Visual Object Recognition using Template Matching

Luke Cole[1,2], David Austin[1,2], Lance Cole[2]

December 8, 2004

[1]Robotic Systems Lab, RSISE
Australian National University,
ACT 0200, Australia

[2]National ICT Australia,
Locked Bag 8001,
Canberra, ACT 2601

Quick Overview of Template Matching

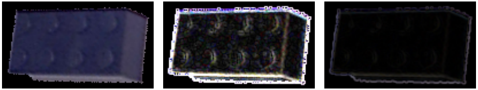
- This is an old well established technique.
- A simple task of performing a correlation between a template image (object in training set) and a new image to classify.
  - Sum of All Differences (SAD)
  - Sum of Square Differences (SSD)
  - Normalised Cross Correlation (NCC)
- Below: Raw Template (left), Edge Based Template (right). For each image set: test image (left), template (right).



The Research

- Template Matching is a rich object detector. Captures entire essence of an object (not the case for many "higher-order" techniques).
- Some object have no or poor internal features so they are not well suited to "higher order" techniques. E.g. aspect graphs use edge features. It's not always possible/easy to detect edges.
- So what is the problem with Template Matching? It's expensive!
- This research addresses this scaling problem with results based on 91 classes and 140 000 extracted blobs each of size 680x480.
- Biologically inspired for real time long-term visual robotic systems.

Not always easy to detect edges



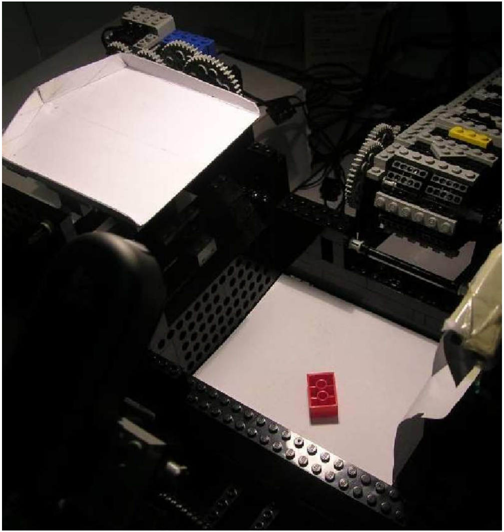
Approach Introduction

- Training database acquisition and extraction.
- Training database reduction to create template images.
- Random classification via NCC's as it is the best form of correlation and the most expensive.

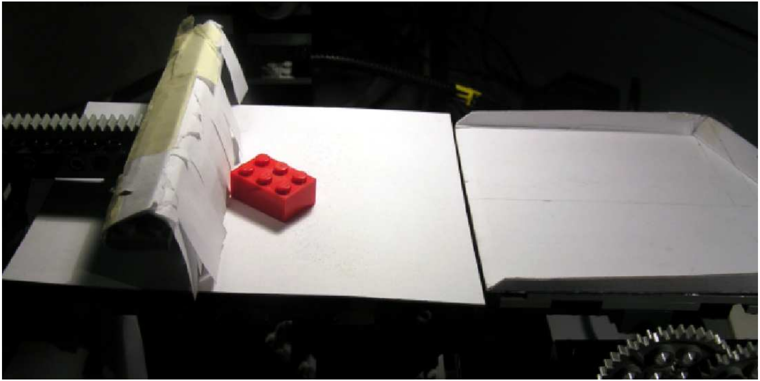
The Object Database

- **Lego Bricks**
- 140 000 image with 91 bricks, approximately 1000 different views for each class.
- Why is it a good database?

