PGM Interactive Object Recognition Project

Chet Corcos, Karol Hausman, Timmy Mbaya

April 8, 2014

Contents

1 Introduction

The goal of this project is to create an object recognition algorithm that determines optimal actions for recognition. We will train our model on a variety of objects in a variety of poses with known actions between the poses, measuring/extracting/computing a variety of features. Certain features associated with an object may only be visible in a certain pose. The model in this document will be a simplified model with discrete poses. Future work can incorporate continuous pose and action variables.

2 Related Work

Research in perception has traditionally focused on static images and recognized objects based on a set of visual features such as SIFT [6], SURF [1] or ORB [7]. There has also been plethora of work on static object recognition that resulted in various systems.

One of the most efficient and robust object recognition systems was developed by Tang et al. [8]. It is worth mentioning that the system took the 1st place in the ICRA 2011 Solutions in Perception instance recognition challenge. The system can be divided into several steps. In the training phase a full object model is created, then it is transformed into a mesh, and finally, the RGB data is projected onto the object model acquired in the previous steps. The authors decided to use SIFT features and color histograms in order to fully describe the textural property of the object. In the testing phase plane extraction and euclidean clustering algorithms are employed in order to segment the object. As next steps color model matching and SIFT feature matching are used to recognize the object. The algorithm also performs geometric pose estimation as well as final scene verification and refinement using global scene consistency checks.

A different approach was discussed by Weijer and Khan [9] where the authors compare various bag-of-words based recognition algorithms. The algorithm represents an image as a set of local regions where each of them is represented as a visual vocabulary. Different objects correspond to different histograms (called bags-of-words) over the created vocabulary. An extracted bag-of-words histogram can be compared to all the histograms stored in the memory and thus, an object can be labelled as one of the previously seen objects.

Although the above mentioned methods are established as the state-of-the-art object recognition systems, we still cannot say that the problem of object recognition has been solved. We claim that the reason for that is already included in the problem formulation. Given static images there are cases where it is impossible to recognize the object simply because of lack of distinctive features. One way to overcome this problem has been presented in the area of object segmentation where robot-objects interaction was introduced.

Segmentation of rigid objects from a video stream of objects being moved by a robot has been first addressed by Fitzpatrick et. al. [3] and Kenney et. al. [5]. Katz and Brock [4] address the problem of segmenting the articulated objects. Another technique was presented by [2], where the authors propose an approach to interactive segmentation that requires initial labeling using a 3D segmentation through fixation which results in a rough initial segmentation. The robot interacts with the scene to disambiguate the hypotheses.

All these approaches do not take into consideration action choices, action is assumed to be known and performed either by a human or a robot. In our approach we tackle interactive object recognition problem but we look for the action that will minimize the entropy over objects distribution. Therefore we want to be optimal in the number of actions that will lead us to successful object recognition.

There are two especially ambiguous recognition cases to consider in evaluating this algorithm.

3 Example cases

There are two especially ambiguous recognition cases to consider in evaluating this algorithm.

3.1 Book Ambiguity

Suppose there are two similar books with different covers. If you are observing a one of the books upside-down, we do not know which book it is. However we know that the most discriminative features between these books will be exposed if we flip the book over, and thus that is the optimal action.

3.2 Ball Ambiguity

Suppose there are two very similar textureless balls, except one ball has a mark on it. The pose of the unmarked ball is always ambigious. The pose of the marked ball is only unambigious if we observe the mark. If observing a ball with no mark on it, the optimal action would be to rotate the ball through all poses until finding a mark or deducing the absense of the mark. This case requires a history of actions and observed features in order to deduce recognition of the markless ball.

4 Recording Data

We will record data for N objects in I poses measuring M features of J different types. In addition to some standard household items, we will record data for ambiguous object such as the balls and books from the ambigous cases discusses earlier.

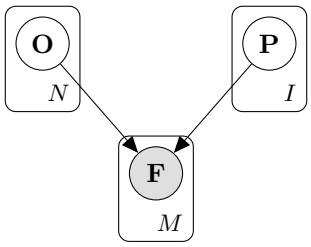
This algorithm should work for any features provided a matching error function for the feature which will be computed. Some example features to extract:

- SURF
- Color spectrum
- 3D point cloud

Since we do not have access to a robot arm, I propose $M = 8$ poses, right-side up and up-side down, with 4 orientations 90° apart for each, $\mathbf{P} = \{p_{+0}, p_{+90}, p_{+180}, p_{+270}, p_{-0}, p_{-90}, p_{-180}, p_{-270}\}$. This leaves $C = I-1 = 7$ pairwise *relative* actions between poses, $\mathbf{A} = \{a_{+90}, a_{+180}, a_{+270}, a_{-0}, a_{-90}, a_{-180}, a_{-270}\}$.

