Fast Rigid Motion Segmentation via Incrementally-Complex Local Models

Fernando Flores-Mangas Allan D. Jepson
Department of Computer Science, University of Toronto
{mangas,jepson}@cs.toronto.edu

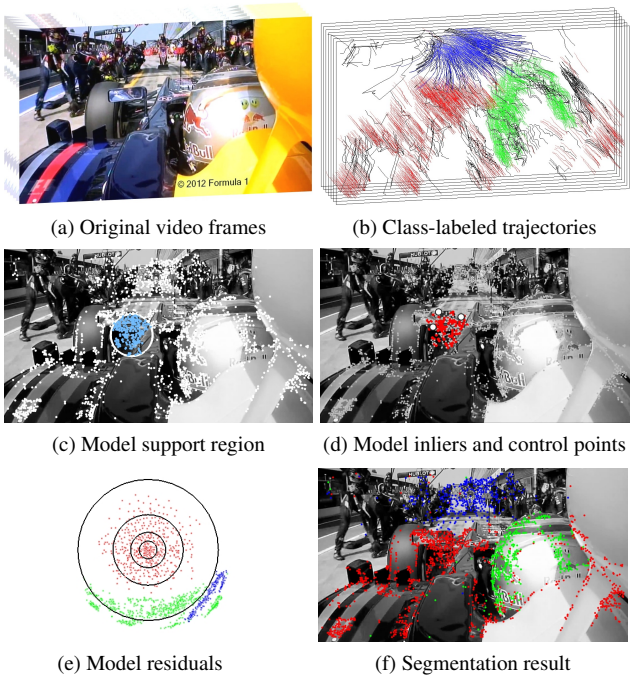
Abstract

The problem of rigid motion segmentation of trajectory data under orthography has been long solved for non-degenerate motions in the absence of noise. But because real trajectory data often incorporates noise, outliers, motion degeneracies and motion dependencies, recently proposed motion segmentation methods resort to non-trivial representations to achieve state of the art segmentation accuracies, at the expense of a large computational cost. This paper proposes a method that dramatically reduces this cost (by two or three orders of magnitude) with minimal accuracy loss (from 98.8% achieved by the state of the art, to 96.2% achieved by our method on the standard Hopkins 155 dataset). Computational efficiency comes from the use of a simple but powerful representation of motion that explicitly incorporates mechanisms to deal with noise, outliers and motion degeneracies. Subsets of motion models with the best balance between prediction accuracy and model complexity are chosen from a pool of candidates, which are then used for segmentation.

1. Rigid Motion Segmentation

Rigid motion segmentation (MS) consists on separating regions, features, or trajectories from a video sequence into spatio-temporally coherent subsets that correspond to independent, rigidly-moving objects in the scene (Figure 1.b or 1.f). The problem currently receives renewed attention, partly because of the extensive amount of video sources and applications that benefit from MS to perform higher level computer vision tasks, but also because the state of the art is reaching functional maturity.

Motion Segmentation methods are widely diverse, but most capture only a small subset of constraints or algebraic properties from those that govern the image formation process of moving objects and their corresponding trajectories, such as the rank limit theorem [9, 10], the linear independence constraint (between trajectories from independent motions) [2, 13], the epipolar constraint [7], and the reduced rank property [11, 15, 13]. Model-selection based methods [11, 6, 8] balance model complexity with modeling accuracy and have been successful at incorporating more of these aspects into a single formulation. For instance, in [8] most model parameters are estimated automatically from the data, including the number of independent motions and their complexity, as well as the segmentation labels (including outliers). However, because of the large number of necessary motion hypotheses that need to be instantiated, as well as the varying and potentially very large number of



(a) Original video frames (b) Class-labeled trajectories

(c) Model support region (d) Model inliers and control points

(e) Model residuals (f) Segmentation result

Figure 1: Model instantiation and segmentation. a) f^{th} original frame, *Italian Grand Prix ©2012 Formula 1*. b) Class-labeled, trajectory data $\mathbf{W}$ (red, green, blue and black correspond to chassis, helmet, background and outlier classes respectively). c) Spatially-local support subset $\hat{\mathbf{W}}^f$ for a candidate motion in blue. d) Candidate motion model inliers in red, control points ($\mathbf{c}_i^f$ from Eq. 3) in white. e) Residuals ($\mathbf{r}_i^f$ from Eq. 11) color-coded with label data, the radial coordinate is logarithmic. f) Segmentation result.

model parameters that must be estimated, the flexibility offered by this method comes at a large computational cost.

Current state of the art methods follow the trend of using sparse low-dimensional subspaces to represent trajectory data. This representation is then fed into a clustering algorithm to obtain a segmentation result. A prime example of this type of method is Sparse Subspace Clustering (SSC) [3] in which each trajectory is represented as a sparse linear combination of a few other basis trajectories. The assumption is that the basis trajectories must belong to the same rigid motion as the reconstructed trajectory (or else, the reconstruction would be impossible). When the assumption is true, the sparse mixing coefficients can be interpreted as the connectivity weights of a graph (or a similarity matrix), which is then (spectral) clustered to obtain a segmentation result. At the time of publication, SSC produced segmentation results three times more accurate than the best predecessor. The practical downside, however, is the inherently large computational cost of finding the optimal sparse representation, which is at least cubic on the number of trajectories.

The work of [14] also falls within the class of subspace separation algorithms. Their approach is based on clustering the principal angles (CPA) of the local subspaces associated to each trajectory and its nearest neighbors. The clustering re-weights a traditional metric of subspace affinity between principal angles. Re-weighted affinities are then used for segmentation. The approach produces segmentation results with accuracies similar to those of SSC, but the computational cost is close to 10 times bigger than SSC's.

In this work we argue that competitive segmentation results are possible using a simple but powerful representation of motion that explicitly incorporates mechanisms to deal with noise, outliers and motion degeneracies. The proposed method is approximately 2 or 3 orders of magnitude faster than [3] and [14] respectively, currently considered the state of the art.

