Technion

Israel Institute of Technology
Department of Eelctrical Engineering

Signal and Image Processing lab

Robust Fitting of Implicit Polynomials with Applications to Contour Coding

Amir Helzer

Supervisors: Dr. Meir Barzohar* and Prof. David Malah

* Meir Barzohar is also with the computer vision group in Rafael

Outline

- Introduction to implicit polynomials (IPs)
- Applications of IPs
- Fitting IPs to object boundaries
- Fitting simulation results
- Boundary reconstruction
- Reconstruction simulation results
- Curve segmentation
- Contour coding simulation results

Introduction to Implicit Polynomials

- d^{th} order 2D Implicit polynomials

$$P_{\bar{a}}(x,y) = a_1 x^d + a_2 x^{d-1} y + \ldots + a_r = 0$$

- 4^{th} order IP

$$a_1x^4 + a_2x^3y + a_3x^2y^2 + a_4xy^3 + a_5y^4 + a_6x^3 + a_7x^2y +$$
$$a_8xy^2 + a_9y^3 + a_{10}xy + a_{11}x^2 + a_{12}y^2 + a_{13}x + a_{14}y + a_{15} = 0$$

- The polynomial is determined by its coefficient vector

$$\bar{a} = [a_1, \ldots, a_r] \quad , \quad r = \frac{(d+1)(d+2)}{2}$$

Introduction to Implicit Polynomials (cont'd)

- An IP can be written as the product of the coefficient vector and a monomial vector

$$\overline{p}(x,y) = \left[x^d y^0, x^{d-1} y^1, x^{d-2} y^2, \ldots, x^1 y^0, x^0 y^1, x^0 y^0\right]$$

$$P_{\overline{a}}(x,y) = \overline{a}\;\overline{p}^T(x,y)$$

- The set of points that solve the IP are referred to as its "Zero-Set"

$$Z_{\overline{a}} = \{(x,y) : P_{\overline{a}}(x,y) = 0\}$$

- The Zero-Set of an IP can describe an object boundary in an image

Applications of Implicit Polynomials

- Object recognition
  - Algebric invariants to affine transformations exist
    - D. Keren - 1994: Using Symbolic computation to Find Algebraic Invariants.
    - J. Subrahmonia, D. Cooper, and D. Keren - 1996: Bayesian Recognition of 2D and 3D Objects.
    - M. Barzohar, D. Keren, and D. Cooper - 1994: Recognizing Intersecting Roads in Aerial Images.
- Contour coding
  - The description power of IPs can be used for contour coding, in region coding applications.
  - Complex shapes may be described by the zero-set of a polynomial, defined only by its coefficients.

Requirements for Fitting IPs to Object Boundaries

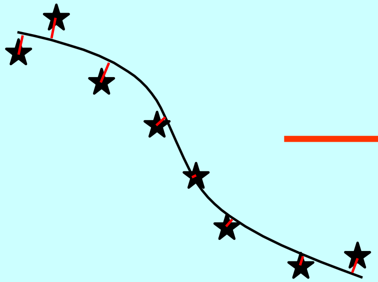
- Minimizing the distance between the zero-set and the data points $\Rightarrow$ tight fit
- Minimizing the sensitivity of the coefficients to noisy data $\Rightarrow$ robustness
- For contour coding: Minimizing the sensitivity to coefficient quantization $\Rightarrow$ Stability

Previous Work on Polynomial Fitting

- Taubin (1991) - Fitting IPs using non-linear iterative optimization for distance minimization.
- Z. Lei, M.M Blane and D.B. Cooper (1997) - Linear IP fitting algorithm with improved performance and stability (3L algorithm).
- D. Keren and C. Gotsman (1999) - IP fitting using special groups of star shaped polynomials.

Previous Work - Taubin (1991)

- The polynomial should be zero at the data.
- For small errors:



$$\sum_{n=1}^{N} dist^2(x_n, y_n) \approx \sum_{n=1}^{N} \left(\frac{P_{\bar{a}}(x_n, y_n)}{\left\| \nabla P_{\bar{a}}(x_n, y_n) \right\|} \right)^2$$

Gradient

- Taubin suggested an iterative optimization:

1. Initialize $\quad w_n = 1 \quad \forall n = 1, \ldots, N$
2. Minimize $\quad \sum_{n=1}^{N} \left(\frac{P_{\bar{a}}(x_n, y_n)}{w_n} \right)^2$
3. Calculate $\quad w_n = \left\| \nabla P_{\bar{a}}(x_n, y_n) \right\|$

Previous Work - Taubin (cont'd)

- The minimization step is performed by:

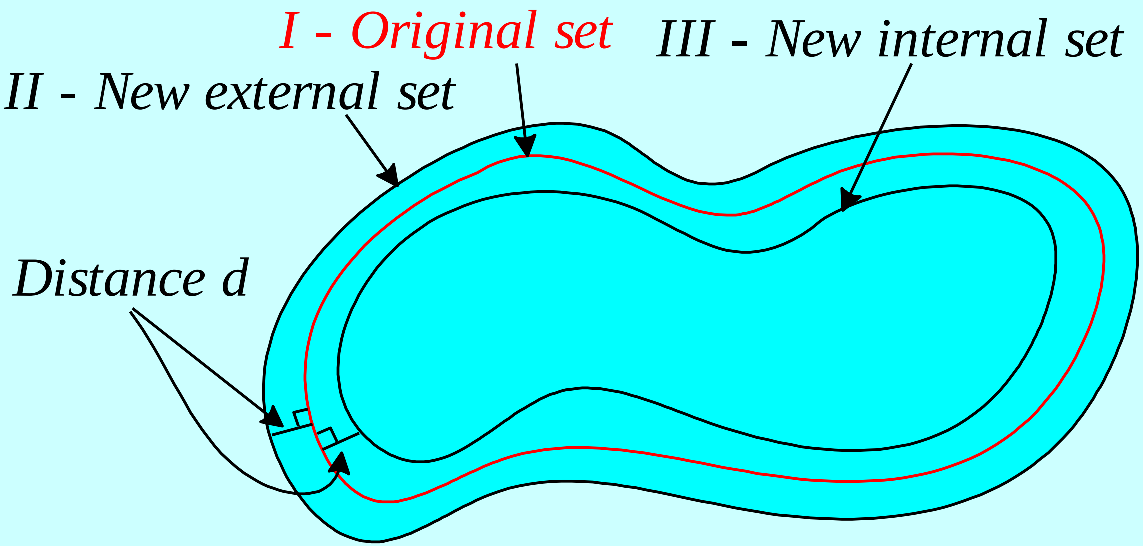
$$\sum_{n=1}^{N}\left(\frac{P_{\bar{a}}(x_n, y_n)}{w_n}\right)^2 = \sum_{n=1}^{N}\left(\frac{\bar{a}\,\bar{p}^T(x_n, y_n)}{w_n}\right)^2 =$$

$$\underbrace{\bar{a}\left(\sum_{n=1}^{N}\frac{\bar{p}^T(x_n, y_n)\bar{p}(x_n, y_n)}{w_n^2}\right)}\bar{a}^T = \bar{a}\, SM\, \bar{a}^T$$

- Minimized when $\bar{a}$ is the eigenvector with the smallest eigenvalue of SM.

