Point Correspondence Performance Evaluation

Robert M. Haralick

Computer Science, Graduate Center
City University of New York
Types of Performance Characterization

- **White Box**
  - Evaluate Each Component: Component Transfer Function
  - Needs Appropriate Random Perturbation Models for
    - Algorithm Inputs
    - Algorithm Outputs

- **Black Box**
  - Empirical Evaluation
  - No Knowledge of Component Transfer Functions
Performance Characterization

An Algorithm has
- Inputs $a$ and their data types
- Outputs $b$ and their data types
- A Relationship Between Input and Output
  - Output $b$ as a function of input $a$
  - Given $a$, $b$ maximizes $F(a, b)$

Random Perturbation Model for Input $a$
Random Perturbation Model for Output $b$
Given Random Perturbation Distribution Acting on $a$
Determine Random Perturbation Distribution Acting on $b$
Determine Robustness
  - Do large perturbations on a small fraction of the input data cause a small perturbation on the output data?
Performance characterization has to do with establishing the correspondence of the random variations and imperfections which the algorithm produces on the output data caused by the random variations and the imperfections on the input data.

(Haralick, 1994)
A system performance characterization has a scoring function that evaluates the goodness of the output. The system performance characterization gives the distribution of the scoring function value as a function of the parameters describing the input perturbation and the tuning parameters.
Experimental Protocols

- Population of ideal inputs
  - Simple Random Sampling
  - Stratified Sampling

- Parameters of random perturbation distribution affecting inputs

- Tuning Parameter Settings

- Scoring Function

- Fix Tuning Parameters
  - Estimate Scoring Function Distribution as a function of Perturbation Parameters

- Fix input
  - Estimate Scoring Function Distribution as a function of Tuning Parameters
Protocols

- Modeling
- Annotating
- Estimating
- Validating
- Propagating
- Optimizing

Haralick
Finding Points To Correspond

There are many methods that are used for finding point correspondences from images of multiple cameras. Among them are:

- Corner Points
- Interest Points
- Dense Subimage Matching
- Image Pyramids
- Correlation
- Distance
Finding Matching Points

Finding Matching points is often posed as an optimization problem and uses sensor projection geometry constraints

- Determine a Window Size
- Maximize Normalized Cross-correlation
- Minimize Normalized Distance
- Minimum Description Length
- Swarming
- Simulated Annealing
- Gradient Descent
- Expectation Maximization
- Mutual Information
- Total Least Squares
- Random Walks with Restart
- SoftPosit
- Energy Minimization
Performance Evaluation

Optimization and triangulation do not give a performance evaluation.
Performance Evaluation includes:

- Estimating the Covariance of the position of each 3D point
- The rule for deciding whether or not to accept the correspondences associated with an estimated 3D point
- The resulting False Alarm - Misdetect Rate
Kinds of Optical Sensor Models

(a) Pinhole
(b) Orthographic
(c) Pushbroom
(d) Cross-Slit
(e) Pencil
(f) Twisted Orthographic

Figure: From Yu, McCillan, and Sturm, 2010
Stereo Correspondence Problem

(a) Epipolar Geometry

(b) Uncertainty

Figure: From Unger and Stojanovic, 2013
Variety of Related Point Correspondence Problems

- Simultaneous Pose and Correspondence
- Sensors calibrated
- Structure from Motion
- Rigidity Checking
- Wide Baseline Stereo Correspondence
- Self-Consistency
Calibrated Sensors

- Ground Truth Point Correspondences
- Perspective Geometry
  - Standard Photogrammetric Procedure
  - Projective Bundle Adjustment
  - Interior Orientation
  - Exterior Orientation
The Multi-Image Point Correspondence Problem

- There are $N > 1$ calibrated sensors
- True but unknown sensor parameters $\theta_1, \ldots, \theta_N$
- 3D Point $q$ whose position is not known
- 2D Corresponding points $x_1, \ldots, x_N$, the sensor projections of $q$ to the $N$ sensors
- $\xi_1, \ldots, \xi_N$ random perturbations of 2D sensor projection points
- Estimated sensor parameters $\hat{\theta}_1, \ldots, \hat{\theta}_N$
- Model:
  - $x_n = P_n(q, \theta_n)$
  - $\hat{x}_n = P_n(q, \theta_n) + \xi_n$, $n = 1, \ldots, N$
  - $\xi_n$ has $N(0, \Sigma_{\xi_n})$
  - $\hat{\theta}_n$ has multivariate uniform
  - $\hat{\theta}_1, \ldots, \hat{\theta}_N, \xi_1, \ldots, \xi_N$ are independent
In the Bayesian setting, true sensor parameters are considered as random variables with independent a priori densities

\[ p_1(\theta_1) \ldots, p_N(\theta_N) \]
Estimate $q = (x, y, z)$ to maximize

$$p(q \mid \hat{x}_1, \ldots, \hat{x}_N, \hat{\theta}_1, \ldots, \hat{\theta}_N)$$

