KALMANSAC: Robust Filtering by Consensus

Andrea Vedaldi† vedaldi@ucla.edu
Hailin Jin‡ hljin@adobe.com
Paolo Favaro§ paolo.favaro@siemens.com
Stefano Soatto† soatto@ucla.edu

† Computer Science Department, University of California, Los Angeles, CA 90095
‡ Office of Technology, Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110
§ Integrated Data Systems Department, Siemens Corporate Research, Princeton, NJ 08540

Abstract

We propose an algorithm to perform causal inference of the state of a dynamical model when the measurements are corrupted by outliers. While the optimal (maximum-likelihood) solution has doubly exponential complexity due to the combinatorial explosion of possible choices of inliers, we exploit the structure of the problem to design a sampling-based algorithm that has constant complexity. We derive our algorithm from the equations of the optimal filter, which makes our approximation explicit. Our work is motivated by real-time tracking and the estimation of structure from motion (SFM). We test our algorithm for on-line outlier rejection both for tracking and for SFM. We show that our approach can tolerate a large proportion of outliers, whereas previous causal robust statistical inference methods failed with less than half as many. Our work can be thought of as the extension of random sample consensus algorithms to dynamic data, or as the implementation of pseudo-Bayesian filtering algorithms in a sampling framework.

1. Introduction

Structure from motion (SFM) is a mature area of computer vision where significant success has been attained during the last decade: We now have commercial products that can estimate 3-D camera pose and point-wise structure of a scene from a collection of images in a fully automatic fashion. Robust statistical inference plays a crucial role in the practical implementation of most SFM systems, since the correspondence mechanisms are often based on low-level assumptions that are prone to errors. In particular, RANSAC [8], along with its many variants, has become the method of choice, owing to its ability to operate in the presence of a large proportion of "outliers[1]." By contrast, vision has so far failed to materialize as a reliable sensory modality in real-time control applications, where data have to be processed in a causal fashion[2] as part of a closed-loop system. We attribute part of this failure to the lack of availability of suitable robust inference techniques that can be applied in causal data processing (there are a few exceptions, upon which we will comment later.) Note that batch-processing based SFM algorithms, together with the associated techniques for handling outliers, cannot be directly applied on these problems as they introduce destabilizing delays in the feedback loop [4]. On the other hand, existing robust filtering techniques, which we review in Section 1.1, either cannot tolerate a large proportion of outliers, or are not suitable for real-time implementation.

Therefore, we turn our attention to *causal* robust statistical inference. This problem arises when there is some hidden variable of interest that evolves over time, and the observations are either related to the hidden states by a simple statistical model, or they are "outliers." The goal is to infer the hidden variables despite outliers, and to do so causally, i.e. only using data up to the current time. For example, the hidden variable could be ego-motion, and the measurements point correspondences from a low-level mechanism. This is important for the real-time estimation and segmentation of structure and motion in vision-based control and robotics, for instance in tracking, manipulation, navigation, and surveillance. The goal is not just to do it fast, but to do it causally to avoid delays in the loop.

1.1. Related work

Robust statistical inference in a dynamic context is a particular type of non-linear filtering problem. Given a probabilistic description of the uncertainty, as well as of the outlier generation mechanism, one can write the equations that govern the evolution of the conditional density of the hidden states (variables of interest as well as inlier/outlier distribution at time t) given all measurements available up to time t. The optimal filter evolves an estimate of such a density

[1] We will define the notion of outlier properly in Section 2; for now an intuitive acception suffices.

[2] Only data up to the current time can be exploited to perform an estimate of the current state.

starting from some initial condition and is easily derived formally (e.g. [10] p. 174). From such a density, one can then construct some *point estimate*, for instance the maximum likelihood, the maximum a-posteriori, or the least mean square or the median estimate.

In practice, the conditional density can only be integrated numerically except for a handful of cases that do not include ours. In general there are a variety of numerical schemes available, including a plethora of particle evolution schemes [7, 14]. However, ultimately we are not interested in the entire conditional density, but in a point estimate. Since the conditional density can only be approximated, a point estimate computed from it is an approximation too. At that point, we might as well settle for a more efficient scheme that yields an (approximate) point estimate at the outset. There is another more philosophical reason to prefer a point estimator: Often if a problem is well formulated the designers have reasons to believe that what they are looking for is a unique entity (e.g. there is *one* ego-motion, not a distribution of them), and the multi-modality of the conditional density is only due to the output statistics (e.g. outliers). Therefore, estimating the entire conditional density would be an overkill. We make the assumption that the posterior density of the hidden variables would be unimodal if it were not for outliers. We therefore design a point estimator that only attempts to model the evolution of the dominant mode, rather than spreading computational resources by evolving particles at the tails of the distribution.