1.1. Affine Motion

Projective Geometry is often used to model the image motion of trajectories from rigid objects between pairs of frames. However, alternative geometric relationships that facilitate parameter computation have also been proven useful for this purpose. For instance, in perspective projection, general image motion from rigid objects can be modeled via the composition of two elements: a 2D *homography*, and parallax residual displacements [5]. The homography describes the motion of an arbitrary plane, and the parallax residuals account for relative depths, that are unaccounted for by the planar surface model.

Under orthography, in contrast, image motion of rigid objects can be modeled via the composition of a 2D *affine* transformation plus epipolar residual displacements. The 2D affine transformation models the motion of an arbitrary plane, and the epipolar residuals account for relative depths. Crucially, these two components can be computed separately and incrementally, which enables an explicit mechanism to deal with motion degeneracy.

In the context of 3D motion, a motion is degenerate when the trajectories originate from a planar (or linear) object, or when neither the camera nor the imaged object exercise all of their degrees of freedom, such as when the object only translates, or when the camera only rotates. These are common situations in real world video sequences. The incremental nature of the decompositions described above, facilitate the transition between degenerate motions and non-degenerate ones.

Planar Model Under orthography, the projection of trajectories from a planar surface can be modeled with the affine transformation:

$$\begin{bmatrix} x^c \\ y^c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{D} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} x^w \\ y^w \\ 1 \end{bmatrix} = \mathbf{A}_{\text{2D}}^{w \to c} \begin{bmatrix} x^w \\ y^w \\ 1 \end{bmatrix}, \quad (1)$$

where $\mathbf{D} \in \mathbb{R}^{2\times 2}$ is an invertible matrix, and $\mathbf{t} \in \mathbb{R}^2$ is a translation vector. Trajectory coordinates (x^w, y^w) are in the plane's reference frame (modulo a 2D affine transformation) and (x^c, y^c) are image coordinates.

Now, let $\mathbf{W} \in \mathbb{R}^{2F\times P}$ be matrix of trajectory data that contains the x and y image coordinates of P feature points tracked through F frames, as in

$$\mathbf{W} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,P} \\ y_{1,1} & \cdots & y_{1,P} \\ \vdots & \ddots & \vdots \\ x_{F,1} & \cdots & x_{F,P} \\ y_{F,1} & \cdots & y_{F,P} \end{bmatrix}. \quad (2)$$

To compute the parameters of $\mathbf{A}_{\text{2D}}$ from trajectory data, let $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3] \in \mathbb{R}^{2f\times 3}$ be three columns (three full trajectories) from $\mathbf{W}$, and let $\mathbf{c}_i^f = [c_i^{2f-1}, c_i^{2f}]^\top$ be the x and y coordinates of the i-th control trajectory at frame f. Then the transformation between points from an arbitrary source frame s to a target frame f can be written as:

$$\begin{bmatrix} \mathbf{c}_1^f & \mathbf{c}_2^f & \mathbf{c}_3^f \\ 1 & 1 & 1 \end{bmatrix} = \mathbf{A}_{\text{2D}}^{s\to f} \begin{bmatrix} \mathbf{c}_1^s & \mathbf{c}_2^s & \mathbf{c}_3^s \\ 1 & 1 & 1 \end{bmatrix}, \quad (3)$$

and $\mathbf{A}_{\text{2D}}^{s\to f}$ can be simply computed as:

$$\mathbf{A}_{\text{2D}}^{s\to f} = \begin{bmatrix} \mathbf{c}_1^f & \mathbf{c}_2^f & \mathbf{c}_3^f \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{c}_1^s & \mathbf{c}_2^s & \mathbf{c}_3^s \\ 1 & 1 & 1 \end{bmatrix}^{-1}. \quad (4)$$

The inverse in the right-hand side matrix of Eq. 4 exists so long as the points $\mathbf{c}_i^s$ are not collinear. For simplicity we refer to $\mathbf{A}_{\text{2D}}^{s\to f}$ as $\mathbf{A}_{\text{2D}}^{f}$ and consequently $\mathbf{A}_{\text{2D}}^{s}$ is the identity matrix.

3D Model In order to upgrade a planar (degenerate) model into a full 3D one, relative depth must be accounted using the epipolar residual displacements. This means extending Eq. 1 with a direction vector, scaled by the corresponding relative depth of each point, as in:

$$\begin{bmatrix} x^c \\ y^c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{D} & \vec{t} \\ \vec{0}^\top & 1 \end{bmatrix} \begin{bmatrix} x^w \\ y^w \\ 1 \end{bmatrix} + \delta z^w \begin{bmatrix} a_{13} \\ a_{23} \\ 0 \end{bmatrix}. \quad (5)$$

The depth δz^w is relative to the arbitrary plane whose motion is modeled by $\mathbf{A}_{2\text{D}}$; a point that lies on such plane would have $\delta z^w = 0$. We call the orthographic version of the plane plus parallax decomposition, the *2D Affine Plus Epipolar* (2DAPE) decomposition.

Eq. 5 is equivalent to

$$\begin{bmatrix} x^c \\ y^c \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_1 \\ a_{21} & a_{22} & a_{23} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^w \\ y^w \\ \delta z^w \\ 1 \end{bmatrix} \quad (6)$$

where it is clear that the parameters of $\mathbf{A}_{3\text{D}}$ define an orthographically projected 3D affine transformation. Determining the motion and structure parameters of a 3D model from point correspondences can be done using the classical matrix factorization approach [10], but besides being sensitive to noise and outliers, the common scenario where the solution becomes degenerate makes the approach difficult to use in real-world applications. Appropriately accommodating and dealing with the degenerate cases is one of the key features of our work.

