end
    for 1 to 5 do
        $\mathbf{P}^w = g(\mathbf{P}^I_{1\to M}|\theta_g)$;
        $[\mathbf{R}|\mathbf{t}]_{1\to M} = \mathrm{EPnP}(\mathbf{P}^I_{1\to M}, \mathbf{P}^w, \boldsymbol{K})$;
        $L = ln(L_{2D}) + \beta * ln(L_{\Delta\mathbf{P}}) + L_{pp}$;
        $\theta_g = \theta_g - \frac{\partial L}{\partial \theta_g}$;
    end
end
```

C. Inference by Alternating Optimization

Compared to previous methods which optimize the 3D shape and intrinsics jointly, we alternate between the optimization of the two networks. We use Adam [16] to optimize the network outputs of Eq. 5 and Eq. 6 in alteration. We depict our optimization process in Alg. 1.

Similar to BPnP [5] we optimize both networks through a PnP algorithm, however we utilize the EPnP algorithm [19] rather than P3P algorithm with RANSAC in [5]. We choose to use the EPnP due to its ability to efficiently handle an arbitrary number of points in $O(n)$ time while being differentiable. By estimating the pose using a closed-form solution with the EPnP, we sidestep the challenge of estimating pose via network. Hence, we are able to finetune the network parameters with emphasis on the 3D shape and focal length, instead of additionally optimizing the pose per frame directly. We denote EPnP as a differentiable function,

$$\mathrm{EPnP}(\mathbf{P}^I_{1\to M}, \mathbf{P}^W, \mathbf{K}) = [\mathbf{R}|\mathbf{t}]_{1\to M}. \tag{9}$$

We implement the EPnP within the automatic differentiation engine of PyTorch [25], in contrast to BPnP [5] which defines their own implicit function for the derivative. Note that the EPnP defines several solutions which can be used to determine the pose. We opt to always use the first linear solution since we observe the initial solution is similar in accuracy to the other solutions. We additionally apply a fixed 10 iterations of Gauss Newton optimization to improve the initial pose estimation. We refer to the original EPnP paper for additional details on the algorithm [19].

Alternating Optimization.

We test both joint optimization of all parameters of $g(\cdot)$ and $h(\cdot)$, but ultimately we find that an alternating optimization strategy is best for this problem. By looking at Eq. 9, it can be observed that the pose of the 3D object is dependent on both $\mathbf{P}^W$ and $\mathbf{K}$. However, both $\mathbf{P}^W$ and $\mathbf{K}$ are highly different quantities with severe differences in both scale and purpose, making the joint optimization of the parameters incredibly difficult. Therefore, we choose to do an alternating optimization of both $g(\cdot)$ and $h(\cdot)$ instead of jointly optimizing both.

To start the optimization, we use the initial network parameters learned during pre-training on synthetic data to predict $\mathbf{P}^w$ and $\mathbf{K}$. We next train the intrinsic network $h(\cdot)$ by predicting the intrinsic matrix $\mathbf{K}$ for 5 iterations, then switch to finetuning the 3D shape network $g(\cdot)$ by predicting $\mathbf{P}^W$ for 5 iterations. We use L_{2D} in Eq. 11 for training both $g(\cdot)$ and $h(\cdot)$. We stop optimization after a fixed global number of iteration q.

Loss. The final loss used in the alternating optimization is

$$L = \ln(L_{2D}) + \beta * \ln(L_{\triangle \mathbf{P}}) + L_{pp}, \tag{10}$$

where β is a fixed variable depending on the magnitude of $L_{\triangle \mathbf{P}}$ at the initial solution.

