Droplet tracking from unsynchronized cameras

L.A. Zarrabeitia, F.Z. Qureshi and D.A. Aruliah
Faculty of Science, University of Ontario Institute of Technology, Oshawa, ON, Canada
{*Luis.Zarrabeitia, Faisal.Qureshi, Dhavide.Aruliah*}*@uoit.ca*

Keywords: Stereo reconstruction, multi-view geometry, nonlinear motion, blood flight, parameter estimation

Abstract: We develop a method for reconstructing the three-dimensional (3D) trajectories of droplets in flight captured by multiple unsynchronized cameras. Triangulation techniques that underpin most stereo reconstruction methods do not apply in this context as the image streams recorded by different cameras are not synchronized. We assume instead *a priori* knowledge about the motion of the droplets to reconstruct their 3D trajectories given unlabelled two-dimensional tracks in videos recorded by different cameras. Our approach also avoids the challenge of accurately matching droplet tracks across multiple video frames. We evaluate the proposed method using both synthetic and real data.

1 Introduction and background

Recovering the three-dimensional structure of a scene requires a system that can match features extracted from at least two distinct images captured by cameras at different spatial locations; once features are matched, reconstruction proceeds using triangulation (Hartley and Zisserman, 2004). A single monocular image is insufficient for scene reconstruction unless some *a priori* information regarding scene geometry is available. When targets in the scene are moving, the videos captured by the cameras must also be synchronized.

The paper (Zarrabeitia et al., 2012) summarizes a method for tracking droplets in flight filmed by a stereo camera pair. The videos consist of experimental simulations of blood-spattering events (a projectile hitting simulated flesh scattering blood droplets through a scene). Following background removal and segmentation of droplets in each video frame, each individual foreground blob is tracked across a sequence of video frames according to a polynomial least-squares model that locally approximates the projection of the global trajectory into each image. This model-based tracking is designed to deal gracefully with the similarity of droplets in shape, size, and color, with the lack of feature points on individual droplets, and, most importantly, with incomplete droplet paths. Even though the cameras are synchronized in this set of experiments, droplet positions are available in every frame due to occlusions, noise, or droplets exiting the view of one—but not both—cameras. In fact, many targets segmented from one camera's video do not correspond to suitable matches in the other's due to false positives and false negatives during background subtraction and 2D tracking.

The 3D trajectories extracted by the methods in (Zarrabeitia et al., 2012) correspond to the subset of 2D image trajectories as reconstructed from each camera's respective videos for which a match could be found. The work in (Murray, 2012) uses those 3D trajectories to estimate physical parameters in an ordinary differential equation (ODE) model associated with each droplet tracked. In the present paper, we demonstrate that the accuracy of ODE-based motion estimation increases significantly with the number of measurements, so it is desirable to maximize the number of available data points. The unmatched points discarded in the reconstruction algorithm of (Zarrabeitia et al., 2012), then, constitute an unexploited potential source of information.

There is a strategy to estimate the 3D world coordinates of a target captured by one camera but not the other. The method of (Park et al., 2010) provides a means to extract motion parameters from a set of monocular images. Unfortunately, this approach relies on approximating the motion model by a linear combination of basis functions specified *a priori*. A nonlinear ODE motion model as in (Murray, 2012) involves the solution of an IVP (initial-value problem); this approach captures motions that cannot be easily modelled by linear models. It does rely on robust IVP solvers which, fortunately, are mature and widely available.

Thus, we propose a method to relax the requirement of synchronized cameras for trajectory reconstruction in scenes with moving targets where nonlinear ODE motion models are appropriate. This approach is necessary in realistic tracking scenarios; for instance, a feature might be occluded (or move out of the frame or be lost due to noise) for a few frames in the view of some cameras but while still being visible to others. Our approach does require an *a priori* model of object motion; in particular, we assume that each object in the scene moves according to an ODE model described by some set of physical parameters. With reasonable estimates of those physical parameters, it is possible to reconstruct the motion of objects not just between successive frames but throughout a longer time interval. Moreover, even when simultaneous measurements are available, if the ODE model faithfully describes the motion, our reconstruction based on estimating the motion parameters can provide better three-dimensional reconstructions than those produced by triangulation alone. We highlight the fact that our technique retrieves the physical motion parameters without explicitly computing the spatial locations of the target. As such, we can avoid bias caused by errors in 3D reconstruction.

The present paper builds on techniques in (Murray, 2012) and in (Park et al., 2010). First, we generalize the extraction of motion parameters from monocular images from the linear to the nonlinear case. Second, we use a nonlinear ODE model to estimate physical motion parameters and the 3D world coordinates of droplets simultaneously. Our generalization loses the elegance and computational efficiency of Park's work but is more appropriate in scenarios where nonlinear ODE models apply. We compare four different motion models: two are ODE-based two are polynomial-based (specifically quadratic polynomials). The accuracy of each model is assessed using synthetic data and real data (as obtained from the video experiments in (Zarrabeitia et al., 2012)).