This work relates to a large body of literature in robust statistics that exploit heuristics related to sample consensus. A prototype algorithm of this class is RANSAC [8], and the many variants that have been proposed to improve its efficiency [5, 6, 19], and robustness [16, 21, 20]. Our work can be interpreted as a way to make RANSAC work in a causal fashion. Our goal is to extend random sample consensus techniques to a dynamic context, so we can use them to handle outliers in real-time applications. We use tracking and SFM as examples, but by no means is our work limited to these applications. We show experiments where up to 85% of the measurements are outliers, where previous robust filters fail, and where a large number of particles would be necessary in order to successfully capture the dominant mode. Although a batch processing of the data would necessarily give better estimates, it would require waiting for all data (or at least for a "large enough" temporal window) to be collected, thereby introducing destabilizing delays in a vision-based control scenario. Our scheme compares favorably with batch RANSAC, in the sense of being in the ballpark in terms of accuracy and robustness, despite only processing data in a causal fashion.

Among other schemes for causal robust inference in the context of computer vision, [18, 17] employ sequential importance sampling techniques [15], which makes real-time operation challenging. Nister [16] provides ways to expedite non-causal sampling schemes so they can be implemented fast enough to be used in real time. However, [16] is limited to processing a window of measurements, with no long-term memory, and introduces an intrinsic delay due to the necessity to collect a buffer of frames in order to run its version of RANSAC.

2. Problem statement

We are interested in some quantity $x_t \in \mathbb{R}^M$ (the "state") that evolves in time but that we cannot measure directly. In the simplest case the time dependency can be described by an ordinary difference equation (ODE), up to some "model uncertainty" v_t that we assume to be well captured by a simple statistical model, say a Gaussian process. In the simplest case the ODE is linear, $x_{t+1} = Ax_t + v_t$, $A \in \mathbb{GL}(M)$, and without loss of generality we can assume the uncertainty to be white and zero-mean[3] $\{v_t\}_{t\in\mathbb{Z}} \sim \mathcal{N}(0, R_v)$. Although we cannot measure x_t, we are given measurements $\{y_t \in \mathbb{R}^N\}_{t\in\mathbb{Z}}$ that come from one of two possible sources: At some time $t \in \mathbb{Z}$, **either** y_t is an instantaneous function of x_t, up to some measurement error that can be described by a simple statistical model, **or** y_t is "completely unrelated" to x_t. In the former case, the simplest instance is a linear model, $y_t = Cx_t + n_t$ where $C \in \mathbb{R}^{N\times M}$ and $\{n_t\}_{t\in\mathbb{Z}} \sim \mathcal{N}(0, R_n)$. In the latter case "completely unrelated" means that there is no statistical dependency between y_t and x_t that is simple enough for us to care to model it explicitly. Instead, in this case we call y_t an *outlier* and we wish to bar it from contributing to the inference of x_t. We do not know a-priori whether y_t is a valid measurement (inlier) or an outlier, so we can model the choice with a stochastic indicator process χ_t: $y_t = \mathrm{diag}(\chi_t)\,(Cx_t + n_t) + (I - \mathrm{diag}(\chi_t))\nu_t$ where diag indicates the diagonal of a matrix, and ν_t is a stochastic process statistically independent of the other variables.

Since we allow for uncertainty in the evolution of x_t and in the measurement process, we have to specify what we mean by inference, which we will do shortly. To that end, we summarize the model that generates the data as follows[4]

$$\begin{cases} x_{t+1} = Ax_t + v_t & v_t \sim \mathcal{N}(0, R_v) \\ \chi_{t+1} = g(\chi_t, \mu_t) & n_t \sim \mathcal{N}(0, R_n) \\ y_t = \mathrm{diag}(\chi_t)\,(Cx_t + n_t) + (I - \mathrm{diag}(\chi_t))\nu_t \end{cases} \tag{1}$$

where the evolution of χ_{t+1} is written formally as a function $g : \{0,1\}^N \times \mathbb{R}^N \longrightarrow \{0,1\}^N$ of some unknown "input"

[3] Lest we can consider the mean to be part of the state and we can prewhiten the filter by simple (linear) projection operators [10]

[4] The generative model can be specified in terms of evolution of the joint density of $\{x_t, y_t\}$, or in terms of the evolution of the generic realization x_t, y_t. We choose the latter because of its simplicity, although the two are entirely equivalent.

μ_t that guarantees that χ_t remains an indicator function. Given measurements ("output" of this model), i.e. realizations of the process $\{y_t\}_{t\in\mathbb{Z}}$, we want to infer the state x_t, by employing only inlier measurements. We are thus in the realm of robust statistical inference, and in particular, since the inference concerns the state of a dynamical model, this problem is known as robust filtering [9]. We indicate with $y_\tau^t \doteq (y_\tau, \dots, y_t)$ a realization of the process $\{y_t\}$ from time τ to t, and we omit the subscript when $\tau = 0$. We denote by $y_t(i)$ the i-th component of the vector y_t.

In the absence of more detailed information on the outlier process ν_t, we will assume that these variables are uniformly distributed and independent, i.e. $p(\nu_t(i)) = 1/\eta$ where η is a nominal spreading value. This simple model has already been proven successful in applications similar to ours [21]. Moreover, we will assume that the processes $\chi^t(i)$ and $\chi^t(j)$ are independent for $i \neq j$. By using these assumptions, we can get the density of $y_t(i)$ conditioned on x_t and χ_{t-1} as

$$p(y_t(i)|x_t, \chi_{t-1}) = p(y_t(i)|x_t, \chi_t(i))P(\chi_t(i) = 1|\chi_{t-1}(i)) + P(\chi_t(i) = 0|\chi_{t-1}(i))/\eta \quad (2)$$

where $p(y_t(i)|x_t, \chi_t(i))$ can be derived from the density of the measurement noise $n_t(i)$ and $P(\chi_{t-1}(i)|\chi_t(i))$ is the transition probability encoded by $g(\cdot, \mu_t)$.