5 Model

Before considering actions, our model can be represented as a Bayesian Network (BN) where features **F** are dependent on objects **O** and poses **P**.



The joint distribution:

$$\mathrm{p}(\mathbf{O}, \mathbf{P}, \mathbf{F}) = \mathrm{p}(\mathbf{O}) \cdot \mathrm{p}(\mathbf{P}) \cdot \mathrm{p}(\mathbf{F}|\mathbf{O}, \mathbf{P})$$

The posterior distribution:

$$\mathrm{p}(o, p|\mathbf{F}) = \frac{\mathrm{p}(o, p) \cdot \mathrm{p}(\mathbf{F}|o, p)}{\mathrm{p}(\mathbf{F})}$$

posterior: $\mathrm{p}(o, p|\mathbf{F})$

prior: $\mathrm{p}(o, p) = \frac{1}{N \cdot I} = \frac{1}{K}$

likelihood: $\mathrm{p}(\mathbf{F}|o, p)$

evidence: $\mathrm{p}(\mathbf{F})$

Each feature type has an associated error function. In the case of SURF features, each feature has an associated descriptor which can be compared against other descriptors. In the case of a color histogram, every color histogram recorded during training gets a node on the graph, and an error function could be something like the mean-square error. A 3D point could feature could be treated the same.

6 Approach 1

6.1 Training

First we need to construct the graph. Every object, pose, and feature in the training data gets a node on the graph.

Lets use a toy example to illustrate. We have two similar books in two poses. We observe 4 SURF features and 4 color histograms. When we generate our graph, we will have 8 feature nodes, 2 object nodes, and 2 pose nodes.

Each feature is a specific parameterization/descriptor and each feature type has a matching function that returns the error between two features.

Lets suppose that for the training data the observed features for each object-pose are given by

$$\begin{array}{l} o_1, p_1\text{: } f_1^1,\ f_5^2 \\ o_1, p_2\text{: } f_2^1,\ f_6^2 \\ o_2, p_1\text{: } f_3^1,\ f_7^2 \\ o_2, p_2\text{: } f_4^1,\ f_8^2 \end{array}$$

The superscript represents the fact that there are two types of features: SURF and color histogram. (Only one SURF feature was found per image)

Then, iterating through each training sample we compute the feature errors. Here is an example for the ambiguous book case where p_1 is the front of the book and p_2 is the back of the book.

	f_1^1	f_2^1	f_3^1	f_4^1	f_5^2	f_6^2	f_7^2	f_8^2
o_1, p_1	0	10	9	10	0	100	40	100
o_1, p_2	10	0	10	1	100	0	100	2
o_2, p_1	8	10	0	10	50	100	0	100
o_2, p_2	10	1	10	0	100	1	100	0

What is represented in this table is the minimum matching error between the feature in the column, and the features observed for the training sample in the row.

Note that the errors returned by different feature types can have vastly different scales.

The back of both books are ambiguous but the front of the books are not. Notice that f_2 and f_6 were observed for o_1, p_2 and thus the error is zero, but due to the ambiguity of the back of these books, f_4 and f_8 also have reasonably small errors.

Lets consider taking a new observation. We observe a set of features for an unknown object-pose $o_?, p_?$ and compute the minimum matching errors between observated features and all of the features in the model, $\mathbf{F}$.

	f_1^1	f_2^1	f_3^1	f_4^1	f_5^2	f_6^2	f_7^2	f_8^2
$o_?, p_?$	9	2	11	1	102	3	100	4

Now we need to determine which feature we are observing. First, we need to know the distribution for $\mathrm{p}(f|o,p)$. To do this, we need to gather a many training samples for each object pose, compute the errors, and learn a distribution.

Now, to learn $\mathrm{p}(f_1^1|o_1,p_1)$, we just need to learn a distribution over the training sample errors, [1,2,1,3]. This could be a Gaussian distribution, or more rigorously a Half-normal distribution.