2 3D reconstruction from synchronized cameras

Let us consider the problem of 3D reconstruction using n synchronized pinhole cameras. Each camera i is described by its 3×4 projection matrix $\mathbf{P}_i$, where $i \in [1,n]$. For each camera we also define a projection operator $\varphi_i : \mathbb{R}^3 \to \mathbb{R}^2$ that maps $\mathbb{R}^3$ to the image plane. Then $\mathbf{P}_i$ and φ_i are related by the following equation:

$$\begin{pmatrix} \varphi_i(\boldsymbol{\lambda}) \\ 1 \end{pmatrix} s = \mathbf{P}_i \begin{pmatrix} \boldsymbol{\lambda} \\ 1 \end{pmatrix}, \tag{1}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^3$ is a 3D point not lying on the plane parallel to the image plane and passing through the center of projection.

Now consider a set of simultaneous measurements $M = \{\langle \mathbf{x}_i, \mathbf{P}_i, t_i \rangle\}$, representing the image coordinates $\mathbf{x}_i$ of an object at location $\boldsymbol{\lambda} = \boldsymbol{\lambda}(t_i)$ when viewed by camera i with projection matrix $\mathbf{P}_i$. Generally speaking, $\mathbf{x}_i$ is a noisy estimate of $\varphi_i(\boldsymbol{\lambda})$ due to sensing limitations of physical cameras, such as occlusions, and inaccuracies present in detection routines. This suggests that the *reconstruction problem* consists of finding the location $\boldsymbol{\lambda}$ that best explains the observations $\mathbf{x}_i$. We can formalize this notion by introducing an error function $e(\boldsymbol{\lambda}, M)$, minimizing which will yield the location $\boldsymbol{\lambda}$ that best explains the observations. One such error function is defined by the Euclidean distance between the observations and the expected locations of the projections given the operators φ_i:

$$e(\boldsymbol{\lambda}, M) = \sum_i \|\mathbf{x}_i - \varphi_i(\boldsymbol{\lambda})\|^2. \tag{2}$$

It is straightforward to minimize the above error function by solving the following *least squares* problem

$$\hat{\boldsymbol{\lambda}} = \min_{\boldsymbol{\lambda}} \sum_i \|\mathbf{x}_i - \varphi_i(\boldsymbol{\lambda})\|^2. \tag{3}$$

It is worth remembering that the error measure (2) is biased if the aspect ratio of a camera is not 1. This, however, can be easily remedied when internal camera parameters are known. The projection operators φ_i are nonlinear (1) and so the least squares problem described by (3) is also nonlinear. We can derive a linear approximation to this least squares problem by observing that vectors

$$\begin{pmatrix} \varphi_i(\boldsymbol{\lambda}) \\ 1 \end{pmatrix} \quad \text{and} \quad \mathbf{P}_i \begin{pmatrix} \boldsymbol{\lambda} \\ 1 \end{pmatrix}$$

differ only by a scaling factor, suggesting that their cross product is 0:

$$\left[\begin{pmatrix} \mathbf{x}_i \\ 1 \end{pmatrix} \right]_\times \mathbf{P}_i \begin{pmatrix} \boldsymbol{\lambda} \\ 1 \end{pmatrix} = 0 \tag{4}$$

$$\left[\begin{pmatrix} \mathbf{x}_i \\ 1 \end{pmatrix} \right]_\times \mathbf{P}_{i,1:3}\, \boldsymbol{\lambda} = -\left[\begin{pmatrix} \mathbf{x}_i \\ 1 \end{pmatrix} \right]_\times \mathbf{P}_{i,4} \tag{5}$$

$$\mathbf{Q}_i \boldsymbol{\lambda} = \mathbf{q}_i, \tag{6}$$

where $[\cdot]_\times$ is the skew-symmetric representation of the cross product (Hartley and Zisserman, 2004). The third equation of (5) is a linear combination of the previous two, so it can be safely discarded. The matrix $\mathbf{Q}_i$ and the vector $\mathbf{q}_i$ in (6) span only the first two equations of (5).

If two or more measurements at time t are available, the 3D location of the object can be retrieved by solving the linear least squares problem

$$(\mathbf{Q}_i \mid_{t_i=t})\, \boldsymbol{\lambda} = (\mathbf{q}_i \mid_{t_i=t}) \tag{7}$$

over the measurements recorded at time t. The solutions of (3) and (7) do not necessarily agree if the measurements are noisy. Specifically, the error associated with (3) is related to the distance between the projections of the guessed 3D location and the observations in the image plane; whereas, the error in (4) depends upon both the sine of the angle between vectors $\begin{pmatrix} \mathbf{x}_i \\ 1 \end{pmatrix}$ and $\mathbf{P}_i \begin{pmatrix} \boldsymbol{\lambda} \\ 1 \end{pmatrix}$ and their lengths.

