Robust Feature Matching in 2.3µs

Simon Taylor
Edward Rosten
Tom Drummond

University of Cambridge

Motivation

- Matching speed is of key importance in real-time vision applications

- Frame-to-frame tracking can be efficient, but requires initialisation
- Therefore fast localisation methods are needed

Local Feature Methods

- Naturally handle partial occlusion and some incorrect correspondences
- Represent a target as a small set of key features (~100s)
- Attempt to identify and match key features in any view of target

Existing Local Feature Approaches

- Descriptor-based, e.g. SIFT
  - Factor out as much variation as possible
  - Soft-binned histograms
- Clasification-based, e.g. Ferns
  - Train classifiers on different views of the same feature
  - Lower runtime computational cost, but high memory usage

Why not make it easier...

- Just require features to match under small viewpoint variations

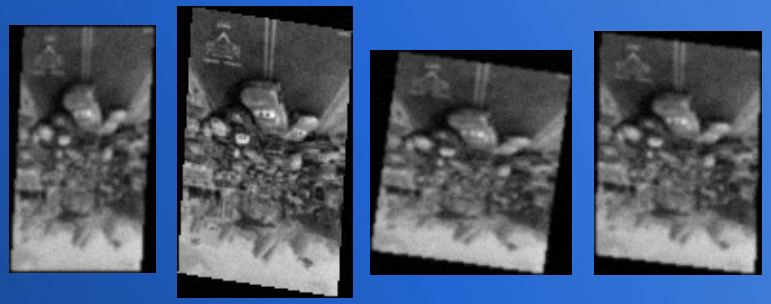
- Simplifies matching problem
- Independent sets of features can handle large viewpoint change
- ...but will need lots of features

Overall Approach

- Classification-based, as runtime speed is key
- Desired runtime operations:
  - FAST-9 Corner Detection
  - Simple “descriptor”
  - Efficient dissimilarity score computation
  - (PROSAC for pose estimation)

Generating Training Images

- Artificially warp a single reference image
- Add blur and noise
- Group into viewpoint “bins”

