Computer Vision and Image Understanding **80,** 172–214 (2000)
doi:10.1006/cviu.2000.0869, available online at http://www.idealibrary.com on IDEAL®

A Critique of Structure-from-Motion Algorithms

John Oliensis

NEC Research Institute, 4 Independence Way, Princeton, New Jersey 08540
E-mail: oliensis@research.nj.nec.com

Received April 27, 1999; accepted June 19, 2000

I review current approaches to structure from motion (SFM) and suggest a framework for designing new algorithms. The discussion focuses on reconstruction rather than on correspondence and on algorithms reconstructing from many images. I argue that it is important to base experiments and algorithm design on theoretical analyses of algorithm behavior and on an understanding of the intrinsic, algorithm-independent properties of SFM optimal estimation. I propose new theoretical analyses as examples, which suggest a range of experimental questions about current algorithms as well as new types of algorithms. The paper begins with a review of several of the important multi-image-based approaches to SFM, including optimization, fusing (e.g., Kalman filtering), projective methods, and invariant-based algorithms. I suggest that optimization by means of general minimization techniques needs to be supplemented by a theoretical understanding of the SFM least-squares error surface. I argue that fusing approaches are essentially no more robust than algorithms reconstructing from a small number of images and advocate experiments to determine the limitations of fusing. I also propose that fusing may be one of the best reconstruction strategies in situations where few-image algorithms give reasonable results, and suggest that an experimental understanding of the properties of few-image algorithms is important for designing good fusing methods. I emphasize the advantages of an approach based on fusing image-pair reconstructions. With regard to the projective approach, I argue that its trade-off of simplicity versus accuracy/robustness needs more careful experimental examination, and I advocate more research on the effects of calibration error on Euclidean reconstruction. I point out the relative lack of research on adapting Euclidean approaches to deal with incomplete knowledge of the calibration. I argue that invariant-based algorithms could be more nonrobust and inaccurate, and not necessarily much faster, than an approach fusing two-image optimizations. Based on recent results showing that two-image reconstructions are nearly as accurate as multi-image ones, I suggest that the authors of invariants methods conduct careful comparisons of their algorithms to two-image-based results. The remainder of the paper discusses the issues involved in designing a generally applicable SFM algorithm. I argue that current SFM algorithms perform well only in restricted domains, and that different types of algorithms do well on quite different types of sequences. I present examples of three domains that are important in applications and describe three types of

1077-3142/00 $35.00

algorithms, each of which performs well in just one of the three domains. I advocate testing current algorithms on a wider variety of sequences to determine their limits of applicability. More generally, I propose that SFM is a messy problem and that it could require a flexible "intermediate-level" system incorporating a variety of different algorithms and sophisticated decision rules for combining them. The paper concludes with a general discussion of experiments, pointing out that real-image experiments are not always the best means of evaluating reconstruction algorithms.

Key Words: Structure from motion; multi-frame structure from motion; projective methods; invariants, self-calibration; fusing; Kalman filtering; optimization; trilinear reconstruction; Bayesian methods; experimental evaluation.

1. INTRODUCTION: PURPOSE AND PHILOSOPHY OF THIS PAPER

Structure from motion (SFM) has been the central problem of computer vision for over 15 years, with more papers devoted to it than to any other area. Relative to this intense interest, researchers have put little effort into theoretical analyses of algorithm behavior. Yet such analyses are important not only for understanding algorithms' properties, but also for conducting good experiments and for developing the best algorithms. Although neglecting theory was appropriate in the early days when researchers were searching for algorithms, the field has matured, and it is time to use more care in defining new algorithms and in evaluating existing ones. This paper redresses the gap in the literature by sketching some preliminary theoretical analyses of algorithm behavior.

The remainder of this Introduction illustrates some of my arguments with examples. More background on the examples appears in the body of the paper, and the reader may want to return to them after reading it.

Theory-based experiments. Most SFM papers neglect to propose a theoretical context for understanding their experiments. They show only that an algorithm performs reasonably on a small, select set of sequences, with no analysis of such questions as:

- How intrinsically easy are the test sequences? Would any algorithm give good results on them?
- What other algorithms would also do well on these sequences and should be run for comparison?
- What type of sequence would give a truly stringent test of the algorithm, displaying its advantages over others?

Without answers to questions like these, it is impossible to tell how effective an algorithm really is. Such questions cannot be answered purely by experiments; *some* theoretical framework is needed. One needs theory both for interpreting the results of experiments and for deciding which experiments are important to do. One of the main goals of this paper is to convince the reader that theoretical analysis is a crucial part of experimentation.

This reminder is needed since many experiments that theory suggests are important have not been performed. For example, multi-image invariant-based algorithms such as [28, 30, 81, 82] do not exploit the image data optimally, and the theoretical arguments in this paper suggest that they might exploit it poorly. Even without this suggestion, it is natural to ask whether these approaches are better than two-image algorithms (or combinations of them), which do use the image data optimally or near optimally [63, 64, 67, 111]. Does

the additional data used by the multi-image algorithms compensate for their imperfect exploitation of the data? Amazingly, the answer to this question is still unknown.

Another example: despite the enormous amount of work on developing projective algorithms, we still do not know when the projective approach is the right tool for its main task of dealing with calibration uncertainty. The projective approach assumes zero knowledge of the calibration. In practice, one invariably has some knowledge about it, and it is natural to ask when assuming complete ignorance is the best strategy for dealing with a partial uncertainty. To answer this, one thing we need to know is whether the calibration uncertainty is large enough in practice to justify assuming that it is total, where "largeness" is measured by its effect on the goal of reconstructing. That is, we need to understand how calibration error affects reconstruction. Amazingly, there are still no thorough experimental studies of this important question.[1] There *are* hints that the answers might be interesting and have algorithmic consequences. Calibration errors often appear to cause small distortions in a reconstruction, or distortions mainly in a few parameters, which suggests that it might be straightforward to isolate and account for these effects by adapting a Euclidean approach. Only experiments can determine when projective methods deal better with calibration uncertainly than such a Euclidean approach.

To illustrate theory's role in guiding experiment, I present theoretical analyses of several current SFM algorithms in this paper. I recognize that these analyses require experimental confirmation—I propose them to suggest important experimental questions and to underline the theoretically important experiment that researchers have neglected. Given the maturity of SFM, I argue that it is time for researchers to begin addressing questions such as these. If others disagree with my analyses, I welcome them to examine them experimentally (that is what they are for) and to develop alternative analyses for grounding experiment.

Although I do want to illustrate how theory raises questions about current algorithms, it is not the purpose of this paper nor my responsibility, to experimentally evaluate algorithms developed by others. But, after completing the first version of this paper, I did carry out experiments with V. Govindu to illustrate the type of studies I am calling for [64, 67, 72]. For example, we studied in [64, 72] my claim that projective reconstruction is less accurate than Euclidean reconstruction and experimentally demonstrated situations in which this becomes important. In [67], I studied the accuracy of two-image algorithms in comparison to multi-image ones showing that two-image reconstructions were nearly as accurate as 15-image ones *if* one analyzed the results based on an appropriate theoretical analysis of the bas-relief ambiguity. Recently, I have found that the Sturm–Triggs algorithm [6, 40, 96] works well for complex camera motions but often fails when the camera moves along a line. This reinforces my point that any one SFM algorithm is unlikely to perform well in all situations and that it is important to analyze and delimit experimentally the situations in which an algorithm fails.

Phenomenology. I advocate a particular kind of theoretical analysis in this paper—what I call a *phenomenological* analysis. Of course, some theoretical work on algorithm behavior already exists, mostly in the form of first-order error-sensitivity analyses [45]. Phenomenology differs from this. I use the term to mean an attempt at modeling the behavior of a "natural" phenomenon that is neither purely theoretical nor purely experimental but, necessarily, some combination of both.

[1] Ma *et al.* [55] and Sturm [93, 94] present initial studies.

One simple example of a phenomenological principle is the bas-relief ambiguity [9, 18, 50, 67, 97]. One can prove this ambiguity rigorously in the limit case of an infinitely small scene and two images. One can also derive first-order estimates indicating that the error sensitivity associated with this ambiguity persists over some reasonable range of conditions. But experiments are needed to show that it occurs in most practical situations (though, without the theory, one might not have noticed the trend in the experimental results). Similarly, Jepson and Heeger [43] and Maybank [54] rigorously analyze this ambiguity and the shape of the least-squares error surface in the limit case of zero field of view, and showing that their results have practical importance requires experiments.

Less familiar is the "principle," which I argue for in Section 3, that the optimal least-squares reconstruction has a complex, nonlinear dependence on the data when the image sequence has large baselines. Though this "principle" is vague, it has the important implication that simplified algorithms may have trouble capturing this dependence: for instance, it suggests that trilinear algorithms [28, 30, 82, 90] might give far from optimal results, worse perhaps than a judicious fusing of optimal two-image reconstructions. I am not claiming that this implication must hold true, but only that the "principle" raises concerns that the authors of the trilinear approach should address experimentally.

This "principle" illustrates that the value of an intuition as a guide to experiments can greatly exceed the precision and depth of its mathematical foundation. The theoretical kernels of the "principle" are the statements that a least-squares reconstruction is the solution to a complex system of polynomial equations and that such systems generically have very complicated dependence on the data. For comparing trilinear and least-squares reconstructions, one should add a rigorous result stating that the variety of trilinear reconstructions differs from, and is in some sense less complex than, that of least-squares reconstructions.[2] However, the detailed form and proofs of these mathematical statements don't matter, since their implications are too imprecise to be useful. What is important is the suggestion that a trilinear reconstruction differs generically from the optimal one, because it suggests that one should investigate experimentally how much these differ in practice.

More recent phenomenological analyses show the existence of a significant new local minimum of the least-squares error surface [13, 64, 83, 84, 86] and demonstrate that many shallow local minima occur for near-forward translations. The theoretical parts of these analyses were important for understanding when the local minima occur. A different type of phenomenology is the analysis of image statistics. For instance, there has been much interest in developing a phenomenological characterization of texture. The body of this paper proposes other principles that could be used as the basis for phenomenological investigations.

One should not be suspicious of phenomenology because it incorporates experiments and is not fully rigorous. Phenomenology is common throughout science; in fact, most scientific theory consists of it. Even in the "exact" sciences such as physics, there is very little that one can predict rigorously. The basic equations and phenomena are too complex to be fully understood—the theories largely consist of intuitions about the implications of the equations that are independent of the equations themselves. Experiments and theory combine to makes these intuitions convincing. Theoretical support for an intuition might come, for example, from a rigorous analysis of simplified equations (a toy model). Some statements from

[2] Another, nonrigorous ingredient of this "principle" is the observation that there is no obvious linearization to apply for large motions. Thus, the natural expectation is that SFM has a fully nonlinear and generic behavior for such motions.

scientific theory do little more than codify descriptions of observed phenomena in abstract language. The principle of universality, which is basic to condensed matter physics, has not been proven except in a few simple cases; essentially, this principle encapsulates an observed fact about most physical systems.

I believe that SFM experimentation has suffered from its lack of a phenomenology. Because the field has not produced one, I offer may theoretical analyses as an example of what one looks like. I hope that others will flesh these out via experiments or else produce an alternative theory.

Algorithm design. I also intend this paper to illustrate that a phenomenological analysis is useful for designing algorithms. I do *not* intend to claim that current algorithms must work badly, or that the approach I propose is the best. I do wish to show how an understanding of the intrinsic properties of SFM estimation suggests algorithmic strategies and to motivate others to think about exploiting a phenomenological analysis to improve their own algorithms. Too many algorithms in the literature were derived by a formal manipulation of equations, with no thought for how the resulting algorithms were likely to behave. Instead, I suggest designing algorithms with the goal of coming as close as possible to a proof that they perform well over some domain. One should understand the performance of an algorithm to have confidence in it.

The clearest example of how theoretical understanding produces better algorithms is the theory of local minima. Local minima and flat regions of the error surface are the two crucial problems that confront any SFM reconstruction algorithm. If one can determine theoretically where the local minima occur, then one can adapt algorithms to avoid them. In [64, 65] I give examples of such theoretical analyses and algorithms.

A better understanding of how the optimal least-squares reconstruction depends on the number of images in a sequence would help in designing multi-image fusing algorithms (Section 2.2). It would determine how many images an algorithm should use to produce each of the intermediate reconstructions that it later combines into a final result—and, especially, how this number should depend on the conditions of the sequence and the number of images already fused.

Last, understanding the situations in which an algorithm can get away with an approximation or simplifying assumption can lead to simpler, more effective, and more robust algorithms for these situations. Some familiar examples of useful approximations include the strategies of approximating the 3D scene as planar [43, 51, 77, 78] or compact [102]. Abandoning all calibration information as in the projective approach[3] often gives a useful simplification.

Reconstruction and correspondence. This paper focuses on the reconstruction problem of SFM, that is, the problem of recovering the 3D scene and motion given known correspondences. This is not because I believe that the correspondence problem is unimportant or that the best approach is to first compute correspondences and then reconstruct. For many SFM applications, correspondence is the critical problem, and the best approach may be a direct algorithm [22–26, 42] that reconstructs from the data itself, for instance the normal flow, rather than from computed correspondences. But reconstruction is a significant subproblem in its own right, as the vast amount of research on it emphasizes (recent examples include

[3] This is equivalent to loosening the orthogonality requirement on the rotations in the Euclidean approach. See Sections 2.3.5 and 2.3.7.

[59, 104]). Reconstructing from known correspondences will clearly remain a crucial problem for some applications, and it is also likely that understanding this subproblem will help in understanding and designing direct algorithms. (For example, in [67] I describe a reconstruction algorithm that extends to a direct one [61].) Finally, one can clearly achieve a deeper theoretical understanding of the reconstruction subproblem than of the full problem including correspondence. This subproblem more clearly highlights the importance of phenomenological analysis, which is one of my main goals.

I believe that phenomenological analysis is generally useful for vision, and I intend my discussion of reconstruction partly as an illustration of this. For example, correspondence algorithms would benefit from a better understanding of how intensities vary between different images of the same scene. Work on object recognition would gain from more study of the only working recognition systems that exist (us). Work on shading *has* benefited from the development of theoretical models of surface reflectance and the classification of observed reflectances using these models [17]. I do not claim that phenomenology is a cure-all. Some problems are too complex to be studied phenomenologically, and the best one may be able to do for these problems is use a learning algorithm to capture some of their aspects automatically. One cannot know this without first attempting an analysis.

Mathematics and phenomenology. This paper contains few equations. This is partly for pedagogical reasons, but I also intend to emphasize that mathematics forms just a part of a useful SFM phenomenology. Although it would be possible to convert the theoretical principles proposed in this paper into rigorous mathematical statements, this would rarely be helpful. For example, the rigorous consequences of the "principle" discussed earlier are so specialized as to be useless. In other cases, as with the bas-relief ambiguity, making a principle rigorous would just amount to specifying enough conditions on the motion sequence so that one could get a proof. But the precise conditions one imposes for a proof are often unimportant and somewhat arbitrary; what matters is how widely the principle applies in practice. If one can come up with the basic intuitions behind a principle and verify their usefulness by experiment, there is little point in formalizing them into a proof for some restrictive and unrealistic limit if the proof is trivial and gives no new insight. Theory supported by experiments, but not strict proof, is usually key.

Also, the mathematics of SFM phenomenology is not sophisticated. By avoiding mathematical notation, I avoid suggesting that the mathematics is deeper than it really is, I highlight the intuitions that underlie the phenomenology rather than hiding them behind the equations, and I emphasize that a phenomenology based on experiment as well as theory is no less valuable or precise simply because it is not wholly mathematical. The SFM field appears to have a bias against theoretical analyses that are not superficially mathematical (and vice versa), and I wish to counter this.

Last, SFM researchers do sometimes confuse mathematics with phenomenology, and I avoid mathematics in this paper to emphasize the separation of the two. For instance, Faugeras and Mourrain [20] and Triggs [107] show that there are no new projective invariants depending on more than four images. Researchers sometimes assume that this result implies that there exist no invariant-type algorithms exploiting more than four images. This is false: one can always derive polynomial constraints on the image data for arbitrary numbers of images and solve for the coefficients of the polynomial constraints using all image data.

