A branch-and-bound algorithm for globally optimal camera pose and focal length

Kyuhyoung Choi, Subin Lee, Yongduek Seo ∗
Department of Media Technology, Sogang University, Seoul, Republic of Korea

A R T I C L E   I N F O
Article history:
Received 18 August 2009
Received in revised form 17 December 2009
Accepted 3 February 2010

Keywords:
Global optimization
Branch-and-bound algorithm
Camera pose and focal length

A B S T R A C T
This paper considers the problem of finding the global optimum of the camera rotation, translation and
the focal length given a set of 2D–3D point pairs. The global solution is obtained under the L∞-norm optimality
by a branch-and-bound algorithm. To obtain the goal, we firstly extend the previous branch-and-bound
formulation and show that the image space error (pixel distance) may be used instead of the angular
error. Then, we present that the problem of camera pose plus focal length given the rotation is a quasi-
convex problem. This provides a derivation of a novel inequality for the branch-and-bound algorithm for
our problem. Finally, experimental results with synthetic and real data are provided.

1. Introduction
Recent development in the L∞-norm minimization method
shows that globally optimal solutions of some geometric vision
problems can be computed via the bisection algorithm based on
the techniques of linear programming (LP) or second-order cone
programming (SOCP) [1–3]. The L∞ method provides a novel way
of obtaining the global optima of diverse problems, but its applica-
tional area has been somewhat limited to the cases, where the residual
function can be expressed in the form of a quasi-convex function, for
which the camera rotation is usually assumed to be known a priori.

Recently, Hartley and Kahl [4] showed that the assumption of
known rotation may be overcome through a branch-and-bound
algorithm, making it possible to find the rotation through an effi-
cient search in the rotation space. The branch-and-bound (BnB)
algorithm tests every cubic sub-domain of the rotation space for
the possibility that it may provide a better (smaller residual) solu-
tion by solving a feasibility problem. The size of the sub-domain is
then reduced repeatedly through a top-down approach. It is shown
that the method can find the optimal solutions to the problems of
camera pose and two-view relative motion under the L∞ sense. The
error metric adopted for the BnB search is the angular error metric
similar to some of the previous works such as [1,5]. That is, the
residual is defined as the angle between the ray of the image mea-
surement and the ray of its parametric projection.

This paper presents that the image space error metric (pixel dis-
tance) can be used in the branch-and-bound algorithm instead of
the angular error metric used in Hartley and Kahl [4]. By far the
most important benefit of this extension is that we can directly
deal with the problem in the unit of pixel distance which is indeed
more intuitive than the angular distance. Based on the pixel dis-
tance formulation, we derive a new bound for the inequality of
the feasibility test, which becomes the key engine of the BnB
search algorithm. The new bound is shown to be a function of
the focal length as well as the size of the cubic sub-domain. But be-
cause the focal length is known a priori for the case of a calibrated
camera, the BnB search is shown to be still possible. This is our first
contribution.

Secondly, we extend the BnB algorithm one step further in such
a way that the focal length is included as an unknown to be com-
tuted together with the pose parameters. For this, we show that
the problem of estimating the focal length and the translation gi-
gen the rotation is a quasi-convex problem. That is, given the rota-
tion, we can obtain a globally optimal L∞ solution of the focal
length and the translation. This result provides another quasi-con-
 vex function which did not appear in the literature yet. The chal-
lenge in the case of unknown focal length lies in the fact that the
focal length term appears in the upper bound of the feasibility
constraints, because it brings about non-convexity. We solve this
problem by deriving a new bound based on a priori information
on the possible range of the focal length. That is, given a range of
the focal length, another new bound is computed using the maxi-
mum and minimum values of it. This makes it possible for the BnB
to do global search over the new parameter space. The assumption
of known range of the focal length is not impractical because, for
example, we know that it usually lies in an interval like [500,2000] for
general web-cams. One strategy is to compute a
new bound based on the entire range of the focal length and use it for the BnB search over the rotation space. The other is to divide the range into several sub-intervals and run the BnB search for each of them. Because each BnB has an additional linear constraint confining the range of the focal length, only the interval having the optimal solution remains through the BnB search and the other sub-intervals will be found to be infeasible.