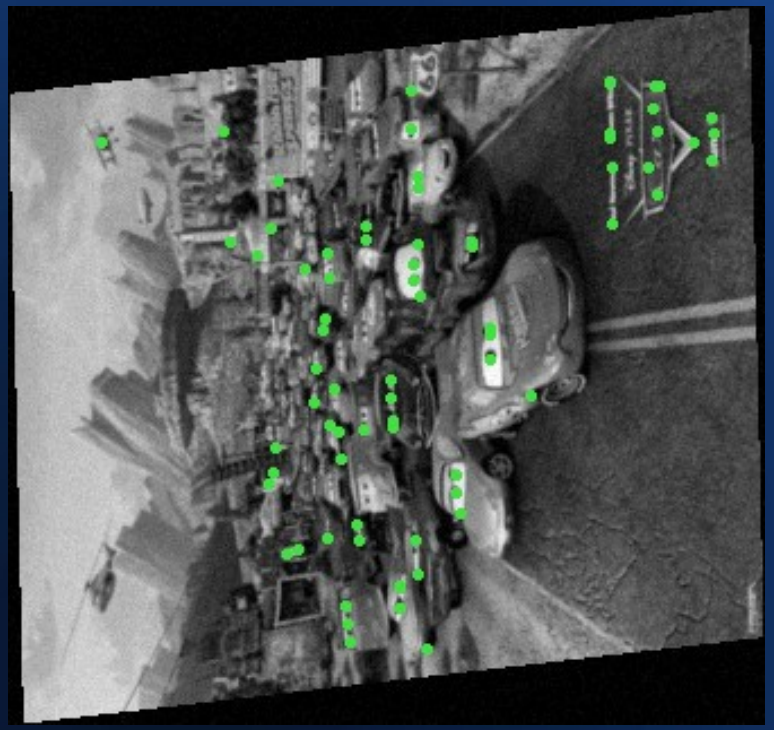


Generating Training Images

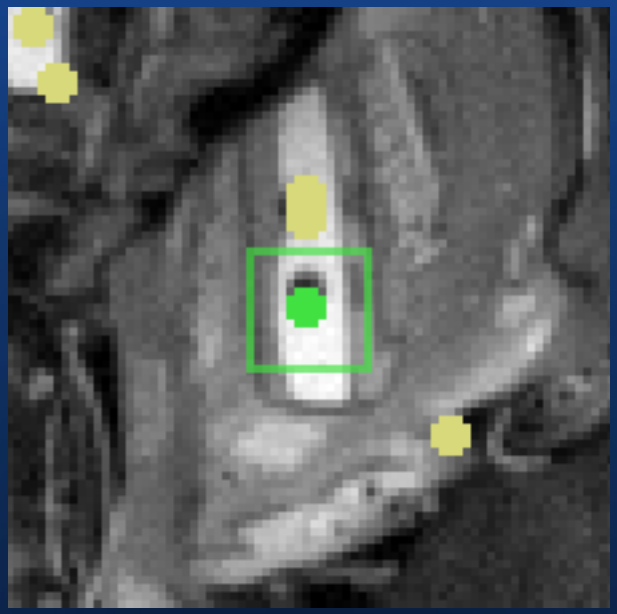
- Each viewpoint bin covers:
  - 10 degrees of camera axis rotation
  - Scale reduction by factor 0.8
  - Affine changes up to 30 degrees out-of-plane
- Have 36 rotation ranges and 7 scale ranges
  - 252 bins in total
- Around 50 features from each viewpoint
  - So around 13000 features for a target

