Supplementary Material: Modeling the Effects of Windshield Refraction for Camera Calibration

Frank Verbiest[1], Marc Proesmans[1], and Luc Van Gool[1,2]

[1] Center for Processing Speech and Images, ESAT-PSI, KU Leuven
[2] Computer Vision Lab, D-ITET, ETH Zürich

1 Introduction

In this document, we provide some additional details on the optimization of the spherical and spline model here below in Section 2. In Section 3, we present additional results using error plots, we illustrate the effect of the amount of spline patches on the final results, and give some insight in the geometry parameters that come out as a result of the optimization process. Finally, in the supplementary material we also included a video of a ray-traced simulation that illustrates the effect of refraction through a spherically curved sheet of glass (with radius $R = 2$). It also shows that the effects of refraction through a flat sheet of glass ($R \to \infty$) are very moderate, but as soon as a bit of curvature is applied, the effects of refraction become substantial and very noticeable.

2 Spherical Model Optimization

2.1 Objective Function

For a given set of 2D-3D correspondences $(\mathbf{m}_k, \mathbf{M}_k)$, the parameters $\boldsymbol{\theta}^*_{sph}$ of the spherical model and the parameters $\boldsymbol{\theta}^*_{cam}$ of the camera behind the windshield are obtained by solving the following non-linear least squares problem,

$$\boldsymbol{\theta}^*_{cam}, \boldsymbol{\theta}^*_{sph} = \arg \min_{\boldsymbol{\theta}_{cam}, \boldsymbol{\theta}_{sph}} E_{repr.sph.} \ , \tag{1}$$

where

$$E_{repr.sph.} = \sum_k \left\| \mathbf{g}(\mathbf{M}_k; \boldsymbol{\theta}_{cam}, \boldsymbol{\theta}_{sph}) - \mathbf{m}_k \right\|^2 \tag{2}$$

is the energy function that is minimized and $\mathbf{g}$ is the projection function for the spherical model which is explained in the Sections 2 and 3 of the paper. Although the problem is easily stated in equation (1), there are a series of steps involved to arrive at a solution. We will go into these steps next.

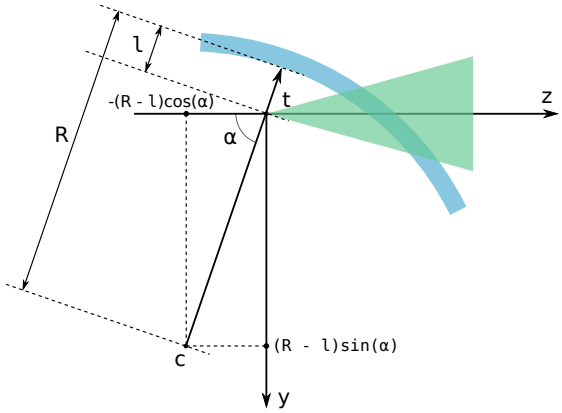

Fig. 1. Initialization of the spherical model: Expressed in camera coordinates, $\mathbf{t}$ is the camera projection center (also the origin in the camera coordinate frame), $\mathbf{c} = [0\ (R-l)\sin\alpha\ \ -(R-l)\cos\alpha]^\top$ is the center of the sphere, R is the radius of the sphere, l is the distance of the camera center to the windshield, and α defines the orientation of the windshield.

2.2 Optimization Procedure

The steps to arrive at a solution for equation (1) are the following:

- Initially, we assume no refraction and solve a regular absolute pose problem with unknown focal length and radial distortion by using the minimal solver [5] in a RANSAC [2] loop. Afterwards, the solution is refined by minimizing the re-projection error [3] formulated as a non-linear least squares problem [1] with a robust loss function (M-estimators [4]). The scale parameters for the robust loss function is set to the *Median Absolute Deviation* (MAD) [6] of the residuals.
- Now that we have an initialization for the camera parameters, we turn our attention to the spherical refraction parameters. With knowledge of the construction of the car, we roughly know how the windshield is configured relative to the camera, which allows to initialize the parameters of the spherical model. A prototypical configuration, in camera coordinates, is shown in Figure 1, where, *e.g.*, $l = 0.05$ (in m), $R = 3$ (in m), and $\alpha = 70$ (in degrees) (for the remaining parameters see below). The introduction of the spherical refraction model causes the image projections to shift.
- With the rough initialization of the spherical refraction parameters, we can improve on the camera initialization by optimizing the non-linear least squares problem in equation (1) over the camera parameters $\boldsymbol{\theta}_{cam}$ only and keep the spherical refraction parameters $\boldsymbol{\theta}_{sph}$ fixed relative to camera (*i.e.*, $\mathbf{c}$ and R are fixed relative to the camera during optimization). As before, to arrive at a robust solution, the non-linear least squares problem is modified with a robust loss function (M-estimators) that has an appropriate scale parameter (see MAD above).