2. Overview of the Method

The proposed motion segmentation algorithm has three stages. First, a pool of M motion model hypotheses $\mathcal{M} = \{\mathbf{M}_1, \ldots, \mathbf{M}_M\}$ is generated using a method that combines a Random Sampling and Consensus (RANSAC) [4] technique with the 2DAPE decomposition. The goal is to generate one motion model for each of the N independent, rigidly-moving objects in the scene; N is assumed to be known a priori. The method instantiates many more models than those expected necessary ($M \gg N$) in an attempt increase the likelihood of generating correct model proposals for all N motions. A good model accurately describes a large subset of coherently moving trajectories with the smallest number of parameters (§3).

In the second stage, subsets of motion models from $\mathcal{M}$ are combined to explain all the trajectories in the sequence. The problem is framed as an objective function that must be minimized. The objective function is the negative log-likelihood over prediction accuracy, regularized by model complexity (number of model parameters) and modeling overlap (trajectories explained by multiple models). Notice that after this stage, the segmentation that results from the optimal model combination could be reported as a segmentation result (§5).

The third stage incorporates the results from a set of model combinations that are closest to the optimal. Segmentation results are aggregated into an affinity matrix, which is then passed to a spectral clustering algorithm to produce the final segmentation result. This refinement stage generally results in improved accuracy and reduced segmentation variability (§6).

3. Motion Model Instantiation

Each model $\mathbf{M} \in \mathcal{M}$ is instantiated independently using RANSAC. This choice is motivated because of this method's well-known computational efficiency and robustness to outliers, but also because of its ability to incorporate spatially local constraints and (as explained below) because most of the computations necessary to evaluate a planar model can be reused to estimate the likelihoods of a potentially necessary 3D model, yielding significant computational savings.

The input to our model instantiation algorithm is a spatially-local, randomly drawn subset of trajectory data $\hat{\mathbf{W}}_{[2F\times I]} \subseteq \mathbf{W}_{[2F\times P]}$ (§3.1). In turn, at each RANSAC trial, the algorithm draws uniformly distributed, random subsets of three control trajectories ($\mathbf{C}_{[2F\times 3]} \subset \hat{\mathbf{W}}_{[2F\times I]}$). Each set of control trajectories is used to estimate the family of 2D affine transformations $\{\mathbf{A}^1, \ldots, \mathbf{A}^F\}$ between the base frame and all other frames in the sequence, which are then used to determine a complete set of model parameters $\mathbf{M} = \{\mathbf{B}, \boldsymbol{\sigma}, \mathbf{C}, \omega\}$. The matrix $\mathbf{B} \in \{0,1\}^{[F\times I]}$ indicates whether the i-th trajectory should be predicted by model $\mathbf{M}$ at frame f (inlier, $b_i^f = 1$) or not (outlier, $b_i^f = 0$), $\boldsymbol{\sigma} = \{\sigma^1, \ldots, \sigma^F\}$ are estimates of the magnitude of the noise for each frame, and $\omega \in \{2\text{D}, 3\text{D}\}$ is the estimated model type. The goal is to find the control points and the associated parameters that minimize the objective function

$$\mathcal{O}(\hat{\mathbf{W}}, \mathbf{M}) = \sum_{f\in F}\sum_{i\in I} b_i^f L_\omega\left(\hat{\mathbf{w}}_i^f \mid \mathbf{A}^f, \sigma^f\right) + \Psi(\omega) + \Gamma(\mathbf{B}) \quad (7)$$

across a number of RANSAC trials, where $\hat{\mathbf{w}}_i^f = (x_i^f, y_i^f) = (\hat{w}_i^{2f-1}, \hat{w}_i^{2f})$ are the coordinates of the i-th trajectory from the support subset $\hat{\mathbf{W}}$ at frame f. The negative log-likelihood term $L_\omega(\cdot)$ penalizes reconstruction error, while $\Psi(\cdot)$ and $\Gamma(\cdot)$ are regularizers. The three terms are defined below.

Knowing that 2D and 3D affine models have 6 and 8 degrees of freedom respectively, $\Psi(\omega)$ regularizes over model complexity using:

$$\Psi(\omega) = \begin{cases} 6(F-1), & \text{if } \omega = 2\text{D} \\ 8(F-1), & \text{if } \omega = 3\text{D}. \end{cases} \quad (8)$$

$\Gamma(\mathbf{B})$ strongly penalizes models that describe too few trajectories:

$$\Gamma(\mathbf{B}) = \begin{cases} \infty, & \text{if } \sum_I \sum_F b_i^f < F\lambda_i \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

The control set $\mathbf{C}$ whose $\mathbf{M}$ minimizes Eq. 7 across a number of RANSAC trials becomes part of the pool of candidates $\mathcal{M}$.

2D likelihoods. For the planar case ($\omega = 2\text{D}$) the negative log-likelihood term is evaluated with:

$$L_{2\text{D}}(\hat{\mathbf{w}}_i^f \mid \mathbf{A}^f, \sigma^f) = -\log\left(\frac{1}{2\pi|\Sigma|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}\mathbf{r}_i^{f\top}\Sigma^{-1}\mathbf{r}_i^f\right\}\right), \tag{10}$$

which is a zero-mean 2D Normal distribution evaluated at the residuals $\mathbf{r}_i^f$. The spherical covariance matrix is $\Sigma = (\sigma^f)^2\mathbf{I}$. The residuals $\mathbf{r}_i^f$ are determined by the differences between the predictions made by a hypothesized model $\mathbf{A}^f$, and the observations at each frame

$$\begin{bmatrix} \mathbf{r}^f \\ \vec{1} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{w}}^f \\ \vec{1} \end{bmatrix} - \mathbf{A}^f \begin{bmatrix} \tilde{\mathbf{w}}^s \\ \vec{1} \end{bmatrix}. \tag{11}$$