Engineering, science, and mathematics. I advocate a scientific approach to SFM in this paper, i.e., the adoption of a theoretical framework for conducting experiments and the basing of algorithms on an understanding of the intrinsic properties of SFM optimal estimation. I am *not* claiming that such an approach is the best or only one. It is possible that even a crude engineering approach, in which one invents algorithms by shuffling, combining, and importing mathematical tools, rather than studying the basic phenomena, and tests algorithms by examining how they work in a few cases of interest, could be good enough. If researchers had invented the perfect algorithm, or at least one with performance clearly good enough for practical applications, there would be no need for careful, theory-based experimentation. Or if SFM were too complex to repay scientific study, then kludging algorithm components together engineering-style might be the only possible approach.

However, so far only the Tomasi and Kanade algorithm [102] has emerged as an obviously superior algorithm, for a restricted domain. It is time to start evaluating claims for algorithms with more care, based on responsible, theoretically founded experiments, and to think harder about SFM itself in designing algorithms.

On the other pole is mathematical, especially algebraic, analysis. Mathematical research in SFM can be justified either because it leads to good new algorithms and practical insights or, sometimes, for its intrinsic interest as mathematics. It is not appreciated enough that scientific understanding has intrinsic value, just as mathematical insight does. For example, researchers sometimes discount the value of a new algorithm because a (partly) effective algorithm already exists for the same purpose. However, if there is little understanding of why or how the old algorithm works (and no guarantee that it will work well in all new situations!), and if the new approach can be derived from fundamental principles and embodies a deeper understanding than the old, then the new approach has value even if it makes no practical contribution—and of course it could have practical impact in the future.

Unfortunately, the engineering, scientific, and mathematical cultures within vision are too often intolerant or uncomprehending of the aims of the others. It is difficult to guess in advance which of these approaches will result in the most progress; researchers should pursue all three.

1.1. Outline of the Paper

In the rest of this paper, I review and discuss the main current SFM approaches according to the criteria of speed, accuracy, robustness, and *reliability*. An accurate algorithm typically reconstructs with small errors; a robust algorithm tends to avoid gross mistakes, and a reliable or trustworthy algorithm signals whenever it is liable to make a large mistake. Since no algorithm is perfectly robust, reliability is an important desideratum. Many current algorithms are not reliable.

Section 2 of the paper discusses optimization, Kalman filtering and fusing, projective methods, and invariant-based algorithms according to these criteria, pointing out unanswered experimental questions. Section 3 proposes new strategies for algorithm design, Section 4 briefly discusses the experimental evaluation of SFM algorithms, and Section 5 concludes. Section 6 cites some recent experimental results that bear on my arguments in this paper.

I focus on multi-image SFM algorithms (MISFM) [80], since two-image algorithms are relatively well understood, and since getting a robust reconstruction does sometimes require more than two images [67].

2. REVIEW OF MISFM ALGORITHMS

2.1. Optimization

An optimization approach defines the "optimal" reconstruction as that minimizing an error function such as the least-squares image error.[4] Algorithms search for the optimal reconstruction by minimizing this error using standard general techniques such as Levenberg–Marquardt (LM). (Note that I use the term "optimization" to mean not just the definition of the computational goal but also the algorithmic approach of relying on a generalized minimization technique.) By definition, an optimization approach gives optimal accuracy if it actually finds the correct global-minimum reconstruction.

Because traditional optimization approaches rely on generalized, brute-force minimization and take no account of the specific properties of the SFM error surface, they are essentially attempts to solve SFM without phenomenology.

Traditional optimization is not reliable. Experimentally, general minimization techniques do sometimes fail to find the global minimum, and when they converge to a local minimum they can produce a grossly wrong reconstruction. The minimization is guaranteed to find the correct minimum only if it starts from a sufficiently good initial reconstruction, and it is difficult to determine in advance how good this starting point needs to be. Also, there is often no way to tell after the fact whether an algorithm has located the correct reconstruction rather than a false minimum.[5] Thus one cannot tell when optimization will work. It is an incomplete approach: to make it robust and reliable, one needs some other method for generating reliable initial reconstructions, or a strategy for dealing with local minima.

In addition, one can compute the exact optimal reconstruction only by minimizing explicitly over all the structure/motion unknowns, and the large number of these unknowns makes optimization *slow*.[6] This is still true even if each iteration of the minimization routine has a computational cost linear in the number of 3D points (as in [33, 37]), since the computation per point for each iteration is significant and convergence often requires a large number of iterations, particularly starting from a bad initial estimate. The best way to alleviate the slowness is to start the minimization from a reconstruction that is already close to correct. Thus, also for speed, one should supplement optimization by a good, fast method for initial reconstruction.

Optimization does become more robust when it is based on more data (i.e., dense correspondences or many images), but since it also becomes slower, and since much faster, nonoptimizing approaches become increasingly accurate and robust, optimization may be unnecessary for many data-rich applications.

Thus optimization is at best half of the solution to SFM. For reliability and robustness as well as speed, it needs to be supplemented by another approach.

Most of the points I make in this section are well known and experimentally demonstrated. However, future experiments will determine whether optimization can be made fast enough for most applications despite its relative slowness. Also, recent work [13, 46, 64] shows

[4] The term "optimal" is not useful until it is given a precise definition in terms of some preference function. For convenience, in this paper I define optimality in terms of the standard least-squares error.

[5] That is, without a global search. In [64], I give an experimental example where one cannot distinguish the true and false minimums using local information, such as the error values at the minimum.

[6] For two-image sequences, one *can* compute a near-optimal reconstruction by minimizing in fewer unknowns [63, 64, 67, 111], but Section 2.4 argues that this is only possible for two images.

that one can make optimization more reliable by combining it with a phenomenological understanding of the SFM error surface. More study of this surface, and of how to adapt optimization to the specifics of SFM, is important.

2.2. Fusing/Kalman Filtering

Fusing, including Kalman filtering, is the second traditional approach to MISFM after optimization [3, 5, 7, 8, 56, 59, 60, 73, 85, 86, 88, 90, 100, 101, 110]. By fusing, I mean any MISFM technique that computes a final multi-image reconstruction from intermediate reconstructions (and estimates of the uncertainties in these) rather than from the image data directly. One type of fusing approach is recursive: it defines an intermediate reconstruction based on a subset of the images and gradually improves this into a full multi-image reconstruction using the information from one after another of the unused images. Another type combines intermediate reconstructions, each computed from a small number of images, into the final reconstruction (e.g., the batch/recursive approach of [59, 60]). I mainly focus on the second type.

I begin by assuming that the motion is completely *unknown and unconstrained* and only consider approaches that fuse estimates of the structure (more precisely, the shape [73, 102]).[7] The structure is all that can be fused for unconstrained motion, since it is the only thing that remains fixed from image to image. I consider this problem since some of the general-purpose multi-image algorithms in the literature (e.g., [59, 60, 81]) are really fusing approaches that do not make assumptions about the motion. I discuss only the issues of fusing's accuracy, robustness, and speed. I do not consider occlusion, correspondence, or real-time SFM, though these are important motivations for the fusing approach.

I suggest that fusing is essentially no more robust than the few-image intermediate reconstructions it is based on: if the few-image reconstructions have large errors, fusing them can give even worse results. In fact, this is well known to be true for a general complex estimation problem,[8] and there is no reason to believe that the situation differs in SFM. Though the results of [3, 5, 7, 8, 56, 59, 60, 73, 85, 86, 88, 90, 100, 101, 110] show that SFM fusing can improve intermediate reconstructions that are already reasonably accurate, this intrinsic and general problem with fusing suggests that it may fail when they are not. It is important to understand experimentally the usefulness and limitations of SFM fusing in practical applications.

Fundamentally, fusing has the same goal and the same local-minimum problem as optimization. To obtain a good fused estimate from subestimates, one must know how to weight the subestimates and their uncertainties so that the final estimate accurately reflects this information. In the worst case, for a general fusing problem, this requires knowing the *full likelihoods* for the subestimates and computing the *full fused likelihood* from them (see footnote 8). Any technique that aims at an optimal estimate must in effect compute this likelihood, which can be arbitrarily complicated. Traditional fusing, e.g., Kalman filtering, simply gives a short cut, a computational method, for computing the likelihood in the special situation when all the subestimate likelihoods are Gaussian. It works because combining Gaussian likelihoods is easy.

[7] I assume that the motion is not even known to be continuous, since this is not true for the multi-image algorithms such as [81].

[8] In the language of filtering theory, making fusing robust in general requires using an infinite-dimensional state representation (see, e.g., [4, p. 375] and [11, 12]), which no real algorithm can achieve.

The reason that fusing is generally useful is that likelihood functions often have good Gaussian approximations. Typically this happens when there are enough data so that the subestimates have small uncertainties. If large errors are unlikely, the measurement functions can be linearized around the true values of the unknowns, which makes it easier to approximate the subestimate likelihoods. More importantly, one only needs to model the likelihoods for small errors in the unknowns, and for this a Gaussian approximation reflecting the variances of the subestimates is often adequate. This is just the law of large numbers: overconstraining data implies a good Gaussian approximation.

Similarly, the law of large numbers implies that the complete fused estimate and the Gaussian approximation to the fused likelihood grow increasingly accurate as more data accumulate and increasingly constrain the estimate. This makes new subestimates easier to fuse. As the overall uncertainty decreases, one only needs to model the likelihood function of a new subestimate over a corresponding, decreasing range of error, which in effect improves its Gaussian approximation so that its information can be fused more accurately. Thus, assuming the errors are small to begin with, fusing will produce a fused estimate that comes increasingly close to an optimal estimate as it incorporates more data.

When all subestimates have large errors, linearization is out of the question. The subestimates can stray from their true values over a wide range, and it is inadequate to approximate the subestimate likelihoods as Gaussian over this whole range—in SFM they have significant non-Gaussian biases at least. In fact, the SFM situation for unconstrained motion is much worse than this. Since one is fusing subestimates of the structure, computing these subestimates and their likelihoods requires integrating out the unfusable motion parameters. This integral is impossible to do exactly, and when there are large errors and so no good guess for the true structure, one can't even approximate the integral by expanding it around the guess. In this situation, even an approximate computation of the subestimate likelihoods is impossible. But one does need complete models of the likelihoods for fusing when the errors are large. For example, the tallest peak of the fused likelihood (the maximum-likelihood estimate or MLE) can emerge from a complicated interaction of nonleading parts of the subestimate likelihoods. Similarly, in recursive fusing, the information from new images can interact with the nonleading parts of the previously computed likelihood.

Since it is impractical to compute and represent the likelihoods for subestimates with large errors, in a general situation one cannot fuse them. If this could be done, it would amount to a guaranteed solution method for optimization. The corollary is if one does combine nonrobust subestimates by some arbitrary rule, the combined estimate will not be robust.

On the other hand, fusing will succeed when starting from a good estimate whose uncertainty is *known* to be small, even if the remaining subestimates have large errors. In this case, it doesn't matter if the subestimate likelihoods are difficult to compute since, as noted above, one only needs to approximate them over the small range corresponding to the expected error. Again, the fused estimate and all Gaussian approximations will improve as new images accumulate. Thus, if a fusing method can be locked onto the correct estimate initially, it will exploit additional data effectively to improve the estimate. Fusing may also be able to work even if the initial estimate is just reasonably good. For example, given a moderately good initial estimate, an enhanced version of the simplest linear fuser such as the (I)EKF (the (iterated) extended Kalman filter, e.g., [3, 4, 8, 56, 110]) might be able to lock onto the correct estimate after accumulating some number of images. Experiments must determine how much such extensions of linear fusing actually help.

Experimental results confirm that fusing accurate two-image reconstructions leads to increased accuracy [73, 100, 101] (see also [5, 8, 27, 60, 85, 86]). When some of the two-image reconstructions are very inaccurate, the fused reconstruction can grow less accurate as more images are acquired [73, 100, 101].[9]

The above arguments imply the following strong claim. Consider a sequence with very many images such that the signal summed over all images is large compared to the noise, but the signal-to-noise ratio is vanishingly small for any small collection of images. Since the fusing approach only exploits a few images at a time, I claim that it will fail on such sequences—that in general it will not converge to the correct reconstruction. If so, it becomes an important experimental question to determine where the fusing approach breaks down in practice. How large must the signal-to-noise ratio be for robust fusing? How small a camera motion can fusing tolerate?

Despite fusing's potential limitations, Section 3.1.1 argues that it may be the best approach for some situations, e.g., for moderate-to-large motions or many correspondences. This suggests that it is important to develop and test fusing approaches tuned for these situations. Also, Sections 2.4 and 3.1.1 propose that fusing two-image reconstructions could work better than fusing intermediate reconstructions based on three or more. Thomas and I [73, 100] already suggested that image-pair fusing has advantages over fusing in one image at a time [5, 15, 85]. Section 2.4 discusses this further. Experiments are important to determine the best number of images to add in at each fusing step.

(This last question matters mostly in the initial stages of fusing. Once the fused estimate incorporates enough images so that it's uncertainly is small, this by itself gives enough constraint for good Gaussian modeling of the likelihoods, or linearization as in the EKF, and one might as well adopt the simpler strategy of fusing a single image at a time. See the discussion above and Section 2.4. Also, single-image fusing becomes more effective when the motion is constrained, as it is in [15, 56, 85, 86].

Finally, it is important to realize that the bas-relief ambiguity often causes few-image algorithms to succeed partially, so that they recover most but not all structure components accurately [1, 9, 18, 50, 67, 71, 97]. In such cases, the discussion above suggests that a fusing approach might be able to enhance accuracy for the part of the structure that the few-image methods recover accurately, even if it cannot recover the whole structure. Such partial-fusing algorithms should be explored in more detail.[10]

I focused till now on unconstrained motion. The situation improves when the motion is at least partly known [3, 5, 7, 8, 56, 60, 85, 86, 110]. It is no longer necessary to integrate out the motion parameters, and the nonlinearities become much more manageable. As discussed in [86], fusing with motion constraints can work even when the assumed model for the motion is stochastic and weak, e.g., when one imposes a small degree of smoothness on the motion by modeling it as a first-order random walk. The strong claim above should still hold true [12], but in practice fusing will be able to cope with lower signal-to-noise.[11]

[9] The reliability of other algorithms such as [2] should be explored in more detail. The algorithm in [2] is in the spirit of the suggestions made in Section 3.1.3—it trades accuracy for a potential increase in robustness—but it too should be fragile in difficult cases with small camera translations.

[10] When the effects of the bas-relief ambiguity are not too severe, a modified fusing approach might be capable of recovering the *whole* structure robustly, for instance by incorporating an optimization over the parameter most affected by the ambiguity.

[11] There do exist recusive methods that can cope with arbitrarily low signal-to-noise. For example, assuming constant translation direction and small rotations, a recursive implementation of the subspace method [44, 65, 86]

A fusing approach based on a particular motion model does require that the true motion partly satisfies that model. How well the algorithm performs clearly must depend on how well its model captures the true motion. Moreover, if for unconstrained motion fusing tends to fail catastrophically at low signal-to-noise, this suggests that the performance of a constrained-motion approach might depend sensitively on the validity of its motion model at small signal-to-noise. Thus, it is important to study experimentally how constrained-motion approaches perform when their motion models break down. It is also important to understand experimentally what motion models are useful in practice.

Many fusing algorithms based on motion models use a Kalman filtering technology that was developed for a different type of problem than SFM—one where the observations directly measure the state, and the focus is on the continuum, in the sense that one wants the algorithm to compute the "derivative" of the estimate with respect to the new data acquired at each time step. For SFM, instead of adopting a previous technology that may not be a perfect fit, I suggest that one may do better to design a fusing algorithm according to the special features of the problem, that is, based on phenomenology. For example, as I discuss in Section 2.4, one could drop the "continuous" form of the classic Kalman approaches and instead fuse image-pair reconstructions, or else adopt the approach of Footnote 11.

2.3. Projective Reconstruction

The projective approach is a framework for SFM rather than an algorithm [21, 37]. Developed mainly because of the difficulty of calibrating cameras precisely, its aim is to carry out SFM without calibration. Computing a projective reconstruction is essentially equivalent to computing a standard Euclidean one except that the linear camera calibration is treated as unknown and *potentially arbitrarily different in each image*.[12] Within this framework, researchers have used various algorithms, including optimization and the invariants approach discussed in Section 2.4.

