Computation of Upper and Lower Bounds of L_2 Error of Multiview Triangulation Using Linear Matrix Inequalities

Zhenjiao Wang[1,a)] Yoshimichi Ito[2,b)] Noboru Babaguchi[1,c)]

Abstract: This paper proposes several methods for computing upper and lower bounds of L_2 error of multiview triangulation. The multiview triangulation is to find the point that minimizes the sum of reprojection errors calculated by points on the image plane observed by multiple cameras. We first show the L_2 optimization for multiview triangulation is reduced to the ones with nonconvex matrix inequality constraints, which are hard to solve. By relaxing the nonconvex matrix inequality constraints, we derive conditions for computing lower bounds of L_2 optimal error. The conditions are represented by linear matrix inequalities (LMI), and they are easy to solve. On the other hand, by tightening the nonconvex matrix inequality constraints, we derive conditions for computing upper bounds of L_2 optimal error, which are also represented by LMI. These methods are easily implemented by MATLAB using tools for LMI such as SeDuMi and YALMIP. The proposed methods are evaluated through numerical examples.

Keywords: multiview triangulation, L_2 optimization, Linear Matrix Inequality (LMI)

1. Introduction

In the field of 3D reconstruction, multiview triangulation problems as a much more important problem than most, have been engaged in research accordingly. In recently years, a feasible way of working with these problems which has been found relies on minimizing the sums of squares (L_2-norm) reprojection error [1]. Furthermore, in a special situation that the L_2-norm of reprojection error equals zero, the reconstructed 3D-point is the same with the real one. To summarize, finding the global minimum of L_2-norm of reprojection error is the main problem in 3D reconstruction, which is defined as multiview triangulation problem. To be more precisely, the multiview triangulation problem is to find the point that minimizes the sum of reprojection errors calculated by points on the image plane observed by multiple cameras. This is also the main problem in this paper. And we call L_2-norm of reprojection error as L_2 error for short.

With the aim of minimizing L_2 error, due to the local minimum problem, global minimum is more efficiently to achieving purpose [2]. In recently years, a large numbers of methods have been proposed besides the method by using bundle adjustment [3] proposed by *F.Kahl* and *R.Hartley* [2]. However, just in three views, this type of solution is under degree 47 [4], not to mention; much less higher degree polynomials are required with more views. High degree make the problem too difficult to calculate in this approach.[5]

In addition, Linear Matrix Inequalities (LMIs) method is also a useful one to overcome the problems. LMIs method is based on the idea that an original non-convex feasible set can be approached by hierarchal convex relaxations to a convex feasible set [6]. On the other hand, the method proposed by Endo shows a way to compute an upper bound of L_2 error. It is a approximation method that the relation to the optimum is not clear. Holding the purpose of overcoming the disadvantages in Endo's paper, a idea that computing but only upper bound, but also lower bound is thought out. Our proposed method is based on the idea proposed in [7] which is also a kind of LMIs method. Instead of calculating L_2 error directly, computation upper and lower bounds is a easier approach for multiview triangulation problems. And with upper and lower bounds, the relation to the optimum can be clear. Especially, when there is no difference between upper and lower bounds, they are equal to optimum.

2. Multiview Triangulation Problem

2.1 Problem formulation

As the simplest model of multiview triangulation for achieving 3D reconstruction, we consider two-view triangulation in this report. The method for general N-view triangulation is derived in a similar fashion. The L_2-norm of the reprojection error, which we call L_2 error, is represented by:

$$E(X) = \sum_{i=1}^{2} \left\| \tilde{x}_i - \frac{A_i X + b_i}{c_i^T X + d_i} \right\|^2 \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean norm, $X \in \mathbb{R}^3$ is a coordinate of a 3D-point X, which is to be estimated, $\tilde{x}_i \in \mathbb{R}^2$ is the observation of X in the image projected onto the i-th camera. $A_i \in \mathbb{R}^{2\times3}$, $b_i \in \mathbb{R}^{2\times1}$, $c_i^T \in \mathbb{R}^{1\times3}$, and $d_i \in \mathbb{R}^{1\times1}$ are block matrices of i-th camera matrix P_i given by:

1 Osaka University, 2-1, Yamada-oka, Osaka 565-0871, Japan
2 Osaka Electro-Communication University, 18-8, Hatsucho, Neyagawa, Osaka 572-8530, Japan
a) wang@nanase.comm.eng.osaka-u.ac.jp
b) ito@isc.osakac.ac.jp
c) babaguchi@comm.eng.osaka-u.ac.jp

© 2016 Information Processing Society of Japan

$$\begin{bmatrix} A_i & b_i \\ c_i^T & d_i \end{bmatrix} = \left[\begin{array}{ccc|c} p_{11}^i & p_{12}^i & p_{13}^i & p_{14}^i \\ p_{21}^i & p_{22}^i & p_{23}^i & p_{24}^i \\ \hline p_{31}^i & p_{32}^i & p_{33}^i & p_{34}^i \end{array}\right] = P_i. \tag{2}$$

The objective function (1) is rewritten as follows:

$$E(X) = \sum_{i=1}^{2} \left\| \frac{\begin{bmatrix} \tilde{x}_i c_i^T - A_i & \tilde{x}_i d_i - b_i \end{bmatrix} \begin{bmatrix} X \\ 1 \end{bmatrix}}{\begin{bmatrix} c_i^T & d_i \end{bmatrix} \begin{bmatrix} X \\ 1 \end{bmatrix}} \right\|^2 = \sum_{i=1}^{2} \left\| \frac{U_i \bar{X}}{V_i \bar{X}} \right\|^2. \tag{3}$$

where $U_i \in \mathbb{R}^{2\times4}$, $V_i \in \mathbb{R}^{1\times4}$ and $\bar{X} \in \mathbb{R}^4$ are given by

$$U_i = \begin{bmatrix} \tilde{x}_i c_i^T - A_i & \tilde{x}_i d_i - b_i \end{bmatrix},\ V_i = \begin{bmatrix} c_i^T & d_i \end{bmatrix},\ \bar{X} = \begin{bmatrix} X^T & 1 \end{bmatrix}^T. \tag{4}$$

For ease of notation, we rewrite $\bar{X}$ by X, and thus, the problem of our concern becomes to find $X \in \mathbb{R}^4$ that minimizes $\sum_{i=1}^{2} \left\| \frac{U_i X}{V_i X} \right\|^2$. Note that, since the norm is the Euclidean norm, the following equality holds:

$$\sum_{i=1}^{2} \left\| \frac{U_i X}{V_i X} \right\|^2 = \left\| \begin{matrix} \frac{U_1 X}{V_1 X} \\ \frac{U_2 X}{V_2 X} \end{matrix} \right\|^2. \tag{5}$$

Note also that the problem is reduced to check the following condition for a given γ.

Equivalent Condition for 2-view triangulation (Condition 0):
There exists $X \in \mathbb{R}^4$ satisfying

$$\gamma^2 \geq \left\| \begin{matrix} \frac{U_1 X}{V_1 X} \\ \frac{U_2 X}{V_2 X} \end{matrix} \right\|^2. \tag{6}$$

The minimum γ satisfying the above condition is the optimal value of the 2-view triangulation problem, and X that attains optimal value is the optimal solution. If there exists $X \in \mathbb{R}^4$ satisfying the above condition, γ is set to be a smaller value, and then, turn to check the condition again. If the above condition does not hold for any $X \in \mathbb{R}^4$, γ is set to be larger, and then, turn to check the condition again. Such a procedure is nothing but a bisection method.

2.2 Schur complement argument

Schur complement plays a central role to derive the results in this paper, which shows that, for symmetric matrices Q and R, the following three conditions are equivalent:

$$\text{(i)} \quad \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \geq 0, \tag{7}$$

$$\text{(ii)} \quad \begin{bmatrix} Q - SR^{-1}S^T & 0 \\ 0 & R \end{bmatrix} \geq 0, \tag{8}$$

$$\text{(iii)} \quad \begin{bmatrix} Q & 0 \\ 0 & R - S^T Q^{-1} S \end{bmatrix} \geq 0, \tag{9}$$

where $A \geq 0$ implies that matrix A is positive semi-definite.

