# Multi-Model Fitting in Computer Vision: Advanced Robust Estimation Techniques

Lecture Notes

July 16, 2025

**Abstract**

This document presents a comprehensive treatment of multi-model fitting techniques in computer vision, focusing on robust estimation methods that can handle multiple geometric structures simultaneously. We explore the evolution from classical RANSAC to modern multi-model approaches, including MSAC, MLESAC, sequential RANSAC, and the Hough transform. The mathematical foundations, algorithmic implementations, and practical applications are thoroughly examined with particular emphasis on the fundamental chicken-and-egg problem inherent in clustering points to fit models while requiring models to define clusters.

# Contents

**1 Introduction to Robust Model Fitting** **2**

1.1 The Outlier Problem . . . . . . . . . . . . 2

1.2 Classical RANSAC Framework . . . . . . . . . . . . 2

**2 Advanced RANSAC Variants** **4**

2.1 M-Estimator Sample Consensus (MSAC) . . . . . . . . . . . . 4

2.2 Maximum Likelihood Estimation Sample Consensus (MLESAC) . . . . . . 4

2.3 Comparative Analysis of RANSAC Variants . . . . . . . . . . . . 5

**3 Multi-Model Fitting Framework** **6**

3.1 Problem Formulation . . . . . . . . . . . . 6

3.2 The Chicken-and-Egg Problem . . . . . . . . . . . . 6

3.3 Outlier Complexity in Multi-Model Settings . . . . . . . . . . . . 7

**4 Sequential Multi-Model Fitting** **8**

4.1 Sequential RANSAC . . . . . . . . . . . . 8

4.2 Theoretical Analysis of Sequential RANSAC . . . . . . . . . . . . 8

4.3 Preference Matrix Formulation . . . . . . . . . . . . 9

**5 Simultaneous Multi-Model Fitting and Hough Transform** 10
5.1 Conceptual Framework for Simultaneous Fitting . . . . . . . . . . . . . . . . . 10
5.2 The Hough Transform . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
5.2.1 Fundamental Principle . . . . . . . . . . . . . . . . . . . . . . . . . 10
5.2.2 Line Detection via Hough Transform . . . . . . . . . . . . . . . . . 11
5.2.3 Mathematical Analysis . . . . . . . . . . . . . . . . . . . . . . . . . 11
5.3 Extension to Circle Detection . . . . . . . . . . . . . . . . . . . . . . . . . 12

**6 Applications and Implementation Considerations** 13
6.1 Geometric Structure Detection . . . . . . . . . . . . . . . . . . . . . . . . . 13
6.1.1 Architectural Scene Analysis . . . . . . . . . . . . . . . . . . . . . 13
6.1.2 Fundamental Matrix Estimation . . . . . . . . . . . . . . . . . . . . 13
6.2 Motion Segmentation and Tracking . . . . . . . . . . . . . . . . . . . . . . 14
6.2.1 Trajectory Clustering . . . . . . . . . . . . . . . . . . . . . . . . . . 14
6.3 Template Matching and Object Recognition . . . . . . . . . . . . . . . . . . 14
6.4 Implementation Considerations . . . . . . . . . . . . . . . . . . . . . . . . 14
6.4.1 Parameter Selection . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
6.4.2 Computational Optimization . . . . . . . . . . . . . . . . . . . . . . 15

**7 Advanced Topics and Current Research** 16
7.1 Multi-Class Model Fitting . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
7.2 Information-Theoretic Model Selection . . . . . . . . . . . . . . . . . . . . . 16
7.3 Sparsity-Regularized Multi-Model Fitting . . . . . . . . . . . . . . . . . . . 17
7.4 Deep Learning Integration . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
7.4.1 Learned Sampling Strategies . . . . . . . . . . . . . . . . . . . . . . 17
7.4.2 Differentiable RANSAC . . . . . . . . . . . . . . . . . . . . . . . . . 17
7.5 Emerging Applications . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
7.5.1 Event-Based Vision . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
7.5.2 Point Cloud Processing . . . . . . . . . . . . . . . . . . . . . . . . . 18

**8 Theoretical Foundations and Conclusions** 19
8.1 Fundamental Theoretical Results . . . . . . . . . . . . . . . . . . . . . . . 19
8.2 Robustness Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
8.3 Convergence Properties . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
8.4 Performance Comparison . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
8.5 Future Research Directions . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
8.5.1 Theoretical Advances . . . . . . . . . . . . . . . . . . . . . . . . . . 20
8.5.2 Algorithmic Improvements . . . . . . . . . . . . . . . . . . . . . . . 20
8.5.3 Application Domains . . . . . . . . . . . . . . . . . . . . . . . . . . 20
8.6 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

**A Mathematical Derivations** 22
A.1 RANSAC Iteration Count Derivation . . . . . . . . . . . . . . . . . . . . . 22
A.2 Hough Transform Accumulator Resolution . . . . . . . . . . . . . . . . . . 22

**B Implementation Guidelines** **22**
B.1 Pseudo-Code for Key Algorithms . . . . . . . . . . . . . . . . . . . . . . . . 22
B.2 Parameter Selection Guidelines . . . . . . . . . . . . . . . . . . . . . . . . . 23

**C Experimental Validation** **23**
C.1 Synthetic Data Experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
C.2 Real Data Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

# 1 Introduction to Robust Model Fitting

The problem of fitting parametric models to data contaminated with outliers represents one of the most fundamental challenges in computer vision and pattern recognition. Classical least-squares estimation, while optimal for Gaussian noise, fails catastrophically in the presence of even small percentages of outliers. This motivates the development of robust estimation techniques that can maintain accuracy even when significant portions of the data do not conform to the assumed model.