However, the projective approach's basic assumption rarely holds—usually something *is* known about the linear calibration. For instance, the x and y image axes are almost always close to perpendicular. Often one knows that a motion sequence was taken with a single camera and that all images have approximately the same calibration (except possibly for the focal length and parameters that covary with it). The image center, the relative scaling of the x and y axes, or the focal length is often known approximately. This approximate information can be useful, especially since errors in the image center or focal length are known to have little effect on the recovered structure [10, 67, 113].

Using the projective approach when its assumptions don't apply amounts to an algorithmic choice. Simply because the calibration is known incompletely or approximately

can reconstruct accurately given enough images, no matter what the signal-to-noise level is for each image. However, this is not a fusing approach according to my definition, since it does not rely on intermediate reconstructions of the translation direction or structure. Because this approach deals easily with occlusions, it may have advantages over traditional Kalman filtering approaches.

[12] In the projective framework, the image point position is determined by $I = MS$, where I is a homogeneous 3-vector representing the image, M is the 3×4 camera matrix, and S is a homogeneous 4-vector representing the structure. The projective framework is equivalent to the Euclidean with unknown calibration (apart from singular cases) in the sense that one can interpret S as a standard Euclidean structure and M as the result of an unknown camera motion and unknown linear calibration. (A nonsingular M has a unique decomposition into motion and calibration matrices.)

is not by itself a reason to jettison all calibration information; an algorithm must use all available information, including approximate information, to get the best reconstructions. Presumably, the projective approach trades away information—and thus accuracy—in the hopes of obtaining an algorithm with increased simplicity or robustness, but there has been little experimental study of the trade-offs involved. Given the extraordinary amount of research on the projective approach, it is time to experimentally investigate the costs it incurs by neglecting information, and to determine what it gains in simplicity and robustness. Also, since worry about the effects of calibration error was the main motivation for the projective approach, it is time for a thorough study of how calibration error does affect reconstruction.

Often the projective approach is used just as the first step toward a full Euclidean result. With this approach, the final Euclidean reconstruction inherits the errors of the initial projective one, unless one eventually abandons the projective reconstruction and recomputes everything, including the projective structure, by purely Euclidean methods. This is seldom done. Since a Euclidean computation is needed eventually to get the best reconstruction, clearly one should start with this approach if an effective algorithm exists.

The next few sections discuss the potential problems of the projective approach in more detail.

2.3.1. Incorrect Modeling

In neglecting calibration information, the projective approach models image formation incorrectly and solves for (approximately) known parameters as if they were unknowns. It is given the freedom to reconstruct wrong values for these artificial unknowns, which in turn can corrupt the recovery of the true unknowns. Using a nonoptimal algorithm can significantly increase this corruption: if an algorithm already tends to overrespond to noise and error, implementing it in the projective framework and allowing it to modify extra unknowns give it new incorrect ways to overrespond. The corruption can also worsen for difficult sequences, when the image data only determine the reconstruction weakly, again since the projective approach has more ways to go wrong in response to noise. Thus, because of its extra artificial unknowns, projective reconstruction can become *nonrobust*. How nonrobust depends on the algorithm and on the sequence. Clearly the projective approach does give good results in some cases [5], but in [64] I described a type of situation where the optimal projective estimate is much more fragile than the optimal Euclidean one.

The problem is most serious for sequences taken with a single camera of fixed calibration, since the number of artificial unknowns is then largest. The projective approach treats the calibration parameters in each image as independent, assuming $5N_I$ calibration unknowns, where N_I is the number of images. But for a single camera the calibration must be (nearly) the same for all images[13]; the true number of calibration unknowns is just 5. Experimental results on synthetic single-camera sequences [72] confirm that a standard Euclidean approach simultaneously computing the reconstruction and calibration gives more accurate results than a projective approach, even for the projective structure.

It is impossible to adapt the projective approach to single-camera sequences without destroying the projective symmetry.

2.3.2. Nonlinear Camera Distortion

Nonlinear camera distortions cause further difficulties for the projective approach. By allowing arbitrary linear calibrations, the projective approach in effect assumes that the linear

calibration errors are likely to be large. It also makes the incongruously strong assumption that there is *zero* nonlinear camera distortion, even though in practice the nonlinear effects can be important. Thus, when nonlinear distortions do occur, they tend to cause larger errors in the projective approach than in the standard one. A projective algorithm can wrongly explain a part of the image distortion by jointly altering the reconstruction and linear calibration; a standard Euclidean algorithm has no freedom to change the calibration, so it interprets this part of the distortion just as image noise and ignores it in reconstructing. As before, a nonoptimal projective algorithm has more ways to overrespond wrongly to the nonlinear distortion and can do even worse compared to its Euclidean counterpart.

Again, there is no way to modify the projective approach, while preserving its essential character, to account for nonlinear as well as linear calibration errors. It is impossible to calibrate away the nonlinear effects without simultaneously doing linear calibration and removing most of the rationale for the projective approach.

Some researchers argue for applying the projective approach even for known calibration. Of course, with this strategy the camera's nonlinear distortion is precorrected and causes no difficulties. I discuss this approach further below.

2.3.3. Self-Calibration

One of the advantages claimed for the projective approach is that it allows self-calibration, that is, one can calibrate the camera without a known 3D calibration object. But it is equally possible to self-calibrate in the Euclidean framework, using optimization, for example, and this gives greater accuracy, since the Euclidean framework exploits the known facts about the calibration from the start (such as the fact that it is fixed[13] when there is a single camera).

The fact that self-calibration allows one to calibrate the camera and simultaneously recover the structure is often claimed as a virtue. But separating these two tasks makes more sense; the situations that are best for self-calibrating are quite different from those that facilitate structure recovery. Self-calibrating is easiest when the camera only rotates and does not translate [34], since this completely factors out the unknwon structure.[14] But structure recovery works best when the translations are large, since the translational image displacements contain the structure information. Rather than simultaneously calibrating and reconstructing, it may often be better to calibrate first using the motion appropriate to this task and recover the structure afterward based on the calibration. (Of course, one does not always have the option of choosing to do this.)

2.3.4. Projective Structure

Researchers sometimes justify applying the projective approach to single-camera sequences (even with known calibration) on the grounds that the projective structure is more robust than the full Euclidean structure—to errors in the motion recovery as well as the calibration. But it is important to distinguish a computation's goal from the assumptions used to achieve it: to best reconstruct the projective structure, an algorithm still needs to use all available information.

[13] Again, with the possible exception of the focal length (changes in which are often relatively easy to handle by Euclidean methods [2, 67]) and covarying parameters.

[14] Unless the FOV is very small, in which case the camera must translate to achieve rotations that are large enough for good accuracy and that also keep the scene in view. Even then, it makes sense to calibrate the relative scaling and skew of the x,y axes using pure rotations.

All the preceding arguments apply. For known calibration, a Euclidean method will compute the projective structure more accurately than a projective one. With a rough calibration, a Euclidean technique can still be better. Govindu and I [72] present examples where the projective structure recovered by Euclidean optimization neglecting miscalibration is as accurate as the projective optimization result even with significant errors in the assumed calibration. This means that *nonoptimal* projective algorithms would probably have done *worse* than the corresponding Euclidean ones. Finally, for a completely uncalibrated single-camera sequence, Euclidean self-calibration by optimization will recover the projective structure more accurately than projective self-calibration.

Also, at least when the calibration is known (and perhaps also when it is not, see the discussion at the end of Section 2.3.6), the projective structure is not necessarily more robust than the full Euclidean structure. Though the Euclidean structure is often noise-sensitive, this sensitivity typically affects just a few components (due to the bas-relief ambiguity), so one can recover most of the structure robustly. I have shown this experimentally as well as theoretically in [67, 71]. Moreover, one can determine from the data which part of the recovered structure is robust,[15] by estimating the error in each structure component [67, 71]. The Euclidean structure plus the estimates of its errors give more information than the projective structure alone, with no loss in robustness. Often it gives much more, since one can usually recover more of the Euclidean structure robustly than the part[16] that corresponds to the projective structure. Thus, it makes sense to let the data determine how much of the recovered Euclidean structure to trust, instead of settling a priori for the projective structure.

2.3.5. Formal Equivalence of Projective and Euclidean Reconstruction

The projective approach models the formation of an image of N 3D points by $I = MS$, where I contains the image points in homogeneous coordinates, M is a 3×4 camera matrix, and S a $4 \times N$ structure matrix. The columns of the structure matrix S give the homogeneous coordinates of the 3D points, and the camera matrix M summarizes the camera rotation, translation, and linear calibration parameters for the given image.

In standard Euclidean SFM, one can model image formation in exactly the same way; only the interpretation of the camera matrix M differs. In this calibrated case, the first three columns of the M give the rotation matrix, and the last column holds the translation. Thus the projective approach is *formally equivalent* to standard Euclidean SFM if one relaxes the constraints on the rotation matrix are relaxed—i.e., if one allows the rotation matrix to be arbitrary rather than restricting it to being orthogonal. Neglecting the rotation matrix constraints is standard in many Euclidean algorithms, for instance, in the "8-point" computation of the essential matrix [53], or Tomasi and Kanade's SVD approach to orthographic SFM [102].

The projective approach therefore does not lead to new SFM algorithms; all projective algorithms translate directly into Euclidean ones. For example, the 8-point algorithm determining the fundamental matrix translates into the familiar 8-point algorithm for the essential matrix. Similarly, plane-plus-parallax [43, 51, 77–79] is the projective equivalent

[15] By "part" of the structure, I mean the projection of the structure onto some subspace of the $3N_p$-dimensional subspace spanned by the coordinates of all N_p points. It is usually better to treat the depths separately from the other structure components and to project in the inverse depths rather than the depths themselves.

[16] Often, for example, the inverse depths can be recovered accurately up to an additive constant (due to the bas-relief ambiguity), which is a one-parameter ambiguity. The projective reconstruction has a larger, four-parameter ambiguity in recovering the inverse depths, since it excludes all components linear in the image coordinates.

of the more familiar rotation-plus-parallax [62, 67, 69]. In the Euclidean framework, one does need to restore the rotation matrix constraints eventually—to force the recovered rotation matrices to be orthogonal. This is straightforward and techniques already exist, e.g., Hartley's method for recovering the rotation matrix from an inexact essential matrix [37], or Tomasi and Kanade's technique [102].

In my view, the main innovation of the projective framework was not that it made possible new algorithms[17] but that it addressed for the first time the problem of reconstructing with uncalibrated, varying cameras. If projective researchers want to attack the problem of reconstructing with partly known or partly fixed calibration, I argue that they should at least consider the alternative algorithmic possibilities to the projective approach, due to the obvious potential problems with the approach that I described above. The next section serves as a reminder that alternative approaches do exist and presents some simple examples for standard applications.

2.3.6. Alternatives to Projective Reconstruction

There are certainly reasonable Euclidean alternatives to the projective approach for single-camera sequences—even, as I have already implied, when the calibration is completely unknown. It is true that, for unknown calibration, any attempt to exploit the single-camera constraint formally complicates the reconstruction problem: the equations lose the simple bilinearity of the projective approach. This causes no difficulties if one is resorting to optimization, as in several of the more practical implementations of the projective approach. Optimization is a flexible approach that can easily incorporate constraints or prior biases; it is equally easy to implement Euclidean and projective self-calibration within this approach.

As already noted, Euclidean self-calibration by optimization gives more accuracy than projective optimization (assuming both approaches converge correctly). Because it involves fewer unknowns, it may also be faster, as Govindu and I have verified experimentally for a few simple examples [72] (but see [98]). In any case, since getting the most accurate Euclidean reconstruction requires doing a complete Euclidean optimization eventually, one can get a faster computation by doing so at the start rather than after an extra stage of projective optimization. At least on the grounds of accuracy or speed, there is little current justification for applying projective rather than Euclidean optimization to single-camera sequences. However, "projective" optimization[18] has less of a local-minimum problem than its Euclidean counterpart [72]. This is one of the most compelling motivations for the projective approach, and I discuss it further in Section 2.3.7.

If one chooses not to use optimization (or needs to compute a starting point for it), then adapting a Euclidean approach will still sometimes be the best way to deal with partial uncertainty about the calibration. Unfortunately, little work has been done on such algorithms. As a reminder that there is more than a single approach to dealing with calibration uncertainty, and that we do not yet know under what conditions the projective approach is the best one, I propose a few suggestions for Euclidean algorithms. My aim is to illustrate that such algorithms represent a valuable and neglected line of research, and that determining when the projective approach is appropriate is an important and unanswered experimental question.

[17] Historically, important new algorithms such as [96] did emerge from the projective approach.

[18] See Section 2.3.7 for an explanation of the quotes.

In many situations the camera has already been calibrated approximately, and one can use the following simple two-stage technique: (1) Initially reconstruct assuming the approximate calibration is correct.[19] This is the standard Euclidean problem, which, because of the formal equivalence of the projective and (relaxed) Euclidean approaches, one can deal with using algorithms that are as simple as the projective ones. (2) Perturbatively correct the calibration and reconstruction. This is similar to the task in the projective approach of extending a projective reconstruction to a Euclidean one. The overall algorithm need not be more complex than starting with a projective reconstruction and making it Euclidean.

The apparent problem with this is that errors in measuring the calibration are often large, and researchers have assumed that this will also cause large errors in a Euclidean reconstruction. If so, the second stage, being perturbative, would be unable to recover from the large errors of the first. But measured calibrations may be reasonably good *at least in terms of their effects on structure recovery*.

It seems almost tautological that errors in the difficult-to-recover calibration parameters (e.g., the camera center) are likely to have little effect on the structure, while the parameters that have strong effects are easy to calibrate accurately [10]. The difficult calibration parameters are difficult to recover even with a precisely measured 3D calibration object. This implies that altering these calibration parameters, and perhaps compensating by changing the motion, does not much affect the images of a fixed 3D structure. Conversely, if the image data robustly determine the structure, as one would expect when there are many images overconstraining the reconstruction, then substantially different structures produce substantially different image data, and errors in these calibration parameters should have little effect on the structure recovery. More generally, if for perfect calibration the image data determine most (see footnote 15) but not all of the structure robustly (due for example to the bas-relief ambiguity), then by the same reasoning, errors in the difficult calibration parameters should have little effect on the robust part of the structure. One could recover this part accurately and use it as a starting point for generating a full reconstruction in the perturbative second stage. The recovery of the rest of the structure might be degraded by calibration errors, but this wouldn't matter since it could already have large errors from the noise. The experiments of [10, 67, 72] suggest that one can accurately recover at least part of the structure despite calibration error.

Because the staged approach exploits the approximate calibration throughout, including the knowledge that it remains fixed, it can be more robust than the projective approach. Past experience with other motion problems indicates that it is important to exploit our expectations for parameter values even at the cost of introducing approximations. The staged strategy of initially treating approximately known parameters as known and then correcting them perturbatively often works well. For instance, in recovering the projective transformation between the images of two planes, it is useful to first find the approximate affine transform and then extend it by solving iteratively for the full projective transform. Similarly, in registering two images taken by a moving camera, it is effective to register first based on an affine transform, then assuming a planar scene, and only at the end to allow general motion/structure [76]. In both cases, the advantage of initially solving for a smaller number of parameters outweighs the disadvantage of computing an approximate

[19] Reconstructing assuming that the calibration is correct can be an effective way of using the calibration measurements. If as argued below there is a rough correspondence between the accuracies of calibration parameters and their effects on structure recovery, then doing so automatically weights the expected errors in the parameters roughly correctly.

reconstruction. This staged approach could also be effective in coping with calibration inaccuracies in MISFM, and it deserves more study as an alternative to projective reconstruction.