The problem of estimating the camera pose and focal length is studied for a practical application of augmented reality in Jain and Neumann [15], Park et al. [16] and Matsunaga and Kanatani [17].

Compared to these works, our approach minimizes the $L_\infty$-norm error using the branch-and-bound search technique of Hartley and Kahl [4]. The contribution is that (1) it gives an explicit formulation of a pixel distance instead of the angular distance, (2) the focal length is included as a variable in addition to the pose, (3) the problem of focal length and pose with known rotation is shown to be a quasi-convex problem, and (4) a fast feasibility test algorithm appropriate for the branch-and-bound is provided.

This paper consists of several sections. Section 2 revises the branch-and-bound algorithm of Hartley and Kahl [4] for the camera pose based on the angular error metric. Then, Section 3 shows how the image space error can be used instead of the angular error. In Section 4, we show that the problem of camera pose plus focal length with known rotation is a quasi-convex problem. A new bound is derived and the global solution is computed by the BnB algorithm in Section 5. The computation speed is improved through an efficient feasibility test algorithm presented in Section 6. Experimental results are given in Section 7, and concluding remarks and discussions in Section 8.

2. Branch-and-bound for camera pose

This section introduces the branch-and-bound algorithm of Hartley and Kahl [4] for the camera pose in order for us to make it easy to provide our work. The definition of the pose problem is: Given a set $\{X_i, v_i\}$ of 3D points with known position and corresponding 2D image points, determine the translation $t$ and rotation $R$ of the camera matrix $P = [R \mid t]$. When the camera is calibrated, the image space may be assumed to be the sphere of unit radius. The image of $X_i$ on to the spherical surface is then given by the re-projection function:

$$v_i = \frac{R X_i + t}{\|RX_i + t\|}$$

The $L_\infty$ solution for $R$ and $t$ is the solution of the following minimization:

$$\min_{R,t} \max_i \angle(v_i, RX_i + t)$$

where $\angle(\cdot, \cdot)$ represents the angle between two vectors, and $v_i \in \mathbb{R}^3$ is the measurement vector representing the direction of its image point corresponding to $X_i$. Unfortunately, this problem is not a convex problem due to the non-linearity of $R$ – it is not easy to obtain the global optimum.

In the method of Hartley and Kahl [4], the domain of rotation space, that is, the 3D sphere of radius $\pi$, is divided into small sub-sets $D_i \cup 0 = \text{domain}(R)$, where each $D_i$ is a cube of half-side length $\sigma$. Given an estimation $(\hat{R}, \hat{t})$ with the $L_\infty$ angular error $e$, the method tests whether a rotation sub-domain $D_i$ may have a rotation that yields a smaller residual than $e$. This test problem, called the feasibility problem, is given as:

$$\text{find } t \quad \text{s.t. } \angle(v_i, \hat{R}X_i + t) < e + \sqrt{3}\sigma, \quad \forall i$$

where $\hat{R}$ is the rotation matrix corresponding to the center of $D_i$. Because this problem is a convex problem, it can be easily solved via any LP or SOCP solver such as GLPK or SeDuMi [18,19]. The value $\sqrt{3}\sigma$ means the maximum possible variation of the maximum residual (right-hand side of the inequality) caused by the variation of the rotation $R$ in $D_i$ whose half-side length is $\sigma$.

Algorithm 1. Branch-and-bound for camera pose

\begin{algorithm}
\begin{algorithmic}
\STATE \textbf{Input:} Find an estimate $(\hat{R}(0), t(0), e(0)); \sigma = \sigma(0)$.
\STATE \textbf{1:} repeat
\STATE \textbf{2:} $\sigma = \sigma/2$.
\STATE \textbf{3:} Divide the domain into cubes $(D_i(\sigma))$.
\STATE \textbf{4:} for each $(D_i)$ do
\STATE \textbf{5:} Solve the feasibility problem $(F)$.
\STATE \textbf{6:} if infeasible then
\STATE \textbf{7:} discard the domain $D_i$.
\STATE \textbf{8:} else
\STATE \textbf{9:} find a new estimate $(\hat{R}(k), t(k), e(k))$.
\STATE \textbf{10:} if $e(k)$ is smaller then update the estimate.
\STATE \textbf{11:} end if
\STATE \textbf{12:} end for
\STATE \textbf{13:} until $\sigma < \sigma_{\text{min}}$
\end{algorithmic}
\end{algorithm}