Previous Work - Z. Lei et al. (1997)

- Construction of two additional data sets, based on the original data set (at a distance 'd')



- The value of the polynomial is required to be zero on the original set, $-\varepsilon$ on the external set and $+\varepsilon$ on the internal set.

Previous Work - Z. Lei et al. (cont'd)

- Minimize: $$E = \boxed{\sum_{n=1}^{N} \left(\bar{a}\, \bar{p}^T (x_n, y_n) \right)^2} +$$ *Error W.R.T original set*

$$\boxed{\sum_{n=N+1}^{2N} \left(\bar{a}\, \bar{p}^T (x_n, y_n) + \varepsilon \right)^2} +$$ *Error W.R.T external set*

$$\boxed{\sum_{n=2N+1}^{3N} \left(\bar{a}\, \bar{p}^T (x_n, y_n) - \varepsilon \right)^2}$$ *Error W.R.T internal set*

- Solved using *Least Squares*.

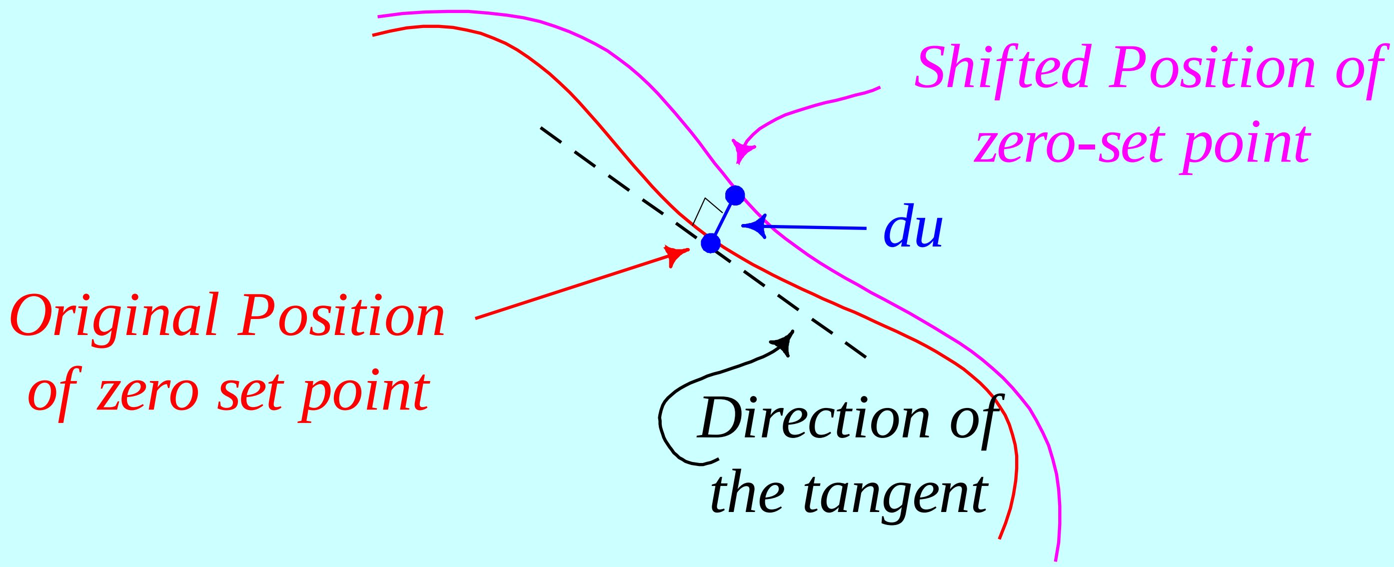
Fast non-iterative calculation !

Proposed Fitting Algorithm - *MinMax*

- Outline
  - Analyze the sensitivity of IPs to coefficient errors (quantization noise).
  - Derive a fitting algorithm that minimizes the sensitivity.
- Characteristics of resulting polynomials
  - Robustness to quantization
  - Improved fitting.
  - “3L” fitting can be viewed as a special case of this algorithm.

Zero-set sensitivity

- A Change in the coefficients causes the zero-set to shift.
- A change in the position of the zero-set is measured in perpendicular direction.



Sensitivity function

- The sensitivity function is defined by:

$$\bar{S}_{\bar{a}}^{u}(x,u) = \frac{du(x,y)}{d\bar{a}}$$

- The change in the position of the zero-set is:

$$\varepsilon_u(x,y) = \bar{S}_{\bar{a}}^{u}(x,y)\bar{\varepsilon}_a{}^T$$

Change in zero-set

Sensitivity function

Coefficients change

Sensitivity function (cont'd)

- The sensitivity function can broken into:

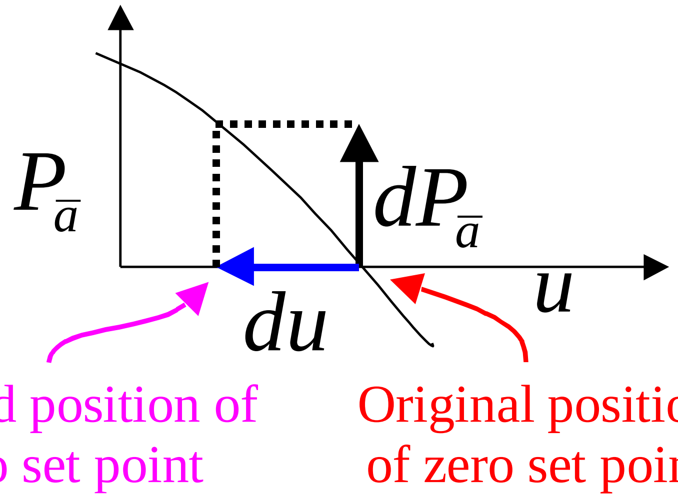
$$\bar{S}_{\bar{a}}^{u}(x,y) = \frac{du(x,y)}{dP_{\bar{a}}(x,y)} \frac{dP_{\bar{a}}(x,y)}{d\bar{a}}$$

Sensitivity of zero-set position to small changes in the value of the polynomial

Sensitivity of the value of the polynomial to small changes in the coefficient vector

Sensitivity function (cont'd)

- Sensitivity of the zero-set position to the value of the polynomial



$$\frac{du}{dP_{\bar{a}}}(x,y) = \frac{1}{\left\|\nabla P_{\bar{a}}(x,y)\right\|} = \frac{1}{\sqrt{\left(\frac{\partial P_{\bar{a}}(x,y)}{\partial x}\right)^2 + \left(\frac{\partial P_{\bar{a}}(x,y)}{\partial y}\right)^2}}$$

Scalar

Gradient

Sensitivity function (cont'd)

- Sensitivity of the value of the polynomial to the coefficients

$$\frac{dP_{\bar{a}}\left(x,y\right)}{d\bar{a}} = \frac{d\left(\bar{a}\;\bar{p}^{T}\left(x,y\right)\right)}{d\bar{a}} = \bar{p}(x,y)$$

Monomial vector

Sensitivity function (summary)

- The sensitivity function evaluates to:

