Implementation of Contracting Curve Density Algorithm for Applications in Personal Robotics

April 27, 2011

Shulei Zhu
Institut für Informatik
Technische Universität München

Outline

Motivation

Some challenging tasks in personal robotics

- Image segmentation
- Pose estimation
- Object recognition and tracking

Model-base method

- require much information external to the image
- Curve-fitting: a crucial part of these problems

Requirements

- Robustness: stable even in the presence of heavy texture, clutter, poor contrast, partial occlusion
- Accuracy: high sub-pixel accuracy
- Efficiency: time-constrained, limited computer hardware resources in personal robotics

How the original CCD algorithm works?

Flowchart of the CCD algorithm

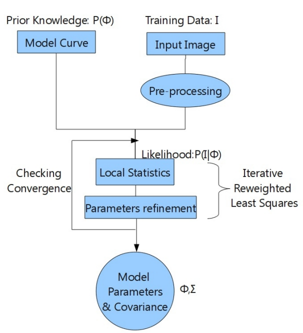


Figure: the CCD algorithm

Basic steps of the CCD algorithm

1. Contour initialization : initialize the model parameter vector Φ (6-DOF or 8-DOF) and covariance matrix Σ_Φ

How the original CCD algorithm works?

Flowchart of the CCD algorithm

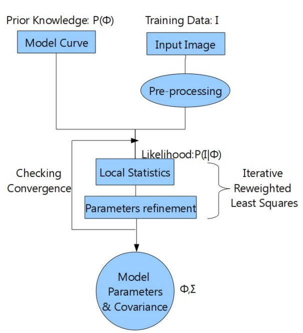


Figure: the CCD algorithm

Basic steps of the CCD algorithm

1. Contour initialization : initialize the model parameter vector Φ (6-DOF or 8-DOF) and covariance matrix Σ_Φ
2. Learning of local statistics : evaluate the likelihood; build the cost function

How the original CCD algorithm works?

Flowchart of the CCD algorithm

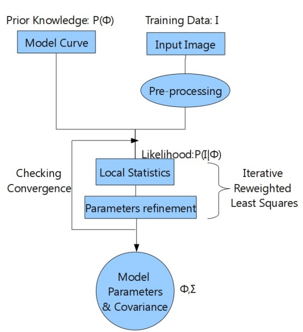


Figure: the CCD algorithm

Basic steps of the CCD algorithm

1. Contour initialization : initialize the model parameter vector Φ (6-DOF or 8-DOF) and covariance matrix Σ_Φ
2. Learning of local statistics : evaluate the likelihood; build the cost function
3. Refinement of model parameters : Maximize the cost function using optimization algorithms

How the original CCD algorithm works?

Flowchart of the CCD algorithm

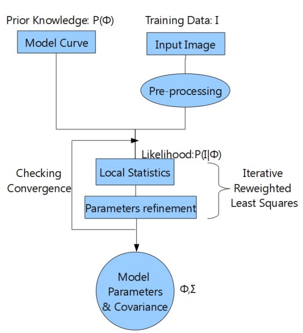


Figure: the CCD algorithm

Basic steps of the CCD algorithm

1. Contour initialization : initialize the model parameter vector Φ (6-DOF or 8-DOF) and covariance matrix Σ_Φ
2. Learning of local statistics : evaluate the likelihood; build the cost function
3. Refinement of model parameters : Maximize the cost function using optimization algorithms
4. Check for convergence, if not, go to Step 2

Sketch of the CCD algorithm

Figure: The contour of a pan

An alternative view of the CCD algorithm

A classification problem

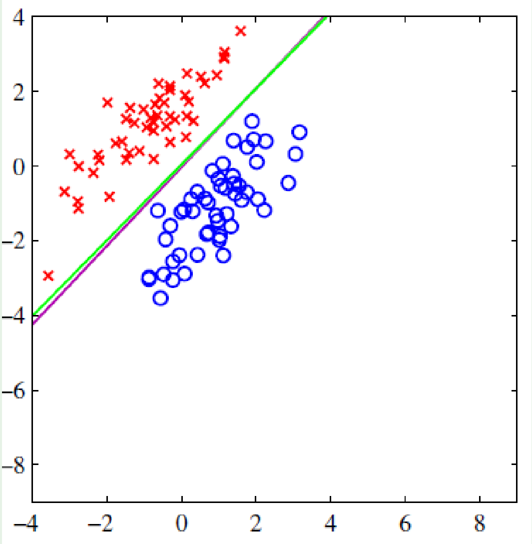


Figure: A classification problem [1]