– Finally, by jointly optimizing the parameters of the camera and the spherical refraction model, the calibration is obtained. Again, the optimization is done robustly.

As for the remaining spherical refraction parameters, the refraction indices n_1, n_2, and the thickness of the glass d, they are either known or can be measured beforehand. Typically, in our case, $n_1 = 1$ for air and $n_2 = 1.5$ for glass, and $d = 5.3$ (in mm) for the thickness of the glass.

We verified that the above initialization values for $l, R, \alpha, n_1, n_2, d, ..$ have been appropriate for all windshields and camera setups we tested.

Also, to prevent negative values for the radius R during optimization, we optimize for γ instead with $R = \gamma^2$. To ensure that the camera projection center is always inside the sphere during optimization, we write the position of the sphere relative to the camera as $\mathbf{c} = \eta R\mathbf{h}$, where $\mathbf{h}$ is a direction vector required to be of unit length and $\eta \in [0, 1]$ denotes a fraction of the radius R, so that $\eta R \in [0, R]$ and $\|\mathbf{c}\| \leq R$. To parametrize the direction vector $\mathbf{h}$, we refer to [3] (see section A6.9.3 p. 625, *Parametrization of the n-sphere*). We keep $\eta \in [0, 1]$, by parametrizing it as follows,

$$\eta = \frac{\frac{\pi}{2} + \tan^{-1}\beta}{\pi} \quad \text{with} \quad \beta \in [-\infty, +\infty] \ . \tag{3}$$

2.3 Objective Function

For a given set of 2D-3D correspondences $(\mathbf{m}_k, \mathbf{M}_k)$, the parameters $\boldsymbol{\theta}^*_{spl}$ of the spline model are obtained by solving the following non-linear least squares problem,

$$\boldsymbol{\theta}^*_{spl} = \arg\min_{\boldsymbol{\theta}_{spl}} E_{repr.spl.} + \lambda E_{thinplate} \ , \tag{4}$$

where

$$E_{repr.spl.} = \sum_k \|\mathbf{g}(\mathbf{M}_k; \boldsymbol{\theta}^*_{cam}, \boldsymbol{\theta}_{spl}) - \mathbf{m}_k\|^2 \tag{5}$$

is the energy function associated with the re-projection error and $\mathbf{g}$ is the projection function for the spline model which is explained in the Sections 2 and 4 of the paper. Here, the camera parameters are kept fixed during the optimization, specifically, they are set to the results of the spherical refraction model optimization. To prevent over-fitting, a thin-plate regularization term $E_{thinplate}$ is used, more details are given below. The λ is determined automatically by splitting the 2D-3D correspondences in a training set and a test set. The training set is used during optimization, and the resulting parameters are verified against the test set to find the optimal λ, *i.e.*, minimal re-projection error on the test set. For a given λ, the steps of a single optimization run of equation (4) are given in the next section.

2.4 Optimization Procedure

The steps to arrive at a solution for equation (4) are the following:

- For the different bi-cubic spline patches, initialise the parameters of the nodes (that define the splines) by setting them to zero. This corresponds to a situation without refraction effects.
- Robustly optimize the non-linear least squares problem of equation (4), where, as before, robust refers to the modification of the re-projection residuals with a robust loss function that has an appropriate scale parameter.

2.5 Implementation Details

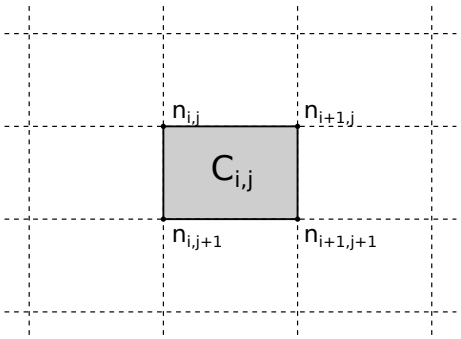

Fig. 2. Illustration of a patch and the nodes that define bi-cubic spline function.