Monomial vector

$$\bar{S}_{\bar{a}}^{u}(x,y) = \frac{du(x,y)}{dP_{\bar{a}}(x,y)} \frac{dP_{\bar{a}}(x,y)}{d\bar{a}} = \frac{\bar{p}(x,y)}{\left\|\nabla P_{\bar{a}}(x,y)\right\|}$$

Vector, with a component for each polynomial coefficient

Gradient

Zero-set error properties

- The fitting error, resulting from small coefficient errors is:

Error of coefficient ‘k’

$$\varepsilon_u(x,y) = \bar{S}_{\bar{a}}^{u}(x,y) \cdot \bar{\varepsilon}_a{}^{T} = \frac{\sum_{k=1}^{r} p_k(x,y)\varepsilon_{a_k}}{\left\|\nabla P_{\bar{a}}(x,y)\right\|}$$

Error, in the perpendicular direction, at the position of a zero set point

Sensitivity to changes in coefficient ‘k’

Zero-set error bounds

- When the maximal coefficient error is bounded by ε_{MAX}, the error is bounded by:

$$\left|\varepsilon_u(x,y)\right| \le \frac{\left|\sum_{k=1}^{r} p_k(x,y)\varepsilon_{a_k}\right|}{\left\|\nabla P_{\bar{a}}(x,y)\right\|} \le \varepsilon_{MAX} \frac{\sum_{k=1}^{r}\left|p_k(x,y)\right|}{\left\|\nabla P_{\bar{a}}(x,y)\right\|}$$

- If all the error components have the same variance - σ_ε^2 than:

$$\operatorname{var}\left(\varepsilon_u(x,y)\right) = \sigma_\varepsilon^2 \frac{\sum_{k=1}^{r} {p_k}^2(x,y)}{\left\|\nabla P_{\bar{a}}(x,y)\right\|^2}$$

A Robust fitting algorithm

- Uniform quantization causes bounded coefficient errors.
- When no boundary point has priority over another, error bounds for all boundary points should have the same values:

$$\frac{\sum_{k=1}^{r} \left| p_k\left(x_n, y_n\right)\right|}{\left\| \nabla P_{\bar{a}}\left(x_n, y_n\right)\right\|} = const \qquad \forall n = 1, \ldots, N$$

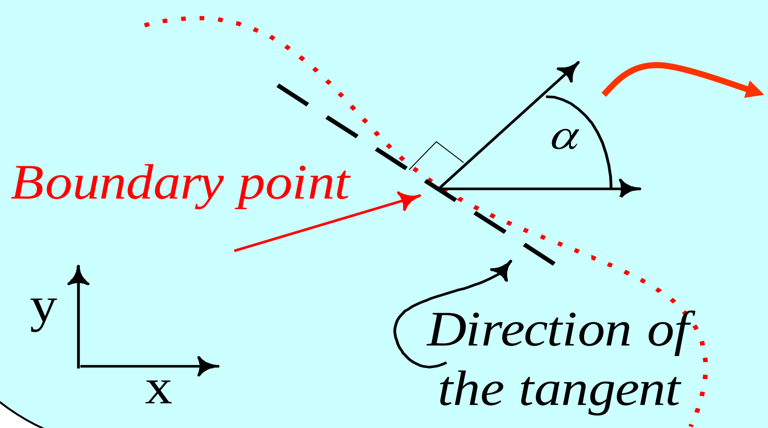
- The actual value of the error bound (*const*) is later normalized and therefore set to ‘1’.

A Robust fitting algorithm (cont'd)

- To fit to the data:
  - The value of the polynomial should be zero at the boundary points.

$$P_{\bar{a}}(x_n, y_n) = 0$$

  - The gradients of the polynomial should point in a direction locally perpendicular to the data.



$$\frac{dy_n}{dx_n} = tg(\alpha_n)$$

(dx_n, dy_n) - required value of gradient

$$\sqrt{\left(dx_n^{\;2} + dy_n^{\;2}\right)} = \sum_{k=1}^{r} \left| p_k(x_n, y_n) \right|$$

IP Fitting Implementation

- For each data point - 3 equations are generated:

$$P_{\bar{a}}(x_n, y_n) = 0 \qquad \frac{dP_{\bar{a}}(x_n, y_n)}{dx} = dx_n \qquad \frac{dP_{\bar{a}}(x_n, y_n)}{dy} = dy_n$$

$$\Downarrow \qquad\qquad \Downarrow \qquad\qquad \Downarrow$$

$$\bar{a}\,\bar{p}^T(x_n, y_n) = 0 \qquad \bar{a}\,\frac{d\bar{p}^T(x_n, y_n)}{dx} = dx_n \qquad \bar{a}\,\frac{d\bar{p}^T(x_n, y_n)}{dy} = dy_n$$

- ‘N’ data points generate 3N equations.
- These equations are put into matrix form, and a least squares solution is used.

IP Fitting Implementation (cont'd)

Minimize the MSE

$$E = \bar{e}\,\bar{e}^T$$

Where,

$$\bar{e} = (\bar{a}M - \bar{b})$$

and

$$\bar{b} = \begin{bmatrix} \bar{0} & \overline{dx} & \overline{dy} \end{bmatrix}$$

$$M = \begin{bmatrix} M_0 & M_X & M_Y \end{bmatrix}$$

Zero at data

X axis derivatives

Y axis derivatives

IP Fitting Implementation (cont'd)

Details:

$$M_0 = \left[\bar{p}^T(x_1, y_1) \quad \dots \quad \bar{p}^T(x_N, y_N)\right]$$
$$M_X = \left[\bar{p}_X{}^T(x_1, y_1) \quad \dots \quad \bar{p}_X{}^T(x_N, y_N)\right]$$
$$M_Y = \left[\bar{p}_Y{}^T(x_1, y_1) \quad \dots \quad \bar{p}_Y{}^T(x_N, y_N)\right]$$

$$\bar{p}_X(x_n, y_n) = \frac{d}{dx}\bar{p}(x_n, y_n)\ ;\ \bar{p}_Y(x_n, y_n) = \frac{d}{dy}\bar{p}(x_n, y_n)$$