3D likelihoods. The negative log-likelihood term for the 3D case is based on the the 2DAPE decomposition. The 2D affinities $\mathbf{A}^f$ and residuals $\mathbf{r}^f$ are reused, but to account for the effect of relative depth, an epipolar line segment $\mathbf{e}^f$ is robustly fit to the residual data at each frame (please see supplementary material for details on the segment fitting algorithm). The 2DAPE does not constrain relative depths to remain constant across frames, but only requires trajectories to be close to the epipolar line. So, if the unitary vector $\mathbf{e}_\perp^f$ indicates the orthogonal direction to $\mathbf{e}^f$, then the negative log-likelihood term for the 3D case is estimated with:

$$L_{3\text{D}}(\hat{\mathbf{w}}_i^f \mid \mathbf{A}^f, \sigma^f) = -2\log\left(\frac{1}{\sqrt{2\pi}\sigma^f}\exp\left\{-\frac{\left(\mathbf{r}_i^{f\top}\mathbf{e}_\perp^f\right)^2}{2(\sigma^f)^2}\right\}\right), \tag{12}$$

which is also a zero-mean 2D Normal distribution computed as the product of two identical, separable, single-variate, normal distributions, evaluated at the distance from the residual to the epipolar line. The first one corresponds to the actual deviation in the direction of $\mathbf{e}_\perp^f$, which is analytically computed using $\mathbf{r}_i^{f\top}\mathbf{e}_\perp^f$. The second one corresponds to an estimate of the deviation in the perpendicular direction ($\mathbf{e}^f$), which cannot be determined using the 2DAPE decomposition model, but can be approximated to be equal to $\mathbf{r}_i^{f\top}\mathbf{e}_\perp^f$, which is a plausible estimate under the isotropic noise assumption.

Note that Eq. 7 does not evaluate the quality of a model using the number of inliers, as it is typical for RANSAC. Instead, we found that better motion models resulted from optimizing over the accuracy of the model predictions for an (estimated) inlier subset, which also means that the effect of outliers is explicitly uncounted.

Algorithm 1: Motion model instantiation

Input: Trajectory data $\mathbf{W}_{[2F\times P]}$, number of RANSAC trials K, arbitrary base frame b
Output: Parameters of the motion model $\mathbf{M} = \{\mathbf{B}, \boldsymbol{\sigma}_n, \omega\}$

// determine the training set $\hat{\mathbf{W}}$
$c \leftarrow \texttt{rand}(1, P)$; $r \leftarrow \texttt{rand}(r_{min}, r_{max})$ *// random center and radius*
$\hat{\mathbf{W}}_{[2F\times I]} \leftarrow \texttt{trajectoriesWithinDisk}(\mathbf{W}, r, c)$ *// support subset*
$\mathbf{X} \leftarrow \texttt{homoCoords}(\hat{\mathbf{W}}^b)$ *// points at base frame*
for *K RANSAC trials* **do**
  $\mathbf{c} \leftarrow \texttt{rand3}(1, I)$ *// three random control trajectory indices*
  for $f \in \{1, \ldots, F\} - \{b\}$ **do**
    $\mathbf{Y} \leftarrow \texttt{homoCoords}(\hat{\mathbf{W}}^f)$ *// points at frame f*
    $\mathbf{A} \leftarrow \mathbf{Y_c}\mathbf{X_c}^{-1}$ *// 2D affine model*
    $\mathbf{R} \leftarrow \mathbf{AX} - \mathbf{Y}$ *// residuals*
    $[\mathbf{B}_{2\text{D}}^f, \sigma_{2\text{D}}^f] \leftarrow \texttt{compute2DInliers}(\mathbf{R})$
    $\mathbf{L}_{2\text{D}}^f \leftarrow \texttt{compute2DLikelihoods}(\mathbf{R}, \sigma_{2\text{D}}^f)$
    $[U, S, V] = \texttt{svd}(\texttt{weightedCov}(\mathbf{R}, \mathbf{B}_{2\text{D}}))$ *// $s_1 \geq s_2$*
    if $\frac{s_1}{s_2} > 1 + \lambda_{3D}$ **then**
      $[\mathbf{B}_{3\text{D}}^f, \sigma_{3\text{D}}^f] \leftarrow \texttt{compute3DInliers}(\mathbf{R})$
      $\mathbf{L}_{3\text{D}}^f \leftarrow \texttt{compute3DLikelihoods}(\mathbf{R}, \sigma_{3\text{D}}^f)$
  // complete penalized neg-loglikleihoods
  $l_{2\text{D}} \leftarrow \sum_f \sum_i \mathbf{B}_{2\text{D}}(f,i)\mathbf{L}_{2\text{D}}(f,i) + \Psi(2\text{D}) + \Gamma(\mathbf{B}_{2\text{D}})$
  $l_{3\text{D}} \leftarrow \sum_f \sum_i \mathbf{B}_{3\text{D}}(f,i)\mathbf{L}_{3\text{D}}(f,i) + \Psi(3\text{D}) + \Gamma(\mathbf{B}_{3\text{D}})$
  // keep the best model overall
  if $(\min(l_{2D}, l_{3D}) < l^\star)$ **then**
    if $(l_{2D} < l_{3D})$ **then**
      $l^\star \leftarrow l_{2\text{D}}; \boldsymbol{\sigma}_n^\star \leftarrow \boldsymbol{\sigma}_{2\text{D}}; \omega^\star \leftarrow 2; \mathbf{B}^\star \leftarrow \mathbf{B}_{2\text{D}}$
    else
      $l^\star \leftarrow l_{3\text{D}}; \boldsymbol{\sigma}_n^\star \leftarrow \boldsymbol{\sigma}_{3\text{D}}; \omega^\star \leftarrow 3; \mathbf{B}^\star \leftarrow \mathbf{B}_{3\text{D}}$
return $\mathbf{M} = \{\mathbf{B}^\star, \boldsymbol{\sigma}_n^\star, \omega^\star\}$

Figure 1.b shows an example of class-labeled trajectory data, 1.c shows a typical spatially-local support subset. Figures 1.d and 1.e show a model's control points and its corresponding (class-labeled) residuals, respectively. A pseudo-code description of the motion instantiation algorithm is provided in Algorithm 1. Details on how to determine $\hat{\mathbf{W}}$, as well as $\mathbf{B}$, $\boldsymbol{\sigma}$, and ω follow.