Using Eq. 4–6, and 9, we solve for the right hand side of Eq. 1 along with the scale λ. We supervise the network θ_g and θ_h using a reprojection loss defined as:

$$L_{2D} = \frac{1}{M} \sum_{j=1}^{M} ||\mathbf{P}^I_j - \lambda \mathbf{A}_j \mathbf{P}^w||_2. \tag{11}$$

We further apply a consistency loss to encourage a smaller range of motion. The consistency error is meant to ensure that the solution across all video frames is consistent with a person's realistic range of motion. Minimizing the reprojection error of Eq. 11 does not necessarily ensure the average motion per frame is as expected for a person, and can cause predictions where the average motion is no longer feasible. The predicted motion can be calculated by projecting the 3D world coordinates to the camera coordinates using the currently predicted pose by Eq. 9: $\mathbf{P}^c_j = \left[\mathbf{R}|\mathbf{t}\right]_j \mathbf{P}^w$. We then define the motion consistency loss as:

$$L_{\triangle \mathbf{P}} = \frac{1}{M} \sum_{j=2}^{M} ||\mathbf{P}^c_j - \mathbf{P}^c_{j-1}||_2. \tag{12}$$

It is typically assumed that the principal point is close to the center of the image. Therefore, we add a strategy proposed by [12] which adds a bias towards the center of the image *a-priori*:

$$L_{pp} = \omega_{pp} || \left[p_x, p_y\right] - \left[\overline{p_x}, \overline{p_y}\right] ||_2. \tag{13}$$

L_{pp} equals to zero when p_x, p_y are at the center of the image $\overline{p_x}, \overline{p_y}$. We set the weight w_{pp} to a small value of $1e^{-5}$.

Similar to Grabner et al. [10] who solves for the logarithmic parameterization of the focal length, we minimize the logarithmic loss of Eq. 11 and Eq. 12. We find that minimization of Eq. 11 has an inherent bias towards larger focal lengths due to an inverse relationship between the gradient magnitude and the focal length. Taking the logarithmic loss removes this bias and ensures a larger gradient at the minimum solution. See supplementary Sec. 5 for more details.

Dataset	Walk	GT int.	GT 3D	# Vid.	# Sub.	Face size (pix^2)	Depth (meters)
Synthetic	✓	✓	✓	50	50	85.53 ± 54.23	1.83 ± 0.67
BIWI	✗	✓	✓	24	24	78.24 ± 10.04	0.95 ± 0.14
BIWI-ID	✓	✓	✗	105	50	25.64 ± 6.84	3.71 ± 1.78
CAD-120	✓	✗	✗	40	4	41.19 ± 6.89	2.11 ± 0.31
Human3.6M	✓	✗	✓	720	6	27.34 ± 8.47	5.08 ± 0.83

TABLE II: Summary of synthetic and real datasets for evaluation.

IV. Experiments

Synthetic Dataset. As our method only works with 2D landmark sequences, it is convenient to generate synthetic data for training and testing. To do so, we randomly generate 3DMM coefficients α from a uniform random distribution with a deviation of 3. The coefficients are then used to compute $\mathbf{P}^w$ via Eq. 4 and project $\mathbf{P}^w$ onto image coordinates using a random camera intrinsic and pose. By defining a random initial and ending pose, we use a smooth spherical linear interpolation of the rotation matrix and a linear interpolation of the translation to generate a 100-frame video of 2D landmarks. We limit pose to a max pitch, yaw, and roll to 30°, and limit translation to ensure faces $1-4$ meters from the camera. We generate all synthetic videos *on the fly* during training. For testing we synthesize 5 videos for each focal length within $[500, 1400]$ pixels at intervals of 100 totaling 50 videos. For each video, we sample the principal point from a 2D Gaussian distribution at the image center with a 10-pixel STD in both px and py. Similar to [10], all evaluations is the median error over all video sequences.

Real Datasets. While there are many public face datasets, few come with GT in either the intrinsics, depth or 3D shapes, which are necessary for quantitative evaluation. We therefore utilize 4 RGBD human datasets which we preprocess using automatic face detection [36], cropping [34], and alignment [42]. A summary of each dataset and the data it provides is shown in Tab. II.

While Human3.6M contains 11 subjects, we only test on 6 subjects where GT depth and 3D shape are available. For Human3.6M, the dataset surrounds subjects with 6 cameras including angles behind the individual. We therefore remove any cameras and frames where the subject is not facing the camera.

