Skeleton based Visual Pattern Recognition: Applications to Tabletop Interaction

informatika fakultatea facultad de informática

INNOVAE VISION
Comunicación Interactiva

Andoni Beristain Iraola

PhD Advisor:

Dr. Manuel Graña Romay

Contents

1. Introduction
2. Motivation
3. Theoretical Background
4. Theoretical Results
5. Experimental Results
6. Conclusions

1. **Introduction**
2. Motivation
3. Theoretical Background
4. Theoretical Results
5. Experimental Results
6. Conclusions

1. Introduction

- PhD. carried out in an university-company collaboration context:
  - Grupo de Inteligencia Computacional GIC (research group from The University of The Basque Country).
  - Innovae Vision (spin-off company created by early GIC members).

1. Introduction

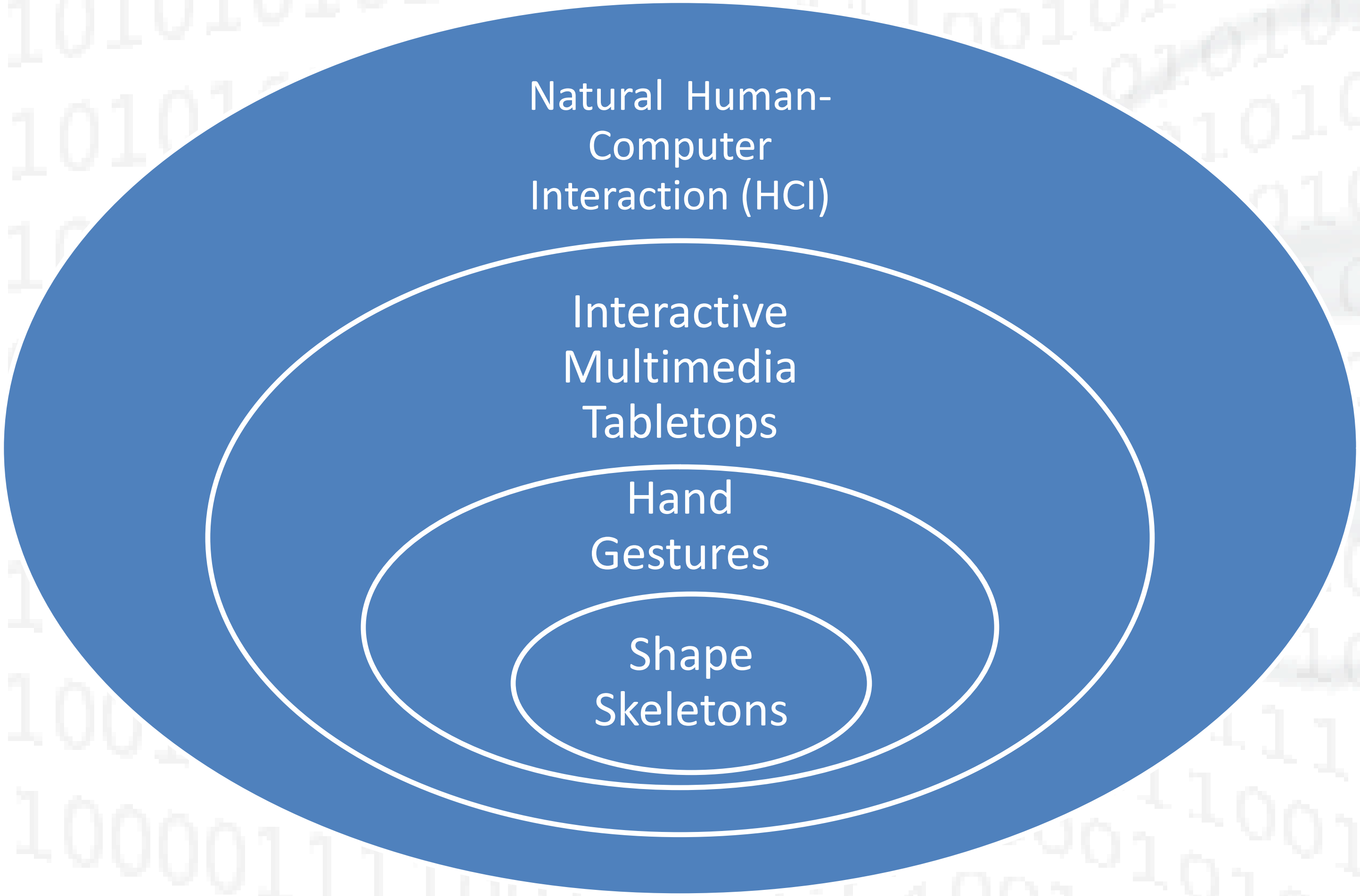


Fig 1.1: Research context.