3.1. Local Coherence

The subset of trajectories $\hat{\mathbf{W}}$ given to RANSAC to generate a model $\mathbf{M}$ is constrained to a spatially local region. The probability of choosing an uncontaminated set of 3 control trajectories, necessary to compute a 2D affine model, from a dataset with a ratio r of inliers, after k trials is: $p = 1 - (1 - r^3)^k$. This means that the number of trials needed to find a subset of 3 inliers with probability p is

$$k = \frac{\log(1-p)}{\log(1-r^3)}. \tag{13}$$

A common assumption is that trajectories from the same underlying motion are locally coherent. Hence, a compact region is likely to increase r, exponentially reducing

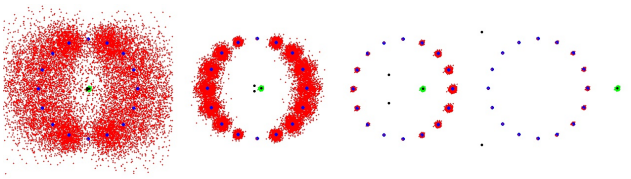


Figure 2: Predictions (red) from a 2D affine model with standard Gaussian noise (green) on one of the control points (black). Noiseless model predictions in blue. All four scenarios have identical noise. The magnitude of the extrapolation error changes with the distance between the control points.

k, and with it, RANSAC's computation time by a proportional amount.

The trade-off that results from drawing model control points from a small region, however, is extrapolation error. A motion model is extrapolated when utilized to make predictions for trajectories outside the region defined by the control points. The magnitude of modeling error depends on the magnitude of the noise affecting the control points, and although hard to characterize in general, extrapolation error can be expected to grow with the distance from the prediction to the control points, and inversely with the distance between the control points themselves. Figure 2 shows a series of synthetic scenarios where one of the control points is affected by zero mean Gaussian noise of small magnitude. Identical noise is added to the same trajectory in all four scenarios. The figure illustrates the relation between the distance between the control points and the magnitude of the extrapolation errors. Our goal is to maximize the region size while limiting the number of outliers.

Without any prior knowledge regarding the scale of the objects in the scene, determining a fixed size for the support region is unlikely to work in general. Instead, the issue is avoided by randomly sampling disk-shaped regions of varying sizes and locations to construct a diverse set of support subsets. Each support subset is then determined by

$$\hat{\mathbf{W}} = \{\mathbf{w}_i \mid (x_i^b - o_x)^2 + (y_i^b - o_y)^2 < r^2\}, \qquad (14)$$

where (o_x, o_y) are the coordinates of the center of a disk of radius r. To promote uniform image coverage, the disk is centered at a randomly chosen trajectory $(o_x, o_y) = (x_i^b, y_i^b)$ with uniformly distributed $i \sim \mathcal{U}(1, P)$ and base frame $b \sim \mathcal{U}(1, F)$. To allow for different region sizes, the radius r is chosen from a uniform distribution $r \sim \mathcal{U}(r_{min}, r_{max})$. If there are I trajectories within the support region, then $\hat{\mathbf{W}} \in \mathbb{R}^{2F \times I}$. It is worth noting that the construction of the support region does not incorporate any knowledge about the motion of objects in the scene, and in consequence $\hat{\mathbf{W}}$ will likely contain trajectories that originate from more than one independently moving object (Figure 3).

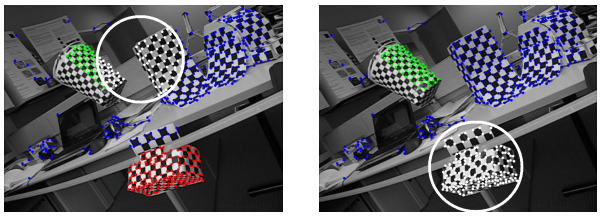


Figure 3: Two randomly drawn local support sets. *Left:* A mixed set with some trajectories from the blue and green classes. *Right:* Another mixed set with all of the trajectories in the red class and some from the blue class.

4. Characterizing the Residual Distribution

At each RANSAC iteration, residuals $\mathbf{r}^f$ are computed using the 2D affine model $\mathbf{A}^f$ that results from the constraints provided by the control trajectories $\mathbf{C}$. Characterizing the distribution of $\mathbf{r}^f$ has three initial goals. The first one is to determine 2D model inliers $\mathbf{b}_{2\text{D}}^f$ (§4.1), the second one is to compute estimates of the magnitude of the noise at every frame $\sigma_{2\text{D}}^f$ (§4.2), and the third one is to determine whether the residual distribution originates from a planar or a 3D object (§4.3). If the object is suspected 3D, then two more goals need to be achieved. The first one is to determine 3D model inliers $\mathbf{b}_{3\text{D}}^f$ (§4.4), and the second one is to estimate the magnitude of the noise ($\sigma_{3\text{D}}^f$) to reflect the use of a 3D model (§4.5).

4.1. 2D Inlier Detection

Suppose the matrix $\hat{\mathbf{W}}$ contains trajectories $\hat{\mathbf{W}}_1 \in \mathbb{R}^{2F \times I}$ and $\hat{\mathbf{W}}_2 \in \mathbb{R}^{2F \times J}$ from two independently moving objects, and that these trajectories are contaminated with zero-mean Gaussian noise of spherical covariance $\eta \sim \mathcal{N}(\mathbf{0}, (\sigma^f)^2\mathbf{I})$:

$$\hat{\mathbf{W}} = \left[\hat{\mathbf{W}}_1 | \hat{\mathbf{W}}_2\right] + \eta. \qquad (15)$$