Optimal coefficient vector

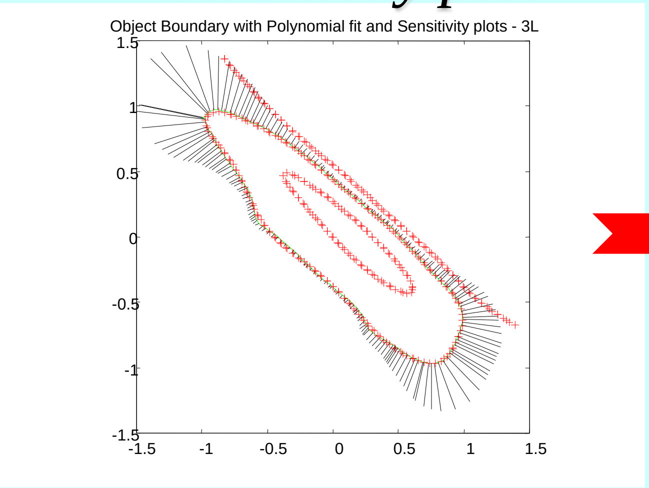
Least Squares solution

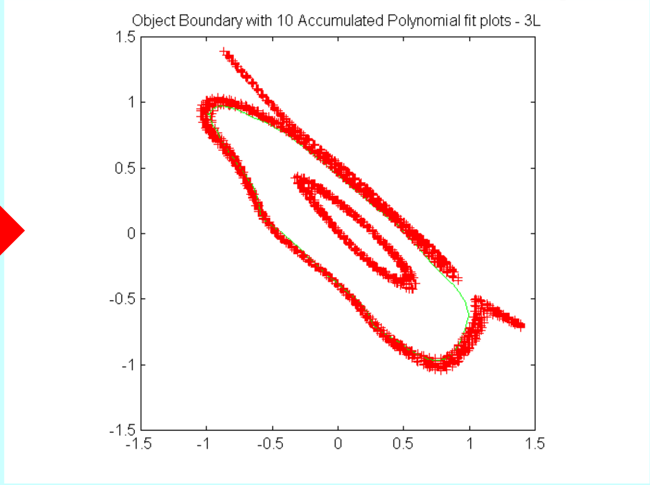
$$\bar{a}_{OPT} = \bar{b} M^T \left(M M^T\right)^{-1}$$

Simulation results
Sensitivity to quantization

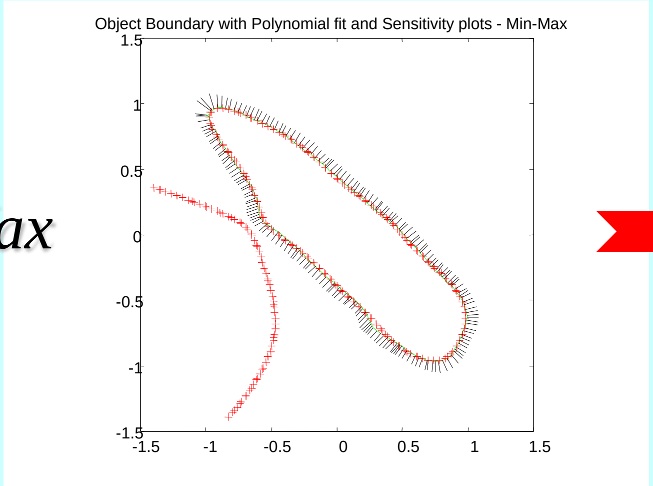
Sensitivity plot *Accumulated zero set plots*

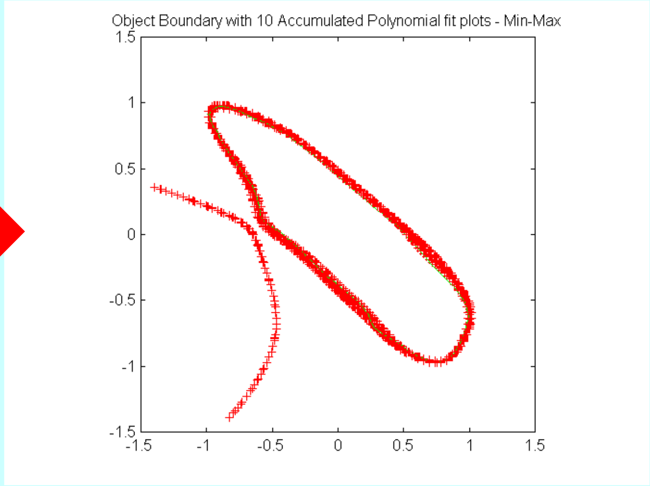
3L



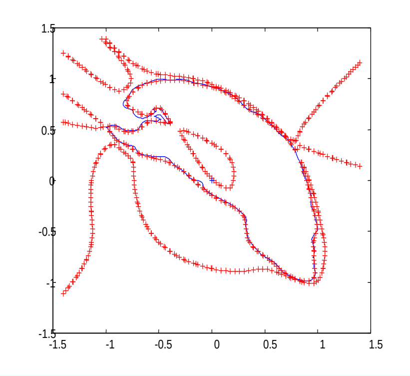


Min-Max

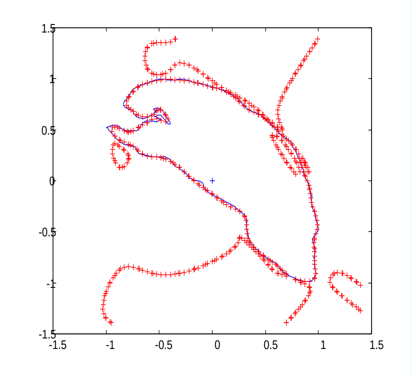




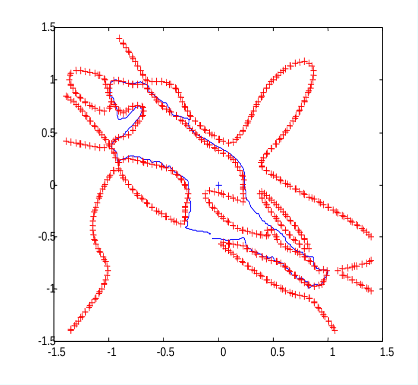
Simulation results
14^{th} order polynomial fitting
with 3L and Min-Max algorithms



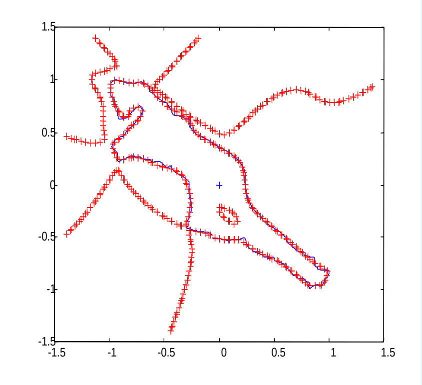
(a1)



(a2)



(b1)



(b2)

3L *Min-Max*

Data Reconstruction

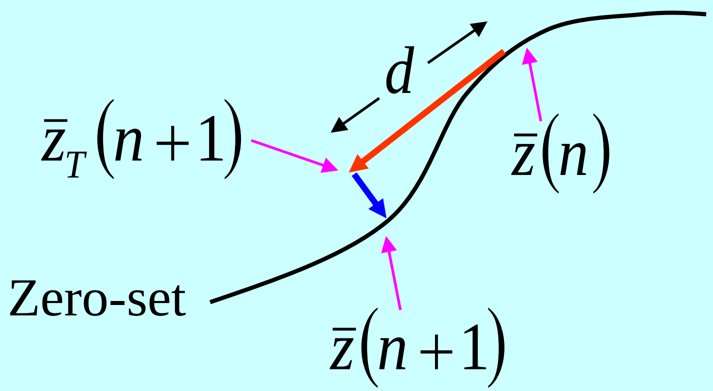
- To achieve a complete coding system, data needs to be reconstructed.
- Reconstruction is done by:
  - Scanning the polynomial zero-set, starting at a known point on it.
  - From each point, a numerical calculation of the next zero-set point is made.