## 1.1 The Outlier Problem

Consider a dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ where each point $\mathbf{x}_i \in \mathbb{R}^d$ represents a measurement. In the presence of outliers, we can partition the data into two disjoint sets:

$$\mathcal{D} = \mathcal{I} \cup \mathcal{O} \tag{1}$$

$$\mathcal{I} \cap \mathcal{O} = \emptyset \tag{2}$$

where $\mathcal{I}$ represents the set of inliers that conform to the underlying model, and $\mathcal{O}$ represents the outliers that do not.

**Definition 1.1** (Breakdown Point)**.** *The breakdown point of an estimator is the smallest fraction of outliers that can cause the estimator to produce arbitrarily large errors. Formally, for an estimator $\hat{\theta}$ and dataset size $n$, the breakdown point $\epsilon^*$ is:*

$$\epsilon^* = \max\left\{\frac{k}{n} : \sup_{\mathcal{O},|\mathcal{O}|=k} \|\hat{\theta}(\mathcal{D}) - \theta_{true}\| < \infty\right\} \tag{3}$$

## 1.2 Classical RANSAC Framework

The Random Sample Consensus (RANSAC) algorithm, introduced by Fischler and Bolles, provides a paradigm for robust model fitting. The algorithm iteratively selects minimal sample sets, fits models, and evaluates their support in the data.

Key Insight: Least squares has a breakdown point of $\frac{1}{n}$, while robust estimators can achieve up to 50%.

**Algorithm 1.2** (RANSAC)**.** *Given dataset $\mathcal{D}$, model class $\mathcal{M}$, and parameters $\tau$ (inlier threshold) and $K$ (maximum iterations):*

```
1. For $k = 1, 2, \ldots, K$:
    (a) Randomly select minimal sample set $\mathcal{S}_k \subset \mathcal{D}$ with $|\mathcal{S}_k| = m$
    (b) Fit model $\theta_k$ to $\mathcal{S}_k$
    (c) Compute consensus set: $\mathcal{C}_k = \{\mathbf{x}_i \in \mathcal{D} : d(\mathbf{x}_i, \theta_k) \leq \tau\}$
    (d) If $|\mathcal{C}_k| > |\mathcal{C}_{best}|$, set $\mathcal{C}_{best} = \mathcal{C}_k$ and $\theta_{best} = \theta_k$
2. Refine $\theta_{best}$ using least squares on $\mathcal{C}_{best}$
```

The number of iterations $K$ is typically chosen to ensure a high probability of sampling at least one outlier-free minimal set:

$$K = \frac{\log(1 - p)}{\log(1 - (1 - \epsilon)^m)} \tag{4}$$

where $p$ is the desired success probability and $\epsilon$ is the outlier ratio.

# 2 Advanced RANSAC Variants

While classical RANSAC provides a robust framework, several limitations motivate the development of more sophisticated variants. These include the binary nature of the consensus evaluation, sensitivity to threshold selection, and suboptimal use of geometric information.

## 2.1 M-Estimator Sample Consensus (MSAC)

MSAC addresses the limitation of RANSAC's binary consensus evaluation by incorporating the distance of inliers from the fitted model. Instead of simply counting inliers, MSAC minimizes a more informative loss function.

**Definition 2.1** (MSAC Loss Function)**.** *For a model $\theta$ and data point $\mathbf{x}_i$, the MSAC loss is defined as:*

$$\rho_{MSAC}(r_i, \theta) = \begin{cases} r_i^2 & \textit{if } r_i \leq \tau \\ \tau^2 & \textit{if } r_i > \tau \end{cases} \tag{5}$$

*where $r_i = d(\mathbf{x}_i, \theta)$ is the residual distance from point $\mathbf{x}_i$ to model $\theta$.*

The total loss for a model $\theta$ is:

$$L_{\text{MSAC}}(\theta) = \sum_{i=1}^{N} \rho_{\text{MSAC}}(r_i, \theta) \tag{6}$$

**Theoretical Analysis:** The MSAC loss function is a truncated quadratic, which provides several advantages:

1. *Outlier Robustness*: Points beyond threshold $\tau$ contribute a constant penalty, preventing outliers from dominating the loss
2. *Inlier Refinement*: Points within the threshold are penalized proportionally to their squared distance, encouraging better fits
3. *Bounded Influence*: The loss function is bounded above by $\tau^2$, ensuring stability

Advantage: MSAC provides more discriminative power by considering how well inliers fit the model, not just their count.

## 2.2 Maximum Likelihood Estimation Sample Consensus (MLESAC)

MLESAC further refines the evaluation criterion by adopting a probabilistic framework based on the median of absolute residuals, providing additional robustness against outliers.

