Response to comments of reviewer 2

We thank the reviewer for the helpful comments and suggestions, and for careful reading of the manuscript. Listed below are our itemized responses, with the original comment/question displayed in italics.

GENERAL COMMENTS

====

The paper describes a minimum curvature based regularization schemed for deriving 2-D trace gas concentrations from optical remote sensing and tomography. The chosen regularization scheme is sensible and the method seems to be an improvement over the state of the art in the field.

The topic fits the journal.

The textual description is good and the different methods are well introduced.

The paper may be published after addressing the following minor issues.

Thanks for the comments. We have revised the manuscript according to your comments. Please see the responses below for specific changes we have made to the manuscript.

SPECIFIC COMMENTS

=====

Eq. 4

The finite difference operator requires the division by the third power of the grid distance. The results are likely correct as one needs to multiply with \delta x \delta y in Eq. (5) to properly approximate the 2-D integral over the third order derivative. However, in case \delta x unequal to \delta y, an error would be made.

Thanks for the correction. We have changed the division to the third power of the grid distance in Eq. 4 to make it more accurate.

Eq. 7

In what way are the matrices T and D_3 similar/different?

Matrix T contains the third derivatives of all the pixels in both x (row) and y (column) direction. For the derivatives in x direction, the matrix is defined by D_3. For those in y direction, each row of the matrix still contains the coefficients [-1 3 -3 1], but they are separated by n (the number of pixels in x direction) "zero" elements to correctly define the difference in the y direction, because in y direction, the difference of index numbers of two neighbor pixels is n.

Eq. 8

Here the fourth symbol is introduced for a weight factor after w, W, and μ. Is this necessary, or can one simplify the employed notation?

They all can be viewed as weight factor in this study. But they may have different interpretations in different fields. For example, the w in the LTD approach represents the weight for a linear equation and was not described as regularization parameter. W is a matrix containing the weights for all linear equations. The weights for different equations can also be different. μ is a single value to determine the weight of the regularization term in the regularized inversion problem. In this view, ω in Eq. 8 and Eq. 10 and λ in Eq. 16 are also regularization parameter and have been replaced by μ according to the reviewer's suggestion,

Eq. 10

Here, a needless approximation error is introduced via the c_i. You derive, effectively, a continuous representation f of the 2-D distribution, which could be used for a better "forward model" F mapping the continuous distribution f onto measured values b. You might think about being more general here before justifying your choice of forward model based on the pixel-based approach.

Thanks for the suggestion. In a general case, the *PIC* of a path is a line integral over concentration field. We have added this integral formula to the manuscript.

Eq. 12

The grid distance is missing in the discretized integral. Please differentiate. Also, I is now both a function of f and a sum. As the "new" I is differentiated later, it would be good to introduce it as a function here as well.

Thanks for the correction. The grid distance has been added to the equation. To distinguish continuous and discrete form, we have changed the discrete total squares curvature I to $S(C)$, which is a function of pixel concentrations.

Eq. 16

I still do not understand why Eq. 14 and the higher derivative was introduced. Eq. 16 describes a minimization problem and it should be fully sufficient to minimize \omega I(f). So why introduce Eq. 14? That the higher derivative is zero is a necessary condition for the existence of a minimum, but that is a detail of the employed minimizer. Unless a Newton-type-method is used, Eq.14 is not necessary and seems to complicate the computation. I assume that the algorithm will still work with this seemingly needless complication.

Eq. 16 is the result of the derivation. If Eq. 14 is not introduced, then we will not get the matrix M in Eq. 16. It is true that we can also work on the original form of I(f) instead of M, and use a different optimization method to solve the minimization problem. In this study, Eq. 16 is actually the result after using the Newton-type-method. We use this form because it is similar with the formula of LTD algorithm. The only difference for the two approaches is the matrix M comparing with D_3 which is used in the LTD approach. Thus, the same technique can be used to solve the minimization problem, which makes the comparison more meaningful.

Response to comments of reviewer 3

We thank the reviewer for the helpful comments and suggestions, and for careful reading of the manuscript. Listed below are our itemized responses, with the original comment/question displayed in italics.

General

As compared to the first version, the paper by Sheng Li, Ke Du has been significantly changed. Many of the concerns I pointed out in the first review have been addressed. I apologize that while reading the new version of the paper, unfortunately, I found some new relevant issues. Some of them are crucial and, to my opinion, should be carefully considered before publication.

In general the language has been greatly improved, however, I would recommend the authors to double check if the editing process has always conserved the original meaning of what they wanted to say.