An alternative view of the CCD algorithm

Probit regression

- Evaluation of conditional distribution $p(\Phi|\mathbf{I})$

$$p(\Phi|\mathbf{I}) \propto \underbrace{p(\mathbf{I}|\mathbf{m}_\Phi, \Sigma_\Phi)}_{\text{local statistics}} \times \underbrace{p(\Phi)}_{\text{prior distribution}}$$

  Local statistics (likelihood): a probit function with respect to Φ

- Goal: MAP (maximum a posteriori probability) solution of cost function $\mathcal{Q}(\Phi)$

$$\mathcal{Q}(\Phi) = \arg\max_{\Phi} \ln(p(\Phi|\mathbf{I}))$$

  Approach: iterative reweighted least squares (IRLS) e.g. Gaussian Newton method, Gradient decent and SVM Least Squares.

Quadratic and Cubic B-spline curves

B-spline curves

$$\mathbf{C}(u) = \sum_{i=0}^{m-n-2} P_i B_{i,n}(u) \ , \ u \in [u_n, u_{m-n-1}]$$

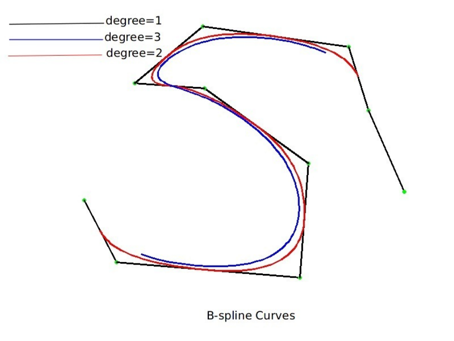


Figure: B-spline curves of degree = 1, 2, 3

Logistic and Probit function

Logistic function

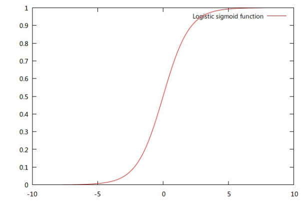


Figure: Logistic function

$$f(\cdot) = \frac{1}{1 + e^{-x}}$$

Logistic and Probit function

Logistic function

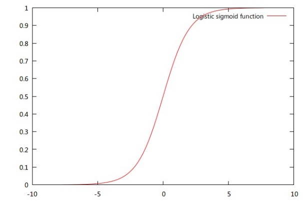


Figure: Logistic function

$$f(\cdot) = \frac{1}{1 + \mathrm{e}^{-x}}$$

Probit function

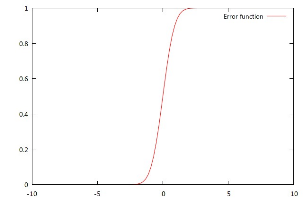


Figure: Probit function

$$f(\cdot) = \frac{1}{2}(\frac{1}{\sqrt{2}} erf(x) + 1)$$

Logistic and Probit function

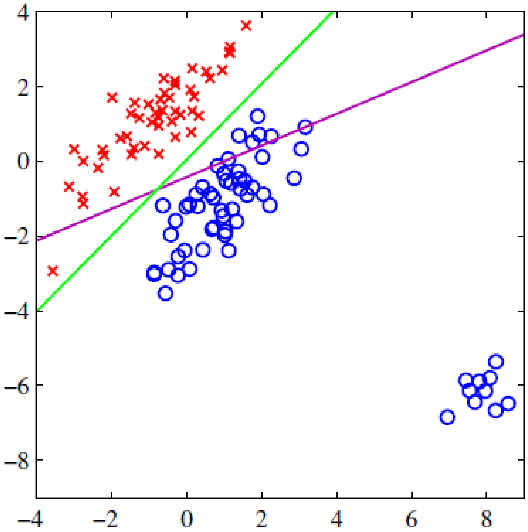


Figure: Probit function is highly sensitive for outliers [1]

Three-dimensional Affine Shape-space

Parallax effect in two-dimensional affine shape-space

Figure: Parallax effect

Three-dimensional affine shape-space

Figure: Three-dimensional affine shape-space

Automated initialization methods (I)

Initialization from SIFT Features

Figure: Initialization from SIFT Features

Automated initialization methods (II)