1. Introduction

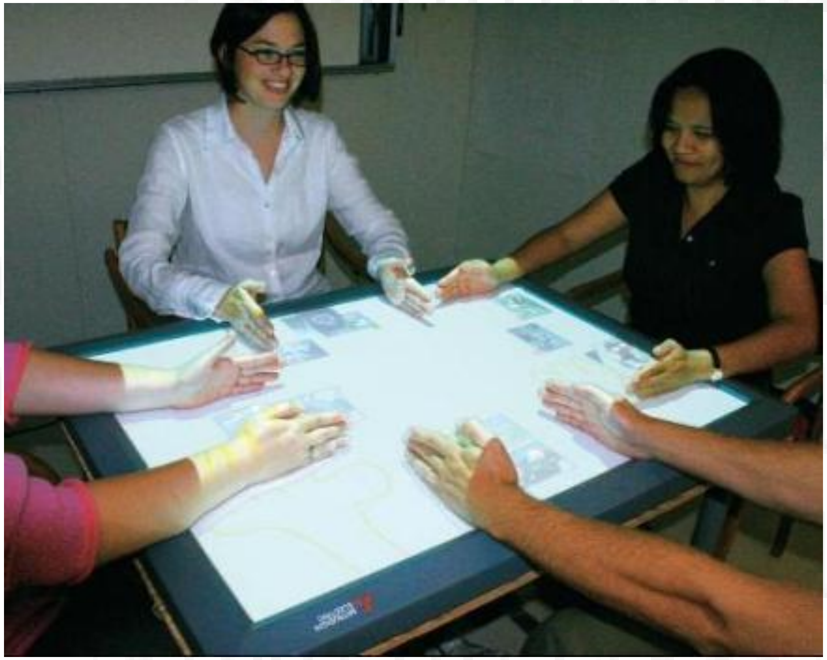
- Research objectives:
  - Real-time hand gesture recognition in a Tabletop context.
  - Design of a **stable** and **efficient** skeleton computation algorithm.

1. Introduction
2. **<u>Motivation</u>**
3. Theoretical Background
4. Theoretical Results
5. Experimental Results
6. Conclusions

2. Motivation

- Interactive Multimedia Tabletops (IMT) as a new paradigm in natural HCI.
  - Computer system with a tabletop shape.
  - Natural interaction methods, usually multimodal.
  - Multiple simultaneous users.
  - Oriented to multimedia as the basic data type, with a visually appealing GUI.

2. Motivation



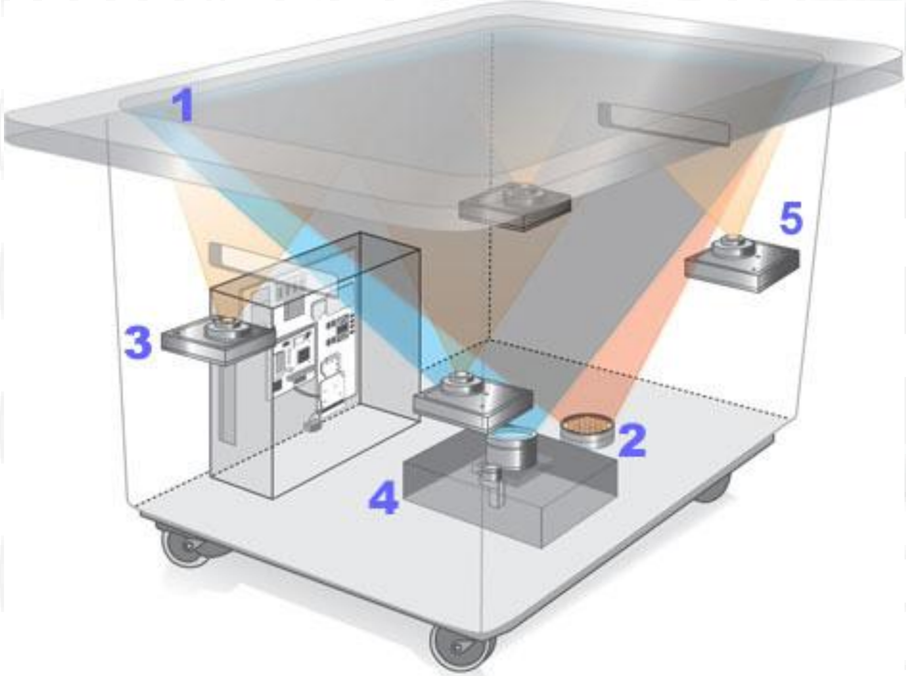


Fig. 2.1: Tabletop examples.

2. Motivation

- Tabletop design proposal including hand gesture interaction.
  - Hand gestures are natural and powerful.
  - They are contactless.
  - They can be combined with multi-touch interaction.

2. Motivation

- The background removal problem is avoided by our physical system design using polarizing filters.
  - – Removes the dynamic GUI from the camera view.
  - – Mostly avoids environmental lighting issues.
  - – Foreground segmentation becomes a trivial task.