In this section, we provide some details on the evaluation of the spline functions and the corresponding thin-plate energy. Consider a 3D point $p = [X\,Y\,Z]^\top$ expressed in camera coordinates (see equation (P.3)[3]) and using equation (P.7), assuming no refraction, we obtain the normalized projection or normalized image coordinates as

$$u = X/Z \tag{6}$$

$$v = Y/Z \tag{7}$$

Now, each of the horizontal and vertical components of $\mathbf{f}_1$ and $\mathbf{f}_2$ (see Section 4 of the paper) that require evaluation represent a grid of bi-cubic spline patches, let $s(u, v)$ be such function. To evaluate $s(u, v)$, we need to access the specific bi-cubic spline patch (and the associated nodes holding function values and derivatives) that corresponds to the normalized projection $[u\,v]^\top$. For this, we

[3] Here, we refer to an equation (x) in the paper as (P.x)

first convert the normalized image coordinates to "spline coordinates" as follows,

$$u' = N_i \frac{u - u_{min}}{u_{max} - u_{min}} \tag{8}$$

$$v' = N_j \frac{v - v_{min}}{v_{max} - v_{min}} \ , \tag{9}$$

where $[u_{min}, u_{max}] \times [v_{min}, v_{max}]$ is the region of normalized image space covered by the spline function $s(u, v)$, and N_j and N_i are the number of rows and columns of the grid of spline patches, respectively. This makes that $u' \in [0, N_i]$ and $v' \in [0, N_j]$. The column index i and row index j of the spline patch of interest are given by (using the operator $\lfloor x \rfloor$ for the floor of x),

$$i = \lfloor u' \rfloor \tag{10}$$

$$j = \lfloor v' \rfloor \ . \tag{11}$$

Similarly, we define "patch coordinates" as follows,

$$u'' = u' - \lfloor u' \rfloor \tag{12}$$

$$v'' = v' - \lfloor v' \rfloor \ , \tag{13}$$

so that $u'' \in [0, 1]$ and $v'' \in [0, 1]$. The bi-cubic spline function associated with this patch is denoted with $p_{i,j}(u'', v'')$, an illustration is given in Figure 2. For notational simplicity, we will drop the indices and use u and v for patch coordinates instead. The patch function is then given by,

$$p(u, v) = \left[u^3\, u^2\, u\, 1\right] \mathbf{C} \left[v^3\, v^2\, v\, 1\right]^\top \ , \tag{14}$$

where the coefficients of the 4×4 matrix $\mathbf{C}$ are completely determined by the function values and the derivatives stored at the four corners of the patch, the corners being $(0, 0)$, $(1, 0)$, $(0, 1)$, and $(1, 1)$. By evaluating p and the derivatives $\frac{\partial p}{\partial u}$, $\frac{\partial p}{\partial v}$, and $\frac{\partial^2}{\partial u \partial v}$ at the 4 corners and matching them with the values stored in the corresponding nodes yields 16 linear equations in the 16 coefficients,

$$vec(\mathbf{C}) = \mathbf{A}^{-1}\mathbf{z} \tag{15}$$

with

$$\mathbf{A} = \begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1\\
0&0&0&3&0&0&0&2&0&0&0&1&0&0&0&0\\
0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0\\
0&0&3&0&0&0&2&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1\\
0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&3&2&1&0\\
0&0&0&0&0&0&0&0&3&2&1&0&0&0&0&0\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
3&3&3&3&2&2&2&2&1&1&1&1&0&0&0&0\\
3&2&1&0&3&2&1&0&3&2&1&0&3&2&1&0\\
9&6&3&0&6&4&2&0&3&2&1&0&0&0&0&0
\end{bmatrix} \quad \text{and} \quad \mathbf{z} = \begin{bmatrix} \mathbf{n}_{0,0} \\ \mathbf{n}_{1,0} \\ \mathbf{n}_{0,1} \\ \mathbf{n}_{1,1} \end{bmatrix} \quad , \tag{16}$$

where $vec(\mathbf{C})$ is the row-wise vectorization of the matrix $\mathbf{C}$, and $\mathbf{n}_{0,0}$, $\mathbf{n}_{1,0}$, $\mathbf{n}_{0,1}$, and $\mathbf{n}_{1,1}$ are 4-vectors or nodes holding the values and derivative parameters for the different corners. Specifically, for some corner, the 4 values stored in any such node $\mathbf{n}$ are matched against $[p \ \frac{\partial p}{\partial u} \ \frac{\partial p}{\partial v} \ \frac{\partial^2}{\partial u \partial v}]^\top$. Given the parameters stored in the nodes, the patch function can be evaluated using equations (14) and (15).

To compute the thin-plate energy $E_{thinplate}$ in equation (4), we need to, for each of the horizontal and vertical components of $\mathbf{f}_1$ and $\mathbf{f}_1$, sum the individual thin-plate energies for all the patches of the grid. The individual thin-plate energy for a patch is given by,

$$E = \iint_\Omega \left(\left(\frac{\partial^2 p}{\partial u^2} \right)^2 + 2 \left(\frac{\partial^2 p}{\partial u \partial v} \right)^2 + \left(\frac{\partial^2 p}{\partial v^2} \right)^2 \right) du \, dv \ . \tag{17}$$