The staged approach could be particularly useful as an alternative to projective methods that are nonoptimal. As argued in Section 2.3.1, the projective approach's incorrect modeling can significantly worsen the results of a nonoptimal algorithm; the effects of the bad modeling can combine with the algorithm's own bias to produce large errors. On the other hand, in the Euclidean approach, the main effect of a moderate error in the assumed calibration is to bias the reconstruction in a well-defined way. A theoretical error analysis can determine this bias; by propagating the effects of miscalibration through the reconstruction algorithm, one can estimate and characterize the resulting errors [67, 93]—and perhaps even correct them after the calibration has been determined accurately. For the projective approach, it is harder to characterize the errors induced by incorrect modeling. For instance, in [64] I describe a type of situation where using the projective approach flattens out the error surface so that noise sometimes makes the reconstruction error very large. Because the error depends sensitively on the noise, it cannot be estimated. Thus, good performance of the Euclidean approach can be easier to guarantee and its performance limits easier to understand: it can be more reliable.

One can extend Euclidean methods to deal with cases where the calibration is known only partially—e.g., where the focal length is unknown and varying [67].

I have suggested that the staged approach will give good results for typical calibration errors. The results of Svoboda and Sturm [95] and Vieville and Faugeras [108] suggest that large errors or changes in the calibration, might corrupt Euclidean reconstruction mainly in a few structure/motion components. For single-camera sequences, if it turns out that the Euclidean approach robustly computes all unknowns that don't depend on the calibration parameters, then it would have little disadvantage compared to a projective one even when the calibration was completely unknown.

2.3.7. Robustness of "Projective" Optimization

In recent experiments on calibrated single-camera sequences, Govindu and I [72] found that "projective" optimization had less of a local-minimum problem than a pure Euclidean approach—i.e., LM minimizations over the structure and unconstrained camera matrices tended to converge to the global minimum from a wider range of starting points than if the camera matrices were constrained according to the Euclidean fixed-calibration assumption. This may be partly because the extra unknowns in the projective approach destabilize some of the Euclidean local minima, as researchers have found in other contexts. (For example, using extra artificial modeling parameters in neural-network learning can avoid local minima [52].) It may also be because "projective" optimization imposes no constraints on the rotation matrices and thus avoids some of the nonlinearities of purely Euclidean optimization.[20] In [64] I recently gave a more concrete explanation, for a camera translating in a fixed direction.[21]

[20] The 8-point algorithm [53] is a good example. This "projective" algorithm relaxes the rotation-matrix constraints and thus in effect minimizes a quadratic error function with a single minimum; it has no local-minimum problem. Of course, avoiding local minima is not the whole story; the global minimum of the 8-point algorithm does not necessarily give a good approximation of the true reconstruction.

[21] This explanation might work in general. For a general multi-image sequence, the extra Euclidean local minima may originate from the two-image local minimas (for all possible image pairs) of the sort described in [64].

This result does *not* imply that the projective framework gives better results than the Euclidean one. As I emphasize here by the use of quotes, "projective" optimization is a computational technique that one can apply in either the Euclidean or projective frameworks. Recall that I defined the projective approach as Euclidean reconstruction with unknown, possibly different calibrations for each image. This is the definition implicit in most projective work; the standard way to make sense of a projective reconstruction by relating it to the Euclidean world is to compute the calibrations as well as the motions from the projective camera matrices M. From Section 2.3.5, the projective and Euclidean frameworks are formally equivalent under the approximation of neglecting the Euclidean rotation-matrix constraints. Alternatively then, one can think of "projective" optimization as the first stage of a Euclidean algorithm which initially uses the approximation of neglecting the rotation constraints. The second stage would compute the motions from the camera matrices M by finding the projective transform making the first three columns of the M as close as possible to orthogonal rotational matrices. Note how this differs from the standard projective technique.

Also, in [58] I identified a commonly occurring situation where "projective" optimization is much less robust than Euclidean optimization despite having fewer local minima. I gave a concrete example where the "projective" error surface has a grossly wrong global minimum, roughly in the position of a Euclidean local minimum, while the Euclidean global minimum is approximately correct.

For easy SFM sequences, where the data strongly determine the reconstruction (i.e., for large camera translations, large 3D depth range, large field of view, or many correspondences), either optimization technique should be robust—in fact, even the 8-point algorithm should give reasonable results, as discussed in Section 3.1.1. In these cases, the speed and accuracy of Euclidean optimization probably justify its direct application.

For difficult sequences (e.g., for small camera translations, small field of view, small range of depths, and few correspondences), local minima become a real danger, and it may sometimes be advisable to do a "projective" optimization first and then use the result to initialize a Euclidean optimization. But not always—for difficult sequences, noise can displace the projective global minimum far from its true location [64]; Alternatively, if phenomenological analysis can predict where the extra Euclidean local minima occur, one could avoid them by incorporating this prediction into an Euclidean approach (e.g., [64, 65]). Getting robustness on difficult problems also depends on finding a good starting point for optimization, which one has to compute with a nonoptimal algorithm. The projective approach becomes relatively more inaccurate on difficult sequences, and using a nonoptimal projective algorithm compounds the problem, since the poor modeling of the projective approach and the nonoptimality work together to make the errors larger (Sections 2.3.1 and 2.3.6). Thus, on difficult problems, it could be best to compute the initial reconstruction using a Euclidean algorithm; the bias of the Euclidean approach due to miscalibration may be less of a problem than the increased and unpredictable error of the projective alternative [64].

Lastly, the fact that relaxing the rotation constraints eliminates local minima does not generlize to other nonoptimization algorithms: algorithms that do not minimize cannot be trapped in local minima. For nonoptimal algorithms, in fact, the projective approach's relative inaccuracy on calibrated, single-camera sequences can translate into decreased robustness.

2.3.8. Experimental Implications

The arguments of this section raised many points that need experimental investigation:

- When is it worthwhile to forgo all knowledge of the calibration, as the projective approach does? When is it safe to resort to the projective approach's simplicity? When is it better to use a Euclidean approach and exploit whatever knowledge is available? To avoid inaccurate reconstruction, how much more data does the projective approach require than the Euclidean one?
- How do nonlinear distortions of the images affect projective versus Euclidean reconstructions?
- How much calibration error can a Euclidean approach tolerate? How do errors in the image center and focal length affect Euclidean depth recovery?
- How can one design an effective Euclidean approach for dealing with calibration error?
- For single-camera sequences, what is the cost in accuracy of computing the projective structure in a projective approach rather than a Euclidean one? How much do errors in the projective structure affect a Euclidean reconstruction computed from it?
- How much does the projective approach neglect of information hurt the accuracy of nonoptimal projective algorithms?
- What causes the extra local minima in the Euclidean least-squares error surface? Do all of these originate in the phenomenon analyzed in [64]? Can one design Euclidean optimization algorithms to avoid such local minima?
- What are the relative accuracies of self-calibrations obtained by rotating versus translating cameras? When the camera translates, how does the unknown 3D structure affect the accuracy of self-calibration?
- For self-calibration on single-camera sequences, what are the disadvantages to using Euclidean optimization directly as opposed to first computing a projective reconstruction? Does Euclidean optimization suffer from important new local minima when it solves for the calibration as well as the structure/motion?

2.3.9. Motivations

The purpose of the previous section was not to challenge the projective approach, which I believe is useful and sometimes necessary. Instead, I wanted to clear up misunderstandings about this approach, point out the lack of phenomenology in projective research, and illustrate the value of a more phenomenological approach.

For instance, more emphasis on phenomenology might have led researchers to study the benefits of using the direct and accurate Euclidean optimization approach to self-calibration, instead of an approach based on projective optimization. With a wider understanding of the equivalence of the projective and Euclidean frameworks (Section 2.3.5), duplicate publications of algorithms in their projective and Euclidean versions could have been avoided. The proposal of projective algorithms that were worse than the existing Euclidean versions of these algorithms would have been curtailed. Important new approaches that happened to be described in Euclidean language (e.g., [91]) might not have been overlooked. Lastly, we might have had more careful studies of the appropriateness of the projective approach for situations where the projective assumptions do not hold, more research on error sensitivity, and fewer algorithms designed without error analysis.

I wrote this section to address neglected theoretical issues. The results of [64], and the problems I recently observed with the Sturm/Triggs [96] approach, demonstrate that these issues have practical importance.

2.4. Invariant-Based Approach

The invariant approach is based on deriving polynomial constraints on the image data by explicit algebra. Usually, one algebraically eliminates either the structure or motion unknowns, yielding polynomial constraints with coefficients that are functions of the unknowns that weren't eliminated. For two-image SFM, for example, the image points satisfy the fundamental or essential matrix equation—a polynomial bilinear in the coordinates of the image points with coefficients depending on the motion alone. Similarly, the trilinear constraint coefficients for three images depend just on the motion [31, 80, 90].

Invariant-based algorithms typically begin by fitting the derived polynomial constraints to the data, producing estimates of the constraint coefficients. For example, a two-image algorithm would begin by estimating the essential matrix. These algorithms then recover some of the unknowns (usually the motion) in a second step by inverting the theoretically derived functional dependence of the coefficients on these unknowns. This corresponds to recovering the rotation and translation from the estimated essential matrix. Often, although not for the essential matrix, this second stage also involves explicit algebraic manipulations such as the solving of polynomial equations. Finally, a third stage computes the remaining unknowns (e.g., the structure).

Although invariant methods are often associated with projective reconstruction there is nothing intrinsically projective about them—one can implement these algorithms just as well in the standard Euclidean framework (e.g., the original trilinear formulation of [90] was Euclidean.

The problem with the invariant approach is that it relies on polynomial manipulations, which are typically extremely unstable [57]. In polynomial data fitting, the estimates of the polynomial coefficients can be strongly noise-sensitive and biased; this is true even for the simplest tasks, such as ellipse fitting, unless the data points cover most of the curve. In SFM, large errors in the constraint coefficients cause large errors in the recovered motion or structure. Moreover, there is no reason why the algebraic derivations of this approach should faithfully model the effects of image noise, as is necessary for an accurate and robust algorithm.

Many researchers have pointed out the noise sensitivity of invariant-based algorithms (e.g., [31, 54]). The simplest such algorithm—the 8-point algorithm for two images—is known to be inaccurate.[22,23] This is so despite the fact that the second, motion-recovery

[22] This is experimentally true for difficult problems where the camera translation is relatively small, as in traditional robot navigation. As argued and demonstrated experimentally, e.g., in [70], when the camera translation is large and the 3D scene extended, two-image SFM is easy, and many algorithms work well, even the 8-point algorithm [53].

[23] Hartley has pointed out that "balancing" in the projective framework makes the 8-point algorithm more accurate and reliable [32]. In this technique, one transforms the homogeneous image coordinates before the 8-point algorithm is applied so that all three components have roughly the same magnitude. In the Euclidean approach, the image coordinates are already relatively balanced. On the other hand, for small field of view, balancing does help to redress what would otherwise be a bias toward reconstructing the translation as parallel to the viewing direction [67]. Nevertheless, balancing does not cure completely the inaccuracy widely reported for the Euclidean

stage of this algorithm is nonalgebraic and nearly optimal [37]. For more than two images, the problem seems likely to worsen: the first-stage polynomial-fitting problem becomes harder and more complex, and a close-to-optimal second stage may no longer be achievable without some kind of iterative improvement and consequent slowdown.

In fact, Hartley states in [35] that an algorithm based on the trilinear constraints is "very unstable." This is an algorithm strictly following the invariant approach: it solves for the trilinear coefficients by fitting the data and then uses them to determine the motion by explicit algebra. Hatley [31, 35] obtains a more stable algorithm by partly abandoning the results of the initial polynomial fitting. The improved algorithm uses the trilinear coefficients to determine just a subset of the motion parameters. Then, with these motion parameters taken as known, it solves for the remaining motion parameters *directly* from the image data rather than from the trilinear coefficients, departing from a strict invariant approach. Though this new algorithm is more stable than the previous one, it apparently has been tested mostly on relatively easy sequences where the camera translations or the number of correspondences are large [70]. Such sequences don't give a stringent test of the trilinear algorithm's robustness, since the inaccurate 8-point algorithm can also work well on these (see footnote 22). On more difficult sequences with small translations, small depth range, and few correspondences [70], the fact that polynomial fitting still plays a role in the revised algorithm may well cause nonrobustness or inaccuracy (e.g., in comparison to two-image optimization). It is important to test this experimentally.

Note that the trilinear constraints operate very differently for points versus lines. For lines, Hartley's technique of using a subset of the trilinear coefficients to determine a subset of the motion parameters could be robust, since the expression for the latter in terms of the former comes close to a solution directly from the image data. This relates to the fact that the trilinear constraints are "minimal" for lines—in the same sense that the essential matrix gives the minimal constraint on the image data for the point case. This reasoning also suggests that the 8-point algorithm may be more accurate and robust than other point-based invariant methods.

Another problem with the invariant approach is that in practice one can apply it just to a small number of images or feature points. First, one must derive the polynomial image constraints for a specific number of images or points, which limits the method's flexibility. Second, this number must be small, because the constraints involving large numbers of images or points quickly become too complicated and high order to be useful.[24] Thus, the standard invariant approach does not yield multi-image algorithms.

Recently, some researchers (e.g., [39, 74, 81]) have computed multi-image reconstructions by combining invariant-based reconstructions, especially trilinear ones, using linear-algebra constraints. This is a fusing approach, though one where the technique for combining intermediate estimates is dictated by mathematical simplicity rather than by an attempt to approximate optimal fusing. As I discussed in Section 2.2, fusing is liable to fail unless it starts from accurate and robust intermediate reconstructions, and it will give poor accuracy unless it weights the intermediate estimates properly in combining them. As argued above, the invariant-based intermediate reconstructions in [81] are nonoptimal and potentially

8-point algorithm [67]. Part of the reason Hartley finds the 8-point algorithm to be accurate may be that he applies the algorithm to large-translation image pairs (see previous footnote).

[24] Using a large number of images is impossible in practice, but not in principle. See the discussion of [20, 107] in the Introduction.

inaccurate, and there is no reason to believe that the techniques in [81] or similar ones combine these reconstructions with the correct weighting (doing this would be difficult due to the strong nonlinearity of the invariant-based techniques). This suggests that it is important to study experimentally how well techniques such as those in [81] do on difficult sequences where the few-image, invariant-based reconstructions become fragile.

The same holds for the Sturm–Triggs algorithm [96]. Though this aims to be a multi-image approach, its first stage uses *small* image sets and algebraic methods to reconstruct the projective depths, and all its later computations depend on the reconstructed depths. Thus, for difficult sequences, where one cannot reconstruct reliably from a small number of images, it is unclear how well the algorithm will perform. The main virtue of a multi-image approach should be its ability to give good results when few-image algorithms cannot!

(The factorization part of the Sturm–Triggs algorithm is often stable under changes in the projective depths, so the algorithm has some protection against bad computations of these. This is not always true: I have found experimentally that the algorithm depends sensitively on the projective depths when the camera moves along a line.)

Optimal fusing revisited.[25] Instead of fusing $\geq$3-image reconstructions as in [81], one may do better to combine image-pair reconstructions—and this might work better than trilinear reconstruction on just three images. The point is that the two-image approach has a special status for point correspondences. If one neglects the positive-depth requirement on the 3D points, as is standard, the only constraint on a two-image reconstruction comes from coplanarity [41] the depths don't matter.[26] As a result, for two images one can eliminate the structure and approximately compute the least-squares error as a function of the motion alone [63, 64, 67, 111]. Optimizing just over the motion unknowns is fast, and given the recovered motion, one can recover the optimal structure *algebraically* for two images [36]. Moreover, recent experimental work [67, 111] shows that two-image optimization is robust and accurate as well as fast, and recent theory suggests that it can be made even more robust—due to the simplicity of the two-image error surface, one can identify and avoid local minima [64] or even locate the global minimum directly [91]. Thus, for two images, there exist *fast*, robust methods for computing the optimal least-squares reconstruction.

Also, the errors in two-image reconstruction are fairly well understood and easy to model: the main effect comes from the confounding of rotational and translational motions due to the bas-relief ambiguity. The ease of modeling helps make fusing more accurate (Section 2.2; see also footnote 10).

Lastly, though researchers sometimes argue against two-image methods because their reconstructions are ambiguous for special scenes (e.g., planes), one can generally resolve these ambiguities by comparing reconstructions from different image pairs.

For $\geq$3 images, in addition to coplanarity, the requirement of rigidity (that the 3D points remain fixed in place over the sequence) determines the reconstruction. To exploit rigidity an algorithm must represent the structure implicitly. In effect, it must impose the requirement that the depths recovered from different image pairs are the same. Thus, an optimization

[25] I again focus on unconstrained motion.