2.3 Conditions for computing L_2 error using matrix inequalities

In [7], based on Schur complement, an equivalent condition to Condition 0 has been given as follows:

Condition 1:
There exist $Y = Y^T \in \mathbb{R}^{4\times4}$ and $X \in \mathbb{R}^4$ satisfying the following conditions:

$$\begin{bmatrix} V_1 Y V_1^T I & 0 & U_1 X \\ 0 & V_2 Y V_2^T I & U_2 X \\ X^T U_1^T & X^T U_2^T & \gamma^2 \end{bmatrix} \geq 0, \tag{10}$$

$$Y = XX^T. \tag{11}$$

Condition 1 is very hard to check because it includes a quadratic term of decision variables, i.e., $Y = XX^T$. In [7], Endo has introduced the following condition:

Condition 2:
$Y = XX^T$ holds, where $Y = Y^T \in \mathbb{R}^{4\times4}$ and $X \in \mathbb{R}^4$ are the optimal solutions of the following LMI1 Problem:

minimize trace(Y)
subject to

$$\begin{bmatrix} V_1 Y V_1^T I & 0 & U_1 X \\ 0 & V_2 Y V_2^T I & U_2 X \\ * & * & \gamma^2 \end{bmatrix} \geq 0, \tag{12}$$

$$\begin{bmatrix} Y & X \\ X^T & 1 \end{bmatrix} \geq 0. \tag{13}$$

(12) and (13) are linear constraints with respect to the decision variables X and Y. In other words, Condition 2 is a linear matrix inequality constraint. Therefore, it is easy to check.

Here, it should be noted that if X and Y satisfy Condition 2 for some γ, they also satisfy Condition 1 for the same γ. Therefore, condition 2 is a sufficient condition of Condition 1. This implies that the following relation holds:

$$\min_{\gamma \in S_1} \gamma \leq \min_{\gamma \in S_2} \gamma \tag{14}$$

where S_i is a set of γ satisfying Condition i.

Since $\min_{\gamma \in S_0} \gamma = \min_{\gamma \in S_1} \gamma =: \gamma_{\mathrm{opt}}$ is the optimal value of the original problem, $\min_{\gamma \in S_2} \gamma$ is an upper bound of γ_{opt}. We refer to $\min_{\gamma \in S_2} \gamma$ as Endo's upper bound, and define $U_{\mathrm{E}} := \min_{\gamma \in S_2} \gamma$.

With method proposed in [7], an upper bound of optimum of L_2 error can be calculated. However, the relation between the upper bound and the optimum is not clear.

3. Computation of Upper and Lower Bounds Using Linear matrix Inequalities

3.1 Computation method for a lower bound derived from rank one LMI

In order to overcome the insufficient result of [7], an idea of computing not only upper bound but also lower bound to estimate optimum have been thought out. By further applying Schur complement to (10), we obtain the following condition:

Condition 3:
There exist $Y = Y^T \in \mathbb{R}^{4\times4}$ and $X \in \mathbb{R}^4$ satisfying the following conditions:

$$\gamma^2 \begin{bmatrix} V_1 Y V_1^T I & 0 \\ 0 & V_2 Y V_2^T I \end{bmatrix} - \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} Y \begin{bmatrix} U_1^T & U_2^T \end{bmatrix} \geq 0, \quad (15)$$

$$Y = XX^T. \quad (16)$$

It should be noted that there exists $X \in \mathbb{R}^4$ satisfying (16), if and only if Y is positive semidefinite and its rank is one. Therefore, we can obtain the following equivalent condition:

Condition 4:

There exists $Y = Y^T \in \mathbb{R}^{4\times4}$ satisfying the following conditions:

$$\gamma^2 \begin{bmatrix} V_1 Y V_1^T I & 0 \\ 0 & V_2 Y V_2^T I \end{bmatrix} - \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} Y \begin{bmatrix} U_1^T & U_2^T \end{bmatrix} \geq 0, \quad (17)$$

$$Y \geq 0, \quad (18)$$

$$\mathrm{rank}(Y) = 1. \quad (19)$$

The set of the constraints (17), (18), and (19) is referred to as rank one LMI. It is straightforward that Condition 4 is equivalent to Conditions 0, 1, and 3.