This is equivalent to estimate $q$ to maximize

$$p(\hat{x}_1, \ldots, \hat{x}_N, \hat{\theta}_1, \ldots, \hat{\theta}_N, q)$$

(Bedekar and Haralick, 1995)
The Bayesian Estimation Problem

\[ p(\hat{x}_1, \ldots, \hat{x}_N, \hat{\theta}_1, \ldots, \hat{\theta}_N, q) = p(\hat{x}_1, \ldots, \hat{x}_N \mid \hat{\theta}_1, \ldots, \hat{\theta}_N, q) \times p(\hat{\theta}_1, \ldots, \hat{\theta}_N, q) \]

Given \( \hat{\theta}_1, \ldots, \hat{\theta}_N \) and \( q \), the sensor projections \( \hat{x}_1, \ldots, \hat{x}_N \) are conditionally independent. Hence,

\[ p(\hat{x}_1, \ldots, \hat{x}_N \mid \hat{\theta}_1, \ldots, \hat{\theta}_N, q) = \prod_{n=1}^{N} p_n(\hat{x}_n \mid \hat{\theta}_1, \ldots, \hat{\theta}_N, q) \]

The sensor projection \( \hat{x}_n \) only depends on the 3D point \( q \) and its associated sensor parameters \( \theta_n \). Hence,

\[ \prod_{n=1}^{N} p_n(\hat{x}_n, \mid \hat{\theta}_1, \ldots, \hat{\theta}_N, q) = \prod_{n=1}^{N} p_n(\hat{x}_n \mid \hat{\theta}_n, q) \]
The calibration that established the estimates $\theta_n$ are certainly independent of each other and independent of the 3D point $q$. Hence,

$$p(\hat{\theta}_1, \ldots, \hat{\theta}_N, q) = p(q) \prod_{n=1}^{N} p_n(\hat{\theta}_n)$$
The Optimization

\[
p(\hat{x}_1, \ldots, \hat{x}_N, \hat{\theta}_1, \ldots, \hat{\theta}_N, q) = p(q) \prod_{n=1}^{N} p_n(\hat{x}_n | \hat{\theta}_n, q)p_n(\hat{\theta}_n)
\]

\[
\log p(\hat{x}_1, \ldots, \hat{x}_N, \hat{\theta}_1, \ldots, \hat{\theta}_N, q) = \log p(q)
\]

\[
-\frac{1}{2} \sum_{n=1}^{N} (\hat{x}_n - P_n(q, \hat{\theta}_n))' \Sigma_{\hat{x}_n}^{-1} (\hat{x}_n - P_n(q, \hat{\theta}_n))
\]

\[
+ \sum_{n=1}^{N} \log p_n(\hat{\theta}_n)
\]

where

\[
\Sigma_{\hat{x}_n}(q, \hat{\theta}_n) = \Sigma_{\xi_n} + \frac{\partial P_n}{\partial \theta}(q, \hat{\theta}_n) \Sigma_{\hat{\theta}_n} \frac{\partial P_n}{\partial \theta}(q, \hat{\theta}_n)'
\]

\(p(q)\) is the prior for 3D point \(q\), taking into account all the 3D points that have already been triangulated and \(p_n(\hat{\theta}_n)\) is a multivariate uniform.
Robustification

Objective Function

\[
\log p(q) - \frac{1}{2} \sum_{n=1}^{N} (\hat{x}_n - P_n(q, \hat{\theta}_n))^\prime \Sigma_{x_n}(q, \hat{\theta}_n)^{-1} (\hat{x}_n - P_n(q, \hat{\theta}_n)) + \sum_{n=1}^{N} \log p_n(\hat{\theta}_n)
\]

- The perturbations are small
- The optimization is only correct if in fact each \(\hat{x}_n\) does correspond to the 3D point \(q\)
- But sometimes the correspondence is not correct
- Robustify the objective function
The optimization provides an estimate $\hat{q}$ of $q$

The covariance $\Sigma_q$ needs to be estimated

The consistency of $\hat{q}$ with respect to $\hat{x}_1, \ldots, \hat{x}_N$ has to be checked

(Haralick, 1994)
Covariance Propagation

- \( X = (x_1, \ldots, x_N) \)
- \( \hat{X} = (\hat{x}_1, \ldots, \hat{x}_N) \)
- \( F(q, X) = 0 \)
- Minimize \( F(\hat{q}, \hat{X}) \)
- \( G(q, X) = \frac{\partial F}{\partial q} \)

\[
\Sigma_q = \left( \frac{\partial G}{\partial q} \right)^{-1} \frac{\partial G}{\partial X} \Sigma_X \left( \frac{\partial G}{\partial X} \right)^\prime \left( \frac{\partial G}{\partial q} \right)^\prime^{-1}
\]

- \( \frac{\partial G}{\partial q}(\hat{q}, \hat{X}) \)
- \( \frac{\partial G}{\partial X}(\hat{q}, \hat{X}) \)
Covariance Propagation