[26] The only usable constraint besides coplanarity is the constraint that the 3D points must be in front of the camera. Unless the motion is very small, this constraint typically has force only for a small number of points and does not significantly affect the reconstruction. I know of no implementation of two-image optimization that fully exploits the positive-depth constraint as well as the coplanarity constraint.

approach must al least minimize over all the structure/motion unknowns, which is slow.[27] Conversely, simplified approach that does not represent the structure explicitly is likely to be far from optimal, since the nonlinear transformations involved in representing the structure implicitly tend to introduce strong biases. Thus, if one wants to design an algorithm that does not reconstruct from image pairs, one confronts a dilemma: either one uses optimization and gets a slow algorithm, or one likely gets a strongly nonoptimal approach which gives poor results on difficult sequences.

Also, even an optimal $\geq$3-image reconstruction gives errors that depend in a complex way on the various motions and are difficult to model, especially for the error correlations between different points. Using a nonoptimal algorithm makes this modeling more difficult. Since accurate fusing does depend on having good models of the subestimate likelihoods, a good fusing approach from $\geq$3-image reconstructions will be difficult to achieve.

In summary, a fusing approach using image pairs has the advantage that its intermediate reconstructions are near-optimal, quick to compute, and easy to fuse accurately due to the ease of modeling their error correlations. Such an approach can be much faster than a direct optimization over all images, since its intermediate optimizations are fast. In contrast, it is unclear whether a fusing approach that uses optimization to compute intermediate reconstructions from $\geq$3 images will be faster than direct optimization, since the intermediate optimizations already involve the structure and are slow. Also, because $\geq$3-image optimization is much more complex than image-pair optimization, it is unclear whether theoretical analyses like [64, 91] can help with its robustness.

Nevertheless, for sequences where image triplets give much better reconstructions than image pairs, fusing *optimal* triplet reconstructions could give more accurate and robust results than image-pair fusing. The greater accuracy of the triplet reconstructions would produce more nearly Gaussian likelihoods, and in fusing this could compensate for the greater complexity of the motion dependence of these likelihoods. Greater accuracy would also produce better linearization within the extended Kalman filter (EKF) framework.

The *same* arguments suggest that it may be better to fuse image pairs than to fuse in one image at a time [5, 15, 85, 113], as Thomas and I [67, 92] argued previously. Since an image pair contains more information than a single image, one will get a better Gaussian model for the pair constraint on the structure than that for the single-image constraint on it, and EKF linearization will be more accurate. (Again, constraints on the motion as in [5, 15, 85, 113] can improve the effectiveness of single-image fusing.)

Current fusing approaches that use intermediate estimates based on $\geq$3 images compute these estimates by *nonoptimal* invariant-based methods and also fuse them nonoptimally (e.g., [81]). It is unknown how these approaches compare in accuracy and robustness to approaches fusing (near) optimal image-pair reconstructions. Their main apparent advantage is speed. This advantage may disappear if one uses an invariant-based algorithm incorporating iterative improvement [28]. Such algorithms gain in robustness but are still nonoptimal, and it is unknown how the iterative stage affects their speed relative to image-pair optimization.

(Note that I do not intend to address the question here of precisely how one should divide up the images in a batch/recursive approach such as that of [59, 60]. For instance, if one starts with image-pair reconstructions, one may want to first fuse pairs into quartets, then

[27] The optimization approach of [105] does in effect minimize over the structure, although it uses an implicit representation based on imposing constraints on the image measurements.

fuse these into octets, etc. I do wish to suggest that the issue of how many images to combine at each stage of reconstruction and fusing is an important methodological and experimental one.)

Summary. The arguments of this section suggest that it is important for authors of invariant-based multi-image algorithms to experimentally compare their approaches to two-image algorithms and two-image fusing techniques. We need a better understanding of the trade-offs between the simplicity of the invariant-based approaches and the optimality of the two-image ones. How far from optimal are the invariant-based approaches? For the more nearly optimal versions that incorporate a stage of iterative improvement, how do the speed and accuracy compare to those of a fusing approach based on image pairs? Is there a good compromise 3-image technique that is robust and accurate enough for fusing but still preserves some of the speed of the trilinear approach? How much do fusing approaches suffer in accuracy and robustness when their rule for combining intermediate reconstructions is strongly nonoptimal? What happens when one applies invariant-based algorithms to sequences with small baselines, few correspondences, small fields of view, and little depth range? Does the original Sturm–Triggs algorithm [96] exploit multiple images effectively on this kind of difficult sequence?

It is also important to study the intrinsic question of how reconstruction accuracy and robustness increase with the number of images used. The answer to this question will determine whether fusing approaches should add in one image, two, or more than two at each fusing step (at least in the initial stages of fusing). This could depend on conditions such as the size of the motions, and one should carry out experiments with this in mind. More generally, it is important to determine experimentally the best order for recursively combining reconstructions in a batch/recursive fusing approach [59].

Additional experimental questions include: how robust and accurate is trilinear reconstruction for lines as opposed to points? How fast can fusing approaches be made to perform [59, 60]? How can one best model the errors in reconstructions based on ≥ 3 images? For modeling the error in image-pair reconstructions, can one adequately represent the error correlations as due to the effects of the bas-relief confusion between rotations and translations? How important is good error modeling for accurate fusing? When does one encounter difficulties in resolving the ambiguities of image-pair reconstructions using different image pairs?

3. A FRAMEWORK FOR MISFM

How then should one solve the problem of multi-image reconstruction? I suggest in this section that one should design algorithms based on phenomenological analysis. There is clearly a potential to improve algorithms by adapting them to exploit an understanding of the intrinsic properties of SFM estimation. Also, I argue that one should aim at algorithms for which one can come close to proving good performance over some domain.[28] The process of coming up with a "proof" often helps to identify an algorithm's weaknesses and improve it. More important, algorithms designed with an accompanying "proof" tend to be

[28] For example, such a "near proof" could combine a rigorous proof, applying in a narrow limiting case, with a perturbative analysis, indicating that the rigorous result holds approximately over a domain of reasonable size. By adding experiments to such a theoretical analysis, one can in effect attain a phenomenological "proof" of good performance.

reliable as well as robust: for a given sequence, one can check the algorithm's behavior to see whether it conforms to the theoretical analysis, and check the data directly to verify that the sequence belongs to the right domain. If both checks pass muster, the sequence is very likely to belong to the right domain, and the algorithm's results are likely to be trustworthy.

To illustrate phenomenology usefulness, I present examples of algorithms suggested by it. I make no claims for these: experiment alone can determine what algorithms work best. Nevertheless, I hope my discussion convinces the reader that neglect of phenomenology analysis has caused the field to ignore interesting avenues of research. My main points are general. I argue that SFM is a potentially messy problem, and that it may require a messy system combining a variety of different algorithms to work well on general sequences. I also stress that any single algorithm is unlikely to work well on all sequences, and that it is important to understand what domain a given algorithm does work well in.

What I am suggesting in this section is that one should design algorithms *expressly* for the task of computing accurate and robust reconstructions. Previous research makes it clear that SFM is a difficult estimation problem and that if algorithms are not designed for robustness they can be fragile. Similarly, SFM algorithms are unlikely to give the best accuracy if one doesn't design then to. To design for accuracy and robustness, one must understand what information can be robustly extracted from noisy data—i.e., one needs an *error analysis*. Also, robust and accurate performance is not enough; one needs to *know* that an algorithm is robust—to understand what errors it is likely to produce for a given amount of noise. And one should be able to tell from its behavior when it is not working. These too amount to error analysis. Error analysis should be the basis for algorithm development and evaluation in SFM as in any other estimation problem.

This error analysis should be more than the standard, first-order analysis around a computed reconstruction. A first-order analysis gives a local estimate of how sensitively the reconstruction depends on the noise. But, to have confidence in a reconstruction, one needs to know that it is globally correct—that it is better than very different reconstructions as well as infinitesimally different ones. No local analysis can determine this: if one conducts a local analysis around a computed reconstruction that is far from the correct one, the results are meaningless. Also, a first-order analysis around a specific reconstruction is tied to that reconstruction and is useless for designing algorithms, since one designs for a range of problems, not just one.

The error analysis should also do more than simply define an estimate like the MLE as the goal of the algorithm—rather, it should lead to an understanding of the behavior of the algorithm itself. For maximum usefulness, it should qualitatively predict an algorithm's errors over a range of problems and loosely characterize the domain over which the algorithm will work well.

Ideally, one would like an algorithm and error analysis that work for every situation. I think this is impossible to achieve. An algorithm should approximately compute the optimal estimate. This is an extraordinarily complex function of the image data—much too complex for any conceivable method to compute it.[29] An algorithm that always reconstructed the optimal estimate, even in extreme cases, would be computing it in its full complexity, and

[29] For any reasonable definition of the optimal estimate. For the MLE, for instance, small changes in the image points can cause large jumps in the estimate by raising the least-squares error of the previous global minimum slightly above that of one of the local minima. Also, the image data determine the local and global minima through a hopelessly complex system of polynomial equations in many unknowns.

thus must be unrealizable. It is no more conceivable that an algorithm could always give a good approximation of the estimates:[30] any SFM algorithm will sometimes deviate from it.

SFM algorithms probably work well only under relatively restricted conditions. Since the optimal estimate is such a complex function—since it depends on the image data so differently in different domains—no one technique will always approximate it well even in practice. (For example, an algorithm that works well for large camera translations can fail completely when the translations are small.) Similarly, since the problem domain strongly determines how noise affects the optimal estimate, probably no single error analysis applies to all domains.

If no algorithm works in all domains, then *in different domains one must use different algorithms*.[31] And if, as I've argued, one should design algorithms expressly to cope with error, then since errors and algorithm performance are domain-dependent this means that *one should design algorithms specifically for their problem domains*. Finally, by exploiting the special characteristics of each domain one can get more robust, reliable, and accurate algorithms [47, 68, 69, 103, 104]. If an algorithm is designed so that it is guaranteed to work well within some domain, one can determine from its performance whether a given sequence belongs to this domain and thus establish whether the algorithm is likely to succeed. Thus, as with many complex functions, the optimal estimate can be best approximated *piecewise*, using different local approximations over different small domain patches, rather than by any one global method.

To summarize the suggestions of this section: (1) Since it seems likely that no effective general-purpose algorithm exists, to best understand and cope with error one should design algorithms for specific problem domains. (2) One should use different algorithms for different domains. (3) Using a variety of algorithms tuned for different domains may give a better approximation of the optimal estimate.

It may turn out that typical applications are easy enough that a single algorithm will do well on most of them. It is important to examine this experimentally. Even if this is true, one might still get better accuracy and robustness by combining specialized algorithms than by using one.

3.1. Domain Constraints

I have argued for designing algorithms for specific domains. In fact, typical applications produce highly constrained motion sequences, and algorithms can often exploit these constraints.

For example, motion sequences typically have limited noise and many points per image, so that the information in the image data is highly redundant—especially if the sequence also contains a large number of images. This by itself suggests an important approach to designing algorithms: at least for an initial reconstruction, *use just the part of the image information that can be exploited simply and effectively*. When the total information is

[30] For example, an invariably good approxmation method would have to track the estimate precisely as it shifted through large jumps stemming from small image changes. See previous footnote.

[31] Often researchers make a distinction between two types of approximate algorithms: those that compute the exact answer for zero noise, for example the 8–point algorithm, and those that do not, fr example the Tomasi Kanade approach [102]. But this distinction has no importance, since all sequence have noise. Both types of algorithms simplify the problem by deviating from the optimal estimate, and both give good approximations to the optimal estimate only for certain types of sequences. For both, the aim is to get a simplified aalgorithm that is more practical and robust over some range of problems.

overredundant using even a part of it will give good results, while it is usually difficult or impossible to use all of it initially. In fact, I believe that in a strict sense this *is* impossible. Using all the information means computing the optimal estimate, and it is difficult to see how one could compute this directly as an initial estimate, without iterating.

One example of this partial-information approach is an algorithm that initially omits a number of points because they are occluded in some images and thus difficult to use. Similarly, Tomasi and Kanade's algorithm [102] neglects the higher order (and thus small and difficult to exploit) perspective effects and utilizes just the first-order image displacements, since these determine the structure/motion directly and simply.

(Often researchers make a distinction between two types of approximate algorithms: those that compute the exact answer for zero noise, for example the 8-point algorithm, and those that do not, for example the Tomasi–Kanade approach [102]. But this distinction has no importance, since all sequences have noise. Both types of algorithms simplify the problem by deviating from the optimal estimate, and both give good approximations to the optical estimate only for certain types of sequences. For both, the aim is to get a simplified algorithm that is more practical and robust over some range of problems.)

Redundant information is only one of the typical conditions for motion sequences. Typical applications lead to a relatively small number of standard scenarios:

1. In robot navigation, the interimage camera motion is smooth and/or small compared to the camera's distance from the scene; scene points typically vary significantly in their depths from the camera.
2. In sequences featuring independently moving objects, the objects are typically compact compared to their distances from the camera. At least for rotating objects, the effective camera translations are typically large.
3. When the goal is to recover a 3D model of an object, the camera usually translates by distances comparable to its distance from the object. The object typically has an extent in depth comparable to (or larger than) its distance from the camera. (One might as well assume this, since otherwise this, case devolves into case 2 above.)[32]

In addition to these generic constraints, one often has specific information about the nature of the scene. For example, the scene may be dominated by the ground plane, or (for an interior scene) it may contain mostly planar surfaces and rectangular solids. Psychophysical evidence indicates that humans do use recognized objects in judging ego- and independent-object motion [16]. However, to avoid the details of dealing with many specific scenes, I will only discuss the generic constraints 1–3.

Appropriately designed algorithms can expoit all three of these constraints. For example, I [68, 69] and Sturm and Triggs [96] describe how to exploit the small translational motion in robot navigation. Crudely, this approach initially neglects small, higher-order corrections in the ratio of the translation size to the distance of the scene from the camera. The strategy resembles the domain-2 approach of Tomasi and Kanade [102], but the algorithms differ characteristically because they are adapted to different domains. In each case the domain determines what image information is most easiest to exploit.

[32] I assume known correspondence in domain 3, as I do throughout this paper. For applications that start with intensity images, it is certainly easier to compute correspondence when the motions are small instead of big as in domain 3. However, I remind the reader that there do exist applications that provide precomputed correspondences. Also, experimentally, algorithms such as in [104, 112] can compute correspondences from intensity images for motions that are large enough to qualify for domain 3, e.g, with $|T|/|Z| \sim 1/3$.

3.1.1. Domain 3: Nonlinear SFM

The domains 1 and 2 are associated with a small physical parameter and algorithms can exploit this by linearizing in these parameters. In domain 3, there is no characteristically small parameter to expand in, and the image data's dependence on the unknown structure/motion cannot be linearized. Inverting this dependence to get the optimal estimate is a fully nonlinear problem, implying that the optimal estimate's dependence on the image data will be extremely complex.

Despite this, in [70] I have argued and shown experimentally that SFM is easy in domain 3. The image data determine the reconstruction strongly and unambiguously—typically just two (or a few) images are enough to get a good reconstruction, and any standard algorithm for a few-image reconstruction is likely to work well [70]. Since the few-image reconstructions are robust and accurate, one can robustly combine them into more accurate multi-image reconstruction by fusing (Section 2.2).

The reason for SFM's tractability in domain 3 is that the signal-to-noise ratio is large even for a sequence of few images. The large translations and scene depth variations cause the translational image displacements—the signal—to be bigger than the noise and clearly distinguishable from rotational displacements [70]. (It may be possible to make these statements precise—for instance, one might be able to show the robustness of two-image optimization theoretically.[33] Indeed, the results of [64, 91] demonstrate that two-image SFM is simple enough that it is possible to understand many properties of the least-squares error.)