6.2 Inference

After an observation, we first want to compute the posterior for all object-poses.

	f_1^1	f_2^1	f_3^1	f_4^1	f_5^2	f_6^2	f_7^2	f_8^2
o_1, p_1	1	10	9	9	2	102	42	98
o_1, p_1	2	8	11	11	2	101	40	101
o_1, p_1	1	9	10	11	2	99	41	101
o_1, p_1	3	11	10	10	1	100	40	100

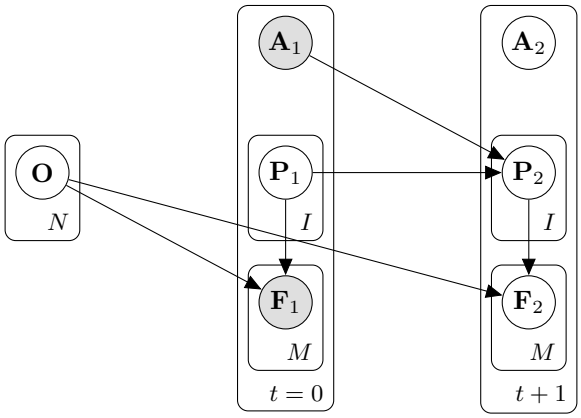
$$\mathrm{p}(o, p|\mathbf{F}) = \frac{\mathrm{p}(o, p) \cdot \mathrm{p}(\mathbf{F}|o, p)}{\mathrm{p}(\mathbf{F})}$$

We can do this by computing

$$\mathrm{p}(\mathbf{F}|o, p) = \prod_f \mathrm{p}(f|o, p)$$

$$\begin{aligned}
\mathrm{p}(\mathbf{F}) &= \sum_{n,i} \mathrm{p}(o_n, p_i, \mathbf{F}) \\
&= \sum_{n,i} \mathrm{p}(\mathbf{F}|o_n, p_i) \cdot \mathrm{p}(o_n, p_i) \\
&= \frac{1}{K} \sum_{n,i} \mathrm{p}(\mathbf{F}|o_n, p_i)
\end{aligned}$$

After the observing of an object for the first time, we want to determine an optimal action which leads to a new pose and a new set of observed features. Thus, our model is actually a dynamic BN.



Lets consider the case of two ambigious books. Suppose there are 3 poses: front, back, and side. And suppose the backs and sides of both books are ambigious.

We observe the back of one of these books. We are 50% sure its book 1 and 50% sure its book 2. What is the best action to take? If we flip the book on the side, we are confident what features

we will see because these features are also ambigious. However, we are uncertain what features we will see if we flip the book over because we do not know if we will see the book 1 cover features or the book 2 cover features.

So the big question is how can we determine this optimal action?

One way of going about this is to first consider, what are the features I might expect to see given an action and my previous observation.

If the action is to flip the book over, we expect to see the cover features for book 1 with 50% and book 2 with 50%. If the action is to turn the book to its ambigious side, we would expect to see the side feature with 100% probability. And thus, the measure that distinguishes the better action is the maximum entropy.

$$a^* = \underset{a}{\operatorname{argmax}} \, \mathrm{H}[\mathbf{F}_2 | \mathbf{F}_1, a]$$

By conditioning over the first observation and the action, we are marginalizing over all objects and poses incorporating all the information of the model.

Another approach to finding the optimal action is a sampling-based approach. The previous approach is optimal but also a huge computation – we must maginalize over all objects and poses!

Suppose we sample a feature f_2 and assume that after some action we observe this feature with zero error. What is then the entropy across all objects? Whatever feature minimizes the entropy is a feature we want to observe so then we determine an action to get us to the appropriate pose to view that feature.

$$f_2^* = \underset{f_2}{\operatorname{argmin}} \, \mathrm{H}[\mathbf{O} | \mathbf{F}_1, f_2]$$
$$a^* = \underset{a}{\operatorname{argmax}} \, \mathrm{p}(f_2^* | \mathbf{F}_1, a)$$

One thing to consider with this approach is how to optimally sample features. For example, we should not sample a feature that is associated with an object that is not probable considering previous observations.