For Human3.6M and CAD-120 which does not provide GT camera intrinsics, we utilize the depth information to acquire pseudo ground truth camera intrinsics. We list further details on preprocessing on supplementary materials.

Implementation Details. For training and inference, we use Adam [16] optimizer, with a learning rate of $1e$-4 for $h(\cdot)$, and $5e$-1 for $g(\cdot)$. For all experiments, we set $\beta = 1e$-3 for walking sequences and $1e$-1 for still sequences as walking has a larger range of motion. To make sequence-level decision on walking vs. still at inference, we compute Eq. 12 under the initial solution and classify videos as still if $L_{\triangle \mathbf{P}}$ is less than 40mm between frames. for both networks, we utilize the point net implementation by [27] with our added modifications for handling 2D points instead of 3D points.

Baselines. We select baselines that are representative of both object-based calibration and self-calibration. Some baselines do not estimate a particular error metric due to a lack of

Method	e_f	e_d	e_{3D} (mm)	e_{px} (pix)	e_{py} (pix)	e_{2D} (pix)
DLT [13]	0.546	0.436	$\bar{\mathbf{S}}$	0.386	1.069	0.633
UPnP [26]	0.427	0.579	$\bar{\mathbf{S}}$	–	–	2.908
Hartley [14]	0.302	–	–	0.548	0.566	–
Louraki [14]	0.103	–	–	0.168	0.220	–
Fetzer [14]	0.137	–	–	0.022	0.034	–
BPnP [5]	0.165	0.142	2.500	0.016	0.021	0.029
NN (ours)	0.171	0.226	3.470	0.105	0.293	0.911
NN+BPnP	0.167	0.166	3.459	0.001	0.001	0.882
NN+JO	0.170	0.164	3.211	0.001	0.003	1.151
NN+SO	0.212	0.192	3.212	0.001	0.001	0.709
NN+AO (ours)	0.090	0.107	2.988	0.006	0.017	0.265

TABLE III: Synthetic test set evaluation. For methods that do not output 3D shapes, $\bar{\mathbf{S}}$ denotes using the mean 3D face to localize landmarks. Our pretrained network is denoted by NN, and BPnP, JO, SO, AO denote optimization strategies. [Key: Best, Second Best]

3D estimation or pose estimation, and cannot handle the scenario we care for (any video with a human face facing the camera). Additionally, certain baselines are excluded if they cannot be handle camera calibration on videos of faces. Current pedestrian based camera-calibration methods [15], [21] cannot be applied in indoor scenarious where full view of the ground plane is not visible along with multiple pedestrians due to the methods being based on pedestrian height distribution.

Object-based calibration baselines [13], [26] are included to show the effects of an incorrect 3D shape when performing object-based calibration. Self-calibration methods [5], [9], [14], [22], [24] can consume the detected facial landmarks on unconstrained scenes without assumption on the 3D shape. However, these methods only estimate the camera intrinsics and do not additionally predict the 3D shape, depth, or reprojection error. Methods such as [9], [14], [22], [24] are SOTA traditional solutions without deep learning. The recently introduced BPnP [5] proposes a SOTA deep learning solution to camera calibration from 2D correspondences. We show that our method performs better against all previous algorithms on both our synthetic and publicly available real data. To compare with BPnP optimization, NN+BPnP uses our pretrained face network to provide an initial solution, then we optimize $f(\cdot), g(\cdot)$ using BPnP.

In order to show the advantage of our alternating optimization (AO) strategy, we additionally perform joint optimization (JO) and sequential optimization (SO) on our initial NN solution. Instead of optimizing both the 3D shape and the camera intrinsic in an alternating fashion, (SO) looks to first optimize the 3D shape until convergence, and then the camera intrinsic afterwards. (JO) optimizes both the 3D shape and the camera intrinsic at the same time.