**Definition 2.2** (MLESAC Loss Function)**.** *The MLESAC loss is defined as:*

$$L_{MLESAC}(\theta) = \text{median}_{i=1}^{N} |r_i| \tag{7}$$

*where $r_i = d(\mathbf{x}_i, \theta)$ represents the residual for point $\mathbf{x}_i$.*

**Robustness Properties:** The median-based loss function exhibits superior robustness characteristics:

**Theorem 2.3** (Median Robustness)**.** *The median estimator has a breakdown point of* 50%*, meaning it can tolerate up to* 50% *outliers without arbitrary degradation.*

*Proof Sketch.* Consider a dataset with $n$ points where $k < n/2$ are outliers. The median will always be determined by the majority of inliers, regardless of the magnitude of the outlier values. Only when $k \geq n/2$ can outliers influence the median value. □

**Computational Considerations:** While the median provides excellent robustness, its computation requires $O(n \log n)$ time for sorting, compared to $O(n)$ for mean-based estimators. However, efficient selection algorithms can reduce this to $O(n)$ expected time.

## 2.3 Comparative Analysis of RANSAC Variants

| **Method** | **Loss Function** | **Breakdown Point** | **Computational Cost** | **Threshold Sensitivi** |
|---|---|---|---|---|
| RANSAC | Binary count | 50% | $O(n)$ | High |
| MSAC | Truncated quadratic | 50% | $O(n)$ | Medium |
| MLESAC | Median absolute | 50% | $O(n \log n)$ | Low |

Table 1: Comparison of RANSAC variants and their properties

# 3 Multi-Model Fitting Framework

The extension from single-model to multi-model fitting introduces fundamental challenges that require sophisticated approaches. The core difficulty lies in the chicken-and-egg problem: clustering points requires knowledge of models, while fitting models requires knowledge of point clusters.

## 3.1 Problem Formulation

**Definition 3.1** (Multi-Model Fitting Problem)**.** *Given a dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ and a model family $\mathcal{M}$, find:*

$$\Theta = \{\theta_1, \theta_2, \ldots, \theta_K\} \tag{8}$$

$$\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_K\} \tag{9}$$

*where $\Theta$ represents the set of model parameters and $\mathcal{P}$ represents a partition of the data such that:*

$$\bigcup_{k=1}^{K} \mathcal{P}_k = \mathcal{D} \setminus \mathcal{O} \tag{10}$$

$$\mathcal{P}_i \cap \mathcal{P}_j = \emptyset \quad \forall i \neq j \tag{11}$$

*and each $\theta_k$ optimally fits the points in $\mathcal{P}_k$.*

## 3.2 The Chicken-and-Egg Problem

The fundamental challenge in multi-model fitting can be formalized as follows:

1. **Clustering Requirement**: To fit model $\theta_k$, we need to identify the subset $\mathcal{P}_k \subset \mathcal{D}$ of points that belong to model $k$
2. **Model Requirement**: To cluster point $\mathbf{x}_i$ into partition $\mathcal{P}_k$, we need to know model $\theta_k$ to compute the distance $d(\mathbf{x}_i, \theta_k)$

This circular dependency necessitates iterative or simultaneous approaches that can break the cycle through:

- *Sequential strategies*: Fit models one at a time, removing inliers after each fit
- *Simultaneous strategies*: Jointly optimize over all models and partitions
- *Voting strategies*: Use parameter space voting to identify multiple models

Challenge: The number of models $K$ is typically unknown a priori, adding another layer of complexity.

### 3.3 Outlier Complexity in Multi-Model Settings

In multi-model fitting, the effective outlier ratio increases significantly. Consider fitting model $\theta_k$ in the presence of $K$ models:

$$\epsilon_{\text{effective}} = \epsilon_{\text{true}} + \frac{K-1}{K}(1 - \epsilon_{\text{true}}) \tag{12}$$

where $\epsilon_{\text{true}}$ is the true outlier ratio and $\epsilon_{\text{effective}}$ is the effective outlier ratio seen by each individual model.

**Example:** With 10 equally represented models and 10% true outliers:

$$\epsilon_{\text{effective}} = 0.1 + \frac{9}{10}(1 - 0.1) = 0.1 + 0.9 \cdot 0.9 = 0.91 \tag{13}$$

This dramatic increase in effective outlier ratio severely impacts the performance of standard RANSAC, motivating specialized multi-model approaches.

# 4 Sequential Multi-Model Fitting

Sequential approaches address multi-model fitting by iteratively applying single-model fitting techniques, removing inliers after each successful fit. While conceptually simple, these methods can suffer from order dependency and accumulation of errors.

## 4.1 Sequential RANSAC

Sequential RANSAC represents the most straightforward extension of RANSAC to multi-model scenarios. The algorithm repeatedly applies RANSAC, removing inliers after each successful model fit.