Thanks for the comments. We have revised the manuscript according to your comments. Please see the responses below for specific changes we have made to the manuscript.

Specific comments

Sect. 2.1:

the section has been significantly improved. I see only two remaining issues:

a) Linearity of equation (2). *Your statement at line 848 of your replies "The predicted PIC for one beam equals to the summation of the multiplication of the pixel concentration and the beam length passing it" holds only in the "optically thin" regime, i.e. only when a small fraction of the laser signal is absorbed by the pollutant to be measured. If the pollutant concentration is large and/or the medium is not very transparent at the laser wavelength, then for sure linearity does not hold. Since the laser wavelength can be tuned, usually it is possible to find a suitable wavelength at which, for average or expected pollutant concentrations, the medium is sufficiently transparent to fall in the linear regime. To my opinion it would be convenient to state explicitly that you are working in this hypothesis.*

Thanks for the suggestion. We are using open path TDL analyzer, which is already well-tuned equipment. The wavelength of the laser beam is tuned to the absorption line of the target gas and is transparent to other species. The laser absorption is in the linear regime and the attenuation of the laser beam is governed by the Beer–Lambert law. This introduction gives a description how the PIC is calculated by the equipment. So the condition of "optical thin" was satisfied in our field experiments. We also added a sentence on page 5 to state that our calculation was based on linearity between laser absorption and pollutant concentration.

b) Equation (3)*: is it really needed to require c ≥0 ? Assume that the real concentration of the pollutant to be measured is zero. If you obtain the concentration as the average of several measurements, then, because of the analyzer's noise, these measurements should be evenly distributed around zero to give a zero average value. If you constrain the solution c to be greater or equal to zero, then you remove the measurements that otherwise would be negative because of the measurement noise. Thus you introduce a bias (or exposure error) in the average. Of course you don't see this type of error in the tests presented later in the paper, because the test 2D distributions considered do not include a "zero concentration" case, and because all your formulas fully ignore the measurement noise error (that is the noise error on your b vector).*

We understand that non-negative constraint may result in bias when averaging the concentrations where only noise exists. However, this is a trade-off as we need to focus on signals well above noise. Depending on the inverse problem, the optimal solution may contain very unrealistic negative value. This is especially true for the case in the 2-D tomographic mapping of air pollutants, whereas the data is limited, and the system of linear equations is ill-posed (e.g., rank-deficient). Therefore, it is necessary to apply these constraints to eliminate such unrealistic solutions because the resulted negative values are not just at noise level. They may be comparable with the positive peak values. In addition, there are different sources of errors. In the simulated study, one error source is discretization error. Even in this case, the results generated without non-negative constraints are not useable. After using the regularization technique, the optimized solution is robust to the measurement noise, and the reconstruction error is dominated by regularization error. Therefore, the measurement noise is negligible, and we can focus the study on the regularization algorithm itself. We have added a sentence below Eq. (3) to explain the necessity of using non-negative constraints.

Sect. 2.2:

If I understand properly, in practice what you call LTD method is the Tikhonov regularization using the discrete 3rd derivative operator D_3. You tune the regularization strength w using the discrepancy principle. Then I have a question: you can not introduce the two equations (4) at every pixel of your domain, as you would have problems at the edges of the domain (see index values k+2, k+1 and k-1 in eq. 4). Thus, you can actually introduce only $2N_c$- 6 additional equations (not $2N_c$ as you state at line 160). If you do so I think you get in trouble if one or more pixels of your domain are not crossed by a ray path. Actually, in this case L^tL would be singular and, if also T^tT is singular, then you cannot invert the matrix A^tWA mentioned at line 172. In conclusion, I think you are using some extra constraints (also involving 0-th order derivatives) at the edges of your domain (as you seem to suggest in the text after eq. 7). Please clarify. Also, what are the symbols m and n in eq. 7?

***Line 243**: please explain clearly the equations you use at the edges of the domain, see also the analogous comment above.*

They were indicated at line 182 and 243 in the manuscript that the constraints at the edges are second-order and first-order difference operators. According to reviewer's comment, we have added their formulas to the manuscript. *m* and *n* are the row and column number of the pixels. They keep the same definition across the manuscript. It is fine if a pixel is not crossed by a ray path. With the high-resolution grid division, most of the pixels are not crossed by the beam. They are restricted by the smoothness constraints.