2. Motivation

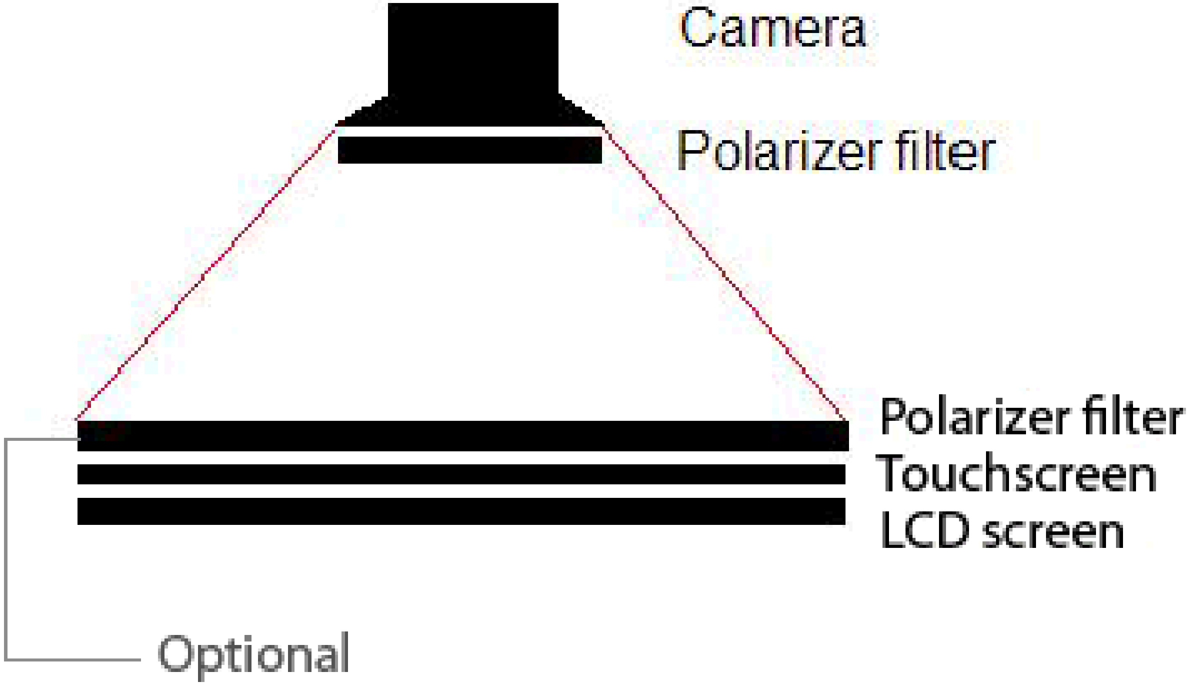


Fig. 2.2: Camera and tabletop setup.

2. Motivation

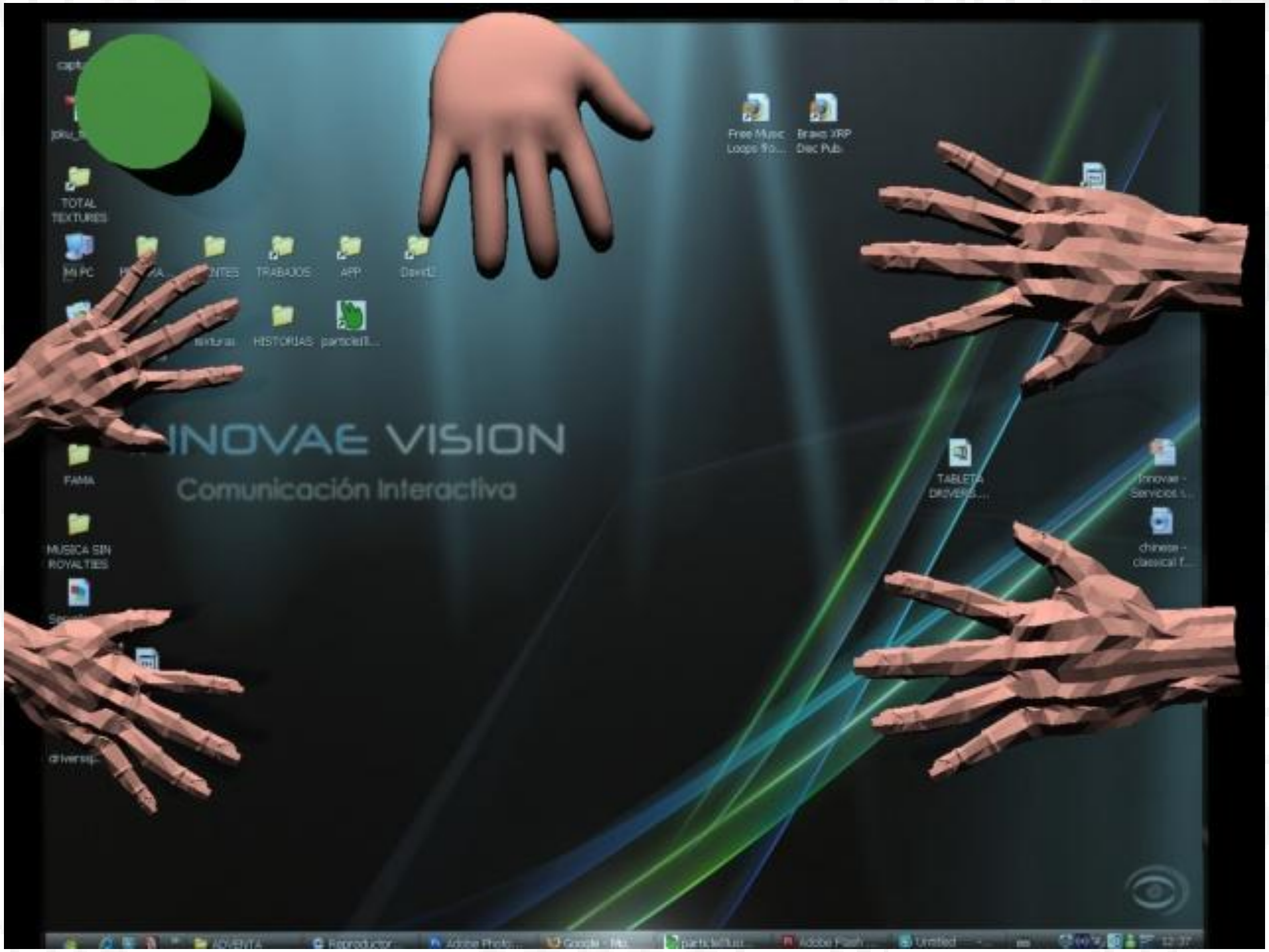


Fig. 2.3: Real view from the camera point of view.