Now the BnB algorithm is shown in Algorithm 1. After dividing the domain of the rotation into small sub-domains (line 2), this algorithm repeatedly checks whether the domain $D_i$ is a feasible candidate or not (line 5). If $D_i$ is found to be feasible, there is a possibility of the existence of a better solution inside the cube $D_i$. So a new estimate is computed via the bisection algorithm (line 9). We give an explanation of the bisection algorithm later. The resolution of the domain is then refined by dividing again the feasible $D_i$ into eight cubes (lines 2 and 3). The stopping condition in line 1 means that the error of the best solution is in the interval: $e' \leq e_{\text{min}} \leq e + \sqrt{3}\sigma$, where $e'$ is the maximum residual obtained in line 9 during the loop. Specifically, the line 9 solves the following quasi-convex problem with the bisection algorithm:

$$\min_{R,t} e \quad \text{s.t. } \angle(v_i, \hat{R}X_i + t) \leq e', \quad \forall i$$

Any algorithm is applicable in finding the initial input estimate because it will eventually be upgraded during the branch-and-bound iteration. Note that the BnB algorithm is not dependent on the initial solution. Indeed, not all the parameters are required as input but only a feasible upper bound for $e(0)$ is sufficient to start the algorithm. So, the simplest way is to choose an $e(0)$ adequately large to allow for the constraints to yield a non-empty solution space.

The angular error in the constraints is replaced by the tangent of it in solving the problems of (3) and (4):

$$\tan(v_i, \hat{R}X_i + t) = \frac{\|v_i \times (\hat{R}X_i + t)\|}{v_i^T (\hat{R}X_i + t)}$$

The problem definition of (3) is then re-written as follows:
find $t$
\[
\begin{align*}
\text{s.t.} \quad & \frac{|v_i \times (\hat{r}X_i + t)|}{|v_i| (\hat{r}X_i + t)} < e^{\epsilon_{\tan}^{\text{old}}} + \tan \sqrt{3}\sigma, \quad \forall i \\
\end{align*}
\]  
where $e^{\epsilon_{\tan}^{\text{old}}}$ corresponds to the best re-projection error obtained during the branch-and-bound iteration by solving the quasi-convex problem
\[
\min e^{\epsilon_{\tan}}
\begin{align*}
\text{s.t.} \quad & \frac{|v_i \times (\hat{r}X_i + t)|}{|v_i| (\hat{r}X_i + t)} \leq e^{\epsilon_{\tan}}, \quad \forall i
\end{align*}
\]
Note that the problems (6) and (7) result in SOCPs if the $L_2$-norm is adopted. They become LPs if the $L_\infty$-norm is used.

3. Branch-and-bound with pixel distance

Since the camera is assumed to be calibrated, we know the principal point which is the intersection of the viewing direction of the camera and the image plane. Fig. 1 shows the geometric relationship of the spherical retina and the planar retina for the image measurement. In the figure, we use another notation $\delta = \sqrt{3}\sigma$ to denote the allowable angle variation in $D$. Let $v_i = [v_{i1}, v_{i2}, v_{i3}]^T$ be a measurement vector ($|v_i| = 1$) in the spherical retina and $\hat{r}i = \hat{r}X_i + t$ be the re-projection ray of $X_i$ corresponding to the measurement $v_i$, where $\hat{r}$ is the central rotation of $D$ and $t$ the translation of the camera.

We are now going to derive a similar but novel inequality corresponding to the one in (3) but defined in the planar image space. Let us look at the inequality in (3) defined with respect to the angular (tangent) error metric:
\[
\angle(v_i, \hat{r}X_i + t) < \epsilon + \sqrt{3}\sigma, \quad \forall i
\]  
If there is not any variation in $\hat{r}_i$, we do have $\delta = 0$ and the image location of $X_i$ is at $\beta_3 = \tan \theta_i$ (Fig. 1), where $\theta_i$ is the incidence angle of $X_i$ with respect to the viewing direction $\mathbf{z} = [0, 0, 1]^T$. If we have $\delta$ due to the volume of $D(\sigma)$, the image intersection of the ray $X_i$ induces a region in the planar image space. The interval $[\beta_1, \beta_3]$ in Fig. 1 corresponds to this. From this observation, the maximum variation at the image pixel location $\beta_2 = \tan \theta_i$ due to $D(\sigma)$ is given by:
\[
\begin{align*}
\tau_i = \tan(\theta_i + \delta) - \tan \theta_i \\
= \tan \theta_i + \tan \delta - \tan \theta_i \\
= \tan \theta_i(1 + \tan \delta) - \tan \theta_i
\end{align*}
\]  
Note that the variation now depends on the location of the measurement as indicated by the index $i$.