Now, assume we know the true affine transformations $\mathbf{A}_1^f$ and $\mathbf{A}_2^f$ that describe the motion of trajectories for the subsets $\hat{\mathbf{W}}_1$ and $\hat{\mathbf{W}}_2$, respectively. If $\mathbf{A}_1^f$ is used to compute predictions for all of $\hat{\mathbf{W}}$ (at frame f), the expected value (denoted by $\langle\cdot\rangle$) of the magnitude of the residuals ($\mathbf{r}^f$ from Eq. 11) for trajectories in $\hat{\mathbf{W}}_1$ will be in the order of the magnitude of the underlying noise $\langle|\mathbf{r}_i^f|\rangle = \sigma^f$ for each $i \in \{1, \ldots, I\}$. But in this scenario, trajectories in $\hat{\mathbf{W}}_2$ will be predicted using the wrong model, resulting in residuals with magnitudes determined by the motion differential $\left|\mathbf{r}_i^f\right| = \left|(\mathbf{A}_1^f - \mathbf{A}_2^f)\hat{\mathbf{w}}_i^b\right|$. If we can assume that the motion differential is bigger than the displacement due to noise:

$$\left|(\mathbf{A}_1^f - \mathbf{A}_2^f)\mathbf{w}_i^b\right| > \sigma^f, \qquad (16)$$

then the model inliers can be determined by thresholding $|\mathbf{r}_i^f|$ with the magnitude of the noise, scaled by a constant ($\tau = \lambda_\sigma \sigma^f$):

$$b_i^f = \begin{cases} 1, & |\mathbf{r}_i^f| \leq \tau \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

But because σ^f is generally unknown, the threshold (τ) is estimated from the residual data. To do so, let $\hat{\mathbf{r}}$ be the vector of residual magnitudes where $\hat{r}_i \leq \hat{r}_{i+1}$. Now, let $\tilde{r} = \texttt{median}\,(\hat{r}_{i+1} - \hat{r}_i)$. The threshold is then defined as

$$\tau = \min\{\hat{r}_i \mid (r_{i+1} - r_i) > \lambda_r \tilde{r}\}, \tag{18}$$

which corresponds to the smallest residual magnitude before a salient magnitude gap. Our experiments showed this test to be efficient and effective. Figure 1.e shows class-labeled residuals. Notice the presence of a (low density) gap between the residuals from the trajectories explained by the correct model (in red, close to the origin), and the rest.

4.2. Magnitude of the Noise, 2D Model

Let $\hat{\mathbf{r}}_{2\text{D}}^f$ contain only the residuals of the inlier trajectories (those where $b_i^f = 1$), and let $USV^\top$ be the singular value decomposition of the covariance matrix of $\hat{\mathbf{r}}_{2\text{D}}^f$:

$$USV^\top = \texttt{svd}\left(\frac{1}{\sum b_p^f}(\hat{\mathbf{r}}_{2\text{D}}^f)^\top \hat{\mathbf{r}}_{2\text{D}}^f\right). \tag{19}$$

Then the magnitude of the noise corresponds to the largest singular value $\sigma^2 = s_1$, because if the underlying geometry is in fact planar, then the only unaccounted displacements captured by the residuals are due to noise. Model capacity can also be determined from S, as explained next.

4.3. Model Capacity

The ratio of largest over smallest singular values (s_1/s_2) determines when upgrading to a 3D model is beneficial. When the underlying geometry is actually non-planar, the residuals from a planar model should distribute along a line (the epipolar line), reflecting that their relative depth is being unaccounted for. This produces a covariance matrix with a large ratio $s_1/s_2 \gg 1$. If on the other hand, if $s_1/s_2 \approx 1$, then there is no indication of unexplained relative depth, in which case, fitting a line to spherically distributed residuals will only increase the model complexity without explaining the residual variance much better. A small spherical residual covariance strongly suggests a planar underlying geometry.

4.4. 3D Inlier Detection

When the residual distribution is elongated ($s_1/s_2 \gg 1$), a line segment is robustly fit to the (potentially contaminated) set of residuals. The segment must go through the origin and its parameters are computed using a Hough transform. Further details about this algorithm can be found in the supplementary material.

Inlier detection The resulting line segment is used to determine 3D model inliers. Trajectory i becomes an inlier at frame f if it satisfies two conditions. First, the projection of $\mathbf{r}_i^f$ onto the line must lie within the segment limits ($\beta \leq \mathbf{r}_i^{f\top}\mathbf{e}^f \leq \gamma$). Second, the normalized distance to the line must be below a threshold ($\mathbf{e}_\perp^{f\top}\mathbf{r}_i^f \leq \sigma_2\lambda_d$). Notice that the threshold depends on the smallest singular value from Eq. 19 to (roughly) account for the presence of noise in the direction perpendicular to the epipolar ($\mathbf{e}_\perp^f$).

4.5. Magnitude of the Noise, 3D Model

Similarly to the 2D case, let $\hat{\mathbf{r}}_{3\text{D}}^f$ contain the residual data from the corresponding 3D inlier trajectories. An estimate for the magnitude of the noise that reflects the use of a 3D model can be obtained from the singular value decomposition of the covariance matrix of $\hat{\mathbf{r}}_{3\text{D}}^f$ (as in Eq. 19). In this case, the largest singular value s_1 captures the spread of residuals along the epipolar line, so its magnitude is mainly related to the magnitude of the displacements due to relative depth. However, s_2 captures deviations from the epipolar line, which in a rigid 3D object can only be attributed to noise, making $\sigma^2 = s_2$ a reasonable estimate for its magnitude.

Optimal model parameters When both 2D and 3D models are instantiated, the one with the smallest penalized negative log-likelihood (7) becomes the winning model for the current RANSAC run. The same penalized negative log-likelihood metric is used to determine the better model from across all RANSAC iterations. The winning model is added to the pool $\mathcal{M}$, and the process is repeated M times, forming the pool $\mathcal{M} = \{\mathbf{M}_1, \ldots, \mathbf{M}_M\}$.