2. Motivation

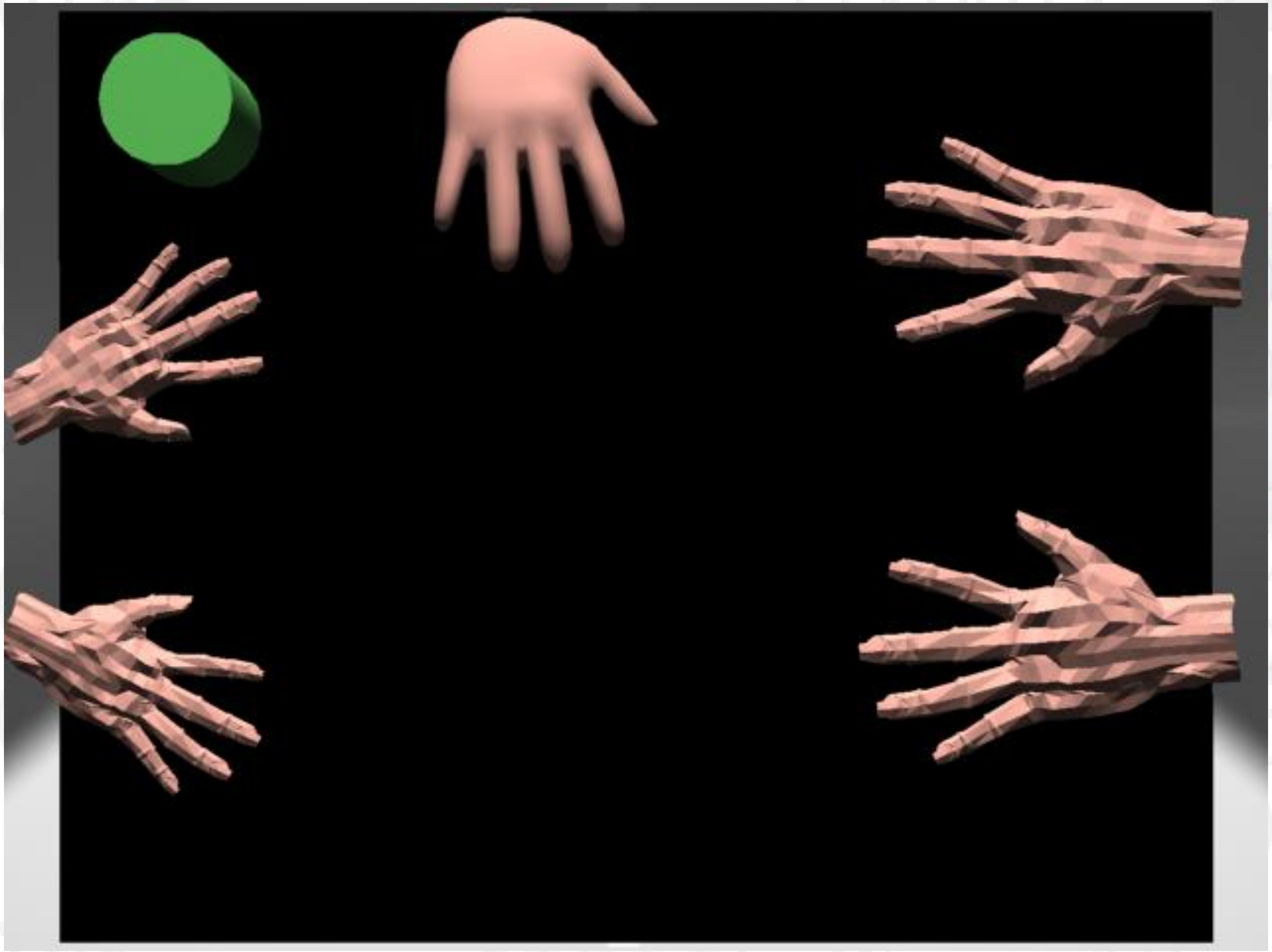


Fig. 2.4: Camera view (polarized).