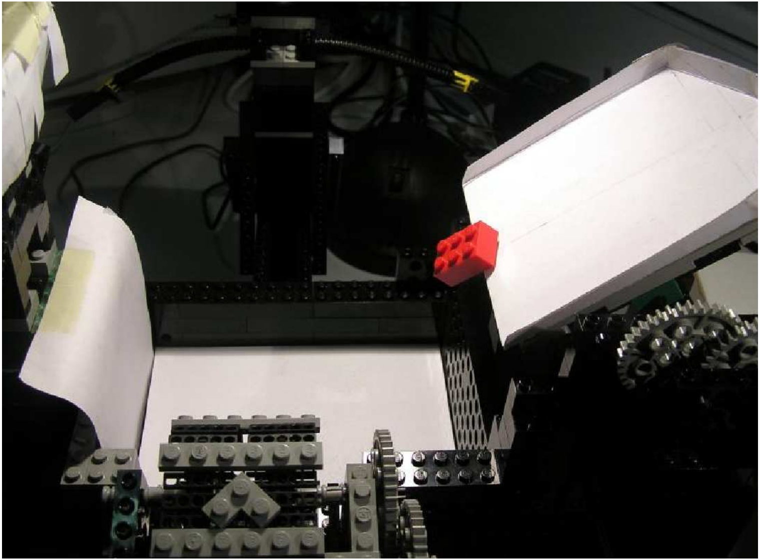
Training Database Acquisition



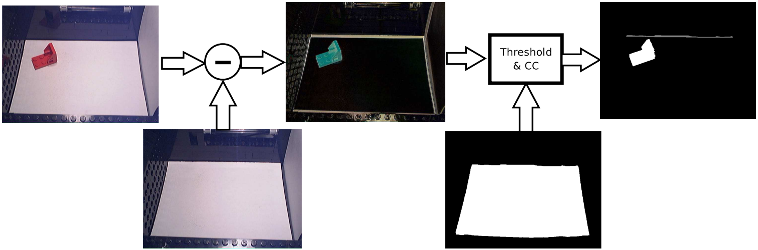
Training Database Acquisition

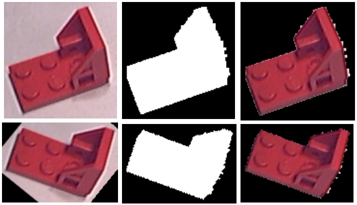


Training Database Acquisition



Training Database Extraction

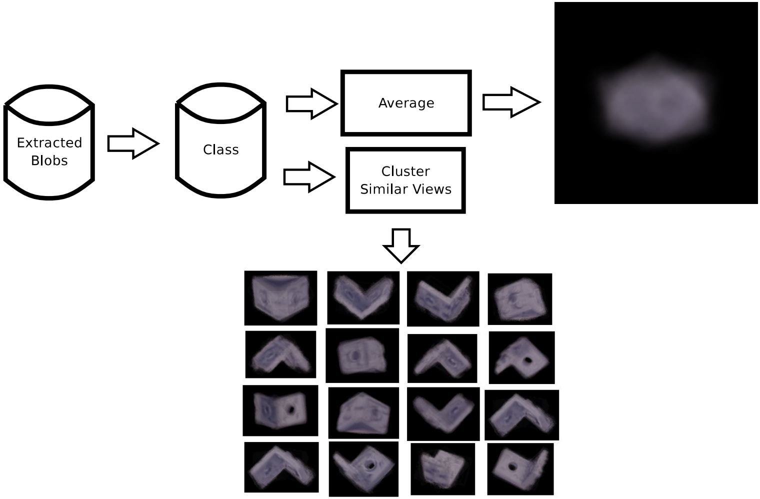




Training Database Reduction

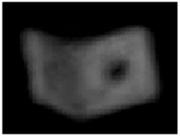
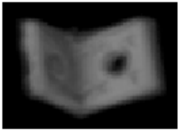
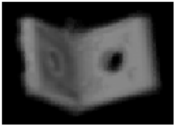
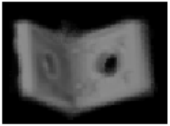
- Classifying a new test image across all the extracted blobs would be computationally infeasible.
- So we reduce the set (since we expect similar and incorrect images).
- If two images are similar, we do not simply keep one image and remove the rest. Instead, a clustering approach was taken.
- Each class is represented by a two-tier hierarchical structure.