As a necessary condition, (19) can be ignored because it is a non-convex constraint which is very hard to solve. And the following condition is obtained:

Condition 5:

There exists $Y = Y^T \in \mathbb{R}^{4\times4}$ satisfying the following conditions:

$$\gamma^2 \begin{bmatrix} V_1 Y V_1^T I & 0 \\ 0 & V_2 Y V_2^T I \end{bmatrix} - \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} Y \begin{bmatrix} U_1^T & U_2^T \end{bmatrix} \geq 0 \quad (20)$$

$$Y \geq 0. \quad (21)$$

Here, it should be noted that Condition 5 is a necessary condition of original problem. This implies that the following relation holds:

$$\min_{\gamma \in S_5} \gamma \leq \min_{\gamma \in S_4} \gamma. \quad (22)$$

Since $\gamma_{\mathrm{opt}} = \min_{\gamma \in S_4} \gamma$, $\min_{\gamma \in S_5} \gamma$ is a lower bound of γ_{opt}. We denote the lower bound by L, i.e., $L := \min_{\gamma \in S_5} \gamma$.

Since Condition 5 only includes LMI constraint, it is easy to check by MATLAB using tools for handing LMI constraints such as SeduMi and YALMIP.

3.2 Computation method for upper bounds by using LMIs

Condition 5 itself is merely a condition for computing a lower bound of γ_{opt}, but the solution Y that attains minimum γ satisfying Condition 5 is used for calculating an another upper bound.

If rank$(Y) = 1$, Y satisfies Condition 4. Therefore, the obtained γ is the optimum, and $X \in \mathbb{R}^4$ satisfying $Y = XX^T$ is the optimal solution. On the other hand, when rank$(Y) \neq 1$, $X \in \mathbb{R}^4$ that minimizes $\|Y - XX^T\|$, say $\widetilde{X}$, would be a good approximation of the optimal solution. Since $E(\widetilde{X})$ must be larger than γ_{opt}, $U_1 := E(\widetilde{X})$ is an upper bound.

The rank of Y, as well as $X \in \mathbb{R}^4$ that minimizes $\|Y - XX^T\|$, is calculated by singular value decomposition of Y.

With a method in a similar fashion to the derivation of Condition 2 from Condition 1, we can obtain the following condition from Condition 3:

Condition 6:

$Y = XX^T$ holds, where $Y = Y^T \in \mathbb{R}^{4\times4}$ and $X \in \mathbb{R}^4$ are the optimal solutions of the following LMI Problem:

minimize trace(Y)

subject to

$$\gamma^2 \begin{bmatrix} V_1 Y V_1^T I & 0 \\ 0 & V_2 Y V_2^T I \end{bmatrix} - \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} Y \begin{bmatrix} U_1^T & U_2^T \end{bmatrix} \geq 0, \quad (23)$$

$$\begin{bmatrix} Y & X \\ X^T & 1 \end{bmatrix} \geq 0. \quad (24)$$

Since Condition 6 is a sufficient condition of Condition 3, we have

$$\min_{\gamma \in S_6} \gamma \geq \min_{\gamma \in S_3} \gamma = \gamma_{\mathrm{opt}}. \quad (25)$$

We denote the this upper bound by U_2, i.e., $U_2 := \min_{\gamma \in S_6} \gamma$.

3.3 Summary

In summary, with these methods proposed in this report, we can get two upper bounds and one lower bounds as follows:

type of bound	condition
upper bound	Condition 2 (U_E); Condition 5 (U_1); Condition 6 (U_2);
lower bound	Condition 5 (L);

Table 1 Relation between Condition and Bounds

The smaller the differences between upper and lower bounds is (especially there is no difference between them), we can estimate the L_2 error more clearly. As a conclusion, in these bounds computed in this report, we should choose the smallest upper bound among U_E, U_1, and U_2 to minimize the difference between upper and lower bounds.

4. Simulation and consideration

4.1 Kahl's example for three-view triangulation

In this section, we have used the example which is given in chapter 5 of [8] to evaluate our proposed methods. However, with the aim of evaluating our proposed method, we have completed program to solve the conditions for searching upper and lower bounds. Programs are completed by MATLAB using tools for handing LMI constraints such as SeDuMi and YALMIP [9].

Otherwise, the hardware of the computer which is used to run our program is as follows:

CPU	Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz 3.40GHz
MEMORY	7.89GB
MATLAB ver.	MATLAB R2011b

Table 2 Properties of computer

Example: Triangulation

Considering the following camera triplet:

$$P_1 = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]; \quad P_2 = \left[\begin{array}{ccc|c} -1 & -1 & -1 & 0 \\ 1 & 0 & -1 & 1 \\ \hline 0 & 0 & 1 & 1 \end{array}\right]$$

$$P_3 = \left[\begin{array}{ccc|c} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ \hline -1 & -1 & 0 & 1 \end{array}\right]$$

$P_i \in \mathbb{R}^{3\times4}$: i-th camera matrix.