3 Linear reconstruction of 3D motion from unsynchronized cameras

Equations (3) and (7) require at least two simultaneous measurements of the target in order to reconstruct its 3D location. In a stereo system, for example, at most two measurements are available for a given target. If one of the measurements is unavailable (due to, say, occlusion) or otherwise corrupted (due to a misdetection or misclassification) the 3D location of the target cannot be estimated. In most cases it is not even possible to determine which one of the two observations is corrupted.

If, however, we are given a motion model for a target, it is sometimes possible to estimate the location of the target even when two simultaneous measurements are not available. (Park et al., 2010) explores this idea by assuming that the target's motion can be approximated by a linear combination of basis functions $\Theta = (\theta_1, \ldots, \theta_k)$; in particular,

$$\boldsymbol{\lambda}_t = \boldsymbol{\lambda}(t) \approx \sum_{j=1}^{k} c_j \theta_j(t) = \Theta_t \mathbf{C} \tag{8}$$

for some coefficients $\mathbf{C} = (c_1, \ldots, c_k)^t \in \mathbb{R}^k$. Using the linearization (6), we can then find the parameters $\mathbf{C}$ by solving the linear least squares problem

$$\mathbf{Q}\bar{\Theta}\mathbf{C} = \mathbf{q}, \tag{9}$$

where

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_1 & 0 & \ldots & 0 \\ 0 & \mathbf{Q}_2 & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & \mathbf{Q}_n \end{pmatrix},$$

$$\bar{\Theta} = \begin{pmatrix} \Theta_1 \\ \Theta_2 \\ \vdots \\ \Theta_n \end{pmatrix} \quad \text{and} \quad \mathbf{q} = \begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \vdots \\ \mathbf{q}_n \end{pmatrix}.$$

4 Nonlinear reconstruction of 3D motion from unsynchronized cameras

Equation (9) is easy to set up and fast to solve. However, it is limited to motion models that can be approximated by a linear combination of a finite set of basis functions Θ. If the subspace $\langle\Theta\rangle$ does not contain a reasonable approximation of the motion model for the target in question, solving Equation (9) yields an erroneous reconstruction. Furthermore, the error measure obtained from (7) is generally different from that of (2). These observations suggest, firstly, that it is possible to estimate the 3D trajectory of a target (given its motion model) from multiple unsynchronized multi-view image measurements and, secondly, that we need to minimize the error (2) directly.

Let $\mathscr{L}_{\boldsymbol{\alpha}}(t)$ be a function that describes the location of the target at time t. $\mathscr{L}_{\boldsymbol{\alpha}}(t)$ is parametrized by $\boldsymbol{\alpha} \in \mathbb{R}^k$ and we assume that $\boldsymbol{\alpha}$ is not known. Given the set of observations $M = \{\langle \mathbf{x}_i, \mathbf{P}_i, t_i \rangle\}$ and a set of parameters $\boldsymbol{\alpha}$, the discrepancy between the model $\mathscr{L}_{\boldsymbol{\alpha}}$ and the observations M is given by

$$e_{\mathscr{L}}(\boldsymbol{\alpha}, M) = \sum_{\langle \mathbf{x}_i, \mathbf{P}_i, t_i \rangle \in M} \|\mathbf{x}_i - \varphi_i(\mathscr{L}_{\boldsymbol{\alpha}}(t_i))\|^2. \tag{10}$$

The optimal set of parameters $\hat{\boldsymbol{\alpha}}$ can be found by minimizing the error $e_{\mathscr{L}}(\boldsymbol{\alpha}, M)$:

$$\hat{\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha}} \sum_{\langle \mathbf{x}_i, \mathbf{P}_i, t_i \rangle \in M} \|\mathbf{x}_i - \varphi_i(\mathscr{L}_{\boldsymbol{\alpha}}(t_i))\|^2. \tag{11}$$

Notice that $\mathscr{L}_{\boldsymbol{\alpha}}$ is a generalization of (8) so we can use (11) to find the parameters $\mathbf{C}$ by minimizing the error measure (2). In this case, we can use the solution $\hat{\mathbf{C}}$ of (9) as the initial guess for (11). This strategy can be generalized for any motion model whenever an initial guess $\boldsymbol{\alpha}$ can be inferred from the coefficients of some linear approximation.

5 Synthetic experiment results

We simulate the motion of an spherical object moving under the effects of drag and gravity after an initial impact. This motion is given by the differential equation (Murray, 2012)

$$\dot{\mathbf{v}} = -\frac{3}{8}\frac{\kappa \rho_f \|\mathbf{v}\| \mathbf{v}}{r \rho_o} - \mathbf{g} \tag{12}$$

where κ represents the drag coefficient, r is the radius, ρ_f and ρ_o are respectively the densities of the fluid and the object, g is the gravitational acceleration vector and v is the velocity vector.