2. Motivation

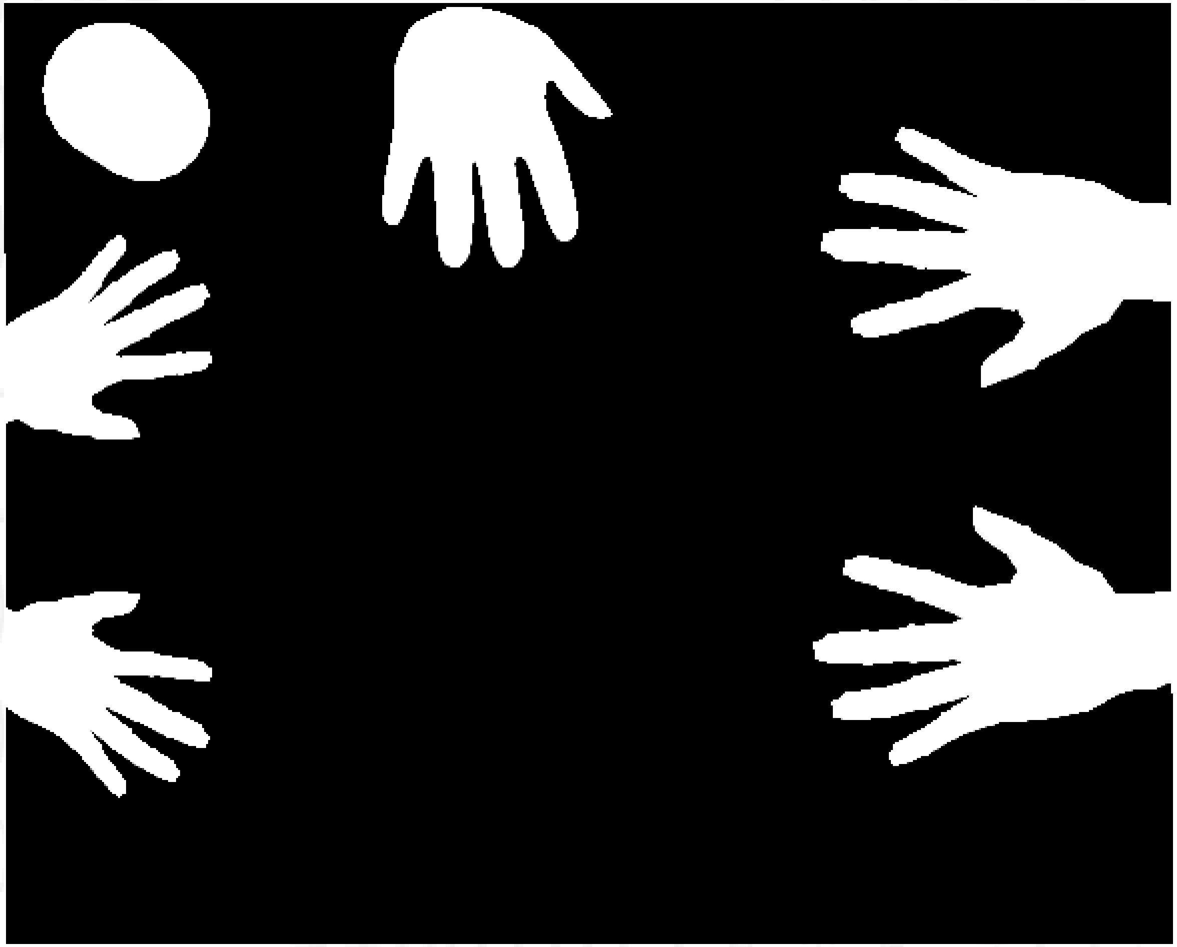


Fig. 2.5: Thresholded binary image.

2. Motivation

- We are interested in dynamic gestures.
  - Hand trajectory.
  - Hand shape variations.

2. Motivation

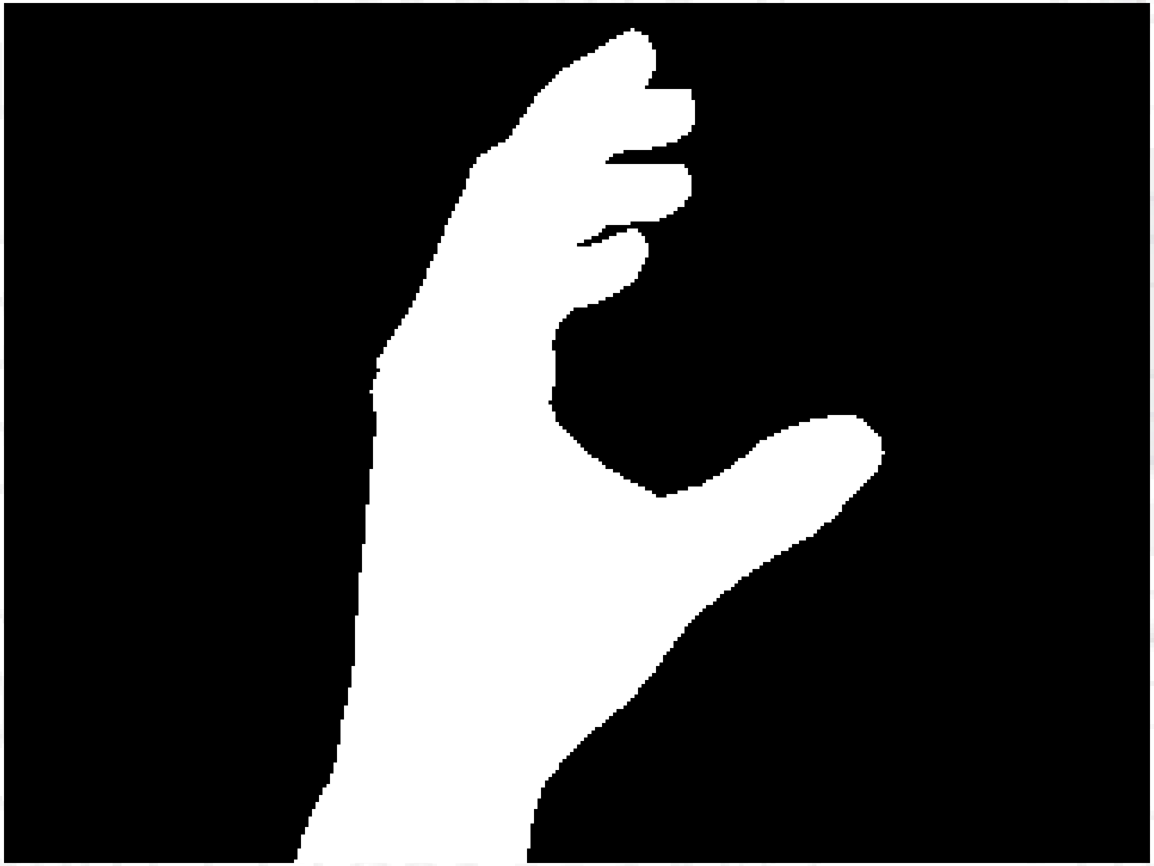


Fig. 2.6: Pose change dynamic gesture example.