7 Approach 2

7.1 Training

In the first step we take pictures of object-pose pairs and save the features that we encountered for each of them. Thus, we prepare a database of features. In this example there are 4 object-poses pairs and at most 6 SURF features per each. The superscript represents a type of the feature (1-SURF descriptor, 2-color histogram, 3-point cloud). The subscript represent the feature number.

	f_1^1	f_2^1	f_3^1	f_4^1	f_5^1	f_6^1	f_7^2	f_8^3
o_1, p_1	$\begin{pmatrix}10 & 2 & \cdots & 5\end{pmatrix}$	$\begin{pmatrix}3 & \cdots & 15\end{pmatrix}$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\begin{pmatrix}200 & \cdots & 100\end{pmatrix}$	$\begin{pmatrix}100 & \cdots & 100\end{pmatrix}$
o_1, p_2	$\begin{pmatrix}1 & 2 & \cdots & 2\end{pmatrix}$	$\begin{pmatrix}13 & \cdots & 5\end{pmatrix}$	$\cdots$	$\cdots$	$\cdots$		$\begin{pmatrix}50 & \cdots & 200\end{pmatrix}$	$\begin{pmatrix}10 & \cdots & 200\end{pmatrix}$
o_2, p_1	$\begin{pmatrix}21 & 12 & \cdots & 3\end{pmatrix}$	$\cdots$	$\cdots$	$\cdots$	$\cdots$		$\begin{pmatrix}150 & \cdots & 70\end{pmatrix}$	$\begin{pmatrix}140 & \cdots & 20\end{pmatrix}$
o_2, p_2	$\begin{pmatrix}1 & 2 & \cdots & 5\end{pmatrix}$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\begin{pmatrix}50 & \cdots & 170\end{pmatrix}$	$\begin{pmatrix}40 & \cdots & 120\end{pmatrix}$

7.2 Likelihood Approximation

In this section we would like to show our method to approximate the likelihood of an observation given the object-pose pair by using a matching function.

During our experiment we observe a set of live image features $F_l = \{f_{l1}^1, f_{l2}^1, \cdots, f_{l5}^1, f_{l6}^2, f_{l7}^3\}$. We calculate errors between observed features and features stored in the database. For that we use norm of a difference of vectors

$$err^t(f_i^t, f_j^t) = |f_i^t - f_j^t|$$

, where the superscript t represents the type of feature.

Calculated errors between all the features in the live image and the features in the database are shown below. Horizontal axis represent live image features and vertical axis shows the database features.

	f^1_{l1}	f^1_{l2}	f^1_{l3}	f^1_{l4}	f^1_{l5}	f^2_{l6}	f^3_{l7}
f^1_{1,o_1,p_1}	0	10	9	10	0		
f^1_{2,o_1,p_1}	10	0	10	1	100		
f^1_{3,o_1,p_1}	8	10	0	10	50		
f^1_{4,o_1,p_1}	10	1	10	0	100		
f^2_{5,o_1,p_1}						1	100
f^3_{6,o_1,p_1}						1	100
f^1_{1,o_1,p_2}	0	10	9	10	0		
f^1_{2,o_1,p_2}	10	0	10	1	100		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
f^2_{5,o_2,p_2}						1	100
f^3_{6,o_2,p_2}						1	100

We normalize all the entries in the above table with respect to columns such that all the errors are numbers between 0 and 1. For each row in the table we are looking for the best match for a given feature f^t_{m,o_n,p_i} where t is the feature type. We claim that the probability of a given match is given by:

$$\mathrm{p}(match(f^t_{m,o_n,p_i}, f^t_{lj})) = \max \mathrm{p}(f^t_{lj}|f^t_{m,o_n,p_i}) = 1 - \min err^t(f^t_{m,o_n,p_i}, f^t_{lj})$$

If the best match results in a probability below a given threshold β^t we disregard this probability, so

if $\mathrm{p}(match(f^t_{m,o_n,p_i}, f^t_{lj})) < \beta^t$ then $\mathrm{p}(match(f^t_{m,o_n,p_i}, f^t_{lj})) = 0$

Now we need to obtain one number for each feature type that will tell us the probability of a good match of all the features of type t. For that we can take the average (weighted average maybe?) of the probabilities of a match for the given object-pose pair.

$$\mathrm{p}(match(f^t_{o_n,p_i}, f^t_l)) = \frac{\sum_m \mathrm{p}(match(f^t_{m,o_n,p_i}, f^t_{lj}))}{m}$$

Now we approximated the probability $\mathrm{p}(F^t_l = F^t|o_n, p_i)$ which gives us a probability for each feature type that the feature we are observing is the feature associated with the given object n and the pose i.