**Algorithm 4.1** (Sequential RANSAC)**.** *Given dataset $\mathcal{D}$, minimum consensus $\tau_{min}$, and maximum models $K_{max}$:*

```
1. Initialize: $\mathcal{D}_{current} = \mathcal{D}$, $\Theta = \emptyset$, $k = 0$
2. While $k < K_{max}$ and $|\mathcal{D}_{current}| \geq \tau_{min}$:
    (a) Apply RANSAC to $\mathcal{D}_{current}$ to find model $\theta_k$
    (b) Compute consensus set $\mathcal{C}_k = \{\mathbf{x}_i \in \mathcal{D}_{current} : d(\mathbf{x}_i, \theta_k) \leq \tau\}$
    (c) If $|\mathcal{C}_k| < \tau_{min}$, terminate
    (d) Refine $\theta_k$ using least squares on $\mathcal{C}_k$
    (e) Add $\theta_k$ to $\Theta$
    (f) Update: $\mathcal{D}_{current} = \mathcal{D}_{current} \setminus \mathcal{C}_k$
    (g) Increment: $k = k + 1$
```

## 4.2 Theoretical Analysis of Sequential RANSAC

**Theorem 4.2** (Sequential RANSAC Convergence)**.** *Under the assumption that models are well-separated and the minimum consensus threshold is appropriately chosen, Sequential RANSAC will find all models with probability:*

$$P_{success} = \prod_{k=1}^{K} \left(1 - (1 - (1 - \epsilon_k)^m)^{T_k}\right) \tag{14}$$

*where $\epsilon_k$ is the outlier ratio when fitting model $k$ and $T_k$ is the number of RANSAC iterations for model $k$.*

**Limitations of Sequential RANSAC:**

1. *Order Dependency*: The sequence in which models are found depends on their relative support and may not reflect the true underlying structure

2. *Error Accumulation*: Misclassified points in early iterations affect subsequent model fitting

3. *Threshold Sensitivity*: Performance heavily depends on the choice of inlier threshold and minimum consensus

4. *Suboptimal Solutions*: Greedy selection may lead to locally optimal but globally suboptimal solutions

## 4.3 Preference Matrix Formulation

Sequential RANSAC can be understood through the lens of preference matrices, which provide insight into simultaneous multi-model approaches.

**Definition 4.3** (Preference Matrix)**.** *The preference matrix* $\mathbf{P} \in \mathbb{R}^{N \times M}$ *is defined as:*

$$P_{ij} = \begin{cases} 1 & \textit{if } d(\mathbf{x}_i, \theta_j) \leq \tau \\ 0 & \textit{otherwise} \end{cases} \tag{15}$$

*where* $N$ *is the number of data points and* $M$ *is the number of model hypotheses.*

Alternatively, the preference matrix can store residuals:

$$P_{ij} = d(\mathbf{x}_i, \theta_j) \tag{16}$$

**Conceptual Insights:**

- Each row represents a data point's affinity to all model hypotheses
- Each column represents a model's support across all data points
- Sequential RANSAC selects the column with maximum support, then removes corresponding rows
- Simultaneous approaches can exploit the full matrix structure

# 5 Simultaneous Multi-Model Fitting and Hough Transform

Simultaneous approaches to multi-model fitting attempt to identify all models jointly, avoiding the limitations of sequential methods. The Hough transform represents a classical voting-based approach that has been widely successful in computer vision applications.

## 5.1 Conceptual Framework for Simultaneous Fitting

Instead of sequentially selecting columns from the preference matrix, simultaneous approaches seek to:

1. *Cluster rows*: Group data points with similar preference patterns
2. *Regularize solutions*: Incorporate sparsity constraints to prefer simpler models
3. *Optimize jointly*: Minimize a global objective function over all models and assignments

**Definition 5.1** (Joint Optimization Problem)**.** *The simultaneous multi-model fitting problem can be formulated as:*

$$\min_{\Theta,\mathcal{P}} \quad \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathcal{P}_k} \rho(d(\mathbf{x}_i, \theta_k)) + \lambda \|\Theta\|_0 \tag{17}$$

$$\textit{subject to} \quad \mathcal{P}_i \cap \mathcal{P}_j = \emptyset \quad \forall i \neq j \tag{18}$$

$$\bigcup_{k=1}^{K} \mathcal{P}_k \subseteq \mathcal{D} \tag{19}$$

*where $\rho(\cdot)$ is a robust loss function and $\lambda$ controls model complexity.*

## 5.2 The Hough Transform

The Hough transform provides an elegant solution to multi-model fitting by transforming the point clustering problem into a peak detection problem in parameter space.

### 5.2.1 Fundamental Principle

The key insight of the Hough transform is the duality between point space and parameter space:

- Each point in data space corresponds to a curve in parameter space
- Each model in parameter space corresponds to a point in data space
- Points lying on the same model generate curves that intersect at the model's parameters

### 5.2.2 Line Detection via Hough Transform

For line detection, we parameterize lines using the normal form:

$$\rho = x \cos\theta + y \sin\theta \tag{20}$$

where $\rho$ is the perpendicular distance from the origin to the line, and $\theta$ is the angle of the normal vector.