5. Optimal Model Subset

The next step is to find the model combination $\mathcal{M}^\star \subset \mathcal{M}$ that maximizes prediction accuracy for the whole trajectory data $\mathbf{W}$, while minimizing model complexity and modelling overlap. For this purpose, let $\mathcal{M}_j = \{\mathbf{M}_{j,1}, \ldots, \mathbf{M}_{j,N}\}$ be the j-th model combination, and let $\{\mathcal{M}_j\}$ be the set of all ${}_M C_N = \frac{M!}{N!(M-N)!)}$ combinations of N-sized models than can be drawn from $\mathcal{M}$. The model selection problem is then formulated as

$$\mathcal{M}^\star = \underset{\{\mathcal{M}_j\}}{\operatorname{argmin}}\, \mathcal{O}_S(\mathcal{M}_j), \tag{20}$$

where the objective is

$$\mathcal{O}_S(\mathcal{M}_j) = \sum_{n=1}^{N}\sum_{p=1}^{P}\pi_{p,n}E\left(\mathbf{w}_p,\mathbf{M}_{j,n}\right) + \lambda_\Phi\sum_{i=1}^{P}\Phi(\mathbf{w}_p,\mathbf{M}_{j,n}) + \lambda_\Psi\sum_{n=1}^{N}\Psi(\mathbf{M}_{j,n}). \tag{21}$$

The first term accounts for prediction accuracy, the other two are regularization terms. Details follow.

Prediction Accuracy In order to determine how well a model $\mathbf{M}$ predicts an arbitrary trajectory $\mathbf{w}$, the affine transformations estimated by RANSAC could be re-used. However, the inherent noise in the control points, and the potentially short distance between them, often render this approach impractical, particularly when $\mathbf{w}$ is spatially distant from the control points (see §3.1). Instead, model parameters are computed with a factorization based [10] method. Given the inlier labeling $\mathbf{B}$ in $\mathbf{M}$, let $\mathbf{W_B}$ be the subset of trajectories where $b_i^f = 1$ for at least half of the frames. The orthonormal basis $\mathbf{S}$ of a $\omega = 2$D (or 3D) motion model can be determined by the 2 (or 3) left singular vectors of $\mathbf{W_B}$. Using $\mathbf{S}$ as the model's motion matrices, prediction accuracy can be computed using:

$$E(\mathbf{w},\mathbf{M}) = \left|\mathbf{S}\mathbf{S}^\top\mathbf{w} - \mathbf{w}\right|^2, \tag{22}$$

which is the sum of squared Euclidean deviations from the predictions ($\mathbf{S}\mathbf{S}^\top\mathbf{w}$), to the observed data ($\mathbf{w}$). Our experiments indicated that, although sensitive to outliers, these model predictions are much more robust to noise.

Ownership variables $\mathbf{\Pi} \in \{0,1\}^{[P\times N]}$ indicate whether trajectory p is explained by the n-th model ($\pi_{p,n} = 1$) or not ($\pi_{p,n} = 0$), and are determined by maximum prediction accuracy (*i.e.* minimum Euclidean deviation):

$$\pi_{p,n} = \begin{cases} 1, & \text{if } \mathbf{M}_{j,n} = \underset{\mathbf{M}\in\mathcal{M}_j}{\operatorname{argmin}} E(\mathbf{w}_p,\mathbf{M}) \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

Regularization terms The second term from Eq. 21 penalizes situations where multiple models explain a trajectory ($\mathbf{w}$) with relatively small residuals. For brevity, let $\hat{E}(\mathbf{w},\mathbf{M}) = \exp\{-E(\mathbf{w},\mathbf{M})\}$, then:

$$\Phi(\mathbf{w},\mathcal{M}_j) = -\log\frac{\max\limits_{\mathbf{M}\in\mathcal{M}_j}\hat{E}(\mathbf{w},\mathbf{M})}{\sum\limits_{\mathbf{M}\in\mathcal{M}_j}\hat{E}(\mathbf{w},\mathbf{M})}. \tag{24}$$

The third term regularizes over the number of model parameters, and is evaluated using Eq. 8. The constants λ_Φ and λ_Ψ modulate the effect of the corresponding regularizer.

Table 1: Accuracy and run-time for the H155 dataset. Naive RANSAC included as a baseline with overall accuracy and total computation time estimated using data from [12].

Algorithm	Average Accuracy [%]	Computation time [s]
SSC [3]	98.76	14500
CPA [14]	98.75	147600
RANSAC	89.15	30
Ours	96.19	217

6. Refinement

The optimal model subset $\mathcal{M}^\star$ yields ownership variables $\mathbf{\Pi}^\star$ which can already be interpreted as a segmentation result. However, we found that segmentation accuracy can be improved by incorporating the labellings $\mathbf{\Pi}^t$ from the top T subsets $\{\mathcal{M}_t^\star \mid 1 \le t \le T\}$ closest to optimal.

Multiple labellings are incorporated into an affinity matrix $\mathbf{F}$, where the $f_{i,j}$ entry indicates the frequency with which trajectory i is given the same label as trajectory j across all T labellings, weighted by the relative objective function $\tilde{\mathcal{O}}_t = \exp\left\{-\frac{\mathcal{O}_S(\mathbf{W}|\mathcal{M}_t^\star)}{\mathcal{O}_S(\mathbf{W}|\mathcal{M}^\star)}\right\}$ for such a labelling:

$$f_{i,j} = \frac{1}{\sum_{t=1}^{T}\tilde{\mathcal{O}}_t}\sum_{t=1}^{T}\left(\boldsymbol{\pi}_{i,:}^t\boldsymbol{\pi}_{j,:}^{t\top}\right)\tilde{\mathcal{O}}_t \tag{25}$$

Note that the inner product between the label vectors $(\boldsymbol{\pi}_{i,:}\boldsymbol{\pi}_{j,:}^\top)$ is equal to one only when the labels are the same.

A spectral clustering method is applied on $\mathbf{F}$ to produce the method's final segmentation result.

7. Experiments

Evaluation was made through three experimental setups.