2. Motivation

- Challenges in hand gesture recognition on tabletops:
  - Hands are biological amorphous objects subject to many kind of deformations.
  - Deal with scale and three-dimensional rotation.
  - Handle occlusions produced in simultaneous interaction.
  - Distinguish between user’s hand, wrist and arm.

2. Motivation

- Skeleton as shape representation method:
  - – Natural.
  - – Flexible.
  - – Robust to non-linear deformation.
  - – Robust to rotation and scale changes.

1. Introduction
2. Motivation
3. **Theoretical Background**
4. Theoretical Results
5. Experimental Results
6. Conclusions

3. Theoretical Background

- **Intuitive Skeleton Def.** (Grassfire transform): *“[*]Imagine that we have a grass field, where the field has the form of a given shape. If one sets fire at all points on the boundary of that grass field simultaneously, then the skeleton is the set of points where two or more wavefronts meet”.*

[*]Extracted from the Wikipedia

3. Theoretical Background

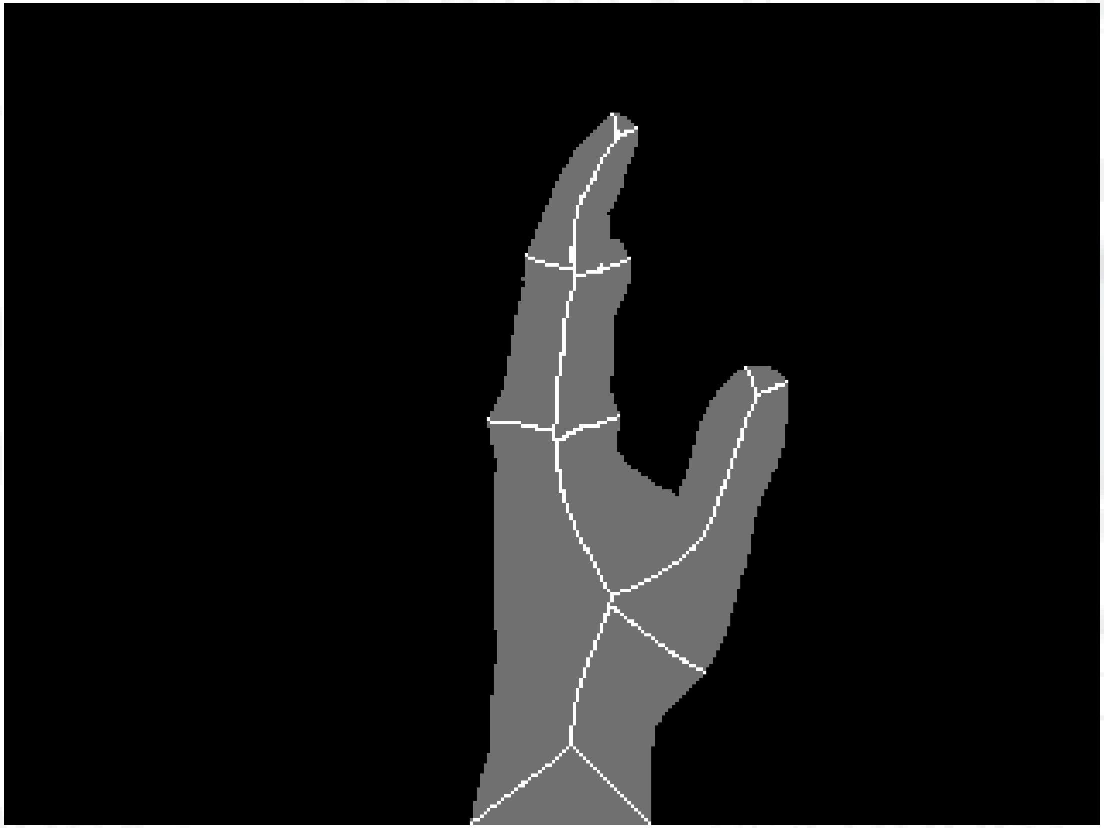


Fig. 3.1: Hand shape skeleton example.

2. Motivation

- Advantages of skeletons for shape representation and recognition:
  - Flexible.
  - Natural.
  - Easily scale and rotational invariant.
  - Robust to some non-linear deformation, which can be very disturbing for other approaches.

2. Motivation

- Disadvantages of skeletons for shape representation and recognition:
  - **Low robustness under noise in the shape boundary.**
  - **Computational complexity.**
  - Ambiguity of the skeleton if the Distance Transform (DT) Value is not included.
  - Complexity of most matching algorithms, based in attributed graph homomorphism.