Combining this with the expression for the patch function in equation (14) yields,

$$E = vec(\mathbf{C})^\top \mathbf{B} \, vec(\mathbf{C}) \tag{18}$$

with

$$\mathbf{B} = \begin{bmatrix}
\frac{1734}{175} & \frac{289}{35} & 6 & 3 & \frac{289}{35} & \frac{13}{2} & \frac{21}{5} & \frac{3}{2} & 6 & \frac{21}{5} & 2\,0 & 3 & \frac{3}{2} & 0\,0 \\
\frac{289}{35} & \frac{272}{35} & \frac{33}{5} & 4 & \frac{13}{2} & \frac{88}{15} & \frac{9}{2} & 2 & \frac{21}{5} & \frac{52}{15} & 2\,0 & \frac{3}{2} & 1 & 0\,0 \\
6 & \frac{33}{5} & \frac{38}{5} & 6 & \frac{21}{5} & \frac{9}{2} & 5 & 3 & 2 & 2 & 2\,0 & 0 & 0\,0 & 0 \\
3 & 4 & 6 & 12 & \frac{3}{2} & 2 & 3 & 6 & 0 & 0 & 0\,0 & 0 & 0\,0 & 0 \\
\frac{289}{35} & \frac{13}{2} & \frac{21}{5} & \frac{3}{2} & \frac{272}{35} & \frac{88}{15} & \frac{52}{15} & 1 & \frac{33}{5} & \frac{9}{2} & 2\,0 & 4 & 2\,0 & 0 \\
\frac{13}{2} & \frac{88}{15} & \frac{9}{2} & 2 & \frac{88}{15} & \frac{232}{45} & \frac{11}{3} & \frac{4}{3} & \frac{9}{2} & \frac{11}{3} & 2\,0 & 2 & \frac{4}{3} & 0\,0 \\
\frac{21}{5} & \frac{9}{2} & 5 & 3 & \frac{52}{15} & \frac{11}{3} & 4 & 2 & 2 & 2 & 2\,0 & 0 & 0\,0 & 0 \\
\frac{3}{2} & 2 & 3 & 6 & 1 & \frac{4}{3} & 2 & 4 & 0 & 0 & 0\,0 & 0 & 0\,0 & 0 \\
6 & \frac{21}{5} & 2 & 0 & \frac{33}{5} & \frac{9}{2} & 2 & 0 & \frac{38}{5} & 5 & 2\,0 & 6 & 3\,0 & 0 \\
\frac{21}{5} & \frac{52}{15} & 2 & 0 & \frac{9}{2} & \frac{11}{3} & 2 & 0 & 5 & 4 & 2\,0 & 3 & 2\,0 & 0 \\
2 & 2 & 2 & 0 & 2 & 2 & 2 & 0 & 2 & 2 & 2\,0 & 0 & 0\,0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\,0 & 0 & 0\,0 & 0 \\
3 & \frac{3}{2} & 0 & 0 & 4 & 2 & 0 & 0 & 6 & 3 & 0\,0 & 12 & 6\,0 & 0 \\
\frac{3}{2} & 1 & 0 & 0 & 2 & \frac{4}{3} & 0 & 0 & 3 & 2 & 0\,0 & 6 & 4\,0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\,0 & 0 & 0\,0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\,0 & 0 & 0\,0 & 0
\end{bmatrix} , \tag{19}$$

where $\mathbf{B}$ is a symmetric semi-positive definite matrix. Using equation (15), this energy can be written as a function of the neighbouring nodes $\mathbf{z}$ as follows,

$$E = \mathbf{z}^\top (\mathbf{A}^{-1})^\top \mathbf{B} \mathbf{A}^{-1} \mathbf{z} \ . \tag{20}$$

Finally, based on this expression, we define the residuals for a patch as,

$$\mathbf{r} = \sqrt{\mathbf{S}} \mathbf{U}^\top \mathbf{z} \quad \text{with} \quad \mathbf{U}\mathbf{S}\mathbf{U}^\top = (\mathbf{A}^{-1})^\top \mathbf{B} \mathbf{A}^{-1} \ , \tag{21}$$

where $\mathbf{U}\mathbf{S}\mathbf{U}^\top$ is the singular value decomposition of the symmetric matrix $(\mathbf{A}^{-1})^\top \mathbf{B} \mathbf{A}^{-1}$, where the left and right singular vectors are given by $\mathbf{U}$ and the singular values are given by the diagonal matrix $\mathbf{S}$. These residuals corresponding to the thin-plate energies of the different patches can then used in the non-linear least squares problem.