Error Metrics. We report the relative *depth* error, $e_d = \frac{1}{M}\sum_{j=1}^{M} ||m(\mathbf{p}_j^c) - m(\tilde{\mathbf{p}_j^c})||_2/||m(\tilde{\mathbf{p}_j^c})||_2$, relative focal length error, $e_f = |f - \tilde{f}|/\tilde{f}$, relative p_x error, $p_x = |p_x - \tilde{p_x}|/\tilde{p_x}$, and relative p_y error, $p_y = |p_y - \tilde{p_y}|/\tilde{p_y}$ following recent work [10], where $m(\mathbf{p})$ computes the mean of 68 landmarks in $\mathbf{p}$. To understand the overall alignment of the predicted pose and 3D shape, we further report the 2D reprojection error in pixels and the 3D shape error, $e_{3D} = ||\mathbf{P}^w - \tilde{\mathbf{P}}^w||_2$, in millimeters. The 2D reprojection error e_{2D} essentially equals to L_{2D} in Eq. 11.

Dataset	Method	e_f	e_d	e_{3D} (mm)	e_{px} (pix)	e_{py} (pix)	e_{2D}
BIWI	DLT [13]	1.209	5.765	$\bar{\mathbf{S}}$	0.595	1.279	0.356
	UPnP [26]	4.117	4.133	$\bar{\mathbf{S}}$	–	–	3.245
	Hartley [14]	0.754	–	–	0.251	0.400	–
	Louraki [22]	0.426	–	–	0.268	0.228	–
	Fetzer [9]	0.538	–	–	0.007	0.013	–
	BPnP [5]	0.664	0.683	9.936	0.013	0.018	0.440
	NN (ours)	0.669	0.691	10.339	0.100	0.288	0.939
	NN+AO (ours)	0.101	0.091	10.176	0.064	0.176	0.745
BIWI RGBD-ID	DLT [13]	1.632	1.543	$\bar{\mathbf{S}}$	0.080	1.193	0.116
	UPnP [26]	1.206	1.178	$\bar{\mathbf{S}}$	–	–	1.007
	Hartley [14]	0.744	–	–	0.250	0.324	–
	Louraki [22]	0.662	–	–	0.387	0.222	–
	Fetzer [9]	0.845	–	–	0.001	0.005	–
	BPnP [5]	0.675	0.672	–	0.322	0.479	0.304
	NN (ours)	0.220	0.379	–	0.088	0.274	0.428
	NN+AO (ours)	0.133	0.231	–	0.026	0.042	0.377
CAD-120	DLT [13]	1.788	1.202	$\bar{\mathbf{S}}$	0.265	0.861	0.274
	UPnP [26]	7.215	6.894	$\bar{\mathbf{S}}$	–	–	4.110
	Hartley [14]	0.784	–	–	0.192	0.306	–
	Louraki [22]	0.732	–	–	0.255	0.180	–
	Fetzer [9]	0.679	–	–	0.001	0.005	–
	BPnP [5]	1.178	1.673	–	0.103	0.129	5.873
	NN (ours)	0.237	0.343	–	0.359	0.376	0.785
	NN+AO (ours)	0.151	0.126	–	0.023	0.063	0.376
Human3.6M	DLT [13]	1.289	1.227	$\bar{\mathbf{S}}$	0.228	0.453	0.182
	UPnP [26]	0.436	0.421	$\bar{\mathbf{S}}$	–	–	3.056
	Hartley [14]	0.779	–	–	0.134	0.377	–
	Louraki [22]	0.618	–	–	0.201	0.202	–
	Fetzer [9]	0.400	–	–	0.359	0.520	–
	BPnP [5]	0.366	0.902	9.714	0.016	0.017	3.114
	NN (ours)	0.409	0.370	6.673	0.410	0.394	1.763
	NN+AO (ours)	0.350	0.318	6.014	0.006	0.006	0.789

TABLE IV: Real dataset evaluation. We estimate e_{3D} whenever GT is available. $\bar{\mathbf{S}}$ denotes we use the mean 3D face to predict the intrinsics and pose. [Key: Best, Second Best].