![Fig. 1. Illustration of two retinal spaces: spherical and planar. The angle between the z-axis and the measurement (or re-projection) vector is represented by $\theta$ and the angle bound due to the rotation cube $D$ of half-side length $\sigma$ is represented by $\delta = \sqrt{3}\sigma$.](image)

Let $u_i = [u_{i1}, u_{i2}] = [v_{i1}/v_{i3}, v_{i2}/v_{i3}]$ be the image pixel coordinates of $v_i$ and $\hat{u}_i = [u_{i1}/u_{i3}, u_{i2}/u_{i3}]$ be the re-projection of $U_i$ into this calibrated image plane:
\[
\hat{u}_i = \left[ \begin{array}{c} r_{i1}X_i + t_1 \\
 r_{i2}X_i + t_2 \\
 r_{i3}X_i + t_3 
\end{array} \right] 
\]  
where $r_i$ and $t_i$ are, respectively, the $i$th row vector of $\hat{r}$ and the $i$th element of the vector $t$. Then, the pixel distance between $u_i$ and $\hat{u}_i$ is defined to be
\[
d(u_i, \hat{u}_i) = \|u_i - \hat{u}_i\| < \epsilon' + \tau_i, \quad \forall i
\]  
and the inequality for this distance corresponding to (8) is given by
\[
d(u_i, \hat{u}_i) < \epsilon' + \tau_i, \quad \forall i
\]  
where the error $\epsilon'$ denotes a residual in the planar retinal space. Note that multiplying it with the focal length gives the residual in the unit of the pixel distance. Let us do this using the known camera calibration parameters. First we multiply the known focal length $f$ to the re-projection equation in (11)
\[
\hat{u}_i = f [r_{i1}X_i + t_1, r_{i2}X_i + t_2] 
\]  
and, then, employing a new notion $w_i$ to represent the pixel unit image measurement, we have the final inequality:
\[
d(w_i, \hat{u}_i) < \epsilon_{\text{pixel}} + f\tau_i, \quad \forall i
\]  
where $\epsilon_{\text{pixel}}$ is the $L_\infty$ re-projection error in the unit of pixels, different from the ones in (13) and (8). Therefore, in the branch-and-bound algorithm, we now deal with the following feasibility problem whose error metric is defined in terms of the pixel unit:

\[
\text{find } t
\begin{align*}
\text{s.t.} \quad & d(w_i, \hat{u}_i) < \epsilon_{\text{pixel}} + f\tau_i, \quad \forall i
\end{align*}
\]

In conclusion, it is now possible to minimize the pixel distance to obtain the global solution through the branch-and-bound algorithm.

4. Pose and focal length problem is quasi-convex

Now let us examine the distance function $d$ in pixel unit:
\[
d(w_i, \hat{u}_i) = \left\| \begin{array}{c} w_i - f^T \hat{u}_i \\
 w_2 - f^T \hat{u}_i 
\end{array} \right\| 
\]  
\[
= \left\| \begin{array}{c} -r_iX_i + ft_1 + w_1f_3 + w_2f_3X_i \\
 -r_iX_i + ft_2 + w_1f_3 + w_2f_3X_i 
\end{array} \right\| 
\]  
\[
= \frac{r_iX_i + t_3}{fX_i + t_3}
\]

By defining new variables $t'_1 = ft_1$ and $t'_2 = ft_2$, we can write down the distance function as follows
\[
d(w_i, \hat{u}_i) = \frac{\|ax + b\|}{cx + d}
\]  
where
\[
A = \left[ \begin{array}{ccc} -r_i & 1 & 0 \\
 -r_i & 0 & -1 
\end{array} \right]
\]  
\[
x = [f, t'_1, t'_2, t_3]^T
\]  
\[
b = [w_1, w_2f_3X_i]^T
\]  
\[
c = [0, 0, 0, 1]
\]  
\[
d = r_iX_i
\]