Variable	Range	Description
$\mathbf{g}$	$(0, 9.80665, 0)^t$ m/s^2	Acceleration due to gravity
$\mathbf{v}_0$	$1\text{m/s} \leq \|\mathbf{v}_0\| \leq 10\text{m/s}$	Initial velocity vector
r	$1\text{mm} \leq r \leq 4\text{mm}$	Sphere radius
ρ_f	1.1839 kg/m^3	Air density at 25°C
ρ_o	1062 kg/m^3	Density of porcine blood
μ_f	$1.8616 \times 10^{-5}\text{N} \cdot \text{s/m}^2$	Dynamic viscosity of air at 25°C

Table 1: Initial values for the synthetic experiments (Raymond et al., 1996).

The drag coefficient κ is computed from the Reynolds number Re, which depends on the velocity $\mathbf{v}$, the radius r, the density of the fluid ρ_f and the viscosity of the fluid μ_f (Murray, 2012; Aggarwal and Peng, 1995; Liu et al., 1993; Peng and Aggarwal, 1996):

$$\kappa = \begin{cases} \frac{24}{\text{Re}}(1 + \frac{1}{6}\text{Re}^{\frac{2}{3}}) & \text{if Re} \leq 1000 \\ 0.424 & \text{otherwise} \end{cases} \quad (13)$$

$$\text{Re} = \frac{2\rho_f \|\mathbf{v}\| r}{\mu_f} \quad (14)$$

A synthetic experiment consists of generating a trajectory using equation (12) for some randomly selected initial values of position, velocity and radius and project them into two pre-selected cameras. The values are chosen according to Table 1.

5.1 Simplified flight models

We attempt to find the flight parameters corresponding to each of the projected trajectories under different flight assumptions. We consider four flight models:

1. *Murray's model.* $\mathscr{L}_{\boldsymbol{\alpha}}$ corresponds to the solution of equation (12), $\boldsymbol{\alpha} = (\boldsymbol{\lambda}_0, \mathbf{v}_0, r) \in \mathbb{R}^7$.
2. *Quadratic drag.* From the well-known quadratic drag equation and Newton's second law of motion, we can derive:

   $$\dot{\mathbf{v}} = -K\|\mathbf{v}\|\mathbf{v} - \mathbf{g},\ \boldsymbol{\alpha} = (\boldsymbol{\lambda}_0, \mathbf{v}_0, K) \in \mathbb{R}^7 \quad (15)$$

   We can also derive this model from equation (12) by assuming a constant drag coefficient κ.
3. *No drag*. If we ignore the effects of drag and only consider gravity, the motion of the a droplet is described by:

   $$\mathscr{L}_{\boldsymbol{\alpha}}(t) = \boldsymbol{\lambda}_0 + \mathbf{v}_0 t + \frac{1}{2}\mathbf{g}t^2,\ \boldsymbol{\alpha} = (\boldsymbol{\lambda}_0, \mathbf{v}_0) \in \mathbb{R}^6 \quad (16)$$

4. *Quadratic polynomial*. A trivial generalization of the *no drag* model is to approximate the flight with a quadratic polynomial. This model corresponds to the unrealistic scenario of the drag force being unknown but constant, in both direction and magnitude:

   $$\mathscr{L}_{\boldsymbol{\alpha}}(t) = \boldsymbol{\lambda}_0 + \mathbf{v}_0 t + \mathbf{a}t^2,\ \boldsymbol{\alpha} = (\boldsymbol{\lambda}_0, \mathbf{v}_0, \mathbf{a}) \in \mathbb{R}^9 \quad (17)$$

Murray's model has 7 degrees of freedom, corresponding to the initial position $\boldsymbol{\lambda}_0 \in \mathbb{R}^3$, velocity $\mathbf{v}_0 \in \mathbb{R}^3$ and radius $r \in \mathbb{R}^+$. Similarly, the quadratic drag model also has 7 degrees of freedom, counting the constant drag coefficient K. The two polynomial models have respectively 6 and 9 degrees of freedom, corresponding to the non-fixed coefficients of the polynomials ($\boldsymbol{\lambda}_0, \mathbf{v}_0, \mathbf{a} \in \mathbb{R}^3$).

Our goal is to evaluate whether equation (11) is able to retrieve the flight parameters of each synthetic experiment, or at least a set of parameters that explain the observations from the synthetic cameras, as well as to evaluate the ability of the simplified flight models to explain the observations.

For each experiment, we solve (11) using the *Levenberg-Marquardt* algorithm. The IVPs (12) and (15) are solved numerically using the LSODA routine, which features support for stiff problems such as these. The initial guesses for $\boldsymbol{\lambda}_0$ and $\mathbf{v}_0$ are obtained by solving the linear problem (9) with the set of basis functions Θ corresponding to the *quadratic polynomial* flight model:

$$\Theta_t = \begin{pmatrix} t^2 & 0 & 0 & t & 0 & 0 & 1 & 0 & 0 \\ 0 & t^2 & 0 & 0 & t & 0 & 0 & 1 & 0 \\ 0 & 0 & t^2 & 0 & 0 & t & 0 & 0 & 1 \end{pmatrix}$$

5.2 Parameter retrieval

The first synthetic experiment attempts to establish the accuracy of the parameter retrieval using equation (11). To this end, we generate a set of trajectories using some random values of initial position, velocity and radius. These trajectories are then projected into two virtual cameras to obtain one set of measurements M per trajectory. To simulate misdetections and non-synchronization, some measurements are randomly eliminated. We then proceed to solve the optimization problem (11) and compare the resulting parameters with the known ground truth.