Line 245*: requiring $c \geq 0$ introduces a bias (i.e. a systematic exposure error) for very small values of the real concentration distributions c, see also the analogous comment above.*

The non-negative constraints are necessary to eliminate the unrealistic negative solutions resulting from measurement errors and ill-posed inverse problem. Please see the response in Sect. 2.1. (b).

Sect. 2.4:

Here you define the "true" concentration distributions that will be considered later to assess the various reconstruction methods. You should say also something regarding the synthetic observations b that you build on the basis of g(x,y). Which is the spatial mesh (or grid) that you use for the calculation of b, starting from g(x,y)? Do you add pseudo-random measurement noise to b, to emulate the real world situation in which the instrument is subject to measurement noise? Your equations discard the measurement error. Mathematically this is equivalent to assume uncorrelated noise equal to 1, thus you should add to each element of b a pseudo random noise with zero average and standard deviation equal to 1.

The concentration filed is discretized with a resolution of 0.2 m×0.2 m. The concentration of each pixel is the average value of the concentrations in that pixel. The discretized concentration map is used as the true concentration distribution. PICs are calculated based on the discretized map by using Eq. (1). We have added this description to the Sec. 2.4.

The purpose of this study was to study the different regularization techniques to produce smooth reconstruction, instead of studying the influence of the measurement noise. After using the regularization technique, the optimized solution is robust to the measurement noise, and the reconstruction error is dominated by regularization error. Therefore, the measurement noise is negligible.

Lines 293 and ff*: the unusual naming conventions used in this part make it very difficult to understand for me and, I guess, also for the whole atmospheric community. G matrices defined at lines 298-300 are usually called "gain matrices". It is true that in this case they also represent the "generalized inverse" of L because the forward model (Le) is linear. Despite of that, I would still call G the "gain matrix", just for uniformity with Rodgers 2000. Equations at lines 299 and 300 should contain a"+" sign within the parenthesis. Matrix R is usually called "averaging kernel", not "resolution matrix", just because it shows how a change in the real concentration field maps onto the concentration estimated by the inversion system. The "resolution" is usually a scalar or vector quantifier computed on the basis of R. Line 304: what you call "regularization error" is usually called "smoothing error".*

The names the reviewer gives are common in the atmospheric sounding problem. But the notations we use are more common in the problem of 2-D or 3-D tomography. It may be necessary to discuss the differences between the atmospheric sounding problem and 2-D gas tomography.

1) One of the differences is apparently the 1-D inversion versus 2-D inversion. In the 2-D tomographic applications, a new factor to consider is the geometric configuration of the beam paths, whereas one does not need to do in the 1-D case. As a result, the path geometry largely affects the inversion. And this influence can be seen from the averaging kernel matrix (see Fig. 4 in the manuscript).

2) For retrieval of vertical profile of an atmospheric quantity, a priori profile (x_a) of the unknowns is usually introduced. The error caused by the priori profile can be measured by using the averaging kernel matrix, which makes it an important tool to evaluate the sensitivity of the inversion. However, in 2-D gas mapping problem, there is no *a priori* distribution of the gas concentration introduced. Therefore, the significance of the averaging kernel matrix is reduced.

3) The averaging kernel matrix is also used to evaluate the regularization error, which is determined by the inversion algorithm instead of the a priori profile. This is what we did in this study. In this case, the regularization term forces the off-diagonal terms to be nonzero, thereby making the estimated concentration of each pixel a weighted average of the concentration of the surrounding pixels. As a result, there is no need to pursue a prefect averaging kernel matrix. The smooth weighting of the matrix is intended and necessary. This is also the focus of this manuscript.

In summary, we think there are differences between the two applications fields. We think it is a hard choice and decide to change the 'resolution matrix' to 'averaging kernel matrix' and keep other names unchanged.

Line 305*: what is the perturbation error G delta_b? Why should the observed quantities be perturbed? Maybe delta_b is the measurement error (noise + calibration, etc.)? Please explain.*

The perturbation is the measurement error due to various noise sources. We have updated the description in the manuscript.

Line 307*: note that the non - sensitivity to the "perturbation error" is not always an appreciated property for the solution. In fact this means that the solution is mostly independent from the observed quantity, therefore it mostly depends on the applied constraints, that is exactly what one usually wants to avoid.*

As indicated in the manuscript (section 2.5) and in the response in Sect. 2.1. (b), the study focuses on the regularization term, which dominates the reconstruction error comparing to the perturbation error. In this case, the regularization error cannot be avoided and can be used to evaluate different regularization techniques. Please refer to the discussion of the differences between the atmospheric sounding problem and the 2-D gas mapping problem (Sect. 2.4. Lines 293 and ff).