Note that Eq. (19) of the distance function in the pixel space is in the form of convex-over-concave. Therefore, it is a quasi-convex function [20]. A global solution to this pose and focal length prob-
lem (known rotation) can be computed via the bisection algorithm shown in Algorithm 2, where the feasibility problem given a constant bound \( z \) is as follows:

\[
\begin{align*}
\text{find} & \quad [t_1, t_2, t_3, f] \\
\text{s.t.} & \quad d(w_i, w_j) < z, \quad \forall i \\
& \quad f > 0
\end{align*}
\] (25)

Among a few ways to solve the feasibility problem we adopt the method of minimizing the maximum infeasibility \( s \):

\[
\begin{align*}
\min & \quad s \\
\text{s.t.} & \quad \| A_i x + b_i \| < \alpha (c_i x + d_i) + s, \quad \forall i \\
& \quad f > 0
\end{align*}
\] (26)

According to the selection of the norm function, this problem may become an SOCP or an LP. In this paper, we choose the latter by adopting the \( L_\infty \)-norm for the constraints. This approach is similar to the one presented previously in [21,4]; using the \( L_1 \)-norm results in an LP as in [3].

Algorithm 2. Bisection algorithm for pose and focal length

\begin{enumerate}
\item \textbf{repeat}
\item \( z = (U + L)/2. \)
\item Solve the feasibility problem (25):
\item \textbf{if} feasible \textbf{then} \( L = z \) \textbf{else} \( U = z. \)
\item \textbf{until} \( U - L \) is small enough
\end{enumerate}

5. Branch-and-bound with unknown focal length

As we presented, the problem of camera pose and the focal length is a quasi-convex optimization problem and we can obtain the global solution if we know the rotation matrix. This section shows that a global search for the rotation can be performed through a branch-and-bound method.

Our new feasibility problem is as follows from (16) and (26), which replaces (3) used in Algorithm 1:

\[
\begin{align*}
\min & \quad s \\
\text{s.t.} & \quad \| A_i x + b_i \| < (c_{\text{pixel}} + f \tau_i (c_i x + d_i) + s, \quad \forall i \\
& \quad 0 < f
\end{align*}
\] (27)

Since the focal length \( f \) is a variable to be estimated in this formulation, it must not appear as a multiplication term with other variables in the right-hand side. Therefore, this is not a convex problem as it is. Furthermore, the upper bound \( \tau_i \) is indeed not a constant but a function of the focal length \( \tau_i = \tau_i(f) \).

Our strategy is to find a new upper bound using the interval information of the focal length. Let us suppose that the focal length \( f \) be delimited by upper and lower bounds \( f_u \) and \( f_l \), then, replacing \( f \) in (27) with \( f_l \) provides a way to obtain an upper bound. So, it remains to find a bound for \( \tau_i(f) \). Since the term \( \tau_i \) in (10) can be written as a function of the focal length

\[
\tan \tau_i = \| w_i \| / f
\] (28)

an explicit form of \( \tau_i \) is provided as follows:

\[
\tau_i(f) = \frac{f \tan \delta + \| w_i \|}{f - \| w_i \| \tan \delta} \quad \text{for} \quad \delta = \sqrt{3}\sigma,
\] (29)

where \( \delta = \sqrt{3}\sigma \), the maximum distance in \( D(\sigma) \) from its central location. We observe that this function is decreasing when

\[
\| w_i \| \tan \sqrt{3}\sigma < f
\] (30)

Therefore, we end up with an inequality

\[
\tau_i(f) \leq \tau_i(f_l)
\] (31)

for a sufficiently small \( \sigma \) that provides the condition (30). Finally, given the interval \([f_l, f_u]\), the feasibility problem (27) becomes a convex problem as follows with additional linear constraints:

\[
\begin{align*}
\min & \quad s \\
\text{s.t.} & \quad \| A_i x + b_i \| < \alpha (c_i x + d_i) + s, \quad \forall i \\
& \quad \max \| w_i \| \tan \sqrt{3}\sigma < f \\
& \quad f_l \leq f \leq f_u
\end{align*}
\] (32)

where \( \alpha_i \) is a constant given as

\[
\alpha_i = \epsilon_{\text{pixel}} + f_u \tau_i(f_l)
\] (33)

This formulation assumes that a feasible range of the focal length is known. We believe that practically this is not a severe restriction because a rough range of the focal length of an ordinary camera is available and it is sufficient to run the BnB algorithm. As for the constraint (30), note that, in practice, a usual value of the focal length is much larger than the lower bound \( \max \| w_i \| \tan \sqrt{3}\sigma \) since \( \sigma \) becomes very small during the BnB iteration.