3 Additional Results

The cameras A and A' from the paper (*i.e.*, CamA and CamA') are both Point-Grey greyscale cameras with a resolution of 1920×1440 pixels, and have ground-truth focal lengths $f_A = 1841.2$, and $f_{A'} = 2173.8$, respectively. For both cameras, the color-coded reprojection error results for different models are shown in Figures 3 (repeated from paper) and 5 (note that in these experiments, for cameras A and A', the laser only scans the scene up to x-coordinates of around ~ 1700). The automotive camera from the paper (*i.e.*, CamB) is a Bayer camera with a resolution of 1280×800 pixels, and a focal length $f_B = 1449.2$ (as determined using the spherical refraction model), the color-coded reprojection error results are shown in Figure 7 (repeated from paper).

All spline refraction related results in the paper use 4×4 number of patches. For this reason, we include an evaluation of the effect of the number of patches,

the numerical results are shown in Table 1. The corresponding color-coded re-projection error results are shown in Figures 4, 6, and 8. Clearly, for these examples, nothing much is gained by going beyond 4×4 number of patches, and it is also faster to optimize.

For the sake of completeness, we also provide some insight in the geometry parameters that are estimated when using the spherical refraction model, these are shown in Table 2. Despite the fact that the geometry of the windshields used with the different cameras is expected to be more or less the same, there is a lot of variation between the estimated radiuses for the different cameras. First of all there is an interchange between the radius of the sphere and its position with respect to the camera. However, ground-truth experiments in the paper show that this does not prevent the camera behind the windshield from being estimated accurately. Furthermore, the radius is estimated only for a rather small patch of the whole "sphere" as observed by the camera. As long as one does not extrapolate beyond that view, the model holds very well.

#patches	1×1	2×2	4×4	8×8	16×16	32×32
σ_{MAD} (CamA)	0.145	0.111	0.0950	0.0856	0.0861	0.0845
σ_{MAD} (CamA')	0.153	0.127	0.110	0.0903	0.0931	0.0911
σ_{MAD} (CamB)	0.278	0.264	0.235	0.238	0.224	0.226

Table 1. For the cameras discussed in the paper, the estimated accuracy of the spline refraction model versus the choice of the number of patches.

	R (in meters)	$\mathbf{c}$ (in meters)
CamA	3.28	$\begin{bmatrix} 0.0549 \\ 2.87 \\ -1.51 \end{bmatrix}$
CamA'	2.11	$\begin{bmatrix} -0.0857 \\ 1.91 \\ -0.764 \end{bmatrix}$
CamB	1.27	$\begin{bmatrix} 0.00267 \\ 1.18 \\ -0.298 \end{bmatrix}$

Table 2. Estimated results of the spherical geometry for the cameras discussed in the paper.

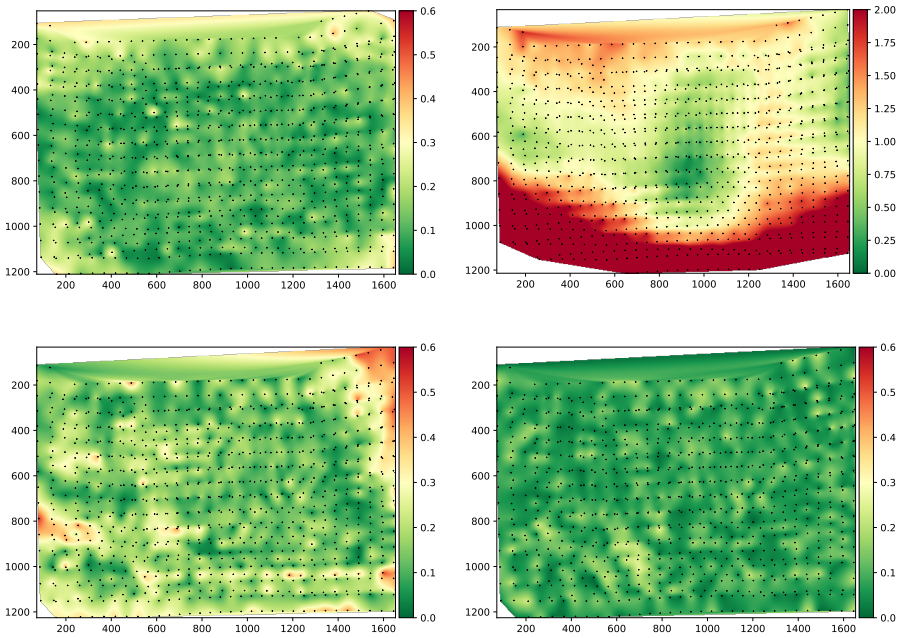

Fig. 3. Color coded reprojection results for CamA (repeated from paper): Top left: Fit of a pinhole model (without windshield), Top right: Fit of a pinhole model (with windshield), Bottom left: Fit of spherical refraction model (with windshield), and Bottom right: Fit of a 4×4 spline refraction model (with windshield).