Eq. 23*: why don't you write explicitly the expression for the Frobenius norm? Which is the advantage of using the Frobenius norm over the more usual trace[R] that, according to Rodgers, 2000 would represent the number of degrees of freedom for the signal? If R is not a orthogonal matrix (like in this case, I guess) then I am not able to find an easy interpretation for your quantifier of eq.(23)*

Frobenius norm is the square root of the sum of the absolute squares of the elements of a matrix. We have added the definition to the manuscript. *R* is not an orthogonal matrix and its row contains the weights of all pixels, which are determined by the path geometry and the regularization algorithm discussed in the response of Lines 293 in Sect. 2.4. As a result, the weights are spread over the pixels, and trace[R] is not useful in the study of either the regularization error or the path geometry, which is why we use the Forbenius norm.

Line 365 and ff: how do you interpret the systematically slightly worse performance of MC as compared to LTD in terms of exposure error? I am also surprised to see such a large exposure error in the NNLS method as compared to the other two methods. Maybe this due to the larger size of the pixels used with NNLS? In this case it would be better to change eq.(20) by multiplying each concentration by the area of the pixel to which it refers (actually eq.(20) should compare the integrals of the concentration in the considered domain).

After carefully examining the results. We found that that large error of NNLS and slightly worse performance of MC were mainly caused by using the spline interpolation after the reconstruction. Therefore, we have reproduced the results by using the nearest which has minimal effect on the original results. The new results illustrate that the MC algorithm shows slightly better performance than LTD in all cases, and the performance of NNLS is greatly improved than the previous results. We have updated Table 3 and the discuss of the results.

Eq. (20) already includes the integrals of all the pixels in the domain. All the reconstructed pixels are compared with the simulated 'true' pixels. Therefore, they are using the same number of pixels no matter what their original grid divisions are.

Sect. 3.4:

at line 144 of the revised paper finally you reveal that you are using the "lsqnonneg" MATLAB routine to minimize the cost functions relating to the various methods considered. I was not able to establish what is exactly the minimization method used by this MATLAB function, however, for sure it does not find the solution mentioned at line 172 of your revised manuscript, because this latter is not bound to be positive. As explained later, I suspect that lsqnonneg is using iterations, which is not optimal at all for the linear case (b = Le) you are dealing with. This implies that the run-time analysis presented in sect. 3.4 is applicable only to the unlucky case in which one uses lsqnonneg. The optimal solution of your inversion would be to implement, directly, in a computer program the matrix operations at line 172. This would make the computation time dependent only on Nb (number of measured PICs) and on Ne (number of unknowns). Both Nb and Ne do not depend on the number of sources that you put in your domain, thus the computation time would not depend neither on the number of sources considered nor on their amplitude and location in the domain. You also claim that the MC method is much faster with respect to the LTD because of the fewer constraint equations. To my experience, if the solution at line 172 would be implemented, most of the computation time would be spent in the inversion of matrix A1WA, of dimension Ne. The number of constraint equations used would impact only the calculation of the product TT (that is needed to compute A'WA), thus the total computation time should depend only very marginally on the number of constraints used.

The NNLS algorithm of 'lsqnonneg' uses the algorithm described by Lawson and Janson (1995). As indicated in the manuscript, it is an active-set optimization method using an iterative procedure to converge on the best fit of positive values. The analytical solution is only demonstrated for the case without non-negative constraints. But it is not applicable in the problems with non-negative constrains and was not used in this study. Because of this, an iterative optimal algorithm needs to be used to solve the constrained inversion problem. Thus, your concern regarding computation time of *lsqnonneg* routine is not applicable in this case. We have added a sentence in the paragraph under Eq. (5) to explain this.

Another discussion point are the absolute values of the computing times shown in Table 4 and their standard deviations: why the computation time changes and thus shows a standard deviation also when the method used and the number of sources do not change? (I suspect this is because lsqnonneg uses iterations whose number changes from test to test). Why is computation time so long? Are you using a very old CPU? Please specify. Assuming your inversion problem dimensions, finding the solution given at line 172 would require less than 1 second with a standard CPU. In conclusion, finding the solution as c = (AT W AJ1 AT W p (as specified at line 172) would be much faster and the savings in computing time achieved with the MC method over the LTD would be much less important

lsqnonneg uses an iterative procedure to converge on the best fit of positive values. Therefore, its speed is affected by the values of the coefficient matrix. This is common for an iterative optimal algorithm. We used a modern computer to do the calculation. The configuration of the computer has been listed according to the comment. The fast speed the reviewer gives is based on the analytical formula, which is not applicable in this study as explained in previous response.