Note that the constant \( \alpha_i \) in (32) may become a large value given an interval \([f_l, f_u]\). A concern is that a large value of \( \alpha_i \) may produce a great number of feasible cubes (sub-domains) of the rotation space. Regarding this, our approach is to divide the interval into several disjoint sub-intervals \( I_k = [f_l, f_u] \):

\[
\{ I_k | I_k = [f_l, f_u], I_k \cap I_j = \emptyset \}
\] (34)

Then, the solution of the original problem can be attained by solving several problems of the same form as (32) but with different intervals. Experiments in the next section use this approach.

Algorithm 3. Fast method for the feasibility problem

\begin{enumerate}
\item \textbf{Input:} of size \( N = |\mathcal{M}| \). Sample size \( N_{\mathcal{S}} = |\mathcal{S}| \).
\item \textbf{Choose a sample set} \( \mathcal{S} \).
\item \textbf{repeat}
\item Solve the feasibility problem (32) with \( \mathcal{S} \).
\item \textbf{if} infeasible \textbf{then} return infeasible
\item \textbf{else} if \( \max (s_j) \leq 0 \) then return feasible
\item \textbf{else}
\item \( f = \arg \max (s_j) \) for \( \mathcal{S} \).
\item \( f = \arg \min (s_1, \ldots, s_{N_{\mathcal{S}}}) \) for \( \mathcal{S} \).
\item Replace \( w_i \) with \( w_f \) in \( \mathcal{S} \).
\item \textbf{endif}
\item \textbf{until} \( \mathcal{S} \).
\end{enumerate}

6. A fast feasibility test

Given many 2D–3D input point pairs, we can solve the feasibility problem (32) by using a partial set of input data instead of the whole data set. The idea of our fast method is as follows. Note that if the feasibility problem is solved with a few \( N_{\mathcal{S}} \) (e.g. 10) measurements and is found to be infeasible (\( \mathcal{S} > 0 \)), then it means that the original problem is infeasible. Our observation is that a partial set \( \mathcal{S} \) of input data \( \mathcal{M} \) may provide the solution for the whole data set. On the other hand, if it is solved to be feasible (\( \mathcal{S} \leq 0 \)), we do not know the feasibility of the original problem as directly as the opposite case. However, because we have an estimate \( \hat{x} \), we may check whether or not the rest of the measurements (\( \mathcal{M} \setminus \mathcal{S} \)) results in feasible constraints. That is, the solution \( x \) of \( \mathcal{S} \) is a feasible solution of the whole set if we have

\[
\left[ A_i x + b_i \right] - \alpha_j (c_i x + d_i) \leq 0, \quad \forall i \in \mathcal{S}.
\] (35)

Finally, when we have neither of the two cases, we choose the measurement corresponding to the maximum of the gaps \( s_j \) and put it into the sample set \( \mathcal{S} \) while the measurement in \( \mathcal{S} \) with the smallest gap is picked out (never to be used again). This procedure is repeated until a solution is obtained.
Our fast feasibility test algorithm is provided in Algorithm 3. Note that our method is very efficient for the BnB algorithm since it solves the feasibility problem with only a few of the inputs. Experimental results show that the feasibility test is finished mostly within only a few iterations and most of the computation time is spent for the evaluation of \( \mathcal{S} \)'s. Therefore, the larger \( N \) is, the more computation time it requires. However, it is much faster than solving the feasibility problem with the whole data set.