Table 2 shows the result with 100 trajectories, each evolved for 0.5 seconds at a simulated framerate of 1300 frames per second. Each row summarizes the results after randomly eliminating some of the measurements. For each row, we list the probability of not removing a measurement, the average number of measurements per path, the mean error in the estimation of the initial location $\boldsymbol{\lambda}_0$, velocity $\mathbf{v}_0$ and radius r, and the mean distance between the reconstruction and

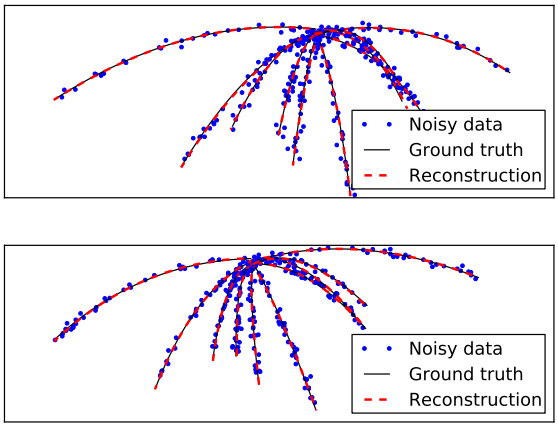

Figure 1: Reprojection into the two virtual cameras of some of the trajectories used in Table 3. The dotted line represents the noisy data, the solid line is the ground truth, and the dashed line shows the trajectory reconstructed from the noisy data.

the ground truth trajectory. In Table 3, normal noise with $\sigma = 5$ was added to the measurements. Figure 1 shows the reconstruction for some of the noisy trajectories from 5% of the measurements. Even with so few points, the reconstructed trajectory is remarkably similar to the ground truth functions.

5.3 Trajectory reconstruction and extrapolation

In (Zarrabeitia et al., 2012), a $2D$ quadratic polynomial approximation was employed to track blood droplets in the image plane. It was found that the quadratic model was able to accurately predict the expected locations of the droplets within a very small time window. It was also noticed that the quadratic predictions broke down quickly after a few frames without a measurement, or if too many measurements were used to estimate the model.

Table 4 compares the accuracy of the reconstruction if only a small window of time is available. The accuracy is measured as the average deviation from the ground truth for the entire simulation time. This allows us to evaluate the predictive capacity of each simplified flight model, that is, how precisely each model allows us to extrapolate from a small set of measurements. We can see that the ODE models behaved significantly better than *quadratic polynomial* model. The *no drag* model also behaved significantly better than the *quadratic polynomial* model in the noisy case for the first two columns. Figure 2 suggests why this happens: having more degrees of freedom, the *quadratic polynomial* model may overfit the noise in the data, resulting in a very inaccurate extrapolation. This is consistent with the prediction breakdown observed in (Zarrabeitia et al., 2012). Without noise (Figure 3), both ODEs were able to accurately reconstruct the entire trajectory using only the first $0.1s$ of flight. The polynomial models also behaved much better with noiseless data, but they were still significantly worse than the ODEs.

With noise				
time (s)	0.1	0.25	0.5	1.0
Murray's model	0.0041	0.00083	0.00022	9.9e-05
Quad. drag	0.005	0.00096	0.00024	9.9e-05
Quad. polynomial	0.12	0.0075	0.0011	0.00046
No drag	0.0048	0.0027	0.0032	0.0041

Without noise				
time (s)	0.1	0.25	0.5	1.0
Murray's model	1e-05	1e-05	1e-05	1e-05
Quad. drag	7.9e-05	5.6e-05	2.6e-05	1.3e-05
Quad. polynomial	0.00086	0.00071	0.0005	0.00032
No drag	0.0018	0.0019	0.0023	0.0029

Table 4: Prediction accuracy, with noise (top) and without noise (bottom). The values represent the mean distance, in meters, between the ground truth and the estimated model over the entire trajectory, when only measurements corresponding to the first $0.1s, 0.25s, 0.5s$ and $1.0s$ of flight are used.