**Algorithm 5.2** (Hough Transform for Line Detection)**.** *Given edge points* $\{(x_i, y_i)\}_{i=1}^{N}$*:*

```
1. Initialize accumulator array $A[\rho, \theta]$ with appropriate discretization
2. For each edge point $(x_i, y_i)$:
   (a) For $\theta = -\pi/2$ to $\pi/2$ with step $\Delta\theta$:
       i. Compute $\rho = x_i \cos\theta + y_i \sin\theta$
       ii. Increment $A[\rho, \theta]$
3. Find local maxima in $A[\rho, \theta]$ above threshold
4. Each maximum corresponds to a line with parameters $(\rho, \theta)$
```

**Advantages of Normal Parameterization:**

1. *Bounded Parameter Space*: $\rho \in [0, \rho_{\max}]$ and $\theta \in [-\pi/2, \pi/2]$
2. *No Singularities*: Unlike slope-intercept form, vertical lines are handled naturally
3. *Uniform Discretization*: Parameter space can be uniformly discretized

### 5.2.3 Mathematical Analysis

**Theorem 5.3** (Hough Transform Optimality)**.** *The Hough transform finds the globally optimal solution to the multi-model fitting problem when:*

1. *The parameter space is exhaustively searched*
2. *The accumulator discretization is sufficiently fine*
3. *The peak detection threshold is appropriately chosen*

**Complexity Analysis:**

- *Time Complexity*: $O(N \cdot M)$ where $N$ is the number of points and $M$ is the number of parameter combinations
- *Space Complexity*: $O(M)$ for the accumulator array
- *Scalability*: Exponential growth with parameter dimensionality (curse of dimensionality)

## 5.3 Extension to Circle Detection

Circle detection requires a three-dimensional parameter space $(x_c, y_c, r)$ where $(x_c, y_c)$ is the center and $r$ is the radius. The circle equation is:

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \tag{21}$$

**Algorithm 5.4** (Hough Transform for Circle Detection)**.** *For known radius $r$:*

```
1. Initialize 2D accumulator $A[x_c, y_c]$
2. For each edge point $(x_i, y_i)$:
    (a) For $\theta = 0$ to $2\pi$ with step $\Delta\theta$:
        i. Compute $x_c = x_i + r\cos\theta$, $y_c = y_i + r\sin\theta$
        ii. Increment $A[x_c, y_c]$
3. Find peaks in $A[x_c, y_c]$
```

For unknown radius, a 3D accumulator is required, significantly increasing computational cost.

# 6 Applications and Implementation Considerations

Multi-model fitting techniques find widespread application across computer vision and robotics. This section examines key application domains and practical implementation considerations.

## 6.1 Geometric Structure Detection

### 6.1.1 Architectural Scene Analysis

In architectural scene analysis, multi-model fitting is used to identify planar structures from 3D point clouds. The typical workflow involves:

1. *3D Point Cloud Acquisition*: Using laser scanners or stereo vision systems
2. *Plane Fitting*: Applying multi-model RANSAC to identify wall and ceiling planes
3. *Structural Analysis*: Extracting architectural elements and their relationships

**Mathematical Formulation:** A plane in 3D space can be parameterized as:

$$\mathbf{n} \cdot (\mathbf{x} - \mathbf{p}) = 0 \tag{22}$$

where $\mathbf{n} = (a, b, c)^T$ is the unit normal vector and $\mathbf{p}$ is a point on the plane.

Alternatively, using the implicit form:

$$ax + by + cz + d = 0 \quad \text{with } a^2 + b^2 + c^2 = 1 \tag{23}$$

### 6.1.2 Fundamental Matrix Estimation

In stereo vision, the fundamental matrix $\mathbf{F}$ encodes the epipolar geometry between two views. Multi-model fitting is essential when dealing with:

- Multiple rigid objects moving independently
- Scenes with both static and dynamic elements
- Degenerate configurations requiring robust estimation

The fundamental matrix satisfies:

$$\mathbf{x}_2^T \mathbf{F} \mathbf{x}_1 = 0 \tag{24}$$

for corresponding points $\mathbf{x}_1$ and $\mathbf{x}_2$ in homogeneous coordinates.

## 6.2 Motion Segmentation and Tracking

### 6.2.1 Trajectory Clustering

In video analysis, points belonging to the same rigid object should follow consistent motion patterns. Multi-model fitting can cluster trajectories by fitting:

- *Affine motion models*: For short-term tracking
- *Subspace constraints*: For long-term trajectory analysis
- *Piecewise linear models*: For trajectory segmentation

**Subspace Clustering Formulation:** Trajectories of points on a rigid object lie in a low-dimensional subspace. Given trajectory matrix $\mathbf{T} \in \mathbb{R}^{2F \times P}$ where $F$ is the number of frames and $P$ is the number of points:

$$\mathbf{T} = \mathbf{MS} \tag{25}$$

where $\mathbf{M} \in \mathbb{R}^{2F \times r}$ contains motion information and $\mathbf{S} \in \mathbb{R}^{r \times P}$ contains structure information, with $r \ll \min(2F, P)$.

## 6.3 Template Matching and Object Recognition

Multi-model fitting enables robust template matching by:

1. *Correspondence Establishment*: Finding point correspondences between template and image
2. *Transformation Estimation*: Fitting multiple geometric transformations (affine, projective)
3. *Outlier Rejection*: Handling mismatched correspondences robustly

**Projective Transformation:** The projective transformation between template and image coordinates is:

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{26}$$

where $(x', y') = (x'/w', y'/w')$ are the transformed coordinates.