With camera matrix P, we can compute the necessary parameters U_i, V_i (i=1,2,3) by our completed program. And assume that the measured image point in each view is at the origin (which is no restriction since it can be accomplished by changing coordinate systems).

In the report [8], with this example, the reprojection error has been got by using 3 formulations such as Polynomial, Schur, Bundle. They are all famous methods for solving multiview triangulation problems proposed in recent years. On the other hand, through our proposed method by using this example, we can get three upper bounds and a lower bound introduced in this report. And the results we have got is shown in Table 3 for comparing with other 3 formulation clearly:

Formulation	reprojection error	variables
Polynomial	0.157291	1286
Schur	0.157282	86
U_E	0.157247	12
U_3	0.156871	9
U_2	0.156325	9
Bundle	0.155998	3
L	0.152554	9

Table 3 Data from the triangulation example by our proposed method

In summary, as we see in this example, in the group which includes the upper bounds, the reprojection error which is computed by using Condition 6 is the smallest. As a conclusion, the method we proposed as Condition 6 may be better than other methods.

However, this is just a simplest example, our method should be further evaluated through more numerical examples.

4.2 Numerical evaluation under Kahl's setting

Before evaluating our proposed methods under Kahl's setting, we should explain the synthetic data in similar to the one supposed in [8] by Kahl. All simulated data was generated in the following manner. Uniformly random 3D points with coordinates within [-1,1] units were projected to cameras with focal lengths of 1 pixel. The position of cameras were randomly, bout the must be at distances of 5 units from the origin in average. The cameras' viewing directions were also random, though biased towards the origin. In addition, the image coordinates were corrupted by zero-mean Gaussian noise with varying levels of standard deviation.[8]

This evaluation is main for 3-views triangulation problems, so that in each experiment, 3 camera matrix were chosen. And with the purpose of getting more ideal results, in this report, we will do 500 times with each level of noise and calculating the mean value which is seen as the final result, the function is both suitable for reprojection error and running time.

4.2.1 Comparison in L_2 error

Regarding bisection method, minimization problems are solved, then verify whether the optimum of minimization problem satisfies $Y = XX^T$ or not. We assume that $Y = XX^T$ is satisfied when the maximum singular value of $Y - XX^T$ is lower than $1.0x10^{-10}$.

Running our program for computing reprojection error and running time, the results we get can be a good element for comparing with our methods just as Schur in 3 orders which is proposed in past time and worked well in the example in section 4.1. And the result of comparing among three types of orders for convex relaxation such as first convex relaxation, second convex relaxation and third convex relaxation, and our proposed methods besides Condition 2, Condition 6, Condition 7, has been given as follows:

Table 4: Average upper bound of L_2 error . The method corresponding relationship with the name showing in following table is like this: (1) The existing method using first convex relaxation called Kahl 1; (2) The existing method using second convex relaxation called Kahl 2; (3) The existing method using third convex relaxation called Kahl 3; (4) Method for computing an upper bound proposed by Endo in [7] called U_E; (5) Our proposed method 2 as Condition 5 called U_1; (6) Our proposed method 3 as Condition 6 called U_2;

Noise	Kahl 1	Kahl 2	Kahl 3	U_E	U_1	U_2
0	0.00987	0.00809	0.00359	0.00344	3.72E-12	0.00636
0.01	0.0102	0.00842	0.00388	0.00379	0.00057	0.00738
0.02	0.0112	0.00939	0.00481	0.0046	0.00334	0.00784
0.03	0.0131	0.0112	0.00631	0.0063	0.0044	0.00806
0.04	0.0159	0.0138	0.00865	0.00886	0.00705	0.01089
0.05	0.0199	0.0176	0.0119	0.0123	0.01116	0.01123
0.06	0.0239	0.02144	0.01533	0.01576	0.01521	0.01271

Table 4 Reprojection error by using 6 methods for computing upper bound in 0-0.06 levels of noise