- $\tilde{x}_n$ reprojection of estimated 3D point

$$\tilde{x}_n = P_n(\hat{q}, \hat{\theta}_n)$$

$$\Sigma_{\tilde{x}_n}(\hat{q}, \hat{\theta}) = \frac{\partial P_n}{\partial \theta} \Sigma_{\hat{\theta}_n} \left( \frac{\partial P_n}{\partial \theta} \right)' + \frac{\partial P_n}{\partial \hat{q}} \Sigma_{\hat{\theta}} \left( \frac{\partial P_n}{\partial \theta} \right)'$$

$$+ \frac{\partial P_n}{\partial \hat{\theta}} \Sigma_{\hat{\theta}} \left( \frac{\partial P_n}{\partial \hat{q}} \right)' + \frac{\partial P_n}{\partial \hat{q}} \Sigma_{\hat{\theta}} \left( \frac{\partial P_n}{\partial \hat{q}} \right)'$$

- $\frac{\partial P_n}{\partial \hat{q}}(\hat{q}, \hat{\theta})$
- $\frac{\partial P_n}{\partial \hat{\theta}}(\hat{q}, \hat{\theta})$
Empirically measure the predictive power of a score with respect to a given algorithm, population of scenes and imaging conditions.

Decide $\hat{x}_n$ is a corresponding point if

$$(\tilde{x}_n - \hat{x}_n)'(\Sigma_{\tilde{x}_n} + \Sigma_{\xi_n})^{-1}(\tilde{x}_n - \hat{x}_n) < \tau_n$$

Acceptance rate: the fraction of corresponding point sets that are close enough to the sensor projection of their estimated 3D points.

If $\hat{x}_n$ is decided as not a corresponding point, look near $\tilde{x}$ for a corresponding point.

(Leclerc and Luong, 2003)
Scene and Image Populations

- Textureless Regions
- Textured Regions
- Unoccluded Corners
- Amount of Noise
- Depth Discontinuities
Performance Characterization

- Specify a population
- Label the true corresponding points: Ground Truth
- Use automatic procedure for finding corresponding points
- Estimate the 3D points
- Determine the projection of the 3D points
- Consider only accepted points
- Matching Error compared to Ground Truth
  - Cumulative Distribution of distance to true position
  - Threshold defining when distance is close enough
  - Misdetect and False alarm rate
Misdetect False Alarm Rate

Misdetect Rate vs False Alarm Rate

- Threshold increasing
- (.071, .201) threshold = 1
- Threshold decreasing

False Alarm Rate
Every algorithm has tuning parameters.

- How the Tuning Parameters are set influences performance
- Some Tuning Parameters are set internal to program
- Some Tuning Parameters can be user set
- Default Settings vs Tuned Settings
- What is the sensitivity of the result to the settings of the tuning parameters?
Performance Surface vs Tuning Parameters

- Ruggedness of Surface
- Smoothness of Surface
- Number of Local Optima
- Ratio of Local Optima values to Global Optima

Crossley, Nisbet, and Amos (2013)
Set a threshold $\theta$ of minimum acceptable performance

Determine a hyperbox having the property

- The fraction $f$ of tuning parameter values in hyperbox yield performance $> \theta$
Estimating Performance Hyperbox Boundaries

- **N Experiments**
- Choose tuning parameter $M$-tuples at random
- Evaluate Performance
- Determine Hyperbox Boundaries
  - Tuning Parameters $(\alpha_{1n}, \ldots, \alpha_{Mn}), n = 1, \ldots, N$
  - Goodness Function $\Psi$
  - Acceptable Set $A(\theta) = \{ n \mid \Psi(\alpha_{1n}, \ldots, \alpha_{Mn}) > \theta \}$
  - $b_{min} = \min_{n \in A} \alpha_{mn}$
  - $b_{max} = \max_{n \in A} \alpha_{mn}$
  - $H = \times_{m=1}^{M} [b_{min}, b_{max}]$
Goodness Fraction

- **N Experiments**
- Choose Tuning Parameter $M$-tuples at random in $\mathcal{H}$
  - $(\alpha_{1n}, \ldots, \alpha_{Mn}), \ n = 1, \ldots, N$
- Evaluate Goodness $\Psi$
- Estimate Goodness Fraction
  - $f = \frac{|\{n \mid \psi(\alpha_{1n}, \ldots, \alpha_{Mn}) > \theta\}|}{N}$
- Estimate Worst Goodness
  - $\psi_{\text{worst}} = \min_{n=1,\ldots,N} \psi(\alpha_{1n}, \ldots, \alpha_{Mn})$
- Find largest Hypercube $\mathcal{H}_C \in \mathcal{H}$ such that
  - $(\gamma_1, \ldots, \gamma_M) \in \mathcal{H}_C$ implies $\psi(\gamma_1, \ldots, \gamma_M) < \theta$
References

References

References


Slides can be found at:

http://haralick.org/conferences/point_correspondence_metrics.pdf