## 6.4 Implementation Considerations

### 6.4.1 Parameter Selection

Effective multi-model fitting requires careful selection of several parameters:

1. *Inlier Threshold ($\tau$)*: Should be set based on:

$$\tau = k \cdot \sigma_{\text{noise}} \tag{27}$$

where $k \in [2, 3]$ and $\sigma_{\text{noise}}$ is the estimated noise standard deviation

2. *Minimum Consensus ($\tau_{\min}$)*: Typically set as:

$$\tau_{\min} = \max(m + 1, \alpha \cdot N) \tag{28}$$

where $m$ is the minimal sample size and $\alpha \in [0.05, 0.1]$

3. *Maximum Iterations*: Based on confidence level and expected outlier ratio:

$$K_{\max} = \frac{\log(1 - \beta)}{\log(1 - (1 - \epsilon)^m)} \tag{29}$$

where $\beta$ is the desired confidence (typically 0.99)

### 6.4.2 Computational Optimization

Several strategies can improve computational efficiency:

1. *Early Termination*: Stop when sufficient consensus is found
2. *Preemptive Scoring*: Evaluate partial models before full fitting
3. *Guided Sampling*: Use prior information to bias sample selection
4. *Parallel Processing*: Distribute iterations across multiple cores

**Algorithm 6.1** (Optimized Sequential RANSAC)**.** *Enhanced version with computational optimizations:*

1. *Initialize data structures for efficient nearest neighbor queries*
2. *For each model fitting iteration:*
   - *(a) Use guided sampling based on local density*
   - *(b) Employ preemptive scoring to reject poor models early*
   - *(c) Update residual statistics incrementally*
   - *(d) Apply early termination when target consensus is reached*
3. *Use spatial indexing for efficient inlier identification*
4. *Employ least squares refinement with iterative reweighting*

# 7 Advanced Topics and Current Research

This section explores cutting-edge developments in multi-model fitting, including theoretical advances, hybrid approaches, and emerging applications in modern computer vision.

## 7.1 Multi-Class Model Fitting

Traditional multi-model fitting assumes all models belong to the same family (e.g., all lines or all circles). Multi-class fitting addresses scenarios where different model types coexist.

**Definition 7.1** (Multi-Class Fitting Problem). *Given model families $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_C$ and dataset $\mathcal{D}$, find:*

$$\Theta = \bigcup_{c=1}^{C} \Theta_c \quad \textit{where } \Theta_c \subset \mathcal{M}_c \tag{30}$$

$$\mathcal{P} = \{\mathcal{P}_\theta : \theta \in \Theta\} \tag{31}$$

*such that each partition is optimally explained by its corresponding model.*

**Challenges in Multi-Class Fitting:**

1. *Model Selection*: Determining which model family to use for each cluster
2. *Parameter Dimensionality*: Different model families have different parameter spaces
3. *Comparison Metrics*: Comparing fits across different model types requires normalized metrics

**Example Application:** Architectural scene analysis where both planar surfaces and cylindrical columns are present requires simultaneous fitting of planes and cylinders.

## 7.2 Information-Theoretic Model Selection

Modern approaches incorporate information-theoretic criteria to balance model complexity with fitting accuracy.

**Definition 7.2** (Bayesian Information Criterion for Multi-Model Fitting). *For a set of models $\Theta = \{\theta_1, \ldots, \theta_K\}$, the BIC score is:*

$$BIC(\Theta) = \sum_{k=1}^{K} \left[ |\mathcal{P}_k| \log \hat{\sigma}_k^2 + m_k \log |\mathcal{P}_k| \right] \tag{32}$$

*where $\hat{\sigma}_k^2$ is the estimated noise variance for model $k$, $m_k$ is the number of parameters in model $k$, and $|\mathcal{P}_k|$ is the size of the corresponding partition.*

**Akaike Information Criterion (AIC):** Alternative criterion that typically favors more complex models:

$$\mathrm{AIC}(\Theta) = \sum_{k=1}^{K} \left[ |\mathcal{P}_k| \log \hat{\sigma}_k^2 + 2m_k \right] \tag{33}$$

## 7.3 Sparsity-Regularized Multi-Model Fitting

Modern optimization techniques incorporate sparsity constraints to promote simpler model explanations.

**Definition 7.3** (Sparsity-Regularized Objective)**.** *The regularized multi-model fitting objective is:*

$$\min_{\Theta,\mathbf{Z}} \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik} \rho(d(\mathbf{x}_i, \theta_k)) + \lambda \sum_{k=1}^{K} \|\theta_k\|_1 + \mu \sum_{k=1}^{K} \mathbf{1}[\|\theta_k\|_2 > 0] \tag{34}$$

*where $\mathbf{Z} \in \{0,1\}^{N \times K}$ is the assignment matrix, $\lambda$ promotes parameter sparsity, and $\mu$ promotes model sparsity.*

**Optimization Approach:** The non-convex problem can be addressed using:

1. *Alternating Minimization*: Alternate between optimizing $\Theta$ and $\mathbf{Z}$
2. *Relaxation Methods*: Relax binary constraints to continuous variables
3. *Proximal Methods*: Use proximal operators for non-smooth terms

## 7.4 Deep Learning Integration

Recent advances combine traditional geometric fitting with deep learning approaches.