Line 378*: what do you infer from fig's 4a and 4b? Are they useful? I was not able to extract information from them. In my previous review I was suggesting to include only Fig.s 4e and 4f. Of course additional figures are welcome, however they should convey useful information that should also be discussed in the text.*

The first two figures are the visualization of the fitness function. It is in 2-D form. Therefore, the values are represented by colors, and we cannot expect a curve like that in the 1-D inversion. We have revised the description.

Line 388*: figures 4e and 4f show diagonal elements significantly smaller than 1, this means that your regularization is actually very strong (w is very large). What do you get for trace[R] / Ne? I guess it is < < 1. In practice I feel that with a much softer regularization you would get solutions with better accuracy and similar smoothness.*

As indicated in the responses of Eq. 23 in Sect. 2.4, the elements in a row of the averaging kernel matrix are determined by the path geometry and regularization terms. The purpose is to evaluate the regularizations and this observation implicates it is working. The trace is not helpful in this case and fitness value was used.

Figure 4: at least, please use the same color scale in the left and right maps, otherwise the comparison of the two methods is very hard ....

The figures have been updated according to the suggestion.

Section 3.6:

the results reported in this section are not useful as they are not general for the experimental setup considered, they depend on the dimension of the assumed sources. From what stated at line 399, I understand that these sensitivity tests are performed assuming 5 Gaussian source functions, whose minimal Full Width at Half Maximum (FWHM) is 2.8 m x 2.355 = 6.59 m (see line 272 and the properties of the Gaussian function). When the dimension of the smallest source function is about 6.6 m, clearly there is no much advantage to go from a 18x18 grid (pixel size = 40m/18 = 2.22 m) to a 40 x 40 grid

(pixel size = 40m/30 = 1.33 m) as in both cases the pixel size is significantly smaller than the source size. Usually, in real situations, the size of the source is unknown and what is useful to know is the spatial resolution of the measuring plus inversion system. The averaging kernels are usually an adequate tool to answer this question: the dimension of the main yellow spots of maps in fig. 4c and 4d quantify the spatial resolution of your measuring system at the location of indexes 16 and 4.

We used the same settings of the sources with the GT-MG, whereas the values are common for mapping of small-scale air pollutants. These results generally show us the trends with changing grid size. It was also indicated in the manuscript that there is a certain threshold value from the trend, which was 24×24 in this case. It may be a different resolution for other application, but the trends are the same. Fig 4. (c) and (d) clearly show the averaging kernels mainly determined by the beam geometry, which makes it a good tool for optimizing the beam configuration.

Lines 410 - 413: from the theoretical point of view, the growth of computation time for this inverse problem should roughly follow a polynomial law as a function of the number of pixels. Did you really prove that the behavior you find is exponential? If the behavior is really exponential then you should give a justification so as why the computation time increases so rapidly in your case.

The exponential trend was concluded from an exponential fit with a R^2=0.9996. Our purpose is to show that the computation time grows fast with increasing pixels and the trend is 'approximately' exponential. Of course, we can fit it with a polynomial (dependent on what order you use). But we are not intended to prove what exactly the trend it should follow, which is related to the details of the optimization algorithm. What we did was to use the same optimization algorithm for all the three reconstruction techniques presented in this study to make sure the comparisons were based on the same standard.

Conclusion section: as outlined above, in my view the time computing savings of the MC method, if any, should be assessed using an optimal solving algorithm for your inversion problem, not the MATLAB lsqnonneg routine. In conclusion, either you renounce claiming the time savings or you implement an optimized solver (like eq. at line 172 implemented in FORTRAN or in C).

As explained in the response of the second question in Sect. 3.4 responses, the analytical formula is for the problems without non-negative constraints and is not applicable to the problems with non-negative constraints in this study. Also, the NNLS algorithm that the lsqnonneg uses is the most widely used inversion algorithm for solving non-negative least squares problems.

Minor corrections

Line 14:

I would say "regularization term" (regularization is not a factor in the inversion formula)

"term' has been used.

Line 22 (and line 113):

It is also simpler to perform .. . maybe you meant "implement" instead of "perform" ?

'implement' has been used.

Line 204:

bending ?

Changed to “potential energy stored in a bended elastic object.”

Line 272:

units of sigma_x and sigma_y should be "m" (meters).

The unit 'm' has been added.

Line 416:

seminors ? ·

'seminorms' has been used.