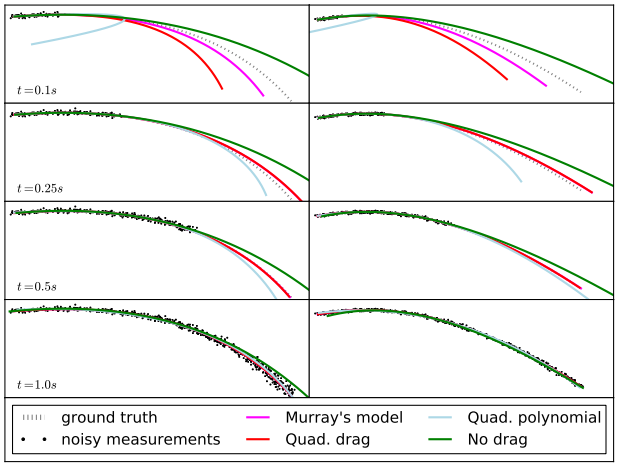

Figure 2: Reconstruction of a trajectory from a small amount of noisy data ($\sigma = 5$). Note how, in the first row, the *quadratic polynomial* model overfits the available data, causing it to behave extremely badly outside of the interval.

6 An application to the blood trajectory reconstruction problem

Our main reason for developing the nonlinear reconstruction method (11) was the study of the motion parameters of blood droplets. Particularly, we wish to construct a framework to validate whether (12) is

Probability of keeping a measurement	Avg. measurements per path	$\boldsymbol{\lambda}_0$ error (cm)	$\mathbf{v}_0$ error, (m/s)	radius error (mm)	Avg. error (m)
0.5	653.16	1.11e-04	4.30e-06	3.45e-06	1.2e-06
0.1	131.08	2.54e-04	1.13e-05	9.85e-05	3.2e-06
0.05	66.15	1.09e-05	1.82e-06	8.81e-05	8.0e-07

Table 2: Accuracy of the estimated parameters from noiseless data, compared with the ground truth. The errors are given in terms of the average distance between the estimated parameters and the true parameters, the average error displays the mean distance between the reconstructed trajectory and the ground truth.

Probability of keeping a measurement	Avg. measurements per path	$\boldsymbol{\lambda}_0$ error (cm)	$\mathbf{v}_0$ error, (m/s)	radius error (mm)	Avg. error (m)
0.5	651.73	4.19e+00	1.71e-01	2.35e-01	5.1e-02
0.1	129.43	8.93e+00	3.67e-01	4.22e-01	1.1e-01
0.05	66.98	1.08e+01	4.09e-01	5.52e-01	1.3e-01

Table 3: Like table 2, but with normal noise of $\sigma = 5$ added to the measurements.

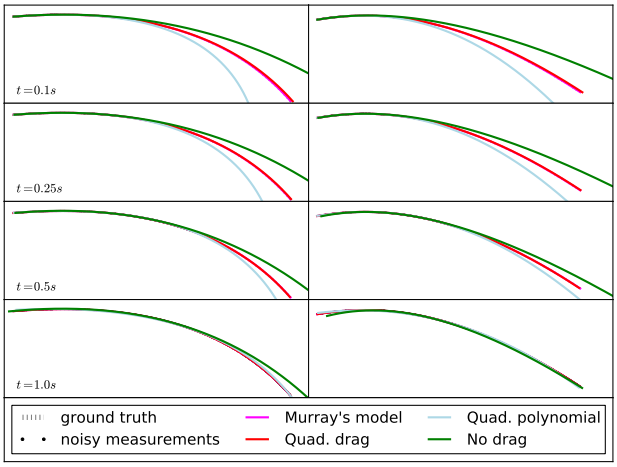

Figure 3: Like Figure 2, but without any noise added to the data.

Method \ Accuracy	0.1	0.25	0.5	0.75	1.0
Triangulation	33.87	34.78	34.39	34.58	34.22
3D optimization	61.63	11.02	3.192	2.154	1.324
Unsynchronized	5.165	2.961	2.015	1.504	1.285

Table 5: Average deviation from the ground truth, in *cm*, using different reconstructions from noisy data, with detection accuracy of 0.1, 0.25, 0.5, 0.75 and 1.0. 50 random trajectories of $1s$ duration with normal noise ($\sigma = 5$) were used.

a good approximation, and whether simpler approximations are suitable under some constraints. An experimental apparatus to measure blood droplet trajectories is described in (Zarrabeitia et al., 2012; Murray, 2012). Furthermore, (Murray, 2012) used these trajectories to develop and validate (12). However, as tables 3 and 4 show, the accuracy of the reconstruction from noisy data depends greatly on the number of measurements available. It is thus essential to maximize the number of useful measurements.

The approach in (Zarrabeitia et al., 2012) is only able to retrieve 3D coordinates for simultaneous measurements. Due to the noise and random misdetections, even perfectly synchronized cameras cannot guarantee finding a pairing for most of the trajectory. Moreover, the sensitivity of the stereo reconstruction with respect to the measurements in the image plane increases with the distance of the target to the camera center. Conversely, this means that there is a region around each reconstructed point that produces very similar measurements, and the size of this region increases with the depth of the target. Fitting a model to these reconstructions ignores that they are not equally accurate.

Table 5 illustrates this issue. The synchronized reconstruction (triangulation) solves the location of each point independently, using equation (7). Observe the very high average deviation from the ground truth. The 3D optimization consists of fitting the model (12) to the synchronized 3D reconstruction. It produces a significant improvement over equation (7), though it still depends heavily on the accuracy of the detector. Finally, the unsynchronized reconstruction proposed in this paper, using *Murray's model* (12) produces a significantly better approximation to the ground truth.