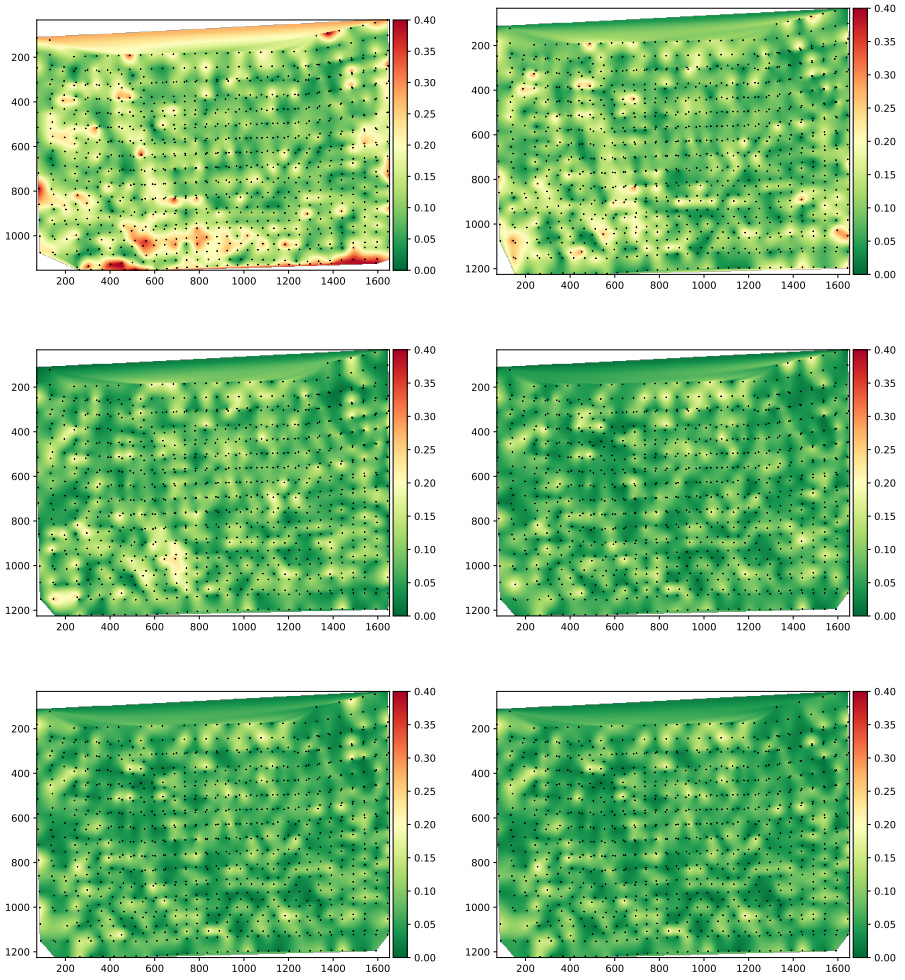

Fig. 4. Color coded reprojection results for CamA for different choices of the number bi-cubic spline patches: From top left to bottom right, 1×1, 2×2, 4×4, 8×8, 16×16, and 32×32 numbers of patches, respectively.

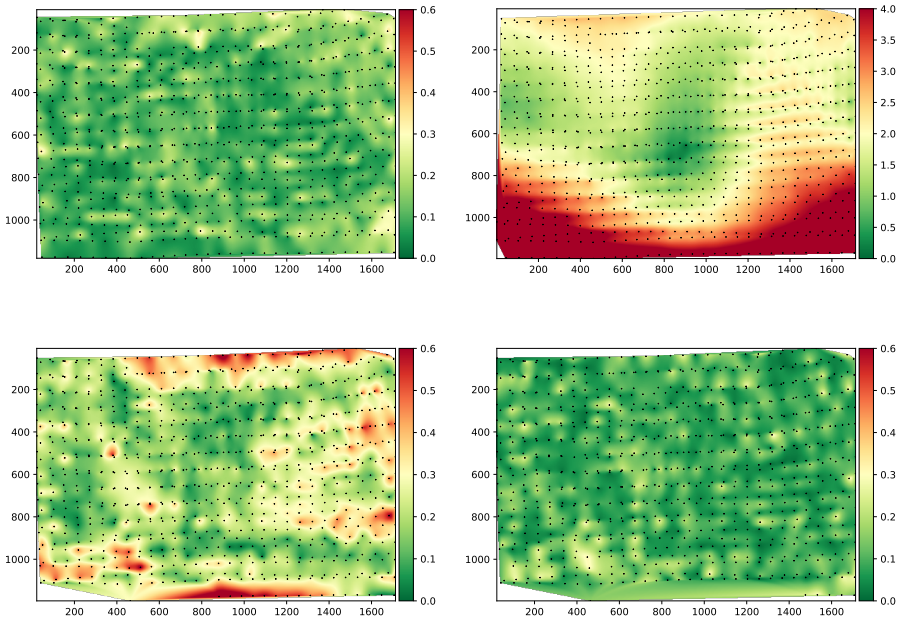

Fig. 5. Color coded reprojection results for CamA': Top left: Fit of a pinhole model (without windshield), Top right: Fit of a pinhole model (with windshield), Bottom left: Fit of spherical refraction model (with windshield), and Bottom right: Fit of a 4×4 spline refraction model (with windshield).