Data Reconstruction (cont'd)

For each new point: $\bar{z}(n) = (x_n, y_n)$

1. Move in the tangent direction → $\bar{z}_T(n+1) = \bar{z}(n) + d \dfrac{\left(\dfrac{\partial P_{\bar{a}}(\bar{z}(n))}{\partial y}, -\dfrac{\partial P_{\bar{a}}(\bar{z}(n))}{\partial x} \right)}{\sqrt{\left(\dfrac{\partial P_{\bar{a}}(\bar{z}(n))}{\partial x} \right)^2 + \left(\dfrac{\partial P_{\bar{a}}(\bar{z}(n))}{\partial y} \right)^2}}$

2. Find the nearest point on the zero-set → *Steepest Descent*

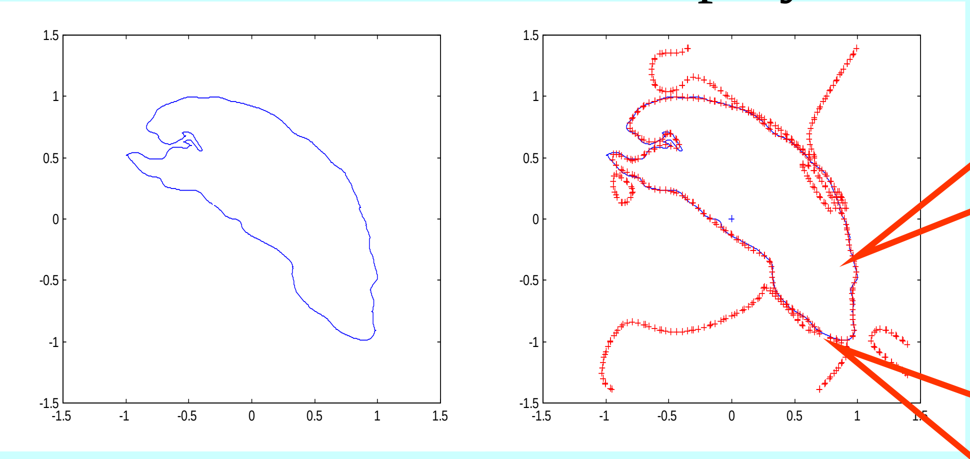


Boundary Reconstruction Problem

- Unconstrained fitting characteristics
  - Optimal coverage of given boundary points.
  - Spurious zero-set points may exist.

Object Boundary

Zero-set of
fitted polynomial



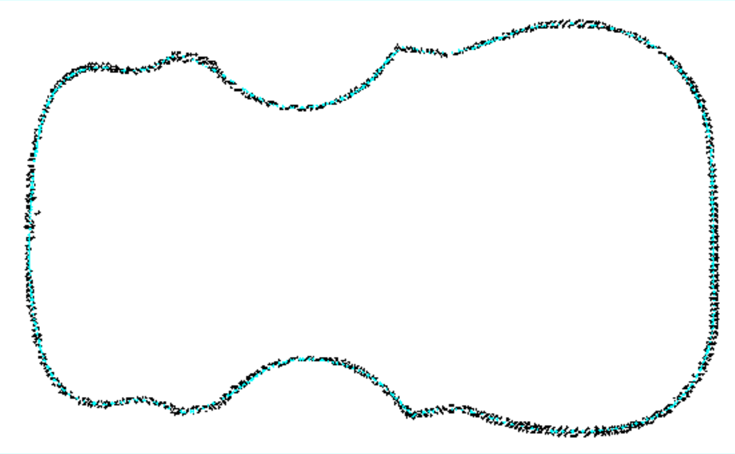
Zero-set
contains boundary

Spurious zero-set
points intersect
object boundary

Previous Work Addressing Spurious Zero-Set Points

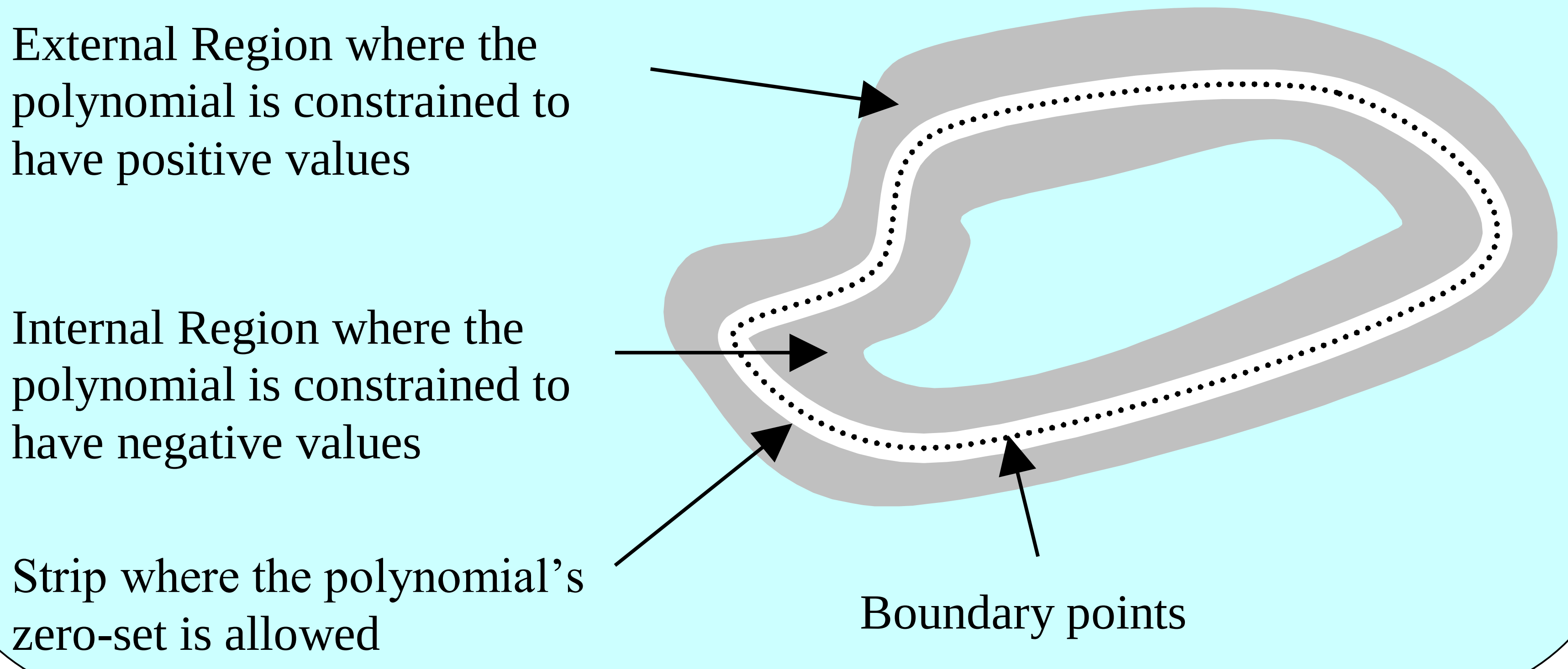
- D. Keren and C. Gotsman [1999]
  - Limits the possible space of polynomials to special groups of star shaped polynomials.
  - Useful for approximation of star shaped objects.

Star shaped polynomials cannot have spurious zeros



Proposed Solution to the Reconstruction Problem

- The zero-set of the polynomial is constrained to lie within a thin strip surrounding the boundary (set empirically).