We show an experimental result in this section. First, we generated 100 sets of synthetic data, each of which has \( N = 100 \) 2D–3D point pairs. The image coordinates are contaminated with a Gaussian noise of standard deviation being 0.5 pixel. Because we adopt the \( L_\infty \) norm for image error, the feasibility problem is a linear programming problem. We implemented our algorithm with GNU linear programming kit (GLPK) in C/C++ [19]. Fig. 2 shows the plot of average computation time against various number of sample elements in \( \mathcal{S} \). When we applied the original version of the bisection algorithm to those synthetic data sets, it took 0.061 s; when we used only two measurements \( |N_\mathcal{S}| = 2 \) for the fast algorithm, it took 0.0125 s, which showed a 4.8 times faster computation. As the number \( N_\mathcal{S} \) increases, the computation time grows linearly in the graph. We use \( N_\mathcal{S} = 2 \) for the experiments in next section.

7. Experiments on the branch-and-bound

7.1. Experiments with synthetic data

We first performed experiments with synthetically generated data sets. Each pixel coordinate of the synthetic measurements were contaminated by Gaussian random noise of standard deviation 0.5 pixels. A data set had 100 2D–3D pairs. Then, the branch-and-bound algorithm was executed to obtain the global solution to the problem of the pose and focal length. The true focal length of the camera was \( f_{\text{true}} = 1000 \). Our search range for the focal length was \([500, 2000]\). We divided this interval into eight sub-intervals of equal length. So, we performed the branch-and-bound rotation search for each of the eight sub-intervals. During a phase of the branch-and-bound, the feasibility of every cube was tested whether or not it could provide a better solution. The next phase then sub-divided feasible cubes into eight pieces before testing their feasibilities, which reduced the side length by half. We did not perform the bisection algorithm to find better solution until the sub-division level reached at the pre-defined value of 8 at which the domain of the rotation had been sub-divided eight times (half-side length = \( \pi/2^8 \)). The branch-and-bound loop was then finished when the sub-division level of the rotation cube reached 16 (half-side length = \( \pi/2^{16} \)).

Fig. 3 shows a result of the experiment for a synthetic data. Among the eight sub-intervals of the focal length, only one sub-interval had feasible rotation cubes remained until the last branch-and-bound rotation search. The other sub-intervals had no feasible rotation cubes after 12th phase. The blue bars in Fig. 3 correspond to the result of the case when we gave the initial error bound as \( e_{\text{init}}^0 = 10 \). When we decreased this initial error bound to \( e_{\text{init}}^0 = 2 \), then we could observe early terminations of the branch-and-bound iteration. The only one sub-interval had feasible cubes until the last phase. Hence, we see that a smaller initial residual may provide a reduced computation since it tightens the bound for the feasibility test. In this experiment, the focal length estimated was \( f' = 999.07 \) with the maximum residual \( e_{\infty} = 1.62 \) pixel and the computation time 427 s.

Next, we generated a hundred sets of synthetic data, each of which was randomly contaminated with a Gaussian noise of standard deviation \( \sigma_c = 0.5 \), and repeated the branch-and-bound algorithm to see its statistical performance. Note that all the solutions we compute are globally optimal; this experiment is to examine the distribution of the focal length through our global optimization algorithm and provide a practical idea of its performance. Fig. 4 shows the focal length estimated against the minimum residual obtained. The residual is in the range of \([0.5, 1.4]\) as shown in the graph, which corresponds to the interval of \([1\sigma_c, 3\sigma_c]\). We believe
that this result is reasonable because the random errors introduced by the Gaussian noise are usually below $3\sigma_C$ and the objective is to minimize the maximum residual. The true focal length is 1000 and the range of the estimated focal lengths is inside the interval [1008,992].

Now, we provide performance statistics for the branch-and-bound algorithm obtained from one hundred experiments using synthetic data. Fig. 5 shows the histogram of the rotation errors calculated by
\[ e = \cos^{-1}\left(\frac{\text{trace}(R^T R_{\text{true}}) - 1}{2}\right) \]
where \(\text{trace}(A)\) is the trace of the matrix \(A\). Average error is 0.055° and the maximum 0.202°. Fig. 6 shows the histogram of relative translation errors defined by
\[ r = \frac{||t - t_{\text{true}}||}{||t_{\text{true}}||} \]

On average the relative translation error was 0.0011 while the maximum was 0.046. Fig. 7 shows the histogram of seconds taken by the computations. The average was found to be 447 s with the maximum of 1892 s. Finally, Fig. 8 shows the histogram of the number of cubes of all the stages through the BnB algorithm. We observe that the maximum number of cubes was as many as 1,128,448 of which similar cases happened a few times during the simulation. Usually the numbers were around $2^{14} \approx 160,000$ as the peak indicates.