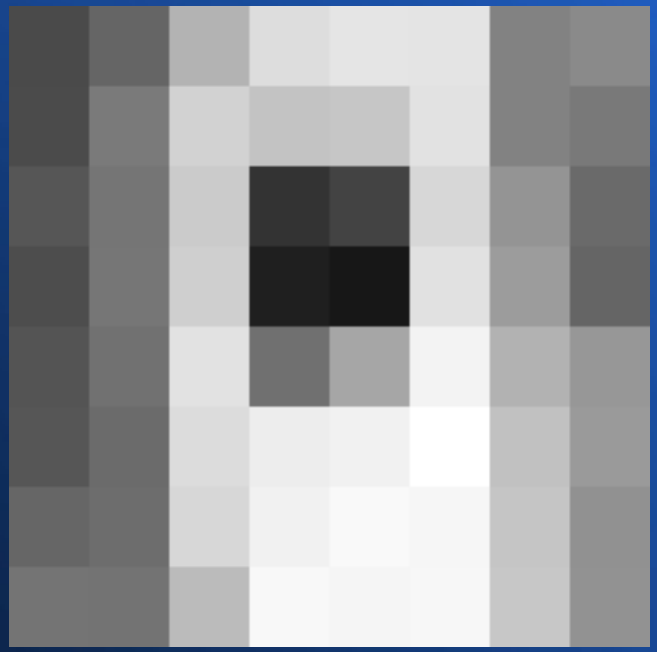
Training Phase

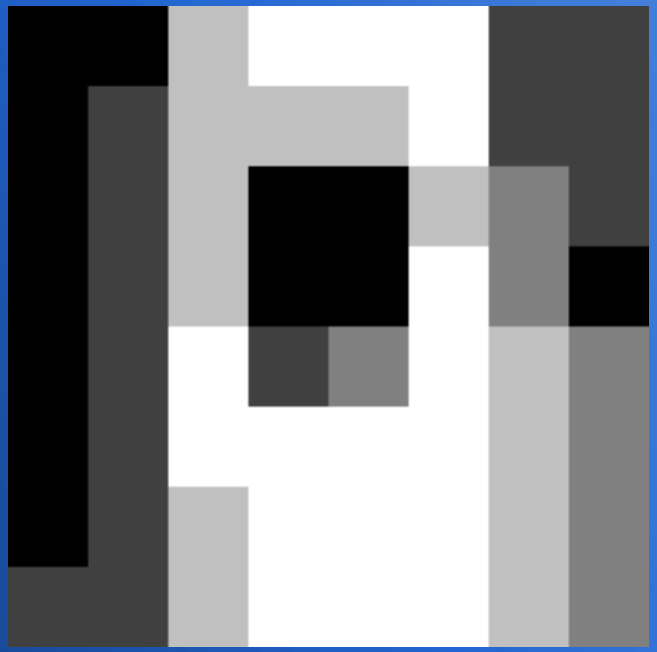
- Detect FAST-9 corners in every training image in viewpoint bin



Quantised Patches







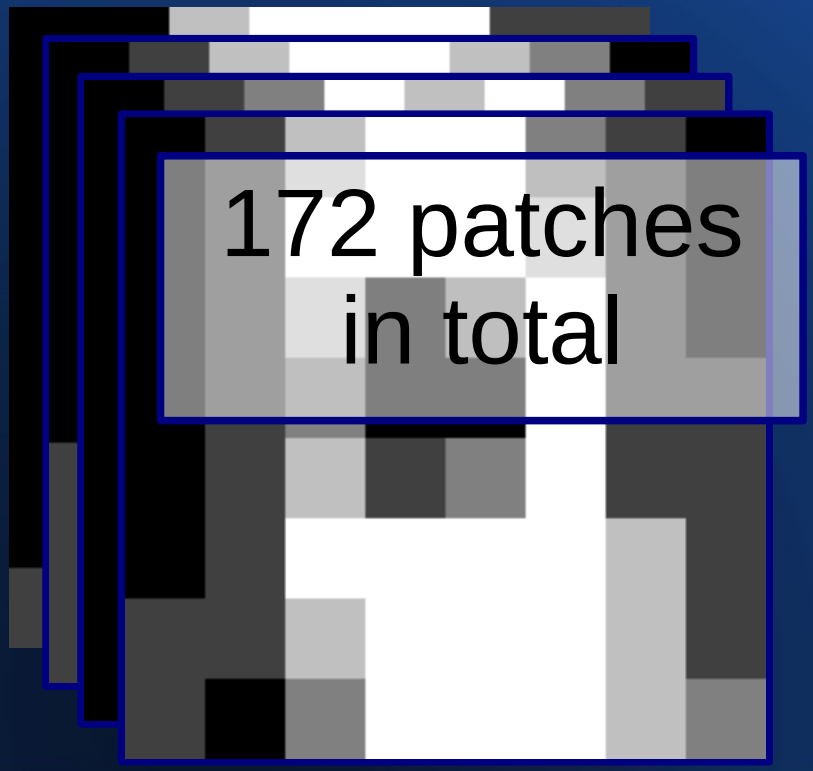
- Sparsely sample patches around corners
- Quantise to 5 levels, relative to mean and standard deviation of samples