Proposed Solution (cont'd)

- Constrained LS solution using *lagrange multipliers*:

Minimize $E = \bar{e}\,\bar{e}^T$ where $\bar{e} = (\bar{a}M - \bar{b})$

Subject to:

$$\bar{a}M_{EXT} > \bar{0}$$
$$\bar{a}M_{INT} < \bar{0}$$

All matrixes and vectors are as presented in the unconstrained solution

The constraint points are sampled in the <u>external</u> / <u>internal</u> regions

$$M_{EXT} = \left[\bar{p}^T\left(x_{EXT_1}, y_{EXT_1}\right) \quad \ldots \quad \bar{p}^T\left(x_{EXT_{N-EXT}}, y_{EXT_{N-EXT}}\right)\right]$$

$$M_{INT} = \left[\bar{p}^T\left(x_{INT_1}, y_{INT_1}\right) \quad \ldots \quad \bar{p}^T\left(x_{INT_{N-INT}}, y_{INT_{N-INT}}\right)\right]$$

Simulation results

14^{th} order polynomial fitting and reconstruction

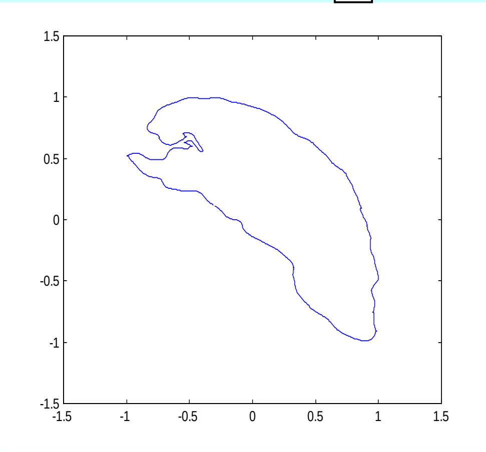
Polynomial Fitting

Polynomial Zero-Set

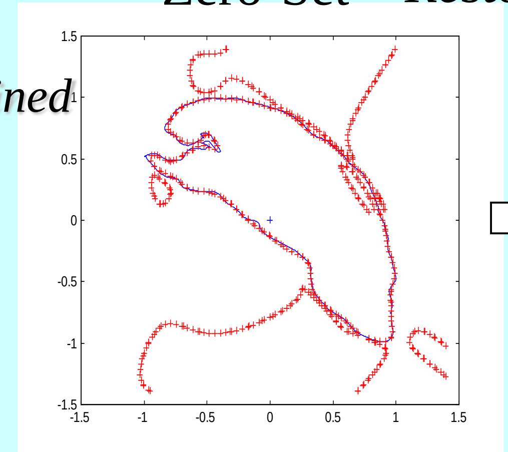
Data Restoration

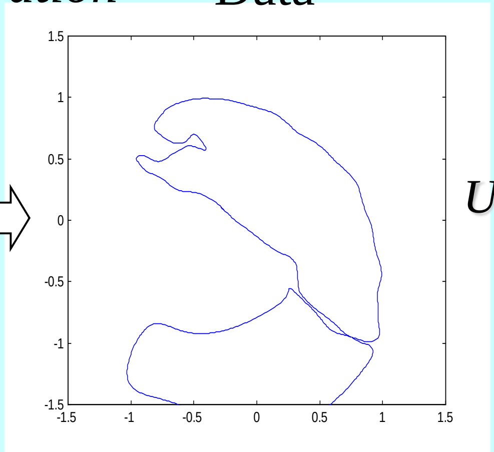
Restored Data

Object Boundary



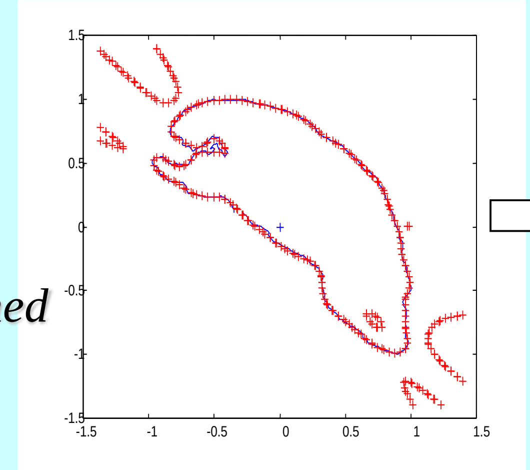
Unconstrained

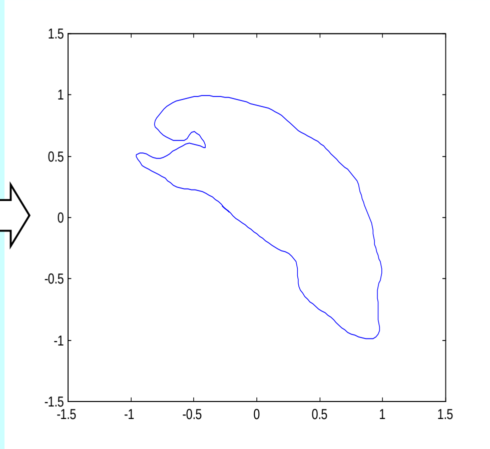




Unconstrained Solution

Constrained





Constrained Solution

Boundary Segmentation

Motivation

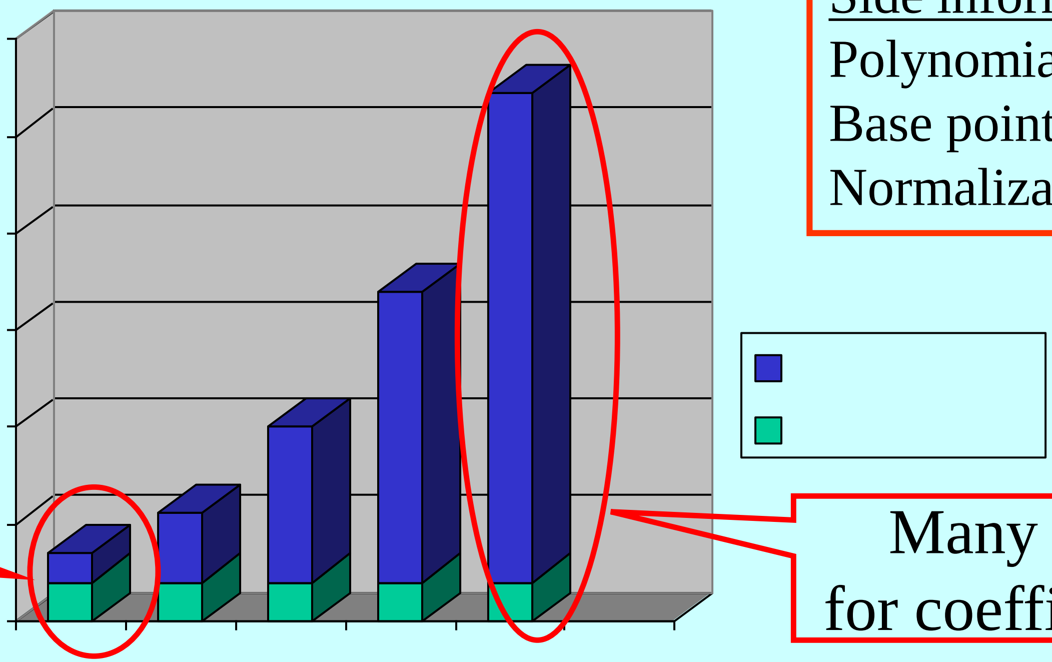
- Complex boundaries may require high order polynomials, requiring a large number of bits.