3. Theoretical Background

- Raw Skeleton computation techniques:
  - Marching front skeleton (simulate grassfire transform):
    - Iterative procedure, high computational cost.
    - Not robust under rotation.
  - Distance Transform function skeleton:
    - Connectivity not guaranteed.
    - Efficient, O(n), with n=number of pixels in shape.
  - Voronoi skeleton:
    - O(nlogn), n=number of Voronoi sites.
    - Especially sensitive to noise in the boundary.

3. Theoretical Background

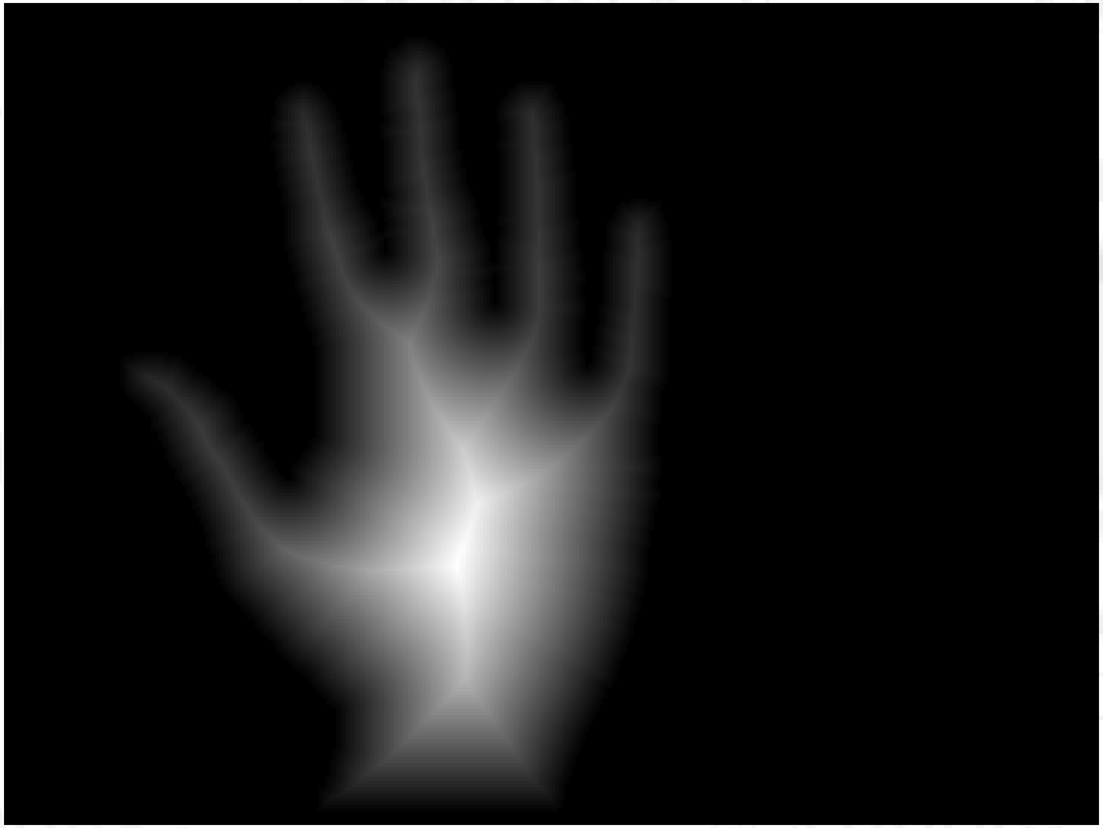


Fig. 3.2: Hand shape Distance Transform Function.

3. Theoretical Background

- **Def.** Voronoi Tessellation of a point set (**V**):
  - Partition of the space into convex regions (Voronoi polygons) around each of the points in the set, called Voronoi sites, so that every point in a region is closer to his Voronoi site than any other Voronoi site.
- **Def.** Voronoi edge ($\mathbf{s}_{\mathbf{ij}}$):
  - Limit between two Voronoi polygons. All the points in the Voronoi edge are equidistant to two Voronoi sites (i,j).