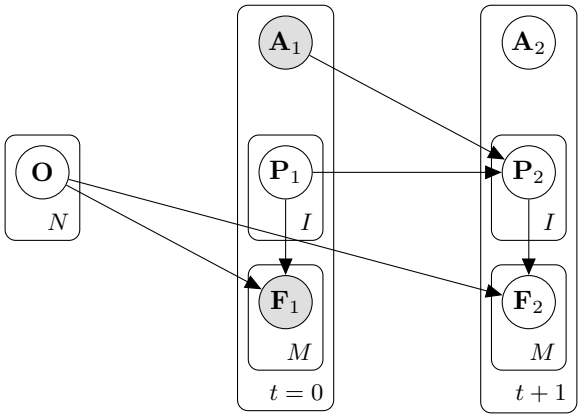
In order to obtain $\mathrm{p}(F|o_n, p_i)$ we need to take take the average (weighted average maybe?) of all feature types, thus:

$$\mathrm{p}(F_l = F|o_n, p_i) \approx \mathrm{p}(match(f_{o_n,p_i}, f_l)) = \frac{\sum_t \mathrm{p}(match(f^t_{o_n,p_i}, f^t_{lj}))}{t}$$

which is an approximation of our likelihood $\mathrm{p}(\mathbf{F}|o, p)$ which says the probability that the feature you are observing is the feature associated with the object o and the pose p

7.3 Inference

After the first observation, we need to determine the optimal action which leads to a new pose and a new set of observed features. Thus, our model is really a dynamic BN.



We want to obtain the optimal action that will minimize the entropy over the posterior distribution:

$$\underset{A_{t+1}}{\operatorname{argmin}} H(\mathrm{p}(\mathbf{O}, \mathbf{P_{t+1}} | \mathbf{F_{1:t}}, \mathbf{A_{t:t+1}}))$$

We can obtain this distribution by:

$$\mathrm{p}(\mathbf{O}, \mathbf{P_{t+1}} | \mathbf{F_{1:t}}, \mathbf{A_{t:t+1}}) = \int \mathrm{p}(\mathbf{O}, \mathbf{P_{t+1}}, \mathbf{F_{t+1}} | \mathbf{F_{1:t}}, \mathbf{A_{t:t+1}}) d\mathbf{F_{t+1}} =$$

$$\int \overbrace{\mathrm{p}(\mathbf{O}, \mathbf{P_{t+1}} | \mathbf{F_{1:t+1}}, \mathbf{A_{t:t+1}})}^{\text{our posterior}} \cdot \mathrm{p}(\mathbf{F_{t+1}} | \mathbf{F_{1:t}}, \mathbf{A_{1:t+1}}) d\mathbf{F_{t+1}}$$

where

$$\mathrm{p}(\mathbf{F_{t+1}} | \mathbf{F_{1:t}}, \mathbf{A_{1:t+1}}) =$$

$$\int \int \mathrm{p}(\mathbf{F_{t+1}}, \mathbf{O}, \mathbf{P_{t+1}} | \mathbf{F_{1:t}}, \mathbf{A_{t:t+1}}) d\mathbf{O} d\mathbf{P_{t+1}}$$

$$\int \int \overbrace{\mathrm{p}(\mathbf{F_{t+1}} | \mathbf{O}, \mathbf{P_{t+1}})}^{\text{observation model}} \cdot \overbrace{\mathrm{p}(\mathbf{O}, \mathbf{P_{t+1}} | \mathbf{F_{1:t}}, \mathbf{A_{1:t+1}})}^{\text{action model}} d\mathbf{O} d\mathbf{P_{t+1}}$$

In our first implementation we assume the action model to be fully deterministic (delta function), so:

$$\mathrm{p}(P_{t+1} | A_{t+1}, P_t) = \delta(p_{t+1})$$

References

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.

[2] Niklas Bergström, Carl Henrik Ek, Mårten Björkman, and Danica Kragic. Scene understanding through interactive perception. In *In 8th International Conference on Computer Vision Systems (ICVS)*, Sophia Antipolis, September 2011.

[3] P. Fitzpatrick. First contact: an active vision approach to segmentation. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2003.

[4] Dov Katz and Oliver Brock. Interactive segmentation of articulated objects in 3d. In *Workshop on Mobile Manipulation at ICRA*, 2011.

[5] Jacqueline Kenney, Thomas Buckley, and Oliver Brock. Interactive segmentation for manipulation in unstructured environments. In *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, ICRA'09, 2009.

[6] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[7] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[8] Jie Tang, Stephen Miller, Arjun Singh, and Pieter Abbeel. A textured object recognition pipeline for color and depth image data. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3467–3474. IEEE, 2012.

[9] Joost van de Weijer and Fahad Shahbaz Khan. Fusing color and shape for bag-of-words based object recognition. In *Computational Color Imaging*, pages 25–34. Springer, 2013.