A. Synthetic Data

Tab. III summarizes comparisons on the synthetic test set. We emphasize that our method can estimate the camera intrinsic, relative depth, and 3D shape at comparable accuracy compared to prior work with only the base algorithm denoted by NN, supporting the usage of a neural network for inference of the camera intrinsic. We validate that our alternating optimization strategy outperforms the recently proposed BPnP, and simple strategy of (JO) and (SO). It can be seen that our method reduces errors e_f, e_d, e_{px}, e_{py} on the entire synthetic dataset compared to SOTA methods. We note that e_{2D} is a commonly used objective function for existing methods [5], [10], [13], [26], but minimizing this error is not always reflective of better camera intrinsic or depth estimation, which are the most significant metrics in practice.

B. Real Data

Our results on real datasets are summarized in Tab. IV. Compared to baselines, our method has lowest errors on e_f and e_d on *all* RGBD face datasets. We consistently find that object based methods such as DLT and UPnP which assumes the 3D shape as the mean 3DMM to perform the worst. Previous SOTA self-calibration methods which require clean 2D correspondences for accurate fundamental matrix estimation cannot handle the noise present in the 2D correspondences of the real dataset. Compared to their synthetic dataset counterpart in Tab. III, self-calibration algorithms perform much worse in Tab. IV. However, our method still provides reliable results despite noisy 2D correspondences. Additionally, our optimization strategy consistently reduces all errors of the initial solution provided by our pretrained model.

Problem	Setting	e_f	e_d	e_{3D} (mm)	e_{px}	e_{py}	e_{2D} (pix)
Calib.	Ideal	0.001	0.002	–	0.000	0.000	0.007
SfM		–	0.003	1.719	–	–	0.141
Calib.	Non-ideal	0.182	0.176	3.813	0.000	0.001	1.340
SfM		1.336	1.285	3.182	0.000	0.000	1.156

TABLE V: Optimization of 3D shape and focal length separately under ideal and semi-ideal setting, on the synthetic test set.

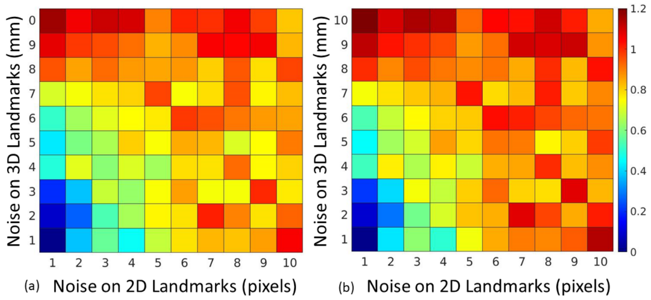

Fig. 3: Relative depth error (a) and focal length error (b) at different amounts of noise in 2D landmarks and 3D shape.

C. Ablation Study

We perform ablation on both synthetic and real data by leveraging the numerous RGBD datasets gathered.

3D Shape vs Calibration. We wish to better understand the importance of predicting 3D shape as part of the camera calibration problem. Therefore, we propose to investigate the optimization of the camera intrinsic and the 3D shape under ideal and non-ideal conditions to get the upper and lower bound of our method. We show this analysis in Tab V. Assuming there is always some noise in 2D, the ideal condition is when 3D, or camera intrinsic is known, and the last unknown variable is estimated. The non-ideal condition assumes a mean 3D shape or a focal length of $2,000$ instead of the ground truth, and again estimates the unknown variable.

Both camera intrinsic and 3D shape optimization perform accurately in the ideal setting. However, both problems break down in the non-ideal case. This supports the idea that the 3D shape and the camera intrinsic problem must be solved jointly in order to reach the preferable lower bound performance. Indeed, our results in Tab. III are closer to the upper bound e_f and e_d of the ideal condition in Tab. V.

