Video-based Car Surveillance:
License Plate, Make, and Model Recognition

UCSD Master’s Thesis 2005
Louka Dlagnekov
Presenter: Brendan Morris

Introduction

• Low Cost License Plate Recognition System
  – No Need for Expensive Hardware for High Quality Video or Other Sensors
• Extend LPR to More General Make and Model Recognition
  – Database Queries Possible with Partial License Plate and Car Visual Description
Outline

• License Plate Detection (LPD)
• License Plate Recognition (LPR)
  – Tracking
  – Optical Character Recognition (OCR)
• Make and Model Recognition (MMR)
• Conclusions and Future Work

License Plate Detection

• Window Search Over Entire Frame
  – 3 Different Sized Windows
  – Independent Classifier for Each Size
• Strong Classifier Constructed from Weak Classifiers Via AdaBoost
  – Computationally Simple
AdaBoost

• Adaptive Algorithm Constructs a Strong Classifier as a Combination of Weaker Classifiers
• Build Initial Classifier Model
• Identify Samples not Explained by Model
  – Mis-Classified Samples
• New Model Built Using New Training Set which Includes the Difficult Mis-Classified Samples from the Previous Model

Optimizations

• Fast Detection Rates
  – 640 x 480 Image Size
  – 10 Frames/sec
• Viola and Jones (2001)
  – Integral Images
  – Cascaded Classifiers
**Integral Images**

- MN Array Accesses for MxN Array (2400 Simple Classifier)
- Use Rectangular Structure to Reduce Accesses to 4
- Sum of Pixels Above and to Left

\[ I'(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y'). \]

\[ D = I'(w) + I'(z) - (I'(x) + I'(y)). \]

**Cascaded Classifiers**

- Group Classifiers into Several Stages in Order of Increasing Complexity
  - Simple Effective Early Classifiers can Reject Most Erroneous Regions
- Train Stages on False Positives of Previous Stage

\[ P = \prod_{i=1}^{K} p_i, \quad N = \prod_{i=1}^{K} n_i, \]
LPD Results

- Detector Trained on Several Scales
- Many False Positives Come From Other Text in Scene

License Plate Recognition

- Use Detection Result to Construct Tracks
  - Robust Plate Detection
  - Enforce Track Smoothness Constraints
  - Multiple Detections for Super-Resolution
- Optical Character Recognition (OCR)
  - NCC Template Matching
Super-Resolution

• Multiple Low Res Samples ($L_k$) Used to Construct Single High Res (H) Image
  $$\tilde{L}_k(x, y) = S \downarrow (h(x, y) + H(T_k(x, y))) + \eta(x, y),$$

• Estimate H Given $L_k$
  – Register Tracks with NCC ($T_k$)
  – Use Gaussian PSF ($h$)
  – Additive Gaussian noise ($\eta$)
  – Down Sample by 2 or 4 ($S$)

• Used to Separate License Characters

Super-Resolution Algorithm

• Maximize

• MLE
  – No Priors – All $\hat{H}$ Equally Likely

• MAP
  – Solved Using Gradient Descent Methods
    - Smoothness Prior
    $$P_{\lambda}(\hat{H}(x, y)) = c_1 e^{-\lambda H(x, y) - \tilde{\lambda}(x, y)},$$
    - Bi-Modal Prior
    $$P_{\lambda}(\hat{H}(x, y)) = c_2 e^{-\lambda |H(x, y) - \mu_1|^2 H(x, y) - \mu_2|^2},$$
LPR Results

• Edit (Levenshtein) Distance for Accuracy Measure
  – Loose Measure – Avoids Penalties for Commonly Mistaken Characters {Z, 2}, {B, 8}

Make and Model Recognition

• Selected Car ROI from LPD
  – Placed in Canonical Position
• Compared Different Recognition Algorithms
  – Appearance-Based Methods
    • Eigencars
  – Feature-Based Methods
    • Shape Context Matching
    • SIFT Matching
Eigencars

- Dimensionality Reduction Using Principle Component Analysis (PCA)
  - Car Image (Pixel Intensities) as Feature Vector
- Project Each Car Image to Lower Dimensional Space
  - Classify Match as Closest ($L_2$ Distance) Database Car