Initialization from projection of point clouds onto the image

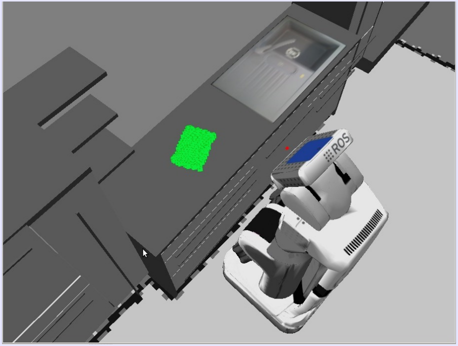


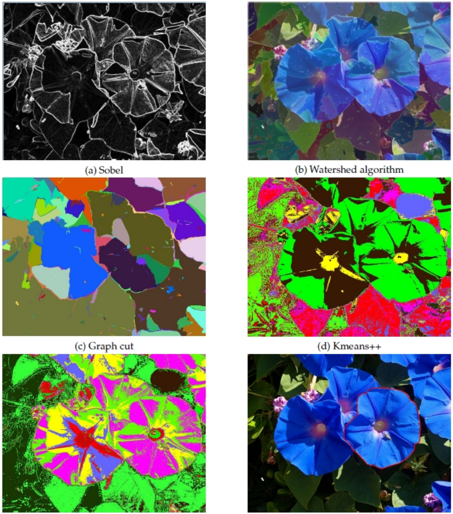
Figure: Initialization from projection of point clouds onto the image

Contracting Curve Density (CCD) Tracker

Algorithm 1 Contracting Curve Density (CCD) tracker

$\Phi \leftarrow 0$
$\mathbf{C} \leftarrow \mathit{contour\_initialization}()$
while *NewFrame* **do**
    $\mathbf{I} \leftarrow \mathit{pre\_processing}()$
    $\mathbf{C} \leftarrow \mathit{contour\_distortion}(\Phi)$
    $\Sigma \leftarrow \mathit{covariance\_initialization}()$
    $\Phi \leftarrow \Phi^{\mathrm{old}}$
    while $\mathit{convergence} = \mathit{FALSE}$ **do**
        $\mathit{local\_statistics\_learning}()$
        $\mathit{cost\_function\_MAP}()$
    end while
    $\Phi \leftarrow \Phi_{MAP}$
    $\Sigma \leftarrow \Sigma_{MAP}$
end while

Segmentation



(a) Sobel

(b) Watershed algorithm

(c) Graph cut

(d) Kmeans++

(e) Expectation and Maximization

(f) CCD

Figure: A Comparison of Image Segmentation Algorithms

Manual initialization

Shadow effects

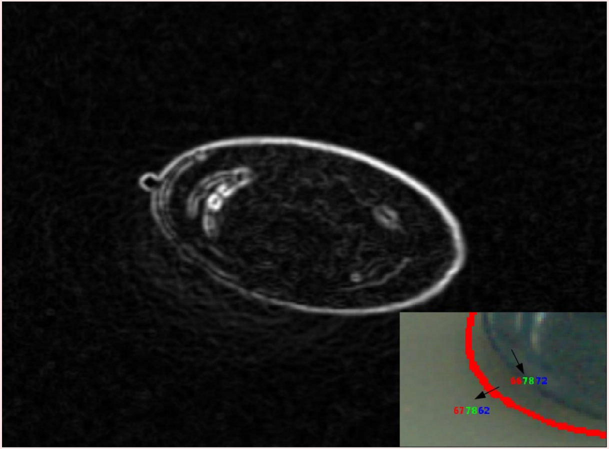


Figure: Shadow effects

Initialization from SIFT Features

Tracking initialized from SIFT features

- Match SIFT keypoints between the template image and the test image
- Discard the false matching points using the RANSAC algorithm
- Compute the homography
- Transform the contour of the template image onto the test image
- Apply the CCD tracker to the video

Summary

Investigate and implement the CCD approach

- Based on the OpenCV
- A ROS package: provide a ROS node interface to the ccd class
  `http://www.ros.org/wiki/contracting-curve-density`
- Released under open source BSD license

Improvements

- B-spline curve and three-dimensional affine shape-space
- Logistic regression
- Automated contour initialization methods: SIFT features and point clouds

Future work

Future work

- Use statistics based on other image features instead of the RGB statistics
- Integration of the CCD algorithm into a more complex tracking framework (e.g. the Lucas-Kanade method (LKM), the extended Kalman filter (EKF))
- B-spline can not precisely represent many useful simple curves such as circles and ellipses, thus, Non-uniform rational B-spline (NURBS) is required for the CCD algorithm.
- Port to Android system to support mobile applications
- ....

Thank you

Thank you for your attention!

References

[Bishop, 2006] Christopher M. Bishop
Pattern recognition and machine learning