Ground truth is not available with real trajectories, such as those measured by (Zarrabeitia et al., 2012) (Figure 4). This makes it impossible to directly evaluate the reconstruction accuracy. Instead, we can evaluate the prediction accuracy of the models. Figure 5 illustrates the reconstruction of a path from real measurements obtained in (Zarrabeitia et al., 2012). The droplet was recorded for about $0.5s$, one of the longest

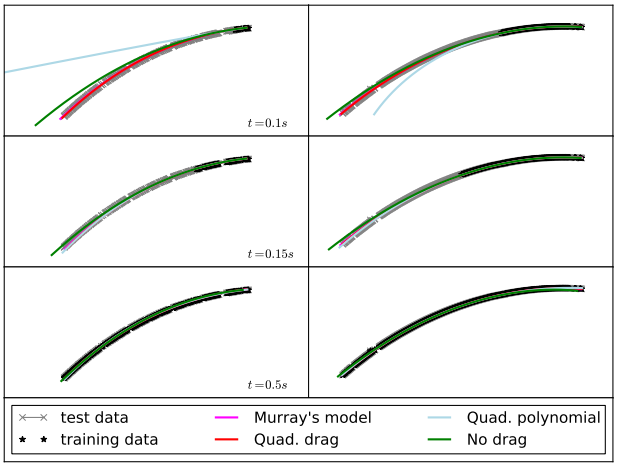

Figure 5: Reconstruction of a real trajectory. Note that, considering only the first $0.1s$ of flight (top), the ODE models were a near perfect fit for the overall trajectory, while the *quadratic polynomial* model behaved badly. Also note that because the noise is significantly lower than that of figure 2, the reconstructions with $t = 0.15$ are nearly indistinguishable.

Model \ time (s)	0.1	0.15	0.5
Murray's model	0.058	0.042	0.016
Quad. drag	0.12	0.048	0.013
Quad. polynomial	0.75	0.2	0.013
No drag	1.4	0.082	0.03

Table 6: Prediction accuracy for real paths. Because ground truth is not available, the errors are given in terms of pixels in the image plane.

trajectories in that experiment. This makes it ideal for evaluating the predictive power of the model, by fitting only a small window of time and comparing the predictions with the actual measurements. The figure shows that even with very few data points, the ODE models were able to accurately predict the entire trajectory.

A more complete test is summarized in Table 6. This test was run over all the trajectories with at least 100 measurements per camera captured during a single experiment. The errors shown are the average of the distances, in pixels, between the predictions and the measurements in the image plane. With incomplete data, the ODE models behaved significantly better than the polynomials.

6.1 A note about outliers

When dealing with a real detector, such as the one described in (Zarrabeitia et al., 2012), outliers are likely to occur. In particular, the parabolic tracker from (Zarrabeitia et al., 2012) tends to produce outliers at the beginning of the trajectories, when there is not enough data to find the quadratic coefficients. Outliers may also appear near the end of the trajectory, when the tracker, unable to find a new measurement, may consume instead some blob near the expected location. Most notably, the this also occurs when the liquid droplet impacts a surface, as the tracker may try to follow the bouncing particles before giving up (Figure 6, top left).

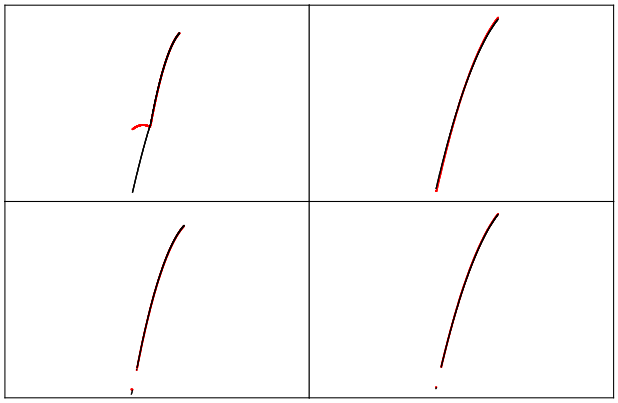

Figure 6: Outlier detection. A first pass fits the ODE parameters using a robustified loss function (top). The outliers are then discarded and a second pass computes the final fit (bottom). The original data is represented in red, the ODE solution is overlaid in black.

Both the linear approach (9) and the nonlinear (11) are sensitive to outliers. In the linear case, they can be handled with *RANSAC*, as we only need to solve a small system of equations for each guess. However, because our use of (9) is limited to computing a rough initial guess for the *Levenberg-Marquardt* iteration, we don't need to filter out the outliers at this stage.