2.1. Optimal filter and its infeasibility

Ideally, given the observation y^t, we would like to obtain the posterior density $p(x_t, \chi_t|y^t)$. This problem has a well known solution in term of a recursive filter: The evolution of the conditional density is immediate to derive using Bayes' rule and Chapman-Kolmogorov's equations ((6.61) [10]):

$$\begin{cases} p(x_{t+1}, \chi_{t+1}|y^{t+1}) \propto p(y_{t+1}|x_{t+1}, \chi_{t+1}) \cdot \\ \qquad \cdot \int p(x_{t+1}, \chi_{t+1}|x_t, \chi_t) dP(x_t, \chi_t|y^t), \\ p(x_0, \chi_0|y^0) \sim p_0, \end{cases} \quad (3)$$

where $p_0 \sim p(x_0, \chi_0|\emptyset)$ is an initial estimate. The integrand $p(x_{t+1}, \chi_{t+1}|x_t, \chi_t)$ can be factored as:

$$p(x_{t+1}, \chi_{t+1}|x_t, \chi_t) = p(x_{t+1}|x_t)p(\chi_{t+1}|\chi_t) \quad (4)$$

under the assumptions made in the previous section, and the transition probability $p(\chi_{t+1}|\chi_t) \doteq \pi_{ij}$ can be further specified as part of the model (1). Unfortunately, computing the integral above in the most general case is out of the question because $p(x_t, \chi_t|y^t)$ depends on the entire history χ^t of χ_t, via y^t, and is therefore a mixture of Gaussians with an exponential number of modes in both time t and the number of observation N. This is where we need to introduce approximations, and several options are available, from "sum-of-Gaussian" filters [1] to "interactive multiple models" [3], to various forms of generalized pseudo-Bayesian filters (e.g. GPB1, [3] p. 451). Each of these filters is based on a different heuristic, and it is impossible to prove general properties or approximation bounds except for special cases. These filters arose in radar signal processing, where switches occur rarely between a small number of models (targets), and are not well suited to our application where the set of possible subsets of inliers is large.

2.2. Naive sampling of filters

By conditioning on *any* choice of putative inliers $\tilde{\chi}^t$, assuming Gaussian priors one could easily (i.e. linearly) solve equation (3) using a Kalman filter. In fact, given $\tilde{\chi}^t$, the posterior density $p(x_{t+1}|y^t, \tilde{\chi}^t)$ is Gaussian. This suggests substituting $p(x_t, \chi_t|y^t)$ with $p(x_t|\hat{\chi}_t, y^t)$ as an estimate of the continuous state x_t given a point estimate of the discrete state $\hat{\chi}^t$. A natural criterion for the choice of $\hat{\chi}^t$ is the one that maximizes the posterior density

$$\hat{x}_t, \hat{\chi}^t \doteq \arg\max_{x_t, \chi_t} p(x_t, \chi^t|y^t). \quad (5)$$

Note that the function $p(x_t, \chi^t|y^t)$ is equivalent to $p(x_t|y^t, \chi^t)p(\chi^t|y^t)$. Once some χ^t is given, say $\chi^t = \tilde{\chi}^t$, our model (1) becomes linear, and the maximum a-posteriori (MAP) estimator $\hat{x}_t(\tilde{\chi}^t) = \arg\max_{x_t} p(x_t, \tilde{\chi}^t|y^t)$ is computed by the Kalman filter [10]. Unfortunately, maximizing in the whole history χ^t is still a doubly exponential problem. The complexity can be reduced drastically by assuming χ_t to be constant $\chi_0 = ... = \chi_t$, which is acceptable as long as the observations tend to preserve their inlier/outlier status. Still the possible assignments of χ_t are exponential in the number of observations. This can be addressed by sampling randomly the solution space, leading to the procedure summarized in Algorithm 1.

The three major shortcomings of Algorithm 1 are (i) the constancy of $\chi_0, ..., \chi_t$, too stringent for large intervals; (ii) the naive assignments for χ_t that, although provably correct in the limit, is too slow in practice; (iii) the complexity that grows linearly with t. These issues will be addressed by the KALMANSAC procedure described in the next section.

3. KALMANSAC

In this section, we present an approximate solution that improves Algorithm 1 by (i) letting χ_t change over time, (ii) using an efficient sampling scheme and (iii) limiting the computational complexity to be constant for all times t.

We start from equation (3) and make our assumption explicit. We are going to assume that, at every instant of time, the best estimate of the inliers $\hat{\chi}_t$ is available, as part of the solution of $\hat{x}_t, \hat{\chi}_t = \arg\max p(x_t, \chi_t|y^t)$. Compounding these choices over time we get the best causal estimate $\hat{\chi}^t$ up to time t. We re-write $p(x_t, \chi_t|y^t) \propto p(x_t|\chi_t, y^t)p(\chi_t|y^t)$.