Noise Robustness. We add varying amount of noise to the GT 3D shape and 2D landmarks of the synthetic test set, to understand the required accuracy of 2D face alignment and 3D estimation. We plot heat maps of focal length and depth error with varying noises in Fig. 3. For each noise combination, we select the focal length which minimizes Eq. 11 by exhaustively searching a focal length range in integer intervals.

In practice the focal length tends to be inaccurate with either noise in 2D or 3D. The worst case appears to be when the 3D noise is the largest and 2D noise is the lowest. In this case, calibration attempts to fit an incorrect 3D shape onto 2D landmarks throughout the video. Despite the correct landmarks, with the wrong 3D shape the predicted focal length under the objective will be incorrect. Thus, 3D shape prediction is crucial for an accurate focal length estimation.

Next, we compare with other self-calibration methods in 2D noise robustness. We add 2D noise from $[0, 5]$ at intervals of 0.5 to our entire synthetic video test set and show the results in Fig. 4. We plot the median and standard deviation over 20 trials for e_f, e_d, e_{3D}, and $e_{pc} = \frac{1}{w}||[p_x \quad p_y] - [\tilde{p_x} \quad \tilde{p_y}]||_2$, where w is the image width. Note that our method remains reliable despite noise added to 2D landmarks, whereas all prior self-calibration algorithms have significantly worse performance when noise is added. Compared to BPnP, our optimization is worse at zero noise for 3D estimation, but significantly outperforms it at noise levels greater than 0.5. This validates our method better handles the noisier real data over previous methods.

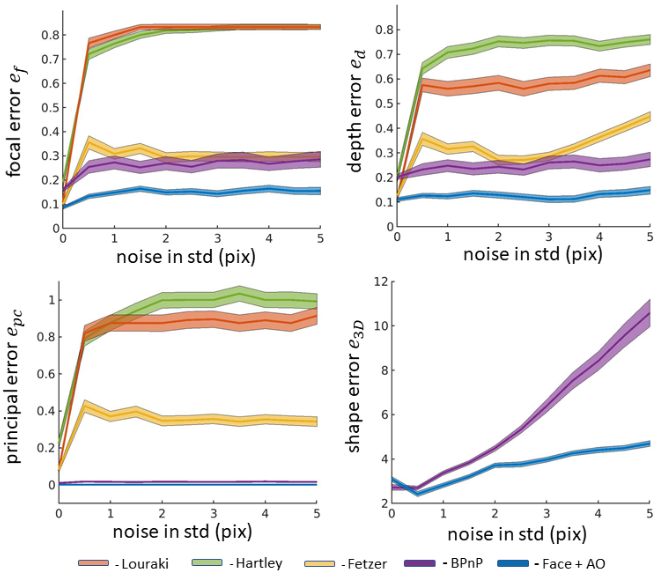

Fig. 4: Focal length, depth, principal point, and shape errors on the synthetic test set at varying levels of noises added to 2D landmarks. Each point shows the median and STD over 20 runs.

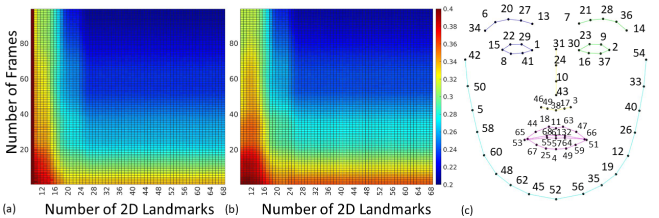

Fig. 5: Relative depth error (a) and focal length error (b) vs video lengths and landmarks, selected w.r.t. the order in (c).

Impact of Video Length and Landmarks on Synthetic. We study the impact of the video length and the number of landmarks on both synthetic and real data. We plot heatmaps of the relative focal length and depth errors by varying the number of video frames and landmarks, using our pre-trained network in Fig. 5. To form a subset of 68 landmarks, we follow a fixed order to evenly sample 5 face regions: left eye, right eye, nose, mouth, and contour. Without surprise, less frames and landmarks lead to lower performance. Further, Fig. 5 suggests a favorable balance is at ≈30 landmarks and ≈50 frames, if low computation is desirable. It is important to have algorithms handling both image and video input for this problem, as current calibration is highly susceptible to outliers within the data. Hence, by leveraging multiple frames and landmarks, our method becomes more robust.