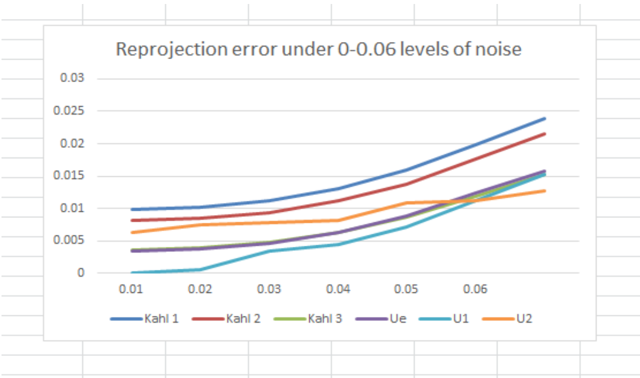

Fig. 1 Reprojection error by using 6 methods for computing upper bound in 0-0.06 levels of noise

Considering the results we have obtained through numerical evaluation: Comparing with the method proposed by Kahl in 3 order, our proposed method as Condition 6 for computing the upper bound may works out the smallest upper bound under Kahl's setting. And in high level of noise, our proposed method 3 may be better than ours.

However, as one and the only lower bound, there is no another lower bound to compare with. Further, we have used the upper bound which are worked well in the experiments done in our report for comparison. And the lower bound of reprojection error which is computed by using our proposed method as Condition 5 is shown as follows:

Table 5: Relation among upper and lower bounds. The method corresponding relationship with the name showing in following table is the same as Table 4. And the only one which has not

© 2016 Information Processing Society of Japan

mentioned in Table 4 is lower bound named method 1.

Noise	L	U_1	U_2
0	7.28E-11	3.72e-12	0.00636
0.01	0.00029	0.000596	0.00738
0.02	0.00090	0.003343	0.00784
0.03	0.002531	0.00316	0.0081
0.04	0.00565	0.007045	0.01089
0.05	0.00595	0.01116	0.0112
0.06	0.00790	0.01521	0.03715

Table 5 Reprojection error of computing lower bound in 0-0.06 levels of noise

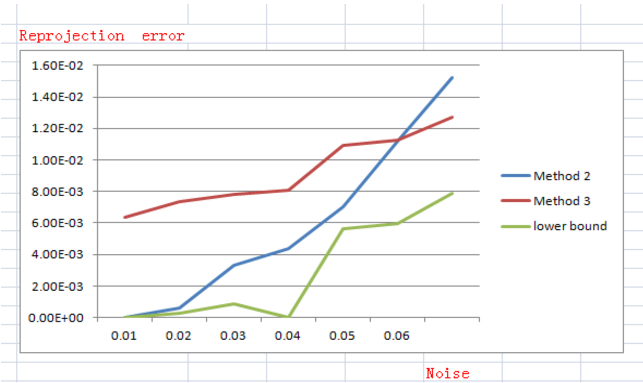

Fig. 2 Lower bound in 0-0.06 levels of noise comparing with upper bound

During computing the upper bound by using proposed method 2, an amazing event had been appeared. That is, bad result which is too large even over 100 had been appeared randomly. It is so inconceivability that one more experiment had been done by using the same reprojection matrix. Through many times of experiments, we obtained the result which is similar to the mean. However, we do not know the reason, we guess is may be cause by property of computer. As a solution, we repeated each experiment in each level of noise three times, and choose rather stable results which are the same in the same condition as the final result.

4.2.2 Comparison in running time

On the other hand, running time is also an important element for evaluating methods and algorithm. And we have obtained the running time of each formulations just as follows:

Noise	Schur1	Schur2	Schur3	Bisection	U_1	U_2
0	1.97862	3.75806	14.5084	4.49283	3.03734	3.94215
0.01	2.04065	4.31047	13.9983	4.27755	3.156	3.90315
0.02	1.93444	4.90249	15.1640	4.44759	3.24482	3.88599
0.03	1.91903	4.899	17.3896	4.34463	3.24950	3.77990
0.04	1.95725	4.85883	18.470	4.40703	3.172	3.65978
0.05	2.04979	4.83403	17.9507	4.48659	3.20582	3.63482
0.06	2.05859	4.7921	18.5777	4.41527	3.187	3.61142

Table 6 Running time by using 6 methods for computing upper bound in 0-0.06 levels of noise