Algorithm 1 Naive sampling of filters.

1: **Initialization:** Randomly choose a set of assignments $\Upsilon \subset \{0,1\}^N$ of inliers/outliers among the measurements.
2: **for all** assignments $\chi^\star$ in Υ **do**
3: Fix $\hat{\chi}^t$ to $\hat{\chi}_0 = ... = \hat{\chi}_t = \chi^\star$.
4: Estimate $\hat{x}_t$ using the Kalman filter.
5: Compute the MAP *score* of the assignment $\chi^\star$ as $p(y^t, \hat{x}_t, \hat{\chi}^t)$ which is proportional to $p(\hat{x}_t, \chi^t | y^t)$.
6: **end for**
7: **Validation:** Choose $\hat{x}_t$ and $\chi^\star$ that yield the maximum score.

Now we assume that

$$p(\chi_t | y^t) = \delta(\chi_t - \hat{\chi}_t), \tag{6}$$

and equation (3) reduces to

$$p(x_{t+1}, \chi_{t+1} | y^{t+1}) \propto p(y_{t+1} | x_{t+1}, \chi_{t+1}) p(\chi_{t+1} | \hat{\chi}_t) \cdot \cdot \int p(x_{t+1} | x_t) dP(x_t | \hat{\chi}_t, y^t). \tag{7}$$

From equation (7), it is easy to check (recursively) that $p(x_t | \hat{\chi}_t, y^t)$ remains Gaussian.

The maximization of equation (7) jointly in x_{t+1} and χ_{t+1} is still problematic because of the exponential number of possible assignments of χ_{t+1}. In the next section we will show an efficient sampling scheme that solves this problem.

3.1. Searching for inliers

We use equation (7) as the basis for our sampling filter. We start with an initial choice of inliers $\hat{\chi}_0$ and with an initial density $\hat{p}_0 \sim p(x_0 | \emptyset)$. At a generic time t, we assume we are given $\hat{\chi}_t$ and $\hat{p}_t \doteq p(x_t | \hat{\chi}_t, y^t)$. We now need to compute the new MAP estimate $\hat{\chi}_{t+1}, \hat{x}_{t+1} \doteq \arg\max p(x_{t+1}, \chi_{t+1} | y^{t+1})$, or, more in general, $\hat{\chi}_{t+1}$ and the entire mode $\hat{p}_{t+1} = p(x_{t+1} | \hat{\chi}_{t+1}, y^{t+1})$ since that will come for free from the Kalman filter and we will need it at the next step $t+1$.

Note that Equation (7) is easy to optimize separately in one of the variables χ_t or x_t given the other. Thus we can use a simple alternating maximization procedure as long as we can get a good *initial estimate* of either variables. This can be done efficiently by RANSAC sampling, as we explain next and is detailed in Algorithm 2.

The best estimate $\hat{x}_t$ is obtained when χ_t is set to its best assignment $\hat{\chi}_t$. However, there are many other assignments that result in good estimates of x_t, as long as they contain a subset of the inliers and none of the outliers. Thus the idea

Algorithm 2 KALMANSAC.

1: **Initialization:** Given the best choice of inliers up to time t, $\hat{\chi}_t$, and the current best estimate of the state conditional density $p(x_t | \hat{\chi}_t, y^t)$, extract a subset $\Upsilon \subset \{0,1\}^N$ of the *minimal set* of assignments of inliers/outliers among the measurements.
2: **for all** $\chi^\star_{t+1}$ in the set Υ **do**
3: Initialize χ_{t+1} with $\chi^\star_{t+1}$.
4: **repeat** {Alternating maximization in χ_{t+1} and x_{t+1}}
5: Given χ_{t+1} compute $\arg\max_{x_{t+1}} p(x_{t+1} | \chi_{t+1}, y^{t+1})$ by reading off the updated state $\hat{x}(\chi_{t+1})$ from one step of the Kalman filter.
6: Given $x_{t+1} = \hat{x}_{t+1}$ compute $\hat{\chi}_{t+1} = \arg\max_{\chi_{t+1}} p(x_{t+1}, \chi_{t+1} | y^{t+1})$
7: Set $\chi_{t+1} = \hat{\chi}_{t+1}$
8: **until** maximum amount of iterations has been reached or until the estimated state $\hat{x}_{t+1}$ and estimated indicator $\hat{\chi}_{t+1}$ do not change.
9: **end for**
10: **Validation** Select $\hat{x}_{t+1}$ and $\hat{\chi}_{t+1}$ that yield the maximum score $p(y^{t+1}, x_{t+1}, \chi_{t+1})$.

is to chose the initial estimate of χ_t by guessing a *minimal*[5] set of inliers (a random sample). Then the algorithm estimates the continuous state $\hat{x}_t$ and then re-estimates $\hat{\chi}_t$ by optimizing the posterior (consensus). This interpretation of RANSAC as an alternate maximization is not new [21].