Impact of Video Length and Depth on Real Data. We

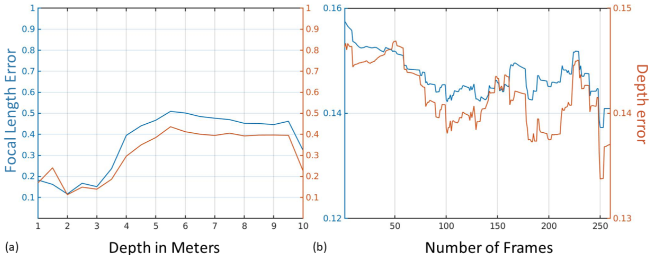

Fig. 6: Relative focal length error (a) and depth error (b) at varying depths and video lengths on all 4 real datasets.

bin the GT depth at intervals of 0.5 from 1 to 10 meters, then computing the average error across *all* real datasets as plotted in Fig. 6(a). Intuitively, errors are higher for frames further away. This effect becomes particularly noticeable after 4 meters, indicating the difficulty of camera calibration in the Human3.6M dataset whose depth is ≈ 5 meters. We attribute this to the small face size (27 pixels as in Tab. II) and the resultant noisy landmarks on low-resolution imagery. Our method may operate on faces with much larger GT depth, as high-resolution videos increase in availability.

We validate the impact of video length on real data in Fig. 6(b). While the overall performance improves with more frames, it does fluctuate more than Fig. 5, likely due to much more short real videos than longer ones, and more "irregular" noise present in real data.

V. Conclusion

This work carries out camera self-calibration using faces without knowing the exact 3D geometry. We propose to predict the 3D shape and camera intrinsic under alternating optimization while using a differentiable EPnP pose estimation derived from both predictions as supervision. By using a face as the calibration object, we determine the metric distance to the person in the camera, and operate on a large range of videos. We compare with current SOTA on self-calibration algorithms on human faces and show that ours is repeatedly best in 3D estimation and calibration performance.

References

[1] A. M. Andrew. Multiple view geometry in computer vision. *Kybernetes*, 2001.

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

[3] A. Bulat and G. Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *CVPR*, 2017.

[4] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. *TVCG*, 20(3), 2013.

[5] B. Chen, T.-J. Chin, and N. Li. Bpnp: Further empowering end-to-end learning with back-propagatable geometric optimization. *CVPR*, 2019.

[6] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *CVIU*, 61(1), 1995.

[7] D. DeMenthon and L. S. Davis. Exact and approximate solutions of the perspective-three-point problem. *TPAMI*, 14(11), 1992.

[8] O. D. Faugeras, Q.-T. Luong, and S. J. Maybank. Camera self-calibration: Theory and experiments. In *ECCV*. Springer, 1992.

[9] T. Fetzer, G. Reis, and D. Stricker. Stable intrinsic auto-calibration from fundamental matrices of devices with uncorrelated camera parameters. In *WCACV*, pages 221–230, 2020.

[10] A. Grabner, P. M. Roth, and V. Lepetit. Gp2c: Geometric projection parameter consensus for joint 3d pose and focal length estimation in the wild. In *CVPR*, pages 2222–2231, 2019.

[11] J. Guo, X. Zhu, Y. Yang, F. Yang, Z. Lei, and S. Z. Li. Towards fast, accurate and stable 3d dense face alignment. In *ECCV*, 2020.

[12] R. Hartley, C. Silpa-Anan, et al. Reconstruction from two views using approximate calibration. In *ACCV*, volume 1, pages 338–343, 2002.

[13] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2 edition, 2003.

[14] R. I. Hartley. Kruppa's equations derived from the fundamental matrix. *TPAMI*, 19(2):133–135, 1997.