For the nonlinear case, *RANSAC* is prohibitively expensive, as it would require to solve a nonlinear optimization problem with a differential equation as the target function, for each candidate subset. Instead, we follow a 2-pass approach (Figure 6): we use a robust loss function, such as Huber loss, rewriting (11) as

$$\hat{\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha}} \sum_{\langle \mathbf{x}_i, \mathbf{P}_i, t_i \rangle \in M} H_\delta(\mathbf{x}_i - \varphi_i(\mathscr{L}_{\boldsymbol{\alpha}}(t_i))) \quad (18)$$

$$H_\delta(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| < \delta \\ \delta(|x| - \frac{\delta}{2}) & \text{otherwise.} \end{cases} \quad (19)$$

to find the outliers (top), followed by another optimization pass with square loss, considering only the inliers and using the solution of (18) as the initial guess (bottom).

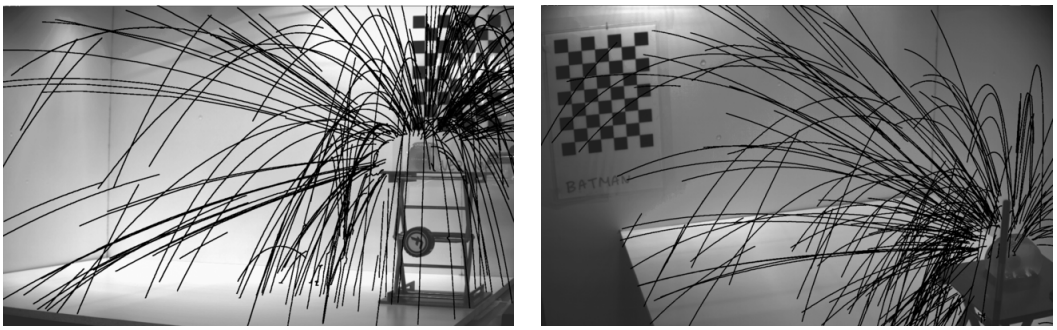

Figure 4: 2D trajectories measured by (Zarrabeitia et al., 2012).

7 Conclusions

We propose a method capable of estimating the 3D path of a particle from a set of unsynchronized multiview image measurements. Our method is able to reliably reconstruct the 3D trajectory of a particle—given a parametrized model $\mathscr{L}_{\boldsymbol{\alpha}}(t)$ describing its motion—even in the presence of occlusions, misdetections and misclassifications. Unlike other attempts that rely upon motion models to improve 3D estimation accuracy, the proposed method makes no linearity assumptions about the motion model.

We applied the proposed method to the problem of estimating the 3D trajectories of blood droplets as they move through the air under the influence of drag and gravity. Physical experiments that were carried out to record these blood droplets are described elsewhere (Zarrabeitia et al., 2012). We also generated synthetic dataset that faithfully mimics those captured from the physical experimental setup mentioned above. The synthetic dataset allowed us to evaluate the proposed approach under controlled settings. We compared the four motion models described in the previous section and we found that the polynomial motion models fair poorly at the task of extrapolation, i.e., estimating the full 3D trajectory of a particle given a small set of initial measurements. ODE based motion models, on the other hand, correctly extrapolated the 3D trajectory of a particle even in the presence of noisy measurements. We achieved similar results for the real dataset.

Our results summarized in Table 5 are both exciting and encouraging. We conclude that 3D reconstruction via triangulation (using synchronized measurements) often exaggerates measurement errors, affecting 3D trajectory estimation. The proposed technique side-steps this issue by solving for both the 3D trajectory of a particle and its motion model concomitantly. We show that the proposed method outperforms the traditional triangulation based 3D trajectory reconstruction approach.

REFERENCES

Aggarwal, S. and Peng, F. (1995). A review of droplet dynamics and vaporization modeling for engineering calculations. *Journal of engineering for gas turbines and power*, 117(3):453–461.

Hartley, R. and Zisserman, A. (2004). *Multiple view geometry in computer vision*. Cambridge University Press, 2nd edition.

Liu, A. B., Mather, D., and Reitz, R. D. (1993). Modeling the effects of drop drag and breakup on fuel sprays. *NASA STI/Recon Technical Report N*, 93:29388.

Murray, R. (2012). Computational and laboratory investigations of a model of blood droplet flight for forensic analysis. Master's thesis, University of Ontario Institute of Technology.

Park, H., Shiratori, T., Matthews, I., and Sheikh, Y. (2010). 3d reconstruction of a moving point from a series of 2d projections. *Computer Vision–ECCV 2010*, pages 158–171.

Peng, F. and Aggarwal, S. K. (1996). Droplet motion under the influence of flow, nonuniformity and relative acceleration. *Atomization and Sprays*, Vol. 6:42–65.

Raymond, M. A., Smith, E. R., and Liesegang, J. (1996). The physical properties of blood—forensic considerations. *Science & Justice: journal of the Forensic Science Society*, 36(3):153–160.

Zarrabeitia, L. A., Aruliah, D. A., and Qureshi, F. Z. (2012). Extraction of blood droplet flight trajectories from videos for forensic analysis. In *ICPRAM 2012 - Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods*, volume 2, pages 142–153. SciTePress.