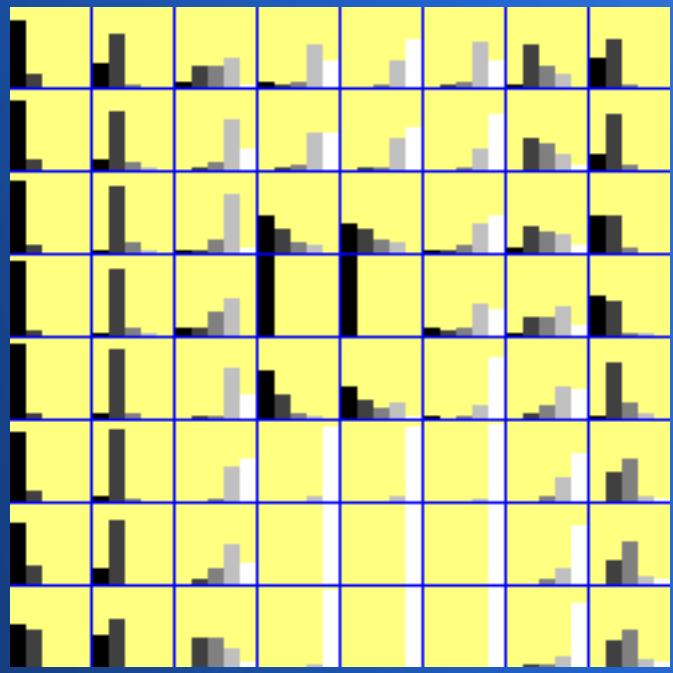
Combining Quantised Patches

- All patches from first training image are added as features
- For remaining training images:
  - Compare FAST corner locations with existing features
    - If within 2px of existing feature, patch added to existing model
    - Otherwise new feature added

Combining Quantised Patches

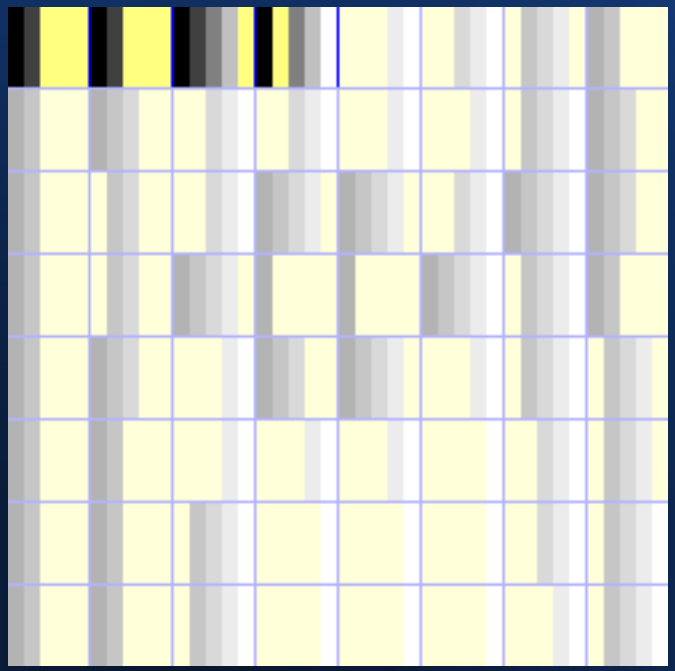
- Feature model uses per-sample histograms





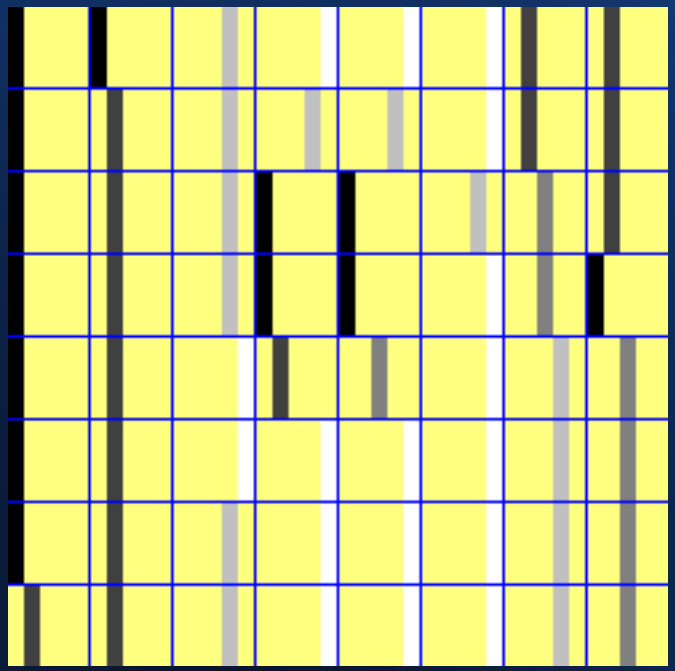
Binary Histogram Representation

- Histograms quantised to binary representation
- 1s represent “rarely” seen intensities



Runtime Phase

- Run FAST, sample patch, quantise intensities
- Represent as binary – exactly 64 1s



1100..0000..0000..0010..0001..