In a sense, at least for the initial recovery, reconstructing from two images can be the *only* viable approach in domain 3. *There is sometimes no direct way to exploit > 2 images effectively.* Two images already determine the optimal estimate accurately [70], so for many images, the estimate's dependence on each additional image becomes increasingly subtle—one would need polynomials of extremely high order to capture it. A direct, i.e., noniterative, algorithm must have a limited algebraic complexity, since otherwise it will not give a practical computation and thus, it cannot reproduce the complexity of the optimal estimate's dependence. The more images such an algorithm attempts to use, the more its biases—its deviations from optimality—will dominate the information in the additional images, and the algorithm will simply waste most of this information.

This argument applies to *any* nonoptimal algorithm, such as an iteratively improved version of a direct technique. Any such algorithm will fail to capture the optimal estimate's dependence on the data if the number of images is large enough. In domain 3, the large nonlinear effects can cause such a failure even for three images.

These arguments again suggest, here for domain 3, that fusing two-image reconstructions might work better than the trilinear approach or other multi-image invariants method (see the discussion at the end of Section 2.4). They imply that the best way to get a multi-image reconstruction in domain 3 is to start with a two-image or few-image reconstruction and then extend it iteratively. Fusing can be a useful technique for this. For example, by dividing the image sequence into image pairs and computing separate initial optimizations

[33] First, one may be able to show that the 8-point method gives good results. Even though this algorithm weights the information in the data nonoptimally, the data determine the correct reconstruction so strongly over domain 3 that the resulting bias may be unimportant. Second, it may be possible to show that the least-squares error has a deep well around the global minimum that is broad compared to the scale of the 8-point estimate's error [70]. Due to the small noise-to-signal ratio, this could require little more than a local curvature analysis of the error around the minimum.

over these pairs, one can get greater speed [60]—and perhaps reliability [64, 91]—than in a global optimization. And since two-image reconstructions are accurate in domain 3, fusing them can give enhanced accuracy while saving the computational cost of going back to the original images for a global optimization [59, 60].

On the other hand, the technique of fusing $\geq$3-image, invariant-based reconstructions confronts a dilemma. Section 2.4 argued that the invariant-based (e.g., trilinear) approach is nonrobust on difficult sequences, which implies that a fusing approach based on it will tend to fail on such sequences (Section 2.2). I just suggested that this approach also fails on easy sequences—even though its accuracy may seem reasonable, it does not capture the optical reconstruction well. These arguments suggest that invariant-based fusing might be advantageous only on in-between sequences that are neither too difficult nor too easy.

3.1.2. Distinguishing Domains

The three domains cited above suggest three quite different algorithms. Clearly, the SFM problem changes significantly depending on the domain; algorithms specialized for one domain will not work throughout another. For example, an algorithm that relies on reconstructing initially from just a few images (as is appropriate for domain 3) will fail if getting a reasonable reconstruction requires many images.

Because image sequences differ characteristically in different domains, it is sometimes possible to determine the domain directly from the image data. For instance, by definition a domain-3 sequence has at least one image pair with a large baseline, i.e., an interimage camera translation that is comparable to the distance between camera and scene. Because the scene depth variation is large, no pure rotation can align these two images without a large residual error. A rough check of whether a sequence belongs to domain 3 is to test whether some images cannot be rotationally aligned.

3.1.3. Mixed-Domain Sequences

Not all SFM problems fall cleanly into one of the domains above. How can these mixed cases be handled? First, it does appear experimentally that the algorithms described above apply broadly [14, 68, 70]: often cases intermediate between two domains can be handled by approaches specialized for either. For instance, the algorithms of [73] and [71] (respectively, domain-3-style and domain-1-style approaches) have both worked on the Martin–Marietta rocket-field sequence [19].

Second, if a sequence spans the extremes of different domains, none of the algorithms suggested above may work on the entire sequence. One can then select parts of the sequence that do belong to single domains. (For example, the previous section describes how to select pairs of images from domain 3.) One can apply specialized algorithms to compute initial reconstructions from these parts, although different algorithms may be needed for parts in different domains. Afterward, one can extend the partial results iteratively to give a full reconstruction from the whole sequence. (I mean to include here the strategy of reconstructing initially from a single part and then using an iterative algorithm to incorporate the remaining data.)

Although dividing up the data and using a mix of different algorithms may seem awkward, I suggest that *it is sometimes unavoidable*. I proposed previously that no single algorithm can cope with all sequences and that the best way to reconstruct robustly is by exploiting constraints. If no constraint applies to the whole sequence, one should simplify the problem

initially by selecting parts that are constrained enough to deal with, and if different parts obey different constraints, the algorithms for these parts should differ too.

The point is that getting *any* robust initial reconstruction can be difficult. SFM is a nonlinear estimation problem, and in general such problems are intractable. Essentially, one can solve a generic nonlinear estimation problem in just two situations: (1) when one can approximate it as linear, and (2) when the estimation is easy—for instance, the error function to be minimized is roughly a single well with one global minimum. One can get good reconstructions in domain 1 or 2 because SFM is approximately linear and in domain 3 because it is easy. This argument suggests that, for any given sequence, one should start by making whatever compromises are necessary—disregarding difficult data, approximating, etc.—to shape the problem into one of these two tractable situations.

The disadvantage of using just part of the available information is that this gives a less constrained and therefore less accurate reconstruction. But initially one only needs an adequate reconstruction, not a perfect one—an iterative technique such as fusing or optimization can compute a good reconstruction from all the data as long as it starts from a reasonable first estimate. (Similarly, if an approximate algorithm gives a reasonable initial reconstruction, an iterative technique can use this to correct the approximation and produce a full reconstruction exploiting all the information, with no approximation.[34]) It is important that the initial reconstruction be adequate—with a poor starting point an iterative-improvement technique can fail. Thus, the problem for a complex sequence is to find a constraint or approximation that is respected well enough by a large enough subset of the data that one of the available direct algorithms can use it to give a good reconstruction.

Since the appropriate methods for getting good initial reconstructions depend on the sequence, so too does the subsequent processing for extending these to a full reconstruction. In general, one needs a *flexible decision rule* for deciding which algorithms to use at each stage and how best to combine their results—for instance, in what order partial reconstructions should be fused [59, 60].

To summarize: (1) I suggest that the key to dealing with a complex sequence is finding a way to generate an initial reconstruction that is good enough. To make initial headway, discarding data and introducing approximations may be necessary. (2) In some cases, a single original sequence will require a range of different algorithms and a flexible approach for combining their results.

3.1.4. An Example

A long egomotion sequence with large depth variations in the scene and camera translations ranging from small to very large is a common example spanning domains 1 and 3. Due to the large translations, applying a domain-1-style algorithm to the whole sequence is likely to fail. Also, the large translations typically cause occlusion, which makes a domain-1 approach more difficult. A domain-3 approach can also fail, for example due to the difficulty of establishing correspondence (or, occlusion may cause the large-baseline image pairs to have too few points in common). Certainly, a domain-3 approach initially neglects most of the information in a long sequence.

[34] For instance, Tomasi's algorithm neglects perspective effects, but an iterative procedure [14] starting from this algorithm's reconstruction can include these effects. I [67, 68] and Sturm and Triggs [96] describe a similar tactic for domain 1.

A good approach might be to form groups of images with small relative translations and to apply a domain-1-style approach to each of these groups. Correspondence becomes easier for such groups, and a multi-image group can produce a better initial reconstruction than one based on just two or three images. Alternatively, one could partition a sequence into a variety of groups, some with small translations and many images, and other consisting of few images with large baselines. In either case, one could fuse the initial reconstructions from different image groups for a full multi-image reconstruction.

In general, it can be a difficult problem to select good initial image groups. One possibility is to group images according to whether they can be approximately rotationally aligned (Section 3.1.2). Also, in practice the images often have a known order corresponding to the camera's motion, and one can divide images into groups according to this.

Since in practice algorithms seem to be robust over wide domains, even crude initial groups could suffice. Also, the fact that grouping might be difficult does not mean that algorithms should avoid dealing with this issue or that useful groupings are impossible to achieve. Grouping is difficult in general vision as well, but it is clearly important to do and humans clearly do it.

In some practical applications, the goal is to create a global map of an environment, most of which is occluded from any single viewpoint. One approach is: (1) Compute good reconstructions from small sets of images; and (2) register, i.e., fuse, these reconstructions based on the (possibly small) number of 3D points that they share. Global optimization over the entire structure and motion would be impractical for this application, at least initially, since optimization needs a starting reconstruction of all points in a single coordinate system, and this can be obtained only by registering the partly disjoint initial reconstructions.

3.2. Bayesian Inference

Sections 3.1.3 suggested that it can be important to segment a sequence, or to choose an approximation for dealing with it, *before* reconstructing. The segmentation or approximation one should choose depends upon the unknown structure/motion, and because of SFM's nonlinearity, there are sometimes no simple image features from which one can draw firm conclusions about the structure and motion without fully reconstructing. For example, establishing the appropriateness of the orthographic approximation requires showing that some 3D points are closely clumped, which is infeasible in general without a reconstruction.

Thus it is necessary to *guess* the appropriate segmentation or approximation. To guess well one must take advantage of prior knowledge about the typical properties of motion sequences, i.e., use a *Bayesian* strategy. Most work on SFM implicitly incorporates some expectation for what the data will look like. For example, the Tomasi–Kanade algorithm [94] assumes that the 3D scene is compact compared to its distance to the camera. For a difficult task like segmentation, using these expectations is essential. Fortunately, given that we do know a lot about the constraints on real-world sequences, making reasonable guesses is often not difficult.

While it is important to use prior knowledge, this is not enough to solve the problem. For instance, what I call a "primitive" Bayesian strategy, which simply defines the goal of the computation to incorporate the prior and then uses optimization to attempt this goal, is often inadequate (Section 2.1). Instead, I argue that a viable approach to Bayesian inference should consider the difficulties of computing the goal as well as the goal itself; it should take into account what algorithms are feasible and practicable. Perhaps one can hope for

some kind of extended Bayesian approach that incorporates computational constraints as well as the prior. However, if the computations are difficult, as they seem to be in SFM, it may not be possible to use the prior in a precise manner. The approach that I have proposed starts with a small list of effective algorithms, dictated by the computational requirements, and attempts to guess, based crudely on the prior, which of these will work best on which part of the sequence data. The flavor of this more resembles Ramachandran's "bag of tricks" approach [75] then traditional statistical inference.

I believe that the vision problem in general also requires this kind of opportunistic and imprecise implementation of the Bayesian approach.

3.3. Summary

This section proposed several strategies for algorithms, with the goal of illustrating how one can exploit a phenomenological analysis of SFM to design then. I do *not* claim that these strategies will necessarily result in the best algorithms. (I suspect, for instance, that one should not spend much computational effort on grouping data into domains, given that algorithms seem to work well far beyond their proper domains; simple heuristics may be enough to give adequate segmentations.) But algorithms based on phenomenology are worth exploring.

I have emphasized SFM's messiness and suggested that dealing with this mess requires a messy algorithm. I also stressed that algorithms are domain-specific. Only experiments can determine how much of the mess algorithms need to deal with in practice, and how messy a good algorithm should be. On the other hand, most SFM papers don't test their algorithms on a wide variety of sequences, and since most also don't identify the type of sequence that their algorithm should do best on, it is difficult to know how general their approaches are. Such experiments leave open the possibility that these algorithms will *not* work well on all sequences, and that only a messy combined system will be able to. (For instance, I have found recently that the Sturm–Triggs approach [96] does not work well when the camera moves along a line.) More recently, a few researchers [47, 48, 104] have begun to advocate combining different, domain-specific algorithms using decision rules into systems of the sort that I propose here. Although researchers have mostly not needed to analyze the domain specificity of their algorithms until now, it may be important to do so in the future to obtain effective, generally applicable algorithms.

The discussion in this section suggests many experimental and phenomenological questions. For instance:

- Can one identify the domains that current algorithms work best in? How well do current approaches work outside their proper domains? Which are the best algorithms in a given domain? In practice, is there a single algorithm that works adequately on all sequences?
- What useful domains can one identify? (The domain of roughly planar 3D scenes is a natural example that I have not discussed here.) What other parameters besides the ones mentioned here can an algorithm expand in to get good approximate reconstructions?
- How much of the data in a sequence should an algorithm attempt to exploit initially? How does this depend on the conditions of the sequence?
- What are the cross-over points at which one should switch between algorithms intended for different domains [48, 104]?
- How well can a direct, noniterative algorithm exploit multi-image data in domain 3? In domain 1?

- Can one segment the data into different domains well enough so that an algorithm will succeed on the data segment it is applied to? What criteria should one use to set the break points between segments? How many segments should be used?
- Is SFM a messy problem? Is segmentation/grouping an important issue? Does obtaining a robust, general purpose SFM system require an "intermediate-level" strategy similar to what appears to be needed for the general vision problem?
- What are the trade-offs between a simple, inflexible approach (e.g., based on a single algorithm) and a sophisticated, more flexible system of the type suggested in Section 3.1.3 and in [104]?

4. GENERAL EXPERIMENTAL ISSUES

This paper has discussed some of the potential problems with current algorithms and raised issues that may be important for the design and experimental evaluation of algorithms.

In this section, I present more general observations on experimental evaluation. First, in evaluating an algorithm's accuracy, one should test it against the MLE or other optimal estimates.[35] An optimal estimate, by definition, represents the ideal of accuracy, and comparing an algorithm against it gives a measure of how closely the algorithm approaches the ideal. Without this comparison, one can fall into the trap of overrating an algorithm's performance due to testing it just on easy sequences, on which many algorithms would do well. Also, in evaluating *multi-image* algorithms, it is crucial to compare them to two-image and fusing approaches, to assess how well they are exploiting the multi-image data. Since two-image algorithms are already optimal, the main point of a multi-image algorithm should be its ability to handle sequences that two-image algorithms cannot deal with—or to do much better when two-image algorithms and their fusing extensions do work (e.g., in domain 3). A multi-image algorithm has little use unless it improves on standard few-image-based approaches. Recent results demonstrating the accuracy of two-image reconstructions [67] reinforce these arguments. One should evaluate algorithms by their reconstruction accuracies, *not* by the size of their image reprojection errors. The first reason for this is that it is typically the reconstruction that one is interested in. Two different algorithms might give small differences in the reprojection errors but large differences in their reconstruction errors. Also, it is conceivable that one algorithm may do better than another at reconstruction but give *larger* reprojection errors. For instance, an approximate algorithm may ignore some component of the image error that is relatively unimportant for reconstruction, but take better account of the rest of the error than a competing algorithm which tries to account for the entire image error.

I have argued that any algorithm will work well only in a limited domain. It is important to test algorithms with an understanding of the domains they are likely to work well in, to avoid tests on inappropriately easy or hard cases, or inappropriate comparisons of algorithms suited for different domains.

To evaluate an algorithm's robustness and accuracy, one must test it on many sequences, but there are too few real-image sequences (especially with ground truth) for adequate statistics. Thus, where possible, reconstruction algorithms should be tested on a large number of

[35] This is preferable to the Cramer–Rao bound [49], which is a lower bound and difficult to compute except by using approximate local information. Evaluating an algorithm in comparison to this bound is probably adequate.

synthetic sequences. In fact, I argue that testing algorithms on synthetic sequences is often preferable and that real-image tests can be misleading.

The main problem with real-image experiments, apart from the crucial lack of statistics, is that their results depend strongly on whatever approach is used for dealing with correspondence. This is fine if one is testing a complete system that determines correspondence as well as reconstructs, e.g., a direct method, or that uses robust statistics to minimize the effects of outliers. But correspondence and reconstruction are often treated as separate problems. If the point of a paper is a new reconstruction algorithm which that makes no contribution toward solving the correspondence problem, then the algorithm should be tested for what it is—for its competence at reconstruction. Testing it on real images will only confuse the issue, since the algorithm will appear to work well or badly depending on the correspondence method it is combined with; a given reconstruction algorithm can be extended in many ways to handle correspondence. It is a basic principle of science that one should conduct experiments under controlled conditions, eliminating all extraneous factors. For testing a reconstruction algorithm, one should eliminate the extraneous and poorly understood effects of correspondence algorithms.

Also, for a strongly overconstrained problem like MISFM, algorithms will behave similarly for any reasonable noise distribution as long as the noise is small, and even with a few large noise values. Synthetic experiments with Gaussian noise will give a good picture of an algorithm's intrinsic accuracy and robustness.

A potentially serious problem with synthetic sequences is that they may not capture the special structures or motions that are typical of the real world. However, we have an increasingly good understanding of the 3D structures (e.g., planes or conics) or motions (sideways or forward) that cause problems for algorithms, and one can create synthetic sequences to incorporate these conditions. One can also run algorithms on sequences ranging near the problem conditions to determine their sphere of influence. It is always possible to use the motions and tracked points recovered from a few "typical" real-image sequences as the models for generating a range of similar synthetic sequences. Synthetic experiments make possible a much more through exploration of the problem domain.

Another claimed difficulty with synthetic images is that they do not reflect the complex miscalibrations of real cameras. But it is easy to add *linear* calibration error to synthetic images. The effects of nonlinear miscalibrations are complicated, and probably the only way to deal with them is to calibrate them away. They are a separate problem, which it makes sense to ignore in designing and evaluating reconstruction algorithms.

In principle, if there existed a large database of tracked correspondences extracted from real-image sequences, then one could compare reconstruction algorithms on this database. But my view is that the choice of techniques for dealing with correspondence outliers will remain essentially the same no matter what algorithm is used to solve the basic problem of reconstruction.[36] Thus, an approach that makes sense is to first develop good reconstruction algorithms, which should be tested on synthetic sequences with Gaussian noise, and then extend these to deal with outliers, where the extensions can be tested on data from real images. There is little reason to think that an algorithm will do worse on Gaussian noise and yet better at outliers, unless it already incorporates some explicit technique for handling the latter.

[36] Large correspondence error will always remain difficult or impossible to model precisely since it depends on the large-scale image context. Probably the best one can do is the sort of crude outlier model embodied in robust statistics.

It is sometimes declared that SFM is a "solved" problem. In fact, there has been relatively little progress on *reconstruction* algorithms since the days when SFM was thought to be hard. SFM has been solved mostly in the sense that faster computers have made it possible to run algorithms exploiting more data—the problems have becomes easier, rather than the algorithms better.[37] If one wants a true "solution" to SFM, i.e., an algorithm that reconstructs as well as possible even on hard problems, then one needs a detailed scientific understanding of how algorithms' behaviors depend on the sequence parameters. This requires controlled synthetic experiments.[38]

I stress again that these arguments do not apply to approaches that deal explicitly with the correspondence issue, such as direct algorithms.

5. CONCLUSION

This paper espouses a scientific approach to designing and evaluating algorithms. I argued that it is important to develop a fundamental phenomenological understanding of how algorithms behave, one which can be neither purely mathematical nor experimental but must be a combination of both. I explained how a basic error-sensitivity analysis of SFM raises natural questions about the robustness of many current approaches, and advocated more careful testing of algorithms in light of these questions. If there is no obvious reason why an algorithm should be robust and accurate, then it often is not. When an algorithm has no clear theoretical justification, good science requires a careful experimental one. Anecdotal evidence in the form of a few real-image experiments is not sufficient.

I have sketched a variety of new algorithms in this paper. These are not necessarily intended to replace current approaches. Rather, they are meant to highlight by contrast some of the potential problems with standard approaches, to show that there are alternatives, and to illustrate how one can embody a phenomenological analysis of SFM in an algorithm.

It is increasingly clear that the error properties of SFM can be qualitatively understood. I believe that SFM algorithms should incorporate such an understanding and that they can be reliable only if they do. For example, I argued that algorithms are robust in limited domains. With an understanding of the physical conditions under which an algorithm is robust, one can check for any given sequence whether these conditions apply and determine whether the algorithm is likely to succeed, thereby achieving reliability.

It may turn out that current approaches, despite their flaws, are adequate for many applications. It is nevertheless an interesting scientific question to find the best possible algorithm.

Selected highlights. I discussed some of the important points in this paper briefly, since they are easy to argue for, and some of the less important ones at length, since they are more subtle. To ensure that the reader does not overlook my main points, I list what I think are the paper's most-valuable or most-needed contributions.

- I suggested that the authors of invariant-based methods should do careful experimental studies of their methods' noise sensitivity. I argued that invariant methods tend to be very sensitive to noise.

[37] Some exceptions include [32, 91, 96, 102, 106].

[38] A rigorous experimental study of SFM reconstruction is appropriate because it has a precisely defined goal. For other vision problems that are less well defined or less mature, rigorous evaluation is often inappropriate.

• I pointed out that two-image reconstruction is an especially effective approach for point correspondences, and I suggested that researchers compare invariant methods to this approach. I argued that the invariant and two-image approaches are likely to work well under the same conditions, and that invariant algorithms may give worse results than two-image reconstruction (or a fusing of such reconstructions). I proposed testing invariant algorithms on "difficult" sequences where two-image algorithms have trouble, i.e., ones with a small number of tracked points, small motions, small field of view, and small depth range.

• I argued that researchers should attempt to understand the effects of the physical conditions of a sequence on the behavior and the robustness of their algorithms, since algorithms perform very differently under different conditions. As an important example, I argued that algorithms that reconstruct directly (or nearly directly) from many images are likely to fail in domain 3. As another, I reported my experimental finding that the Sturm–Triggs [96] approach tends to fail when the camera moves along a line in a roughly forward direction. I emphasized that understanding the conditions under which an algorithm does well or badly is important for conducting good experimental tests of it.

• I suggested that, since algorithms perform well over limited domains, one should try to exploit the conditions of these domains to design the best possible algorithms for them. I argued that one may get a better overall system by combining several such algorithms.

• I stressed that some multi-image algorithms use intermediate results calculated from a few images, and that it is important to check whether these algorithms are robust on sequences that few-image methods cannot deal with. I argued that the main point of a multi-image approach should be its ability to do better than a few-image one.

• I noted the discovery of [64] that the projective approach has much more noise sensitivity than the Euclidean one (for linear motion and candidate epipoles outside the image).[39] I argued that the Euclidean approach's susceptibility to miscalibration is a much less serious flaw, since one can estimate the reconstruction error caused by miscalibration and perhaps correct it given the true calibration. The large errors of the projective approach are wholly due to noise and cannot be corrected.

• I noted that the projective approach is formally equivalent to the Euclidean one, if one makes the standard approximation of relaxing the orthogonality constraints on the rotation matrices.

• I noted that the line-based trilinear approach could be much better than the point-based one.

• My suggestions in Section 2.3.6 about how the bas-relief ambiguity interacts with the effects of calibration error are new and worth investigating.

• I emphasized new research issues in fusing, including the order in which fusing should be done and how one should divide up the computation between direct reconstruction and fusing. I stressed that an experimental understanding of how the robustness of a reconstruction increases with the number of images used could be important for resolving these issues. I proposed that it is important to develop good phenomenological errors models for the motion dependence of the intermediate reconstructions. More generally, I proposed that researchers should consider designing fusing approaches based on phenomenology.

• I pointed out that Euclidean optimization may be a better technique for self-calibration than starting with a projective optimization and upgrading the result to a Euclidean one.

• I proposed that solving SFM for general sequences may ultimately require an "intermediate-level" approach that incorporates several low-level algorithms within it. (This is almost guaranteed if one wants an approach that deals with correspondence error and

outliers.) I suggested that SFM might present a simplified version of the "chicken and egg" problem encountered in the general vision problem.

• I proposed the notion of *reliability* as an important desideratum for algorithms. I stressed the value of science and phenomenology in vision research.

6. UPDATE

A few relevant results have appeared in the literature since I first wrote this paper, and I give a brief overview of them here.

• In [67] I found experimentally that optimal two-image reconstructions were nearly as accurate as the optimal reconstructions for 15 images, apart from the effects of the bas-relief ambiguity. The results of [63, 64, 67, 111] show that one can compute a near-optimal least-squares reconstruction from two images by minimizing just over the five motion parameters, which gives a fast algorithm. These results and point to my argument that the authors of multi-image algorithms need to carefully compare their results to those of two-image algorithms.

• In [72] Govindu and I found experimentally that, for sequences where the camera positions do not lie just or a line or plane, the projective approach is nearly as accurate as a Euclidean one that exploits the correct calibration. Thus, the projective approach may be quite useful in these situations. However, in [64] I show that the projective approach gives inaccurate results when the true translational motion is sideways[39] (when the epipole lies outside the image region). This makes more precise my claim that projective algorithms are less accurate than Euclidean ones. In [64] I also give a concrete explanation for why the "projective" error surface has fewer local minima than the Euclidean one and describe the trade-offs between the two approaches (fewer local minima at the cost of lower accuracy).

• Chiuso *et al.* [13] confirmed experimentally the suggestion in [64] that the SFM error surface has many local minima for epipoles within the image. This appears to be true both for the projective and Euclidean approaches. Srinivasan [91] described an algorithm that can quickly locate the *global* minimum of the error surface for such epipoles, at least under the hypothesis of infinitesimal motion. In [64] I showed that the error for infinitesimal motion often gives a good approximation to the least-squares error for large motion. These results suggest that applying the algorithm of [91] to the error for infinitesimal motion is an effective technique for recovering forward motion and show the practical value of a phenomenological analysis of the error surface.

• Sturm and Triggs [6, 40, 96] developed an effective projective approach for domain-1 sequences (small motions). I have found experimentally that this approach works less well for larger motions. Also, I found recently that, though the approach works well when the camera moves in many directions, it tends to fail when the camera moves forward along a line. This verifies my point that it is important to investigate what domains algorithms work well or fail in.

• Approaches combining separate algorithms optimized for different domains have been developed by Torr *et al.* [103, 104]. These works illustrate the value of my proposed strategy of combining a variety of algorithms optimized for different domains into a system that will give good performance over a range of conditions.

[39] This is part of the explanation for the problems of the Sturm-Triggs approach [96].

• The work of Bougnoux [10] shows experimentally that typical calibration errors do not destroy the usefulness of Euclidean reconstructions. This bolsters my suggestions that it is important to study the effect of calibration errors on Euclidean reconstructions and that it is not necessary to resort to the projective approach to deal with calibration error.

• Several works have underscored my point in this paper that camera rotations, rather than translations, are most useful for determining the calibration [55, 93, 94].

• Fast fusing approaches that approximate the optimal batch reconstruction are being developed by McLauchlan [59]. This paper emphasizes the issue of the order in which fusing should be carried out, which I also stress here.

ACKNOWLEDGMENTS

I thank David Jacobs, Mike Werman, Karvel Thornber, Olga Veksler, Venu Govindu, and Bosco Tjan for their helpful comments on this paper.

REFERENCES

1. G. Adiv, Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field, *IEEE Trans. Pattern Anal. Mach. Intell.* **11**, 1989, 477–489.
2. A. Azarbayejani and A. Pentland, Recursive estimation of motion, structure and focal length, *IEEE Trans. Pattern Anal. Mach. Intell.* **17**, 1995, 562–575.
3. N. Ayache and O. Faugeras, Maintaining representations of the environment of a mobile robot, *IEEE Trans. Robot. Automat.* **6**, 1989, 804–819.
4. Y. Bar-Shalom and X.-R. Li, *Estimation and Tracking: Principles, Techniques, and Software*, Artech House, Norwood, Massachusetts, 1993.
5. P. Beardsley, A. Zisserman, and D. Murray, Sequential updating of projective and affine structure from motion, *Internat. J. Comput. Vision* **23**, 1997, 235–259.
6. R. Berthilsson, A. Heyden, and G. Sparr, Recursive structure and motion from image sequences using shape and depth spaces, *Comput. Vision Pattern Recognit.*, 1997, 444–449.
7. T. J. Broida and R. Chellappa, Estimating the kinematics and structure of a rigid object from a sequence of monocular images, *IEEE Trans. Pattern Anal. Mach. Intell.* **13**, 1991, 497–513.
8. T. J. Broida, S. Chandrashekhar, and R. Chellappa, Recursive estimation of structure and motion using relative orientation constraints, *IEEE Trans. Aerospace Electron. Systems* **26**, 1990, 639–656.
9. P. Belhumeur, D. Kriegman, and A. Yuille, The bas-relief ambiguity, in *Comput. Vision Pattern Recognit. 1997*, pp. 1060–1066.
10. S. Bougnoux, From projective to euclidean space under any practical situation, a criticism of self-calibration, in *ICCV 1998*, pp. 790–795.
11. A. Chiuso and S. Soatto, "MFm": 3-D motion from 2-D motion causally integrated over time Part I: Theory, Washington University technical report, 1999, and Tutorial at ICRA 2000.
12. A. Chiuso and S. Soatto, MFm": 3-D motion from 2-D motion causally integrated over time Part II: Implementation, in *ECCV II, 2000*, pp. 734–750.
13. A. Chiuso, R. Brockett, and S. Soatto, Optimal structure from motion: Local ambiguities and global estimates, Washington University technical report, 1999, and *Internat. J. Comput. Vision*, in press.
14. S. Christy and R. Horaud, Euclidean shape and motion from multiple perspective views by affine iteration, *IEEE Trans. Pattern Anal. Mach. Intell.* **18**, 1996, 1098–1104.
15. N. Cui, J. Weng, and P. Cohen, Extended structure and motion analysis from monocular image sequences, in *ICCV 1990*, pp. 222–229.
16. J. E. Cutting, M. Fluckinger, B. Baumberger, and J. D. Gerndt, Local heading information and layout from full-cue, simulated pursuit-fixation displays, ARVO Abstract 2069, 1996.

17. K. Dana, S. Nayar, B. van Ginneken, and J. J. Koenderink, Reflectance and texture of real-world surfaces, in *Comput. Vision Pattern Recognit. 1997*, pp. 151–157.
18. K. Daniilidis and M. Spetsakis, Understanding noise sensitivity in structure from motion, in *Visual Navigation* (Y. Aloimonos, Ed.), pp. 61–88, Lawrence Erlbaum, 1993.
19. R. Dutta, R. Manmatha, L. R. Williams, and E. M. Riseman, A data set for quantitative motion analysis, in *Comput. Vision Pattern Recognit. 1989*, pp. 159–164.
20. O. Faugeras and B. Mourrain, On the geometry and algebra of the point and line correspondences between *N* images, in *ICCV 1995*, pp. 951–956.
21. O. D. Faugeras, What can be seen in three dimensions with an uncalibrated stereo rig? in *ECCV 1992*, pp. 563–578.
22. C. Fermüller and Y. Aloimonos, Qualitative egomotion, *Internat. J. Comput. Vision* **15**, 1995, 7–29.
23. C. Fermuller and Y. Aloimonos, On the geometry of visual correspondence, *Internat. J. Comput. Vision* **21**, 1997, 223–247.
24. C. Fermuller, Passive navigation as a pattern-recognition problem, *Internat. J. Comput. Vision* **14**, 1995, 147–158.
25. C. Fermüller and Y. Aloimonos, Direct perception of three-dimensional motion through patterns of visual motion, *Science* **270**, 1995, 1973–1976.
26. K. J. Hanna, Direct multi-resolution estimation of ego-motion and structure from motion, in *Motion Workshop 1991*, pp. 156–162.
27. C. G. Harris and J. M. Pike, 3D positional integration from image sequences, *Image Vision Comput.* **6**, 1988, 87–90.
28. R. I. Hartley, Minimizing algebraic error in geometric estimation problems, in *ICCV 1998*, pp. 469–476.
29. R. I. Hartley, In defense of the eight-point algorithm, *IEEE Trans. Pattern Anal. Mach. Intell.* **19**, 1997, 580–593.
30. R. I. Hartley, Lines and points in three views and the trifocal tensor, *Internat. J. Comput. Vision* **22**, 1997, 125–140.
31. R. I. Hartley, A linear method for reconstruction from lines and points, in *ICCV 1995*, pp. 882–887.
32. R. I. Hartley, In defense of the 8-point algorithm, in *ICCV 1995*, pp. 1064–1070.
33. R. Hartley, Euclidean reconstruction from uncalibrated views, in *Second Workshop on Invariants, 1993*, pp. 187–202.
34. R. I. Hartley, Self-calibration from multiple views with a rotating camera, in *ECCV 1994*, pp. 471–478.
35. R. I. Hartley, Lines and points in three views—A unified approach, in *IUW 1994*, pp. 1009–1016.
36. R. I. Hartley and P. Sturm, Triangulation, in *IUW 1994*, pp. 957–966.
37. R. I. Hartley, Estimation of relative camera positions for uncalibrated cameras, in *ECCV 1992*, pp. 579–587.
38. D. J. Heeger and A. D. Jepson, Subspace methods for recovering rigid motion I: Algorithm and implementation, *Internat. J. Comput. Vision* **7**, 1992, 95–117.
39. F. Kahl and Anders Heyden, Affine structure and motion from points, lines, and conics, *Internat. J. Comput. Vision* **33**(3), 1999, 163–180.
40. A. Heyden, Projective structure and motion from image sequences using subspace methods, in *SCIA II*, pp. 963–968, 1997.
41. B. K. P. Horn, Relative orientation, *Internat. J. Comput. Vision* **4**, 1990, 59–78.
42. B. K. P. Horn and E. J. Weldon, Jr., Direct methods for recovering motion, *Internat. J. Comput. Vision* **2**, 1988, 51–76.
43. M. Irani, B. Rousso, and S. Peleg, Recovery of ego-motion using region alignment, *IEEE Trans. Pattern Anal. Mach. Intell.* **19**, 1997, 268–272.
44. A. D. Jepson and D. J. Heeger, Linear subspace methods for recovering translational diection, University of Toronto Technical Report RBCV-TR-92-40, p. 19, 1992.
45. A. D. Jepson and D. J. Heeger, A fast subspace algorithm for recovering rigid motion, in *Motion Workshop, Princepton, NJ, 1991*, pp. 124–131.

46. A. D. Jepson and D. J. Hegger, Subspace methods for recovering rigid motion II: Theory, University of Toronto Technical Report RBCV-TR-90-36, 1990.
47. K. Kanatani, Geometric information criterion for model selection, *Internat. J. Comput. Vision* **26**(3), 1998, 171–189.
48. K. Kanatani, Automatic singularity test for motion analysis by an information criterion, in *ECCV I, 1996*, pp. 697–708.
49. K. Kanatani, *Statistial Optimization for Geometric Computation: Theory and Practice*, Elsevier, Amsterdam, 1996.
50. J. J. Koenderink and A. J. Van Doorn, Affine structure from motion, *J. Opt. Soc. Am.* **8**, 1991, 377–385.
51. R. Kumar, P. Anandan, and K. Hanna, Direct recovery of shape from multiple views: A parallax based approach, in *ICPR A 1994*, pp. 685–688.
52. S. Lawrence, A. C. Tsoi, and C. L. Giles, Local Minima and Generalization, in *International Conference on Neural Networks, 1996*, pp. 371–376.
53. H. C. Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, *Nature* **293**, 1981, 133–135.
54. Q. T. Luong and O. Fauggeras, On the direct determination of epipoles: A case study in algebraic methods for geometric problems, in *ICPR 1994*, pp. 243–247.
55. Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, Euclidean reconstruction and reprojection up to subgroups, in *ICCV 1999*, pp. 773–780.
56. L. Matthies, T. Kanade, and R. Szeliski, Kalman filter-based algorithms for estimating depth from image sequences, *Internat. J. Comput. Vision* **3**, 1989, 209–236.
57. S. J. Maybank, Stochastic properties of the cross ratio, *Pattern Recognit. Lett.* **17**, 1996, 211–217.
58. S. Maybank, *Theory of Reconstruction from Image Motion*, Springe-Verlag, Berlin, 1992.
59. P. F. McLauchlan, The variable state dimension filter applied to surface-based structure from motion, University of Survey technical report, 1999.
60. P. F. McLauchlan and D. Murray, A unifying framework for structure and motion recovery fom image sequences, in *ICCV 1995*, pp. 314–320.
61. J. Oliensis and M. Werman, Structure from motion using points, lines, and intensities, *CVPR 2000*, pp. 599–601, and J. Oliensis, Direct structure from motion for hand-held cameras, in *ICPR 2000*.
62. J. Oliensis and Y. Genc, Fast algorithms for projective multi-frame structure from motion, in *ICCV 1999*, pp. 536–543.
63. J. Oliensis and Y. Genc, New algorithms for two-frame structure from motion, in *ICCV 1999*, pp. 737–744. [The error defined in this paper gives a good approximation to the true error for all motions including forward motion, unlike the error used in [67, 111], but it is not exact as was claimed.]
64. J. Oliensis, A new structure from motion ambiguity. *Trans, Pattern Anal. Mach. Intell.* **22**, 2000, 185–700.
65. J. Oliensis, Recovering heading and structure for constant-direction motion, NEC technical report, 1997.
66. J. Oliensis, Computing the camera heading from multiple frames, in *CVPR 1998*, pp. 203–210.
67. J. Oliensis, A multi-frame structure from motion algorithm under perspective, *Internat. J. Comput. Vision* **34**, 1999, 163–192.
68. J. Oliensis, Multiframe structure from motion in perspective, in *Workshop on the Representations of Visual Scenes, June 1995*, pp. 77–84.
69. J. Oliensis, A linear solution for multiframe structure from motion, in *IUW, 1994*, pp. 1225–1231.
70. J. Oliensis, Rigorous bounds for two-frame structure from motion, in *ECCV 1996*, pp. 184–195.
71. J. Oliensis, Structure from linear and planar motions, in *CVPR 1996*, pp. 335–342.
72. J. Oliensis and V. Govindu, An experimental study of projective structure from motion, *IEEE Trans. Pattern Anal. Mach. Intell.* **21**, 1999, 665–671.
73. J. Oliensis and J. I. Thomas, Incorporating motion error in multi-frame structure from motion, in *Motion Workshop 1991*, pp. 8–13.
74. L. Quan, A. Heyden, and F. Kahl, Projective reconstruction with missing data, in *CVPR II*, pp. 210–216, 1999.

75. V. S. Ramachandran, Interaction between motion, depth, color and form: The utilitarian theory of perception, in *Vision: Coding and Efficiency* (C. Blakemore, Ed.), Cambridge, Univ. Press, Cambridge, UK, 1990.
76. H. S. Sawhney and R. Kumar, True multi-image alignment and its application to mosaicing and lens distortion correction, *IEEE Trans. Pattern Anal. Mach. Intell.* **21**, 1999, 235–243.
77. H. Sawhney, 3D geometry from planar parallax, in *CVPR 1994*, pp. 929–934.
78. A. Shashua and N. Navab, Relative affine structure: Canonical model for 3D from 2D geometry and applications, *IEEE Trans. Pattern Anal. Mach. Intell.* **18**, 1996, 873–883.
79. A. Shashua and N. Navab, Relative affine structure: Theory and application to 3D reconstruction from perspective views, in *CVPR*, 483–489. 1994.
80. A. Shashua, Trilinearity in visual recognition by alignment, in *ECCV 1994*, Vol. 1, pp. 479–484.
81. A. Shashua and S. Avidan, The rank 4 constraint in multiple ($\geq$3) view geometry, in *ECCV 1996*, pp. 196–206.
82. A. Shashua, Algebraic functions for recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* **17**, 1995, 779–789.
83. S. Soatto and R. Brocket, Optimal structure from motion: Local ambiguities and global estimates, in *CVPR 1998*, pp. 282–288.
84. S. Soatto and R. Brocket, Optimal and suboptimal structure from motion, Harvard University technical report, 1997.
85. S. Soatto and P. Perona, Reducing structure-from-motion a general framework for dynamic vision part 2: Implementation and experimental assessment, *IEEE Trans. Pattern Anal. Mach. Intell.* **20**, 1998, 943–960.
86. S. Soatto and P. Perona, Recursive 3D visual-motion estimation using subspace constraints, *Internat. J. Comput. Vision* **22**, 1997, 235–259.
87. M. Spetsakis, A linear algorithm for point and line-based structure from motion, *Comput. Vision Graph. Image Process.* **56**, 1992, 230–241.
88. M. Spetsakis and Y. Aloimonos, A multi-frame approach to visual motion perception, *Internat. J. Comput. Vision* **6**, 1991, 245–255.
89. M. Spetsakis and Y. Aloimonos, Structure from motion using line correspondences, *Internat. J. Comput. Vision* **4**, 1990, 171–183.
90. M. Spetsakis and Y. Aloimonos, A unified theory of structure from motion, in *IUW 1990*, pp. 271–283.
91. S. Srinivasan, Fast partial search solution to the 3D SFM problem, in *ICCV 1999*, pp. 528–535.
92. G. Stein and A. Shashua, Direct methods for estimation of structure and motion from three views, in *CVPR 1997*, pp. 400–406.
93. P. Sturm, Ph.D. thesis, 1997.
94. P. Sturm, Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction, in *CVPR 1997*, pp. 1100–11105.
95. T. Svoboda and P. Sturm, Badly calibrated camera in ego-motion estimation—Propagation of uncertainty, in *CAIP 1997*, pp. 183–190.
96. P. Sturm and B. Triggs, A factorization based algorithm for multi-image projective structure and motion, in *ECCV 1996*, pp. 709–720.
97. R. Szeliski and S. B. Kang, Shape ambiguities in structure from motion, *IEEE Trans. Pattern Anal. Mach. Intell.* **19**, 1997, 506–512.
98. R. Szeliski and S. B. Kang, Recovering 3D shape and motion from image streams using nonlinear least squares, *J. Visual Commun. Image Represent.* **5**, 1994, 10–28.
99. T. Y. Tian, C. Tomasi, and D. J. Heeger, Comparison of approaches to egomotion computation, in *CVPR 1996*, pp. 315–320.
100. J. I. Thomas and J. Oliensis, Dealing with noise in multi-frame structure from motion, *Comput. Vision Image Understand.* **76**, 1999, 109–124.
101. J. I. Thomas, A. Hanson, and J. Oliensis, Refining 3D reconstructions: A theoretical and experimental study of the effect of cross-correlations, *Comput. Vision Graph. Image Process. Image Understand.* **60**, 1994, 359–370.

102. C. Tomasi and T. Kanade, Shape and motion from image streams under orthography: A factorization method, *Internat. J. Comput. Vision* **9**, 1992, 137–154.

103. P. H. S. Torr, A. Fitzgibbon, and A. Zisserman, The problem of degeneracy in structure and motion recovery from uncalibrated image sequences, *Internat. J. Comput. Vision* **32**, 1999, 27–44.

104. P. H. S. Torr, A. Fitzgibbon, and A. Zisserman, Maintaining multiple motion model hypotheses over many views to recover matching and structure, in *ICCV 1998*, pp. 485–491.

105. B. Triggs, Optimal Estimation of Matching Constraints, in *SMILE 1998*.

106. B. Triggs, Factorization methods for projective structure and motion, in *CVPR 1996*, pp. 845–851.

107. B. Triggs, Matching constraints and the joint image, in *ICCV 1995*, pp. 338–343.

108. T. Vieville and O. Faugeras, The first order expansion of motion equations in the uncalibrated case, *Comput. Vision Image Understand.* **64**, 1996, 128–146.

109. D. Weinshall, M. Werman, and A. Shashua, Duality of multi-point and multi-frame geometry: Fundamental shape matrices and tensors, in *ECCV 1996*, pp. 217–227.

110. J. J. Wu, R. E. Rink, T. M. Caelli, and V. G. Gourishankar, Recovery of the 3-D location and motion of a rigid object through camera image (an extended Kalman filter approach), *Internat. J. Comput. Vision* **2**, 1989, 373–394.

111. Z. Zhang, On the optimization criteria used in two-view motion analysis, in *Trans. Pattern Annal. Mach. Intel.* **20**, 1998, 717–729.

112. Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, A. robust techniques for matching two uncalibrated images through the recovery of the unknown epipolar geometry, *Artif. Intell. J.* **78**, 1995, 87–119.

113. Z. Y. Zhang, Q. T. Luong, and O. Faugeras, Motion of an uncalibrated stereo rig: Self-calibration and metric reconstruction, *IEEE Trans. Robot. Automat.* **12**, 1996, 103–113.

Update

Vision and Image Understanding
e 84, Issue 3, December 2001, Page 407–408

https://doi.org/10.1006/cviu.2001.0914

Computer Vision and Image Understanding **84,** 407–408 (2001)
doi:10.1006/cviu.2001.0914

ERRATUM

Volume **80**, Number 2 (2000), in the article "A Critique of Structure-from-Motion Algorithms," by John Oliensis, pages 172–214 (doi:10.1006/cviu.2000.0869): Several omissions and misprints may confuse the reader. The most significant misprints and omissions are listed here.

On page 175, the citations in the top line should include citation [1]. The following citations in this paragraph should be to [46] (Jepson and Heeger) and [58] (Maybank). Also on page 175, in the third paragraph, footnote 2 is positioned incorrectly in the main text. The text should read: "The mathematical kernels of this 'principle' are the statements that a least-squares reconstruction is the solution to a complex system of polynomial equations and that such systems generically have very complicated dependence on the data.[2]"

On page 177, the paragraph just above the section heading *Mathematics and Phenomenology* (which begins "I believe that phenomenological analysis . . .") is misplaced. It should directly precede the last paragraph on page 175. Thus, page 175 should read, in part:

> The body of this paper proposes other principles on which one could base phenomenological investigations.
>
> I believe that phenomenological analysis is generally useful for vision, and I intend my discussion of SFM reconstruction to illustrate this . . . .

On page 178, the citation for multi-image structure from motion (MISFM) in the last paragraph should be to [88], rather than to [80].

On page 183, in the last complete paragraph the fourth sentence should read: "The image center, the relative scaling of the x and y axes, and/or the focal length is often known approximately."

On page 186, footnote 16 should read: "Typically, one can recover the inverse depths accurately up to an additive constant (due to the bas-relief ambiguity). This gives a one-parameter ambiguity. The projective approach has a larger, four-parameter ambiguity in recovering the inverse depths, since it excludes all components linear in the image coordinates."

On page 189, the two citations in the middle of the first complete paragraph (following "resulting errors") should be [67, 95], rather than [67, 93].

On page 190, the second paragraph should begin: "Also, in [64] I identified . . . ."

On page 191, the fourth bullet was omitted. The omitted paragraph was as follows:

> • Do calibration errors affect strongly just a few parameters of a Euclidean reconstruction? Can one characterize the effects of miscalibration? Given a Euclidean reconstruction obtained with a wrong calibration, can one approximately correct the reconstruction just by plugging the right calibration values into this characterization? How can one design an effective Euclidean approach for dealing with calibration error?

1077-3142/01 $35.00

On page 194, in the first paragraph of the section headed *Optimal fusing revisited,* the first sentence should read: "Instead of fusing $\geq$3-image reconstructions as in [81], one may do better to combine image-pair reconstructions—and this might work better than trilinear reconstruction even on just three images." The third sentence should read: "If one neglects the positive-depth requirement on the 3D points, as is standard, the only constraint on a two-image reconstruction comes from coplanarity [41]; the depths don't matter."

On page 194, in the last paragraph several misprints combine to make the text difficult to understand. This paragraph should read as follows:

> For $\geq$3 images, in addition to coplanarity, the *rigidity* constraint (that the 3D points remain fixed in place over the sequence) determines the reconstruction. To exploit rigidity, an algorithm must represent the structure at least implicitly—in effect, it must impose the requirement that the depths recovered from different image pairs are the same. Thus, an optimization approach must minimize over all the structure/motion unknowns, which is slow.[27] Conversely, a simplified approach that does not explicitly represent the structure is likely to be far from optimal, since the nonlinear transformations involved in representing the structure implicitly tend to introduce strong biases. Thus, if one wants to design an algorithm that does not reconstruct from image pairs, one confronts a dilemma: either one uses optimization and gets a slow algorithm, or else one is likely to get a strongly non-optimal approach which gives poor results on difficult sequences.

On page 198, footnote 31 should read: "For sequences where dense image correspondences are available, the amount of information is so great that a single algorithm (e.g., a standard two-image method) may work in most cases. A variety of algorithms become important for the more difficult cases with relatively few correspondences."

Added note. The text refers to experiments showing that the Sturm/Triggs algorithm tends to fail for forward or backward moving cameras. These experiments have been published in "Fast and Accurate Algorithms for Projective Multi-Image Structure from Motion," J. Oliensis and Y. Gene, *IEEE Transactions in Pattern Analysis and Machine Intelligence.* **PAMI-23**, 546–559, 2001.