[15] S. Huang, X. Ying, J. Rong, Z. Shang, and H. Zha. Camera calibration from periodic motion of a pedestrian. In *CVPR*, pages 3025–3033, 2016.

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2014.

[17] A. Kumar*, T. K. Marks*, W. Mou*, Y. Wang, M. Jones, A. Cherian, T. Koike-Akino, X. Liu, and C. Feng. Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood. In *CVPR*, Seattle, WA, June 2020.

[18] J. Lee, H. Go, H. Lee, S. Cho, M. Sung, and J. Kim. Ctrl-c: Camera calibration transformer with line-classification. In *ICCV*, pages 16228–16237, 2021.

[19] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o (n) solution to the pnp problem. *IJCV*, 81(2):155, 2009.

[20] F. Liu, D. Zeng, Q. Zhao, and X. Liu. Joint face alignment and 3d face reconstruction. In *ECCV*. Springer, 2016.

[21] J. Liu, R. T. Collins, and Y. Liu. Surveillance camera autocalibration based on pedestrian height distributions. In *BMVC*, page 144, 2011.

[22] M. I. Lourakis and R. Deriche. *Camera self-calibration using the singular value decomposition of the fundamental matrix: From point correspondences to 3D measurements*. PhD thesis, INRIA, 1999.

[23] F. Lv, T. Zhao, and R. Nevatia. Camera calibration from video of a walking human. *TPAMI*, 28(9):1513–1518, 2006.

[24] P. R. Mendonça and R. Cipolla. A simple technique for self-calibration. In *Cat. No PR00149*, volume 1, pages 500–505. IEEE, 1999.

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *NEURIPS*, pages 8024–8035. 2019.

[26] A. Penate-Sanchez, J. Andrade-Cetto, and F. Moreno-Noguer. Exhaustive linearization for robust camera pose and focal length estimation. *TPAMI*, 35(10), 2013.

[27] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CVPR*, 2016.

[28] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017.

[29] H. Rastgar. *Robust self-calibration and fundamental matrix estimation in 3D computer vision*. PhD thesis, Université d'Ottawa/University of Ottawa, 2013.

[30] Z. Tang, Y.-S. Lin, K.-H. Lee, J.-N. Hwang, and J.-H. Chuang. Esther: Joint camera self-calibration and automatic radial distortion correction from tracking of walking humans. *IEEE Access*, 7:10754–10766, 2019.

[31] L. Tran and X. Liu. Nonlinear 3d face morphable model. In *CVPR*, 2018.

[32] L. Tran and X. Liu. On learning 3d face morphable model from in-the-wild images. *TPAMI*, 2019.

[33] B. Triggs. Autocalibration and the absolute quadric. In *CVPR*, 1997.

[34] D. Wang, C. Otto, and A. K. Jain. Face search at scale: 80 million gallery. *arXiv preprint arXiv:1507.07242*, 2015.

[35] H. Zhang, K. W. Kwan-yee, and G. Zhang. Camera calibration from images of spheres. *TPAMI*, 29(3):499–502, 2007.

[36] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *SPL*, 23(10), 2016.

[37] Y. Zhang, L. Zhou, H. Liu, and Y. Shang. A flexible online camera calibration using line segments. *Journal of Sensors*, 2016, 2016.

[38] Z. Zhang. A flexible new technique for camera calibration. *TPAMI*, 22(11):1330–1334, 2000.

[39] Z. Zhang. Camera calibration with one-dimensional objects. *TPAMI*, 26(7), 2004.

[40] Y. Zheng and L. Kneip. A direct least-squares solution to the pnp problem with unknown focal length. In *CVPR*, 2016.

[41] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. In *CVPR*, pages 146–155, 2016.

[42] X. Zhu, X. Liu, Z. Lei, and S. Z. Li. Face alignment in full pose range: A 3d total solution. *TPAMI*, 41(1), 2017.