Hopkins 155 The Hopkins 155 (H155) dataset has been the standard evaluation metric for the problem of motion segmentation of trajectory data since 2007. It consists of checkerboard, traffic and articulated sequences with either 2 or 3 motions. Data was automatically tracked, but tracking errors were manually corrected; further details are available in [12]. The use of a standard dataset enables direct comparison of accuracy and run-time performance. Table 1 shows the relevant figures for the two most competitive algorithms that we are aware of. The data indicates that our algorithm has run-times that are close to 2 or 3 orders of magnitude faster than the state of the art methods, with minimal accuracy loss. Computation times are measured in the same (or very similar) hardware architectures. Like in CPA, our implementation uses a single set of parameters for all the experiments, but as others had pointed out [14], it remains unclear whether the same is true for the results reported in the original SSC paper.

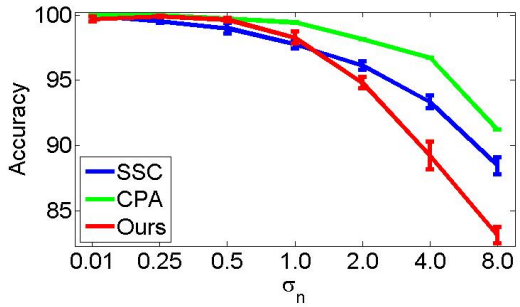


Figure 4: Accuracy error-bars across artificial H155 datasets with controlled levels of Gaussian noise.

Artificial Noise The second experimental setup complements an unexplored dimension in the H155 dataset: noise. The goal is to determine the effects of noise of different magnitudes towards the segmentation accuracy of our method, in comparison with the state of the art.

We noted that H155 contains structured long-tailed noise, but for the purpose of this experiment we required a noise-free dataset as a baseline. To generate such a dataset, ground-truth labels were used to compute a rank 3 reconstruction of (mean-subtracted) trajectories for each segment. Then, multiple versions of H155 were computed by contaminating the noise-free dataset with Gaussian noise of magnitudes $\sigma_n \in \{0.01, 0.25, 0.5, 1, 2, 4, 8\}$. Our method, as well as SSC and CPA were run on these noise-controlled datasets; results are shown in Figure 4. The error bars on SSC and Ours indicate one standard deviation, computed over 20 runs. The plot for CPA is generated with only one run for each dataset (running time: 11.95 days). The graph indicates that our method only compromises accuracy for large levels of noise, while still being around 2 or 3 orders of magnitude faster than the most competitive algorithms.

KLT Tracking The last experimental setup evaluates the applicability of the algorithm in real world conditions using raw tracks from an off-the-shelf implementation [1] of the Kanade-Lucas-Tomasi algorithm. Several sequences were tracked and the resulting trajectories classified by our method. Figure 5 shows qualitatively good motion segmentation results for four sequences. Challenges include very small relative motions, tracking noise, and a large presence of outliers.

8. Conclusions

We introduced a computationally efficient motion segmentation algorithm for trajectory data. Efficiency comes from the use of a simple but powerful representation of motion that explicitly incorporates mechanisms to deal with noise, outliers and motion degeneracies. Run-time comparisons indicate that our method is 2 or 3 orders of magnitude faster than the state of the art, with only a small loss in accuracy. The robustness of our method to Gaussian noise of different magnitudes was found competitive with state of the art, while retaining the inherent computational efficiency. The method was also found to be useful for motion segmentation of real-world, raw trajectory data.

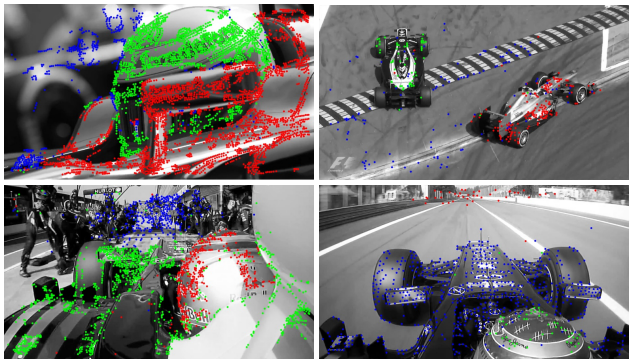


Figure 5: Segmentation results from raw KLT automatic tracks from four Formula 1 sequences. *Italian Grand Prix* ©*2012 Formula 1*. In this figure, all trajectories are given a motion label, including outliers.

References

[1] http://www.ces.clemnson.edu/~stb/klt. 8

[2] J. P. Costeira and T. Kanade. A Multibody Factorization Method for Independently Moving Objects. *IJCV*, 1998. 1

[3] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *Proc. CVPR*, 2009. 2, 7

[4] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981. 3

[5] M. Irani and P. Anandan. Parallax geometry of pairs of points for 3d scene analysis. *Proc. ECCV*, 1996. 2

[6] K. Kanatani. Motion segmentation by subspace separation: Model selection and reliability evaluation. *International Journal Image Graphics*, 2002. 1

[7] H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, MA Fischler and O. Firschein, eds*, 1987. 1

[8] K. Schindler, D. Suter, , and H. Wang. A model-selection framework for multibody structure-and-motion of image sequences. *Proc. IJCV*, 79(2):159–177, 2008. 1

[9] C. Tomasi and T. Kanade. Shape and motion without depth. *Proc. ICCV*, 1990. 1

[10] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 1992. 1, 3, 7

[11] P. Torr. Geometric motion segmentation and model selection. *Phil. Tran. of the Royal Soc. of Lon. Series A: Mathematical, Physical and Engineering Sciences*, 1998. 1

[12] R. Tron and R. Vidal. A Benchmark for the Comparison of 3-D Motion Segmentation Algorithms. In *Proc. CVPR*, 2007. 7

[13] J. Yan and M. Pollefeys. A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *PAMI*, 2008. 1

[14] L. Zappella, E. Provenzi, X. Lladó, and J. Salvi. Adaptive motion segmentation algorithm based on the principal angles configuration. *Proc. ACCV*, 2011. 2, 7

[15] L. Zelnik-Manor and M. Irani. Degeneracies, dependencies and their implications in multi-body and multi-sequence factorizations. *Proc. CVPR*, 2003. 1