Camera pose and intrinsic parameters estimation from n 2D-to-3D point correspondences is a known problem in computer vision and photogrammetry. Depending on the set of unknown parameters, the problem is called Perspective-n-Point (PnP) when only absolute camera pose is unknown or PnPF when focal length is unknown as well. Projection error functions are highly non-convex in focal length, so before methods for PnPf were published, the only choice was to do exhaustive search not suitable for real-time applications. The EPnP method was extended to PnPf problem in [4], we refer to this method as UPnPf. RPnP inspired the authors of [5] to propose a method GnPnPf+GN for PnPf problem. They use angle constraints to build specific polygonal system and solve it, then they use nonlinear refinement with Gauss-Newton algorithm. It gave superior results to [4] both in speed and accuracy in general case, and was more accurate in planar case, although UPnPf [4] was faster.

This paper is devoted to a method for PnPf problem for arbitrary amount of points, more or equal to 6. We consider both planar and non-planar cases. We fix the space of the search as a linear combination of several right singular vectors of the least squares system matrix. We use linear programming techniques to find feasible solutions faster. Then we do nonlinear refinement with Levenberg-Marquardt.

The barycentric representation of 3D points allows to express n 3D points \( \mathbf{p}_i \) as a frame-independent linear combination of 4 basis points \( c_j \):

\[
\mathbf{p}_i = \sum_{j=1}^{4} \alpha_j^{(i)} \mathbf{c}_j, \quad i = 1 \ldots n, \quad \sum_{j=1}^{4} \alpha_j^{(i)} = 1.
\]

Each point projection as described in [2, 4] leads to two independent linear w.r.t. basis points’ coordinates equations. The equations form a system with matrix \( M \). The solution lies in a null-space (kernel) of \( M \) and can be expressed as a linear combination of kernel basis with coefficients \( \beta_i \)

\[
\mathbf{x} = \sum_{i=1}^{N} \beta_i \mathbf{q}_i,
\]

where \( \mathbf{q}_i \) are the right-singular vectors of \( M \) corresponding to the \( N \) null singular values of \( M \). There are distance constraints which need to be satisfied:

\[
\| \mathbf{c}_i - \mathbf{c}_j \|^2 = \| \mathbf{c}_m^i - \mathbf{c}_m^j \|^2 = r_{ij}^2,
\]

where \( r_{ij} \) is a known distance between \( i \)-th and \( j \)-th basis points. Distance constraints are quadratic w.r.t. \( \beta_i \) and in the same time are linear w.r.t. \( \mathbf{b} \):

\[
\mathbf{b} = ( \beta_1^2 \ldots \beta_n^2 \beta_1 \beta_2 \ldots \beta_{N-1} \beta_N )^T.
\]

While the EPnP method tries to solve the constraints system (2), we solve (in a set defined by (1)) a least squares problem

\[
F_R(\mathbf{x}) = \| M \mathbf{x} \|^2 + \gamma \| \mathbf{Lb} - \mathbf{r} \|^2 \rightarrow \min,
\]

where \( \gamma \) is some coefficient. Distance constraints for the PnPf problem have the form:

\[
r_{ij}^2 = \sum_{k=1}^{3} \| \mathbf{c}_i^k - \mathbf{c}_j^k \|^2 = \frac{1}{T} ((\mathbf{d}_i^{(1)} - \mathbf{d}_j^{(1)})^2 + (\mathbf{d}_i^{(2)} - \mathbf{d}_j^{(2)})^2 + (\mathbf{g}_i^{(3)} - \mathbf{g}_j^{(3)})^2),
\]

for definition of \( \mathbf{d}_i, \mathbf{g}_i \) see paper. Constraint is quadratic in \( \mathbf{d}_i, \mathbf{g}_i \), but if we choose the unknowns vector \( \mathbf{b} \) analogously to (3), it becomes linear with \( \mathbf{b} \) equal to \( \mathbf{b} \) given in (3) and:

\[
\mathbf{b}^2 = \mathbf{f}^2 \mathbf{b}_1^1, \quad \mathbf{b}^T = (\mathbf{b}_1^T, \mathbf{b}_2^T)^T.
\]

So, for the PnPf problem we get the analogous function as (4) for PnP problem as described in the paper. In the Algorithm described in the paper, for each \( N = 1, 2, 3 \) we find candidate solutions (1), for which we formulate additional linear constraints and solve using MATLAB’s \texttt{linprog}.

After a loop over \( N \), we choose the solutions which have the least amount of points lying behind the camera, and among these solutions choose the by comparing the value of \( F_R(\mathbf{x}) \) (4). This chosen solution is subsequently refined by Levenberg-Marquardt procedure [3].

Algorithm implementation is available at \texttt{http://sites.google.com/site/alexandervakhitov/projects/epnpfr}.

We made two sets of experiments: comparative test and a test using real images. Synthetic experiments test performance of the method w.r.t. varying noise level and point number. Here we show at the figure the results for some of measured parameters for general point configuration.

The aim is to demonstrate applicability of the algorithm in a real setting. We made three shots of a non-moving scene with Nikon D3100 camera with several focal length settings. We performed standard structure-from-motion reconstruction using one initial frame pair.

We matched the SIFT points from one of the frames in the initial pair and every other frame. We ran our algorithm and the best state-of-the-art GPnPf+GN algorithm with the RANSAC loop, choosing 6 points [1]. When both methods returned results, they were different less than for 1% in focal length and reprojection error, except 105 mm focal length (average error GPnPf+GN 2.53 pix, EPnPfR 0.77 pix).

Proposed algorithm is as accurate and stable as the state-of-the-art methods, and more than 2 times faster.