One limitation of the KALMANSAC algorithm is that it relies completely on the previous estimate $\hat{\chi}_t$ (because of the assumption (6)). Since the recovered $\hat{\chi}_t$ is obtained from an approximate solution, this might prevent the algorithm from converging to the optimal solution. However, by relaxing the assumption on $p(\chi_t | y^t)$, we face a problem of exponential complexity as we have seen in Section 2.1. We address this problem by using a *limited memory filter*, which trades off optimality for constant complexity, as we discuss in the next section.

3.2. Back-tracing: limited memory filter

The algorithm presented in Section 3.1 provides an approximation of the MAP estimate of x_t together with an approximation of its covariance (from the Kalman filter) following a myopic estimate of the set of inliers χ_t based only on the current observations. Our main observation is that the best estimate $\hat{x}_t$ available at time t may be affected by the approximations of our procedure, and therefore at the

[5] The *minimal set* is made of the smallest number of samples that allows to uniquely recover the model parameters. In particular, minimality is related to the smallest number of measurements that make the model *observable* [12].

time step $t+1$ we choose to re-estimate it by using also the new measurements. More in general, we can do so for τ steps back in time. This leads to a limited memory filter ([10], p. 318). The steps of the algorithm are exactly the ones described in the previous section, except that the samples to be drawn are not for χ_{t+1}, but for $\chi_{t-\tau}^{t+1}$ for some $\tau \geq 0$, and the computational step involves a τ-step prediction and update for the process x_t, which are both standard for the Kalman filter, and the Viterbi algorithm for the process χ_t. As for the sampling of χ_t, we exploit the observation of Section 2.2 that the inlier/outlier state tends to be preserved, and we let the samples $\chi_{t-\tau}^{t+1}$ be constant across the time frame and equal to $\hat{\chi}_t$ (line 1 of Algorithm 2). Notice that in the subsequent steps the assignment may change within the time-frame $[t-\tau, t+1]$.

Another benefit of using more steps of back-tracing is that each observation is checked for consistency with the inlier model across consecutive time steps, making the inlier/outlier classification more accurate. This only requires setting the transition probability $P(\chi_t|\chi_{t-1})$ to a function that penalizes switches from inliers to outliers and vice-versa. In the experiments we have chosen $\tau = 2$ as a compromise between accuracy and efficiency.

3.3. Extension to non-linear models

Equation (7) is valid for models far more general than (1). The advantage of a linear model is to allow computing (7) using linear operations to evolve conditional mean and covariance. However, the sampling schemes proposed in Sections 3.1 and 3.2 are valid for any type of model, provided one has at least an approximate procedure to integrate (7). These include various types of approximations, from the extended Kalman filter (EKF, [10] p. 332) to numerical integration, even to particle filters. In the experimental section we will illustrate the performance of our sampling scheme with an EKF used to estimate ego-motion.

3.4. Accelerating convergence

The number of samples required to guess a minimal assignment of inliers can still be large for large amounts of outliers [21], but can be reduced by employing a smarter sampling strategy. We propose a method similar in spirit to [19, 6], that extracts the same set of M samples that standard RANSAC would choose, but taking first the ones that contains observations likely to be inliers. In this way, regardless for the order of samples, the algorithm is no worse than RANSAC [6].

As in [6], we sort the samples in decreasing likeliness of being inliers by exploiting the prediction of the filter. Let $(\sigma_1, ..., \sigma_N) \in S(N)$ be such an ordering, being $S(N)$ the symmetric group of order N. Consider the function $f(j) = \sum_{i=1}^{j} \chi_t(\sigma_j),\ j = 1, ..., N$ (see Figure 1) which

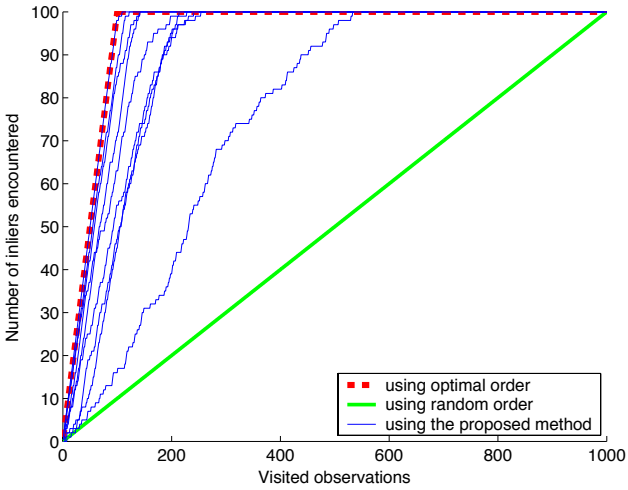

Figure 1. Accelerating the convergence. The figure shows the function $f(j),\ j = 1, ..., N$ (see text) for various orderings: (dashed curve) ideal ordering with all inliers first; (dotted curve) random order; (solid curves) order obtained by the singleton inlier probability for the first 10 iterations of the filter. This figure is obtained based on the tracking problem with 1000 features and 85% outliers (see Section 4).