### 7.2. Experiments with real data

Finally, we test our method for a real data set called the model-house sequence available on the internet.\(^1\) Fig. 9 shows one of ten images of the model-house sequence. The data set provides camera matrices, matches of corner points extracted from the images and 3D coordinates of the matches. Before we applied the branch-and-bound algorithm, the measurements whose residuals were larger than 2 pixels were excluded from the data set. Our goal is to observe some computational behaviour of the branch-and-bound algorithm using a set of measurements taken from real images.

Fig. 10 shows two graphs of the $L_1$ residuals for the ten model-house images. The green curve corresponds to the residuals before

\(^1\) \url{http://www.robots.ox.ac.uk/vgg/data1.html}
the branch-and-bound optimization, and the blue curve shows the residuals after the branch-and-bound. Then, we look at the effect of the setting on the interval of the focal length by running the branch-and-bound algorithm with different parameter sets. One set had the interval of [400, 1000] and the other had [300, 2000]. The intervals were then divided into four sub-intervals. We measured the computation seconds for every image of the model-house sequence. Fig. 11 shows curves of the computation time in seconds for two different choices of initial bounds for the focal length. The red curve corresponds to the case of [400, 1000], whereas the blue curve [300, 2000]. Interestingly, the computation time does not show a particular dependence on the length of the interval of the focal length as long as it includes the optimal one, which we note from the two graphs.

Finally, in order to observe the effect of using multiple sub-intervals for the focal length, we ran the branch-and-bound with two different numbers of sub-intervals. The first did not use a multiple sub-intervals, whereas the second used sixteen. During the experiments, the interval of the focal length was set to [400, 1000]; Fig. 12 shows two histograms of the number of feasible cubes remaining after every branch-and-bound iteration. The left histogram corresponds to the case of only one interval. The maximum number of feasible cubes was 578,584. When the number of sub-intervals was sixteen, the maximum was 259,880 as shown in the right histogram. In addition, the number of feasible cubes was mostly below 20,000 when multiple sub-intervals were used; the average was 6652. In contrast, the number in the case of one interval was largely below 300,000 and the average was 119,850 (18 times larger). This indicates that the strategy of using multiple sub-intervals is appropriate for parallel processing even though we do not include an experiment for it.

8. Conclusion

This paper presented how to apply the branch-and-bound algorithm to find the $L_1$ global solution of the camera’s pose (rotation and translation) and focal length given a data set of 2D–3D pairs. It shows that the problem of the pose and focal length is a quasi-convex problem when the rotation is known. Then we presented how to apply the branch-and-bound algorithm to search for the best (global) solution. The branch-and-bound algorithm is extended in two ways. The image pixel distance is minimized instead of the angle distance, and the focal length is included as a variable. Finally, a quick algorithm is proposed for solving the feasibility problem which needs to be solved many times repeatedly by the branch-and-bound algorithm.

Our idea to include the focal length in the branch-and-bound algorithm is to utilize the upper and lower bounds of the focal length. Once the bounds are provided, the estimate of the focal length is then computed while the feasibility problem is solved. Another way may be developing a search algorithm that does a branch-and-bound over the focal length as well. The search domain then becomes 4D however; avoiding an explosion of sub-domains for the feasibility test is necessary by, for example, finding a tight bound. We leave this as a future research.

An interesting question related to this kind of global optimization algorithms is whether there are really lots of local minima or whether the global solution may be obtained via a local optimiza-
tion without resorting to a time-consuming computation. As presented in Hartley and Seo [22] and Olsson et al. [23], one may prove, for some problems, that a local minimum is in fact a global minimum. Since any such algorithm for the problem considered in this paper is not yet developed, it could be a good direction for a prospective study.

Finally, the algorithm is suitable for parallel computing as pointed out before. Note that multiple core or GPU-based parallel processing is quite appropriate for the branch-and-bound algorithm since it has a simple parallel structure. Even though the proposed algorithm is somewhat slow to be used for on-line applications, we believe that the BnB optimization can be adopted instead of a local method by exploiting the new computing paradigm of parallel computing because of its parallel structure, which is left as a subsequent development.

References