Side information

Polynomial order

Base point for scan

Normalization



Side Info.
is dominant

Many bits
for coefficients

Boundary Segmentation

Optimal Segmentation

- Each curve can be optimally segmented in terms of required number of bits.
- There exist 2^N possible segmentations.
- Exhaustive search is impractical.
- Segmentation can be performed on the basis of "special" points.
- We present a scheme for segmentation based on rate-distortion criteria.

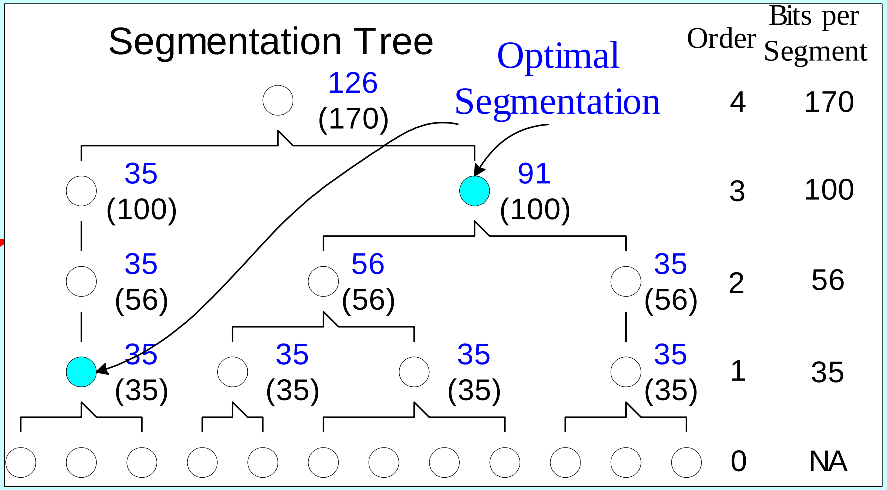
Bottom to Top Segmentation

- Segmentation begins with 1st order, minimal size segments (2 points).
- Merges are performed:
  - The segment pair with the least distortion is merged.
  - Merges are performed when the resulting distortion is below a limit.
- When no more merges are possible, the order of the polynomial is raised for all segments.
- When a maximal order is reached or only one segment remains, the algorithm terminates.

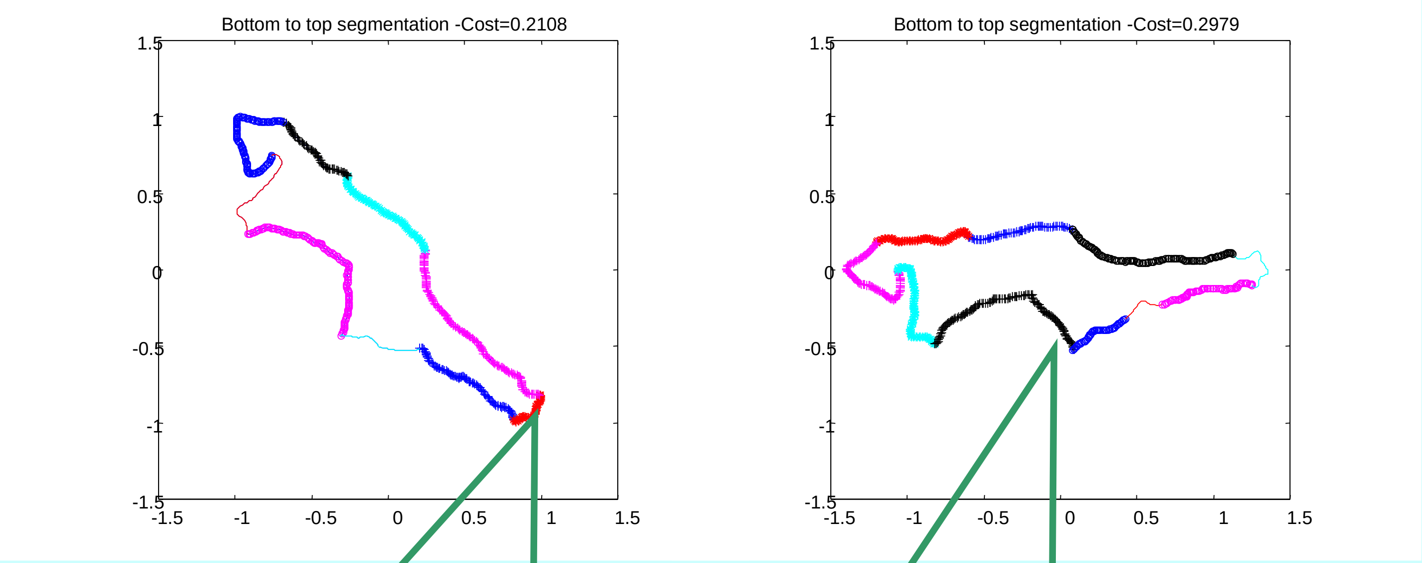
Bottom to Top Segmentation (cont'd)

- Rate is lowered during merges and raised during order increase.
- A scan is made for the best combination of segments, at the end of the merge process, for the optimal segmentation encountered.

Segmentation tree with optimal segmentation highlighted.
Total number of bits for representation - 126



Bottom to Top Segmentation - example



Segmentation captures most high curvature points

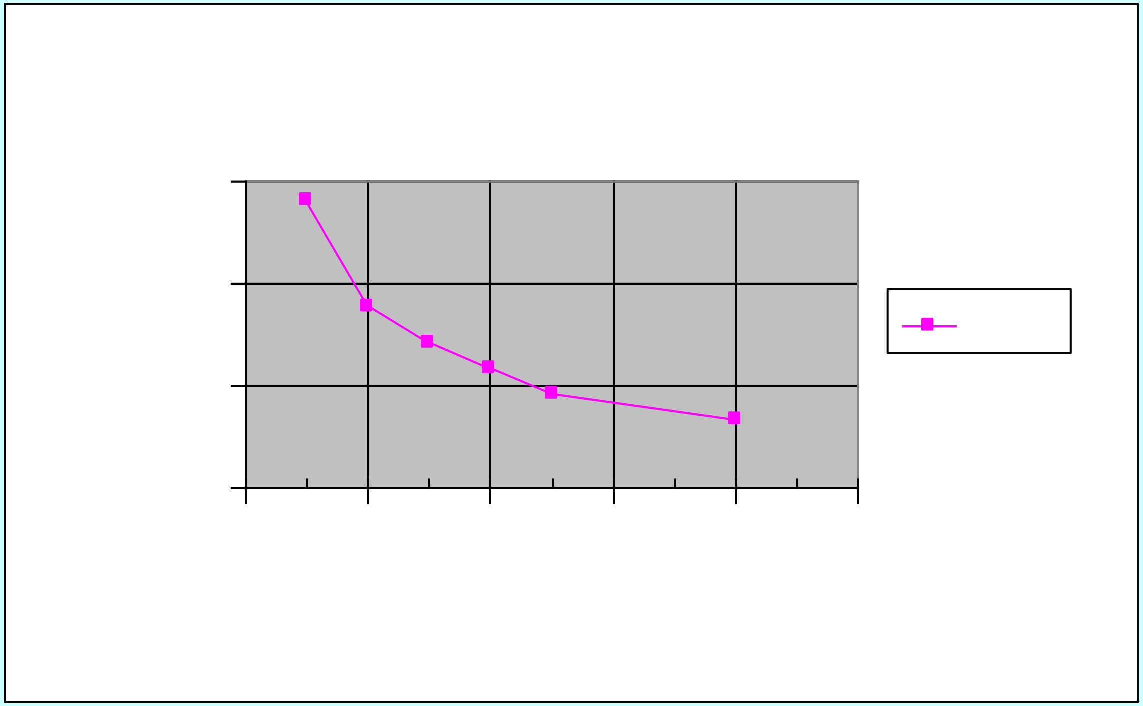
Same segmentation obtained after data rotation