Eigencars Results

- Recognition Rate of 23.7%
  - Recognition Rate of 2.5% for Random Guessing
- Improvements
  - Discard Largest Eigenvalues
    - 44.7 – 47.4% Recognition
  - Fisherface Method
Improved Eigencars

- Using All N Eigencars
- Using N-3 Eigencars

Feature Extraction

- Corner Detectors
  - Harris and Förstner
- Salient Features
  - High Entropy
  \[ H(s,x) = \sum_{i} p_{i}(i) \log p_{i}(i) \]
- SIFT Features
  - Scale Invariant Feature Transform
### Shape Contexts

- **Radial Edge Pixel Count Histogram**

- **Usually Compared Using Chi-Squared or L₂ Distance**

\[
d(h_i, h_j) = \sum_{k \text{ even}} \left( \frac{\|h_i(k) - h_j(k)\|^2}{\|h_i(k) + h_j(k)\|^2} \right)
\]

### Shape Context Matching

- For Database Entries, d, and Query Images, q, Take N Random Sample Points of Corresponding Edge Images and Compute Shape Context Around Each Point
Shape Context Matching

• For Each d
  – For Each Sampled Edge Point, \( p_q \), in \( q \) Find Best Matching Point \( p_d \) Within a Radius Threshold Using Chi-Squared Distance
  – Create Match Cost as Sum of Distances For Every Correspondence
• Choose d with Lowest Cost as Match

Shape Contest Results

• Descriptor Radius – 35 Pixels
• Sampling Size – \( N = 400 \) Points
• 65.8% Recognition
  – 5 x 12 Shape Context
• 63.2% Recognition
  – 9x4 Shape Context
SIFT Features

• 4 Step Procedure
  – Scale-Space Extrema Detection
  – Keypoint Localization
  – Orientation Assignment
  – Descriptor Assignment

• Scale Space
  \[ L(x, y, \sigma) = G(x, y, \sigma) \ast I(x, y), \]
  \[ G(x, y, \sigma) = \frac{1}{2\pi\sigma^2}e^{-\frac{(x^2+y^2)}{2\sigma^2}}. \]

\( \sigma \) – quantized scale factor

SIFT Features

• Keypoint Localization
  – Find Extrema in
  \[ D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma), \]

• Orientation Assignment
  \[ m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \]
  \[ \theta(x, y) = \tan^{-1} \frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)}. \]

• Descriptor Assignment (16 x 8 = 128 Dim)
  – Divide Region Around Keypoint into 16 Symmetric Sub-Regions and Create 8 Orientation Bins
SIFT Descriptor

• Scale and Rotation Invariant
  – $\sigma$ – Scale Factor
  – Keypoint Orientation

SIFT Matching

• Extract SIFT Features for Each Image $d$, Database Entry, and $q$, a Query Image
• For Each $d$
  – For Each Keypoint, $k_q$, in $q$ find $k_d$ with Smallest $L_2$ Distance and is at Least a Factor of $\alpha$ Smaller than Second Nearest Neighbor
  – Count Number of Matched Descriptors
• $d$ with Largest Count as Best Match
SIFT Results

- (After Applying Keypoint Pruning)
- 89.5% Recognition Rate

MMR Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eigencars using all eigenvectors</td>
<td>23.7%</td>
</tr>
<tr>
<td>Eigencars without 3 highest</td>
<td>44.7%</td>
</tr>
<tr>
<td>Shape context matching with 9 x 4 bins</td>
<td>63.2%</td>
</tr>
<tr>
<td>Shape context matching with 5 x 12 bins</td>
<td>65.8%</td>
</tr>
<tr>
<td>SIFT matching</td>
<td>89.5%</td>
</tr>
</tbody>
</table>

- Achieved High Recognition Rates
  - Mis-Classifications had Few (<5) Database Examples
- High Recognition at Cost of Computation
  - 30 sec for Shape Context and SIFT vs 0.5 for Eigencars
Conclusion

- Developed Car Recognition Framework Combining LPR and MMR
- Can Be Used in a Query Based Car Surveillance System
- High Recognition Rates
  - Only LPR Currently Real-Time

Future Work

- MMR
  - Speed Up Recognition (Real-Time Application)
    - Group Database into Vehicle Type {SUV, Truck, ...}
    - Formulate as Text Retrieval (Sivic and Zisserman)
- Add Color Inference
- Database Query Algorithm Development
- Make and Model 3D Structure