counts how many inliers are encountered by visiting the observations in the specified order. The best possible order goes first through all inliers, then through all outliers. Thus, we formulate the choice of the ordering as the N optimization problems (for $j = 1, ..., N$)

$$\max_{\sigma_1,...,\sigma_j} E[f(j)|y_t, \hat{\chi}^{t-1}] = \max_{\sigma_1,...,\sigma_j} \sum_{i=1}^{j} S_t(\sigma_j), \quad (8)$$

where N is the number of observations and $S_t(i) = P[\chi_t(i)|y^t, \hat{\chi}^{t-1}]$ is the *singleton inlier probability*. The N problems are solved *simultaneously* by simply ordering the observations by decreasing $S_t(i)$, which can be computed as

$$S_t(i) \propto P(\chi_t(i)|\hat{\chi}_{t-1}) \times \int p(y_t|\chi_t(i), \hat{\chi}_{t-1}, x_t) p(x_t|y^{t-1}, \hat{\chi}^{t-1})\, dx_t. \quad (9)$$

Although this integral is computationally expensive, it can be approximated by setting $p(x_t|y^t, \hat{\chi}^{t-1}) = \delta(x_t - x'_t)$, x'_t being the prediction of the filter at time t. This results in the score $\tilde{S}_t(i) \propto P(\chi_t(i)|\hat{\chi}_{t-1}) p(y_t|\chi_t^j, \hat{\chi}_{t-1}, x'_t)$, which is very quick to compute and in practice yields excellent ordering results, as one can see in Figure 1.

4. Experiments

In this section we illustrate the features of our algorithm on two examples: a 2-D tracker, where the model used is a

second-order random walk that fits equation (1), and structure from motion, where we adopt a model borrowed from [2, 4], which is non-linear. In the latter case, the computation of the MAP densities are approximated by an extended Kalman filter. In order to perform systematic and controlled tests, we employ synthetic data in the former case. For the more complex SFM case, we show results on both synthetic and real image sequences, although performance is best tested on simulations where the parameters of the experiment, which include a large variety of factors depending on the applications, can be carefully controlled.

4.1. Tracking

In this first set of experiments we choose to test the KALMANSAC on a simple 2-D object tracker. We consider tracking a group of 2-D points $y^i \in \mathbb{R}^2$, $i = 1, \ldots, N$ that evolve in time according to similarity transformations:

$$y_t^i = s_t R_t (y_0^i + T_t) \quad i = 1, 2, \ldots, N \tag{10}$$

where $s_t \in \mathbb{R}$ is the isotropic scaling, $R_t \in SO(2)$ is the 2-D rotation and $T_t \in \mathbb{R}^2$ is the translation. Rather than representing the state as the vector containing s, R and T, we use an equivalent alternative representation that yields a linear system of equations. We define two variables $a_t \in \mathbb{R}^2$ and $b_t \in \mathbb{R}^2$, such that

$$a_t = s_t R_t T_t \quad \text{and} \quad b_t = s_t R_t \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right] + a_t. \tag{11}$$

Then, it is immediate to obtain the expressions of s_t, R_t and T_t as a function of a_t and b_t. By substituting these expressions into equation (10) we obtain that

$$y_t^i = (I - Q(y_0^i))a_t + Q(y_0^i)b_t \quad i = 1, 2, \ldots, N \tag{12}$$

where $Q(y_0^i) = [y_0^i \ \ (y_0^i)^\perp]$. Now, let $P_t = [a_t^T \ \ b_t^T]^T \in \mathbb{R}^4$. We then choose a second-order random walk as a model for the dynamics, i.e.

$$P_{t+1} = P_t + V_t \quad \text{and} \quad V_{t+1} = V_t + n_t \tag{13}$$

where $n_t \sim \mathcal{N}(0, R)$. As a consequence, the dynamical system corresponding to equation (1) is defined as follows:

$$\begin{cases} A = \left[\begin{smallmatrix} I & I \\ 0 & I \end{smallmatrix}\right] \in \mathbb{R}^{8\times 8} \\ C = [C_1^T, \ldots, C_N^T]^T \\ \qquad \text{where } C_i \doteq \left[\begin{smallmatrix} I - Q(y^i(0)) & Q(y^i(0)) \\ 0 & 0 \end{smallmatrix}\right] \in \mathbb{R}^{4\times 8} \end{cases} \tag{14}$$

and the state $x_t \doteq [P_t^T \ \ V_t^T]^T \in \mathbb{R}^8$ with initial conditions $x_0 = [0\ 0\ 1\ 0\ 0\ 0\ 0\ 0]^T$.

To evaluate the performance of KALMANSAC on this dynamical system, we compare it with a classic *Kalman filter (KF)*, and with the *Robust KF* [13]. The robust KF is a modification of the Kalman filter that uses a robust model of the observations [9] to obtain an M-estimate of the state rather than the usual MAP-estimate. Contrary to our algorithm, the robust KF is capable of detecting outliers only based on the current estimate of the measurement prediction error, and, because of the weighting scheme used, is not completely resilient to outliers. Moreover, it does not provide an explicit estimation of the process χ_t.