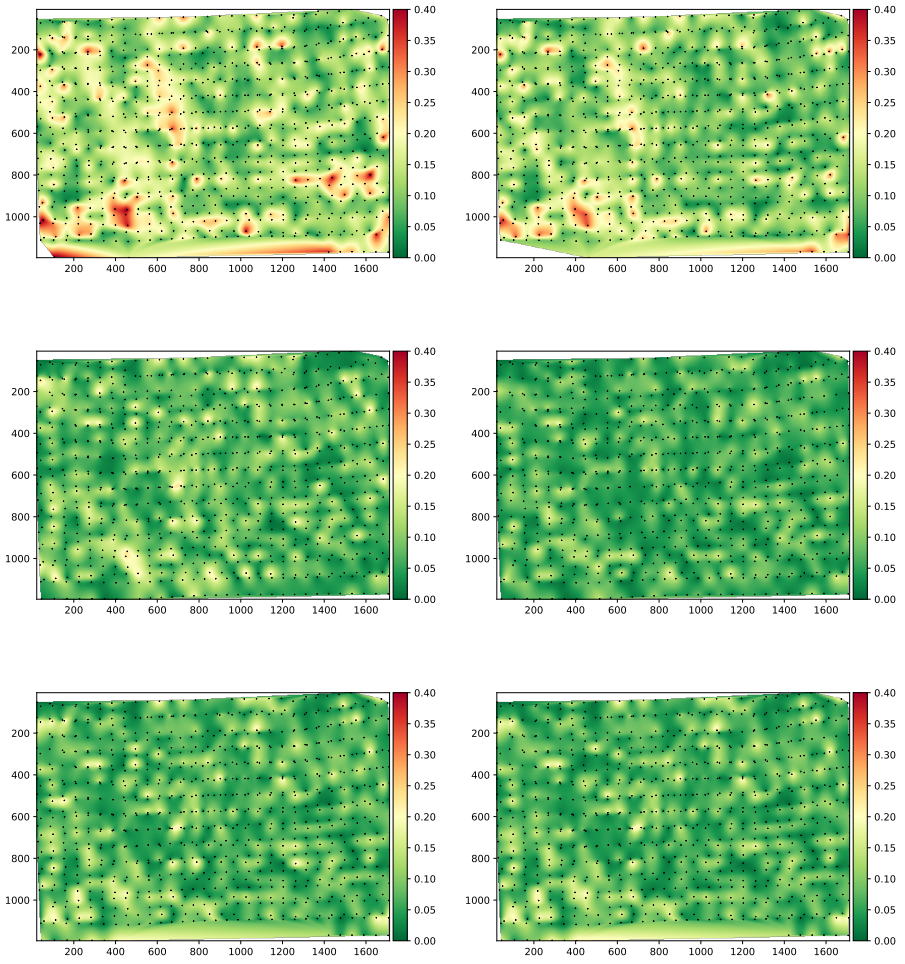

Fig. 6. Color coded reprojection results for CamA' for different choices of the number bi-cubic spline patches: From top left to bottom right, 1×1, 2×2, 4×4, 8×8, 16×16, and 32×32 numbers of patches, respectively.