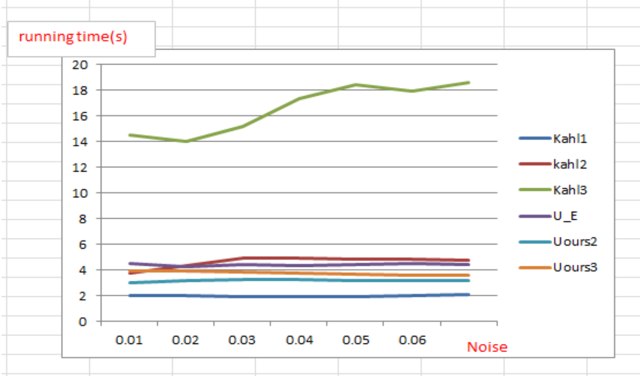

Fig. 3 Running time of each time in 0-0.06 levels of noise

Noise	0	0.01	0.02	0.03
Running time	2.887267	2.906455	2.918779	2.922616
Noise	0.04	0.05	0.06	
Running time	2.925816	2.936469	2.936727	

Table 7 Running time of computing lower bound in 0-0.06 levels of noise

Based on the running time, we know that our proposed methods is faster than the method proposed by Kahl in order 3 which is works well in computing upper bound. On the other hand, the ones proposed by us is much slower than Kahl's method in order 1.

5. Conclusion

In this report, we have proposed several methods for computing upper and lower bounds of L_2 error of multiview triangulation problems. In detail, two methods for computing upper bounds such as the Condition 6 and Condition 7 introduced in the report, and one method for calculating lower bound which is introduced as Condition 5.

Our proposed methods are represented by convex linear matrix inequalities (LMI), and they are easily implemented by MATLAB using tools for handling LMI constraints such as SeDuMi and YALMIP. In this report, we not only proposed our methods for solving multiview triangulation problems, but also evaluating our methods by comparing with Kahl's and Endo's methods which are proposed in recently years. And we have chosen a simple virtual digital example which is given in chapter 5 of Kahl's paper and more numerical examples by using synthetic data which are randomly. With the results, we have realized that our proposed method as Condition 6 for computing upper bound is relative usefulness than other methods in low level of noise. And our method as Condition 7 works well in high level of nose. On the other hand, the method as Condition 9 is the only one for computing lower bound, even though the difference between it and smallest bound is little, it need to validate effectiveness. Further research in finding out the much more lower bound which we would like to deal with in the future.

References

[1] R. Hartley and A. Zisserman: *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004.

[2] R. Hartley and F. Kahl: Optimal algorithm in multiview geometry: *In Asian Conf. Computer Vision*, pp. 13–34, 2007.

[3] W. Triggs, P. F. McLauchlan, R. Hartley, and A. Fitzgibbon: Bundle adjustment for structure from motion, *In Vision Algorithms: Theory and Practice*, pp. 298–372, 2000.

[4] H. Stewenius, F. Schaffalitzky, and D. Nister: How hard is 3-view triangulation really? *In International Conf. on Computer Vision*, pp. 686–693, 2005.

[5] R. Hartley, F. Kahl, C. Olsson, and Y. Seo: Verifying global minima for L_2 minimization problems in multiple view Geometry, *International Journal of Computer Vision*, Vol. 101, No. 2, pp. 288–304,

© 2016 Information Processing Society of Japan

2013.

[6] D. Henrion, and J. B. Lasserre: Convergent relaxations of polynomial matrix inequalities and static output feedback, *IEEE Transactions on Automatic Control*, Vol. 51, No. 2, pp. 192–202, 2006.

[7] T. Endo, Y. Ito, and N. Babaguchi: LMI-based method for 2-norm optimization in triangulation problems, (in Japanese), *IPSJ SIG Technical Report on Computer Vision and Image Media*, Vol.2013-CVIM-187, No. 8, pp. 1–8, 2013.

[8] F. Kahl and D. Henrion: Globally optimal estimates for geometric reconstruction problem, *International journal of Computer Vision*, Vol. 74, No. 1, pp. 3–15, 2007.

[9] J. F. Strurm: Using SeDuMi 1.02, a MATLAB Toolbox for Optimization over Symmetric cones, *Optimization Methods and Software*, Vol. 11, No. 1-4, pp. 625–653, 1999.

© 2016 Information Processing Society of Japan