### 7.4.1 Learned Sampling Strategies

Neural networks can learn to predict promising sample sets for RANSAC:

$$p(\mathcal{S}|\mathcal{D}) = \text{NetworkSample}(\mathcal{D}; \phi) \tag{35}$$

where $\phi$ represents neural network parameters trained on datasets with known ground truth.

### 7.4.2 Differentiable RANSAC

Differentiable formulations enable end-to-end training:

$$\hat{\theta} = \sum_{j=1}^{M} w_j \theta_j \quad \text{where } w_j = \frac{\exp(\beta \cdot \text{score}(\theta_j))}{\sum_{k=1}^{M} \exp(\beta \cdot \text{score}(\theta_k))} \tag{36}$$

where $\beta$ is a temperature parameter controlling the softmax distribution.

## 7.5 Emerging Applications

### 7.5.1 Event-Based Vision

Event cameras generate asynchronous streams of brightness change events, requiring new approaches to multi-model fitting:

- *Temporal Modeling*: Events are timestamped, requiring temporal consistency
- *Sparse Data*: Events are sparse, challenging traditional accumulation methods
- *High-Speed Motion*: Rapid changes require adaptive model updating

**Event-Based Line Detection:** Lines in event space can be parameterized as:

$$\rho(t) = x\cos\theta + y\sin\theta + v_\rho t \tag{37}$$

where $v_\rho$ represents the temporal velocity component.

### 7.5.2 Point Cloud Processing

Modern 3D sensors generate massive point clouds requiring efficient multi-model fitting:

- *Hierarchical Processing*: Multi-resolution approaches for scalability
- *Semantic Constraints*: Incorporating learned semantic information
- *Temporal Consistency*: Maintaining model consistency across time

# 8 Theoretical Foundations and Conclusions

This final section synthesizes the theoretical foundations underlying multi-model fitting and provides conclusions about current capabilities and future directions.

## 8.1 Fundamental Theoretical Results

**Theorem 8.1** (Multi-Model Fitting Complexity)**.** *The multi-model fitting problem with $K$ models from a family with $m$ parameters each is NP-hard when $K$ is unknown and must be determined from the data.*

*Proof Sketch.* The problem reduces to the optimal clustering problem, which is known to be NP-hard. Specifically, determining the optimal partition $\mathcal{P}$ that minimizes the total fitting error across all models requires exploring an exponential number of possible partitions. □

**Corollary 8.2** (Approximation Guarantees)**.** *Sequential RANSAC provides a $\frac{1}{K}$-approximation to the optimal multi-model fitting solution under certain separability assumptions.*

## 8.2 Robustness Analysis

**Theorem 8.3** (Breakdown Point of Multi-Model RANSAC)**.** *The breakdown point of multi-model RANSAC with $K$ models is:*

$$\epsilon^* = \frac{1}{K+1} \tag{38}$$

*This is significantly lower than the* 50% *breakdown point of single-model RANSAC.*

**Implications:** The degradation in breakdown point motivates the development of more sophisticated algorithms that can handle higher outlier ratios in multi-model scenarios.

## 8.3 Convergence Properties

**Theorem 8.4** (Convergence of Iterative Multi-Model Fitting)**.** *Under appropriate conditions on model separability and initialization, iterative multi-model fitting algorithms converge to a local minimum of the multi-model objective function.*

**Conditions for Convergence:**

1. *Model Separability*: Models should be sufficiently separated in parameter space
2. *Initialization Quality*: Initial model estimates should be within the basin of attraction
3. *Noise Characteristics*: Noise should be bounded and have known statistical properties

| **Method** | **Complexity** | **Accuracy** | **Robustness** | **Scalability** | **Generality** |
|---|---|---|---|---|---|
| Sequential RANSAC | $O(KN)$ | Medium | Medium | Good | High |
| Hough Transform | $O(N \cdot M^d)$ | High | High | Poor | Medium |
| Preference Matrix | $O(N^2M)$ | High | Medium | Medium | High |
| Sparsity-Based | $O(N^2K)$ | High | High | Medium | Medium |
| Deep Learning | $O(N)$ | Variable | Variable | Good | Low |

Table 2: Comparison of multi-model fitting approaches across different criteria. $N$ is the number of data points, $K$ is the number of models, $M$ is the number of model hypotheses, and $d$ is the parameter dimension.

## 8.4 Performance Comparison

## 8.5 Future Research Directions

### 8.5.1 Theoretical Advances

1. *Approximation Algorithms*: Development of polynomial-time approximation algorithms with provable guarantees

2. *Sample Complexity*: Tighter bounds on the number of samples required for reliable multi-model fitting

3. *Robustness Theory*: Extended robustness analysis for multi-model scenarios

### 8.5.2 Algorithmic Improvements

1. *Adaptive Methods*: Algorithms that adapt to local data characteristics

2. *Hierarchical Approaches*: Multi-scale methods for handling large datasets

3. *Hybrid Techniques*: Combining multiple approaches for complementary strengths

### 8.5.3 Application Domains

1. *Autonomous Systems*: Real-time multi-model fitting for robotics and autonomous vehicles

2. *Medical Imaging*: Robust structure detection in medical image analysis

3. *Augmented Reality*: Real-time scene understanding for AR applications

## 8.6 Conclusion

Multi-model fitting represents a fundamental challenge in computer vision that requires sophisticated approaches to handle the inherent complexity of simultaneously clustering data and fitting models. While significant progress has been made in developing robust algorithms, several challenges remain:

1. *Scalability*: Handling large datasets with many models remains computationally challenging

2. *Automation*: Reducing the need for manual parameter tuning and threshold selection

3. *Generalization*: Developing methods that work across diverse application domains

The field continues to evolve with contributions from optimization theory, machine learning, and computational geometry. The integration of classical geometric methods with modern deep learning approaches promises to unlock new capabilities in robust multi-model fitting.