3. Theoretical Background

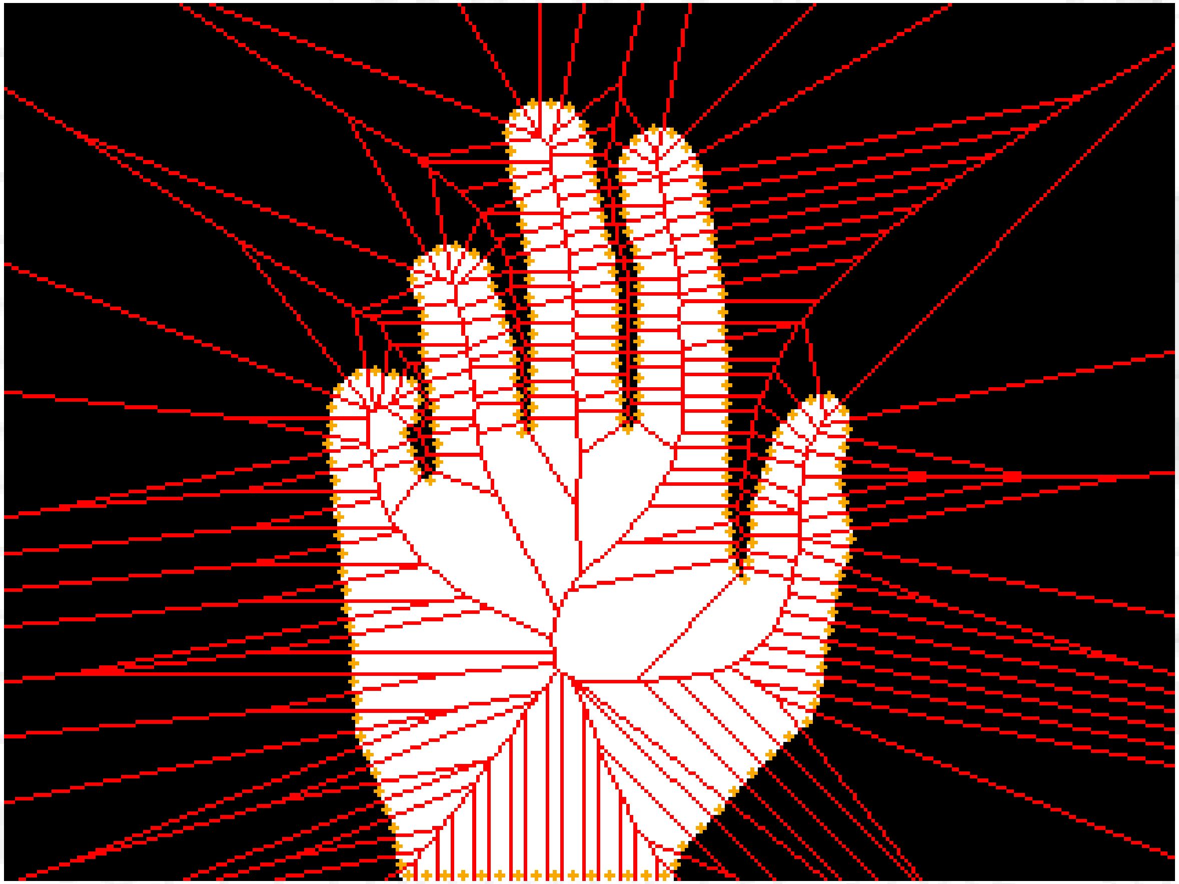


Fig. 3.3: Voronoi Tessellation of the shape boundary curve point set.

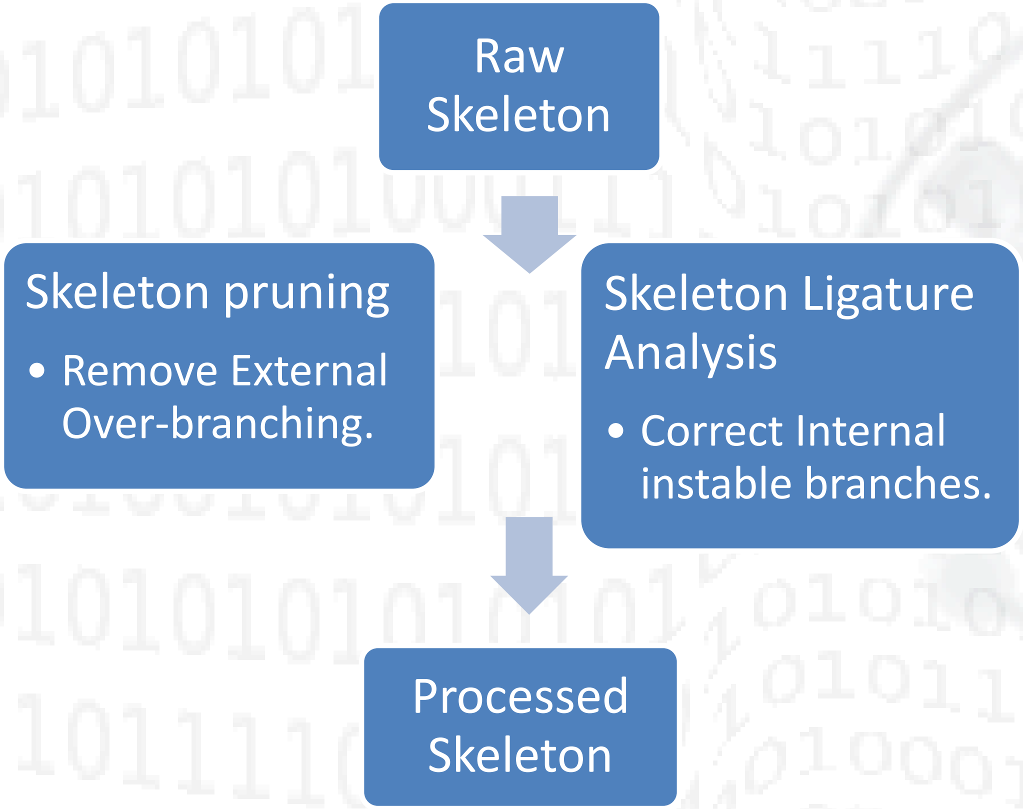
3. Theoretical Background

- **Key Def.** Skeleton stability:
  - *Similar shapes, in terms of the Hausdorff distance, must produce similar skeletons.*

3. Theoretical Background

- There are some post-processing of the raw skeleton:
  - Obtain more **stable** and **concise** skeletons.
  - Post-processing techniques:
    - Skeleton pruning.
    - Skeleton ligature analysis.

3. Theoretical Background



Raw Skeleton

Skeleton pruning

- Remove External Over-branching.

Skeleton Ligature Analysis

- Correct Internal instable branches.

Processed Skeleton

Fig. 3.4: Skeletonization procedure.

3. Theoretical Background