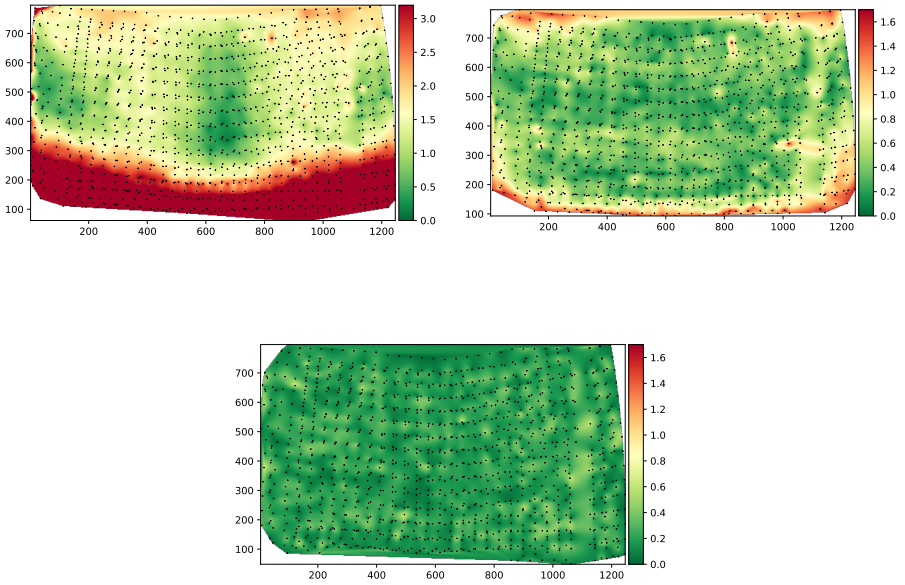

Fig. 7. Color coded reprojection results for the automotive camera (*i.e.*, CamB) (repeated from paper): Top left: Fit of a pinhole model (with windshield), Top right: Fit of spherical refraction model (with windshield), and Bottom: Fit of a 4×4 spline refraction model (with windshield).

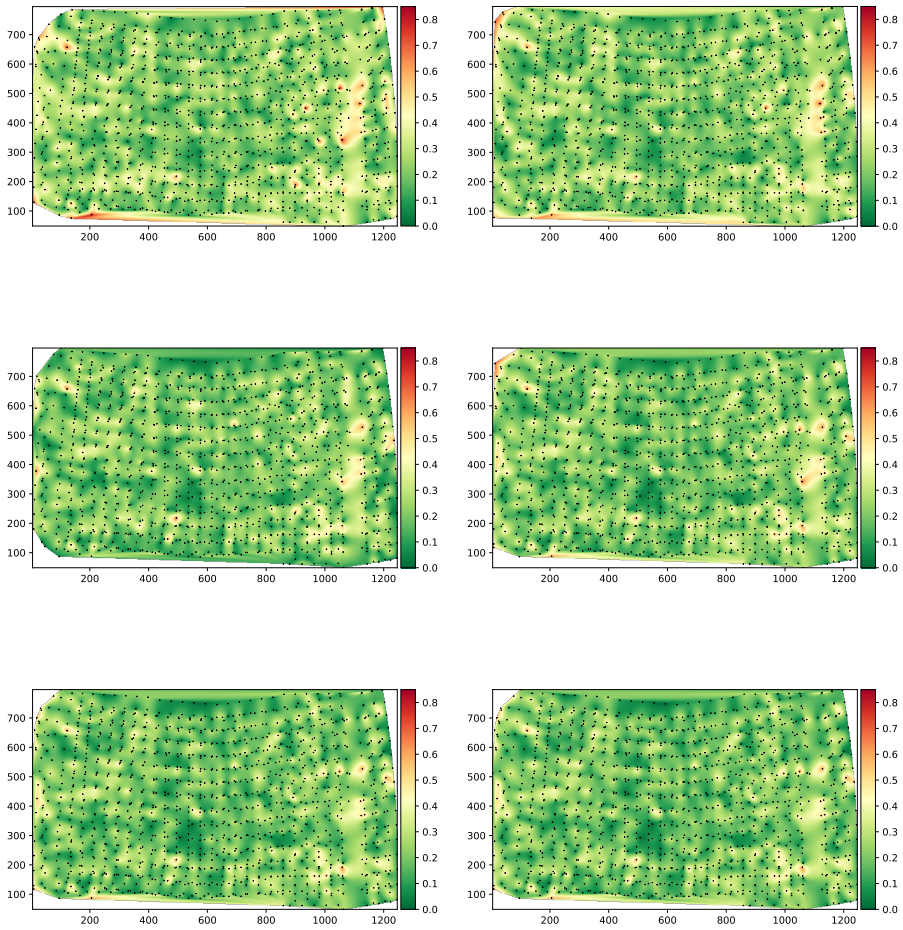

Fig. 8. Color coded reprojection results for the automotive camera (*i.e.*, CamB) for different choices of the number bi-cubic spline patches: From top left to bottom right, 1×1, 2×2, 4×4, 8×8, 16×16, and 32×32 numbers of patches, respectively.

References

1. Agarwal, S., Mierle, K., Others: Ceres solver. http://ceres-solver.org
2. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (Jun 1981)
3. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, second edn. (2004)
4. Huber, P.J.: Robust Statistics, pp. 1248–1251. Springer Berlin Heidelberg, editors Lovric, Miodrag, Berlin, Heidelberg (2011)
5. Kukelova, Z., Bujnak, M., Pajdla, T.: Real-time solution to the absolute pose problem with unknown radial distortion and focal length. In: Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV'13). pp. 2816–2823. Washington, DC, USA (2013)
6. Rousseeuw, P.J., Croux, C.: Alternatives to the median absolute deviation. Journal of the American Statistical Association 88 **88**(424), 1273–1283 (1993)