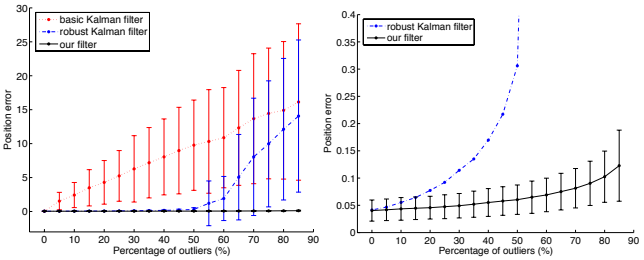

Figure 2. 2-D tracking experiment. Left: performance comparison between the classic Kalman Filter (KF) (dotted), the robust KF (dot-dashed) and the KALMANSAC (solid). As one can see the KALMANSAC can produce very accurate estimates of the state up to 85% outliers, while the robust KF fails at 50% outliers and the classic KF as soon as outliers appear in the measurements. Right: a zoomed-in version of the plot highlights the difference in performance between the robust KF and KALMANSAC.

We choose $N = 100$ and generate 100 sequences of measurements each contaminated with $0\%, 5\%, \ldots, 85\%$ of outliers. On this data we run the classic KF, the robust KF and KALMANSAC. We compute the mean and the standard deviation of the state estimation error over the 100 trials and plot them in Figure 2. Because of the enhanced sampling scheme (Section 3.4), only 20 to 100 samples (depending on the outlier concentration) need to be drawn at each time step. Notice that the classic KF starts to return inconsistent results as soon as some outliers appear in the measurements. The robust KF can instead tolerate up to 50% outliers, but then rapidly degenerates. KALMANSAC proves to be very resistant to outliers, maintaining consistent estimates up to 85% outliers.

4.2. Structure from motion

In this section, we carry out experiments on a non-linear system that is suited to solve structure from motion under the assumption that both translational and rotational accelerations are Brownian motions. As mentioned above, we adopt a model borrowed from [2, 4].

Synthetic data. The synthetic scene is composed of 200 points. The camera rotates around the points, with center of rotation on the center of mass of the structure. We re-scale both translation and structure by fixing the depth coordinate of one of the points to 1. In this experiment, we want to show how the different implementations respond on aver-

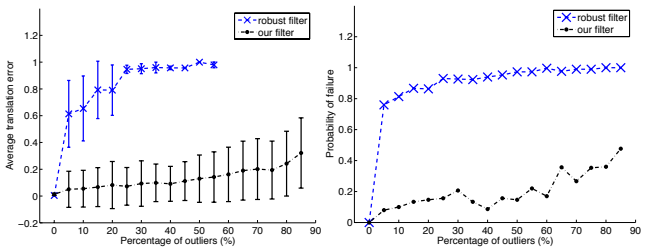

Figure 3. Comparison with synthetic data. Mean and standard deviation error of motion estimation versus increasing proportions of outliers. Left: The robust EKF (blue solid line) diverges immediately as soon as measurements contain 5% of outliers. KALMANSAC (black solid line) returns consistent motion estimates with a very low error even up to 75% outliers. Right: We plot the frequency of failure of both the robust EKF (blue solid line) and KALMANSAC (black solid line). Notice that the robust EKF is confused almost immediately by outliers as opposed to the KALMANSAC that starts to be confused half of the times when outliers are more than 80%.

age to different amounts of outliers. We simulate outliers as 3-D points whose projections follow a second-order random walk (diffusion). We choose $0\%, 5\%, ..., 85\%$ outliers and for each of the filters we run 100 experiments and store the estimated motion. Then, we compute the mean and the standard deviation for each outlier level and for both the robust EKF (an extension of [13] to nonlinear systems) and KALMANSAC and plot the results in Figure 3. As one can see, the robust EKF consistently fails to produce any sensible estimate of motion as soon as some outliers are introduced in the measurements (Figure 3 left). KALMANSAC can produce a sensible estimate of motion up to 75% of outliers (left plot in Figure 3). In the right plot in Figure 3, we show how frequently both filters diverge. We do so because in the case of structure from motion the recovery of motion parameters may be unsuccessful even when there are no outliers. Indeed, the same measurements (up to noise) may be generated by different configurations of points and motion so that the recovery of motion parameters is an ambiguous process. In Figure 3, right, one can see that while the robust EKF fails almost always as soon as we have 5% outliers, KALMANSAC starts to fail half of the times when we have more than 80% outliers.

Real data. In this set of experiments we test the robust EKF and KALMANSAC, on a real sequence (Figure 5) where three independent objects that are moving within a rigid scene. Features are tracked using [11]. These three objects plus additional T-junctions and specular reflections generate more than 60% of outliers in our measurements. Similarly to the case of synthetic data, in Figure 4 we compare the performance of the two filters by comparing the