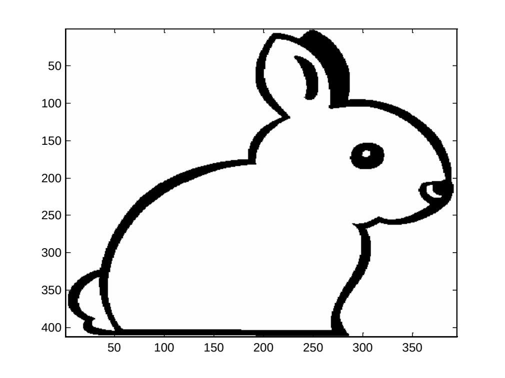
Bottom to Top Segmentation - properties

- ↑ Based on actual rate and distortion-objective.
- ↑ Rate is always decreasing when merges are performed.
- ↑ Low complexity - only merge actions are considered.
- ↑ Results insensitive to initialization.
- ↓ Cost increases when the order is raised.
- ↓ Cannot perform splits.

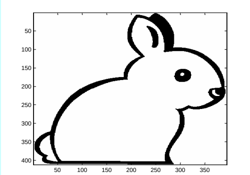
Simulation results

Rate / Distortion Using IPs



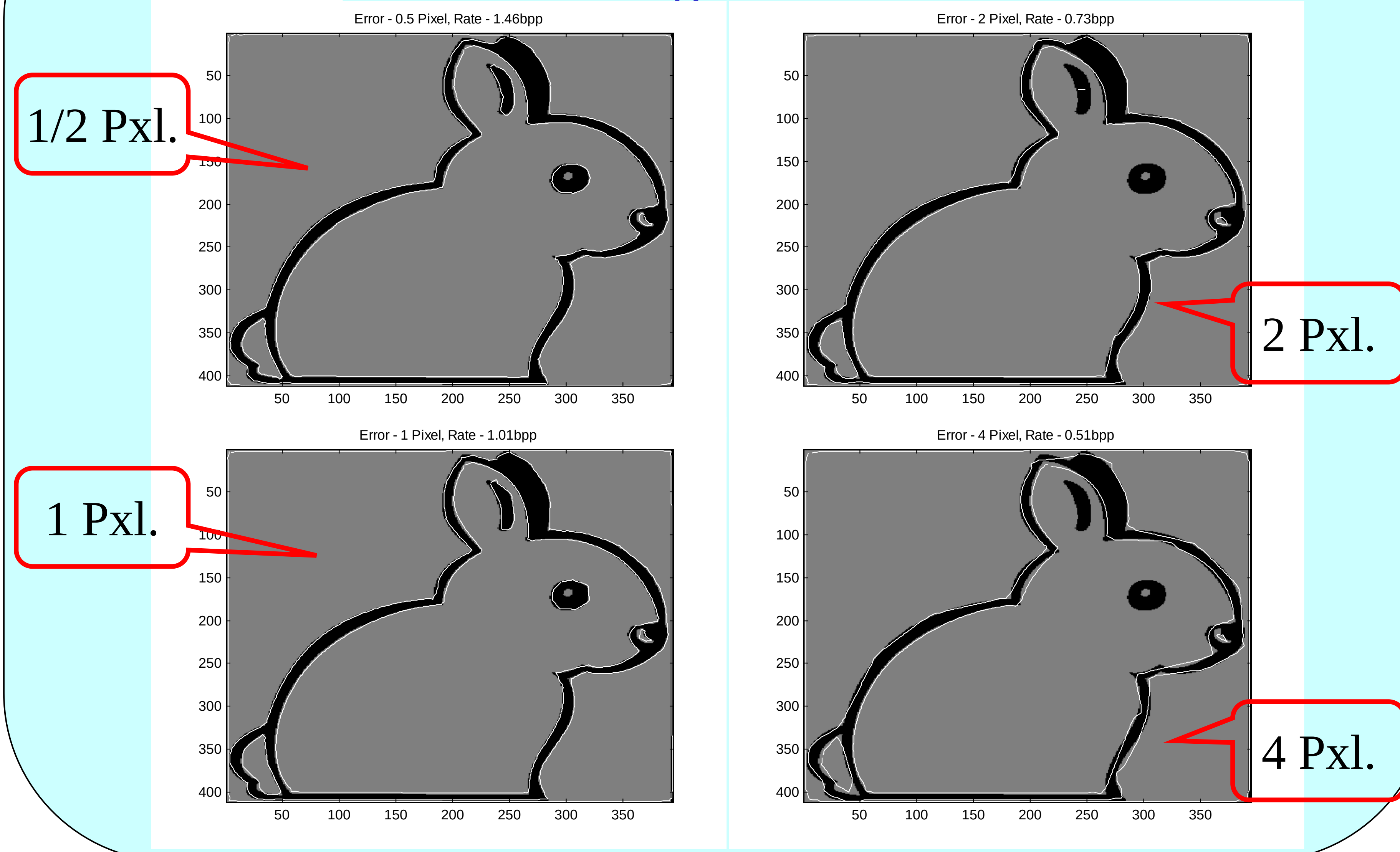


8 Neighbor Chain-Code: 13359 bits
4 Neighbor Chain-Code: 10818 bits



Simulation results

Restored images with different distortion



Summary

- A complete scheme for boundary coding was presented:
  - Segmentation of boundaries into efficiently coded curves.
  - Fitting implicit polynomials to curves, with robustness to coefficient quantization, and constraints that allow reconstruction.
  - Data reconstruction from polynomial coefficients and side information.

Summary (cont'd)

- Robustness to quantization noise also brings about robustness to data noise:

Improved fitting

Reduced $S_{\bar{a}}^{u}$ $\xrightarrow{\text{Improved fitting}}$ Reduced $S_{data}^{\bar{a}}$

Sensitivity of zero set position to coefficient changes

Sensitivity of coefficients to data point changes

- Fitting algorithm extended to 3D surfaces.

Future work

- This work investigated the usage of implicit polynomials for contour coding.
- Where object contours in image sequences are to be coded, 3D IPs may be used.
  - Setting the time as the 3rd axis ('z' axis), 3D IPs can be used to describe the changing contour of an object in several frames.
  - Higher compression should be achieved.