Training Database Reduction

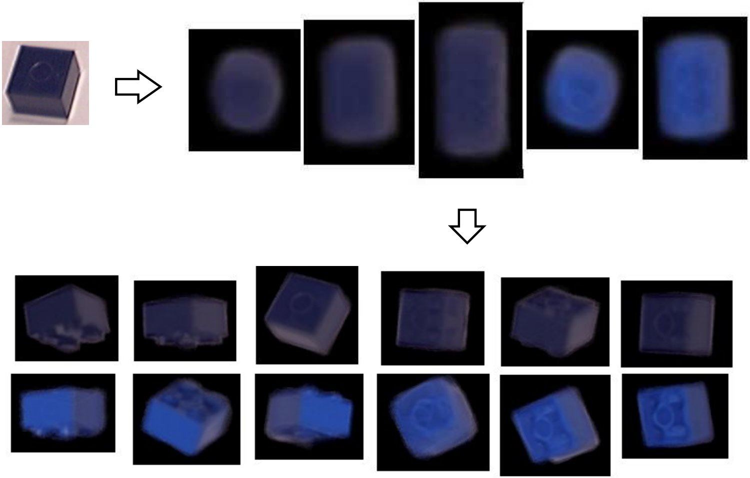


Training Database Reduction

- Obviously determining the correct NCC threshold would be a task in itself. So our results are based on four reduced sets with the NCC threshold equal to 0.75, 0.8, 0.85, 0.9.

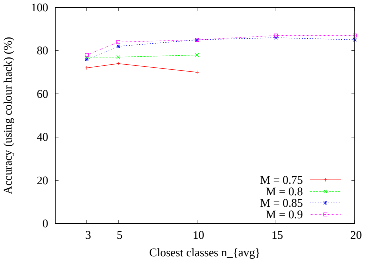
Recognition Procedure

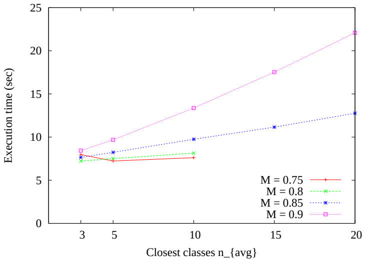


Results

- C/C++ Implementation.
- Images obtained from a standard webcam (640x480).
- Results obtained on a AMD Athon(tm) XP 2700+ with 1GB of memory, running Debian Linux.
- Different reduced sets (M) and closest classes (n_{avg}).

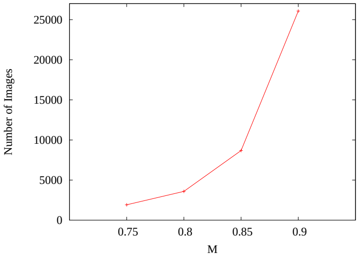
Accuracy and Execution Time

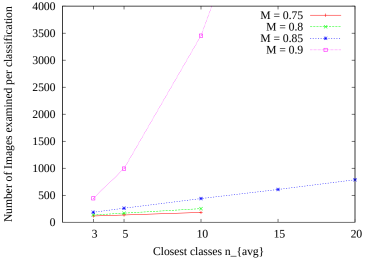




- Lastest result: 90% in 6.75 seconds for $n_{avg} = 15$, $M = 0.9$

Reduced and Examined Images





Conclusions

- Uses all of the information about each object
- Not exactly real-time, however still favorably over more complex methods that take many minutes (NCC optimizations).
- Clustering and averaging seems an interesting way to catalogue and classify an object.
- Large computation required for unsegmented recognition

Future Work

- More rigorous method to extracting and clustering.
- The green factor!
- Hardware implemention to template matching (FPGA).
- More camera views.
- Physical interaction.

Acknowledgements

This work was supported by funding from National ICT Australia. National ICT Australia is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Centre of Excellence program.