Dissimilarity Score

- Count of “rare bits” in runtime patch
  - Bitcount of ANDed model and patch

```
Feature model: 0000..0001..1100..1100..1110..
Runtime patch: 1100..0000..0000..0010..0001..
        ANDed: 0000..0000..0000..0000..0000..
```

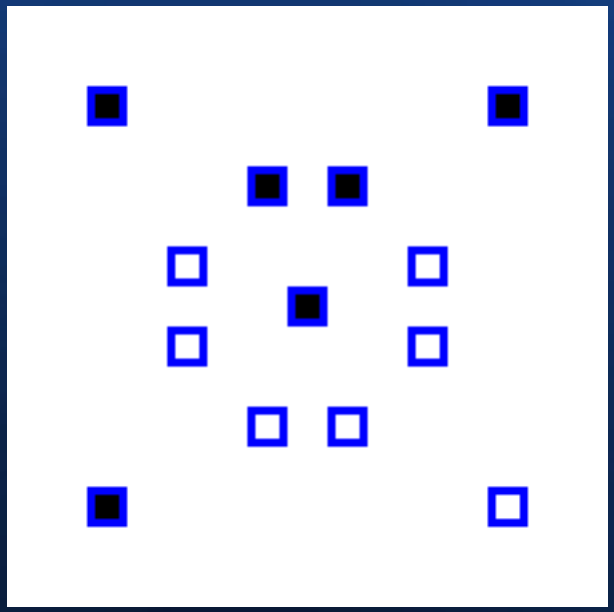
- Bit arrangement means only a 64-bit count is needed
- Low threshold for matches – typically < 5.

It works, but...

- It's a bit slow
  - Individual dissimilarity score just 20ns to compute
  - ...but we have over 10000 features
- Blur is a problem
  - Training set images include random blur
  - ...but FAST repeatability is the main problem

Indexing Scheme

- Assign index number based on 13 samples



000011011101
(445)

000011011101 (445)

000011011101 (445)

000011011100 (444)

000001011101 (189)

Indexing Scheme

- Select most common indices until 80% of samples included
- Feature added to each of these bins
- At runtime only search a single bin

Index Number	Number of Training Samples
445	36
189	32
2365	22
2237	10
3373	9
701	8
957	8
1981	8
444	4
3901	3
17 others	22 total

Image Pyramid

- Attempt to match into full image
  - FAST, extract patches, matching, pose
- If insufficient inliers, half-sample image to obtain more correspondences
- Half-sample again if necessary

- Side effect: Allows matching closer to target

The System in Action...

Results

- Comparison to exhaustive nearest-neighbour SIFT matching

	Frames Matched
Our Method, 50 PROSAC Iterations	625
Our Method, 5000 PROSAC Iterations	**632**
SIFT, 150 PROSAC Iterations, Distance	590
SIFT, 150 PROSAC Iterations, Ratio	629
SIFT, 5000 PROSAC Iterations, Ratio	630

Results

- Wagner et al. modified SIFT and Ferns with focus on mobile platforms

	Our Method	Wagner et al.
Full Frame Time	1.04ms	~5ms
Robustness	99.6%	~96%

Conclusions

- Only requiring limited invariance to viewpoint can enable the use of small and fast features
- Future Work
  - Orientation normalisation
  - Viewpoint consistency constraints
  - Optimisation of parameters
  - Quantised histograms of additional image properties?