The theoretical foundations presented in this document provide a solid basis for understanding current methods and developing new approaches. As computer vision applications become increasingly complex, the importance of robust multi-model fitting techniques will only continue to grow.

Key Takeaway: Success in multi-model fitting requires careful consideration of the application domain, appropriate algorithm selection, and thorough parameter tuning.

# A Mathematical Derivations

## A.1 RANSAC Iteration Count Derivation

The probability that a randomly selected minimal sample set of size $m$ contains only inliers is:

$$p_{\text{clean}} = (1 - \epsilon)^m \tag{39}$$

The probability that at least one clean sample is selected in $K$ iterations is:

$$\begin{aligned} P_{\text{success}} &= 1 - (1 - p_{\text{clean}})^K && (40) \\ &= 1 - (1 - (1 - \epsilon)^m)^K && (41) \end{aligned}$$

Solving for $K$ given desired success probability $p$:

$$\begin{aligned} p &= 1 - (1 - (1 - \epsilon)^m)^K && (42) \\ 1 - p &= (1 - (1 - \epsilon)^m)^K && (43) \\ \log(1 - p) &= K \log(1 - (1 - \epsilon)^m) && (44) \\ K &= \frac{\log(1 - p)}{\log(1 - (1 - \epsilon)^m)} && (45) \end{aligned}$$

## A.2 Hough Transform Accumulator Resolution

For line detection using normal parameterization, the resolution requirements are:

**Angular Resolution:** The minimum angular resolution $\Delta\theta$ should satisfy:

$$\Delta\theta \leq \frac{1}{2\rho_{\max}} \tag{46}$$

**Distance Resolution:** The minimum distance resolution $\Delta\rho$ should satisfy:

$$\Delta\rho \leq \frac{\sigma_{\text{noise}}}{2} \tag{47}$$

where $\sigma_{\text{noise}}$ is the standard deviation of noise in the data.

# B Implementation Guidelines

## B.1 Pseudo-Code for Key Algorithms

**Algorithm B.1** (MSAC Implementation)**.** ***Input:*** *Data points $\mathcal{D}$, threshold $\tau$, iterations $K$*

***Output:*** *Best model $\theta^*$ and inliers $\mathcal{I}^*$*

```
1. Initialize: best_loss = $\infty$, $\theta^* = \emptyset$, $\mathcal{I}^* = \emptyset$

2. For $k = 1$ to $K$:
    (a) Sample minimal set $\mathcal{S}_k$ randomly from $\mathcal{D}$
    (b) Fit model $\theta_k$ to $\mathcal{S}_k$
    (c) Compute loss: $L_k = \sum_{i=1}^{N} \min(d(\mathbf{x}_i, \theta_k)^2, \tau^2)$
    (d) If $L_k <$ best_loss:
        i. best_loss $= L_k$
        ii. $\theta^* = \theta_k$
        iii. $\mathcal{I}^* = \{\mathbf{x}_i : d(\mathbf{x}_i, \theta_k) \leq \tau\}$
3. Refine $\theta^*$ using least squares on $\mathcal{I}^*$
```

## B.2 Parameter Selection Guidelines

| **Parameter** | **Recommended Range** | **Selection Criterion** |
|---|---|---|
| Inlier threshold $\tau$ | $[2\sigma, 3\sigma]$ | Based on noise statistics |
| Min consensus $\tau_{\text{min}}$ | $[m+1, 0.1N]$ | Application dependent |
| Max iterations $K$ | $[100, 10000]$ | Confidence and outlier ratio |
| Hough resolution | $[0.1, 1]$ | Accuracy requirements |

Table 3: Parameter selection guidelines for multi-model fitting algorithms

# C Experimental Validation

## C.1 Synthetic Data Experiments

**Experimental Setup:**

- Generate synthetic data with known ground truth models
- Add controlled amounts of Gaussian noise and outliers
- Evaluate accuracy, robustness, and computational efficiency

**Metrics:**

$$\text{Precision} = \frac{|\text{True Positives}|}{|\text{True Positives}| + |\text{False Positives}|} \tag{48}$$

$$\text{Recall} = \frac{|\text{True Positives}|}{|\text{True Positives}| + |\text{False Negatives}|} \tag{49}$$

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{50}$$

## C.2 Real Data Evaluation

**Datasets:**

- Adelaide RMF Dataset: Multi-model fitting benchmark
- York Urban Database: Urban scene analysis
- KITTI Dataset: Autonomous driving scenarios

**Performance Metrics:**

- Model fitting accuracy
- Computational time
- Memory usage
- Robustness to parameter variations