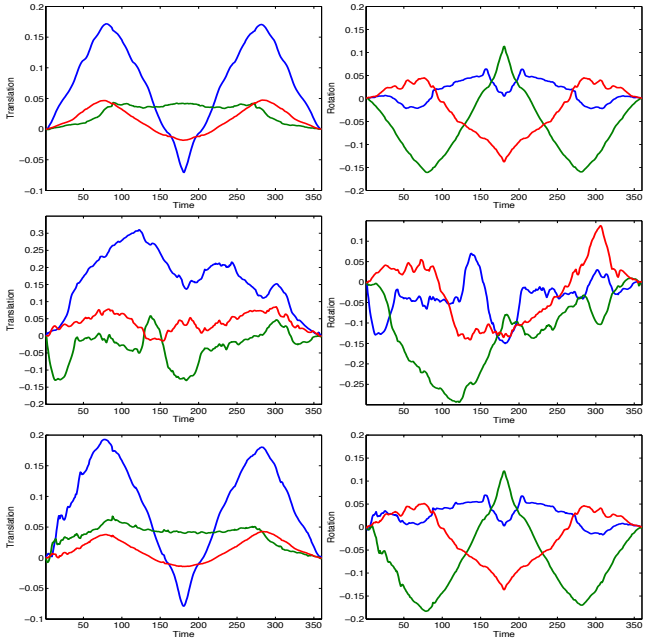

Figure 4. Comparison with real data. We estimated the camera motion playing the sequence back and forth once. The left column shows the translation error and the right column shows the rotation error (expressed in exponential coordinates). Top row: Pseudo-ground truth of the camera motion obtained by manually eliminating the outliers from data. Middle row: camera motion estimated by the robust filter. Bottom row: camera motion estimated by the KALMANSAC algorithm.

estimated motion to motion that has been estimated with a classic EKF by manually discarding the outliers (pseudo-ground truth). As one can see the overall performance reflects the experiments on synthetic data: While the robust EKF fails to recover the motion parameters, KALMANSAC returns an estimate very similar to that based on the pseudo-ground truth.

5. Summary and conclusions

We have presented an algorithm for causal robust statistical inference of the state of a dynamical model. Since the optimal solution is computationally prohibitive, we have proposed a random sampling approach that propagates the best current estimate of the set of inliers $\hat{\chi}_t$, together with the state conditional density $p(x_t|\hat{\chi}^t, y^t)$. We have derived this algorithm from the optimal filter, clearly highlighting the assumptions that underly our approximation. We have validated our scheme experimentally, on both synthetic and real data. We show that our proposed algorithm can operate successfully in the presence of a large proportion of outliers where existing robust filtering schemes fail.

Figure 5. Experiments with real data. Top row: five frames extracted from a real sequence (180 frames in total). The independently moving objects are: The car, the top checkerboard box and the blue box in the top-left corner. The camera is moving sideways while these objects are also moving independently. Bottom row: Corresponding images with tracked features superimposed. The features are marked with either red or green squares. The green squares are the features considered as inliers while the red squares are considered outliers by the KALMANSAC.

Acknowledgements

We wish to acknowledge the support of Adobe Systems Inc., AFOSR F49620-03-1-0095 and ONR N00014-03-1-0850. The work was carried out while Paolo Favaro was at UCLA, has been continued while at Cambridge University, Dept. of Engineering and finalized while at Siemens Corporate Research, Princeton, NJ.

References

[1] D. L. Alspach and H. W. Sorenson. Nonlinear bayesian estimation using gaussian sum approximation. *IEEE Trans. Aut. Contr.*, 17(4):439–448, 1972.

[2] A. Azarbayejani and A. Pentland. Recursive estimation of motion, structure and focal length. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(6):562–575, 1995.

[3] Y. Bar-Shalom and X.-R. Li. *Estimation and tracking: principles, techniques and software*. YBS Publishing, February 1998.

[4] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from motion causally integrated over time. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):523–535, April 2002.

[5] O. Chum and J. Matas. Randomized RANSAC with $t_{(d,d)}$ test. In *British Machine Visison Conference*, pages 448–457, 2002.

[6] O. Chum and J. Matas. Matching with PROSAC – progressive sample consensus. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 2005.

[7] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer Verlag, 2001.

[8] M. A. Fischler and R. C. Bolles. RANSAC, random sampling consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 26:381–395, 1981.

[9] P. J. Huber. *Robust Statistics*. Wiley-Interscience, December 2003.

[10] A. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

[11] H. Jin, P. Favaro, and S. Soatto. Real-time feature tracking and outlier rejection with changes in illumination. In *Proc. Intl. Conf. on Computer Vision*, pages 684–689, July 2001.

[12] T. Kailath. *Linear Systems*. Prentice Hall, 1980.

[13] K. Koch and Y. Yang. Robust Kalman filter for rank deficient observation models. *J. Geodesy*, 72(7-8):436–41, 1998.

[14] J. Liu. *Monte Carlo strategies in scientific computing*. Springer Verlag, 2001.

[15] J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *J. Amer. Statist. Assoc.*, 93:1032–1044, 1998.

[16] D. Nister. Preemptive ransac for live structure and motion estimation. In *Proc. Intl. Conf. on Computer Vision*, pages 199–206, 2003.

[17] G. Qian, R. Chellapa, and Q. Zheng. Bayesian algorithms for simultaneous structure from motion estimation of multiple independently moving objects. 14(1):94–109, January 2005.

[18] G. Qian and R. Chellappa. Structure from motion using sequential Monte Carlo methods. *Intl. J. of Computer Vision*, 59(1):5–31, 2004.

[19] B. Tordoff and D. Murray. Guided sampling and consensus for motion estimation. In *Proc Euro. Conf. on Computer Vision*, pages 82–96, May 2002.

[20] P. H. S. Torr and C. Davidson. IMPSAC: synthesis of importance sampling and random sample consensus. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(3):354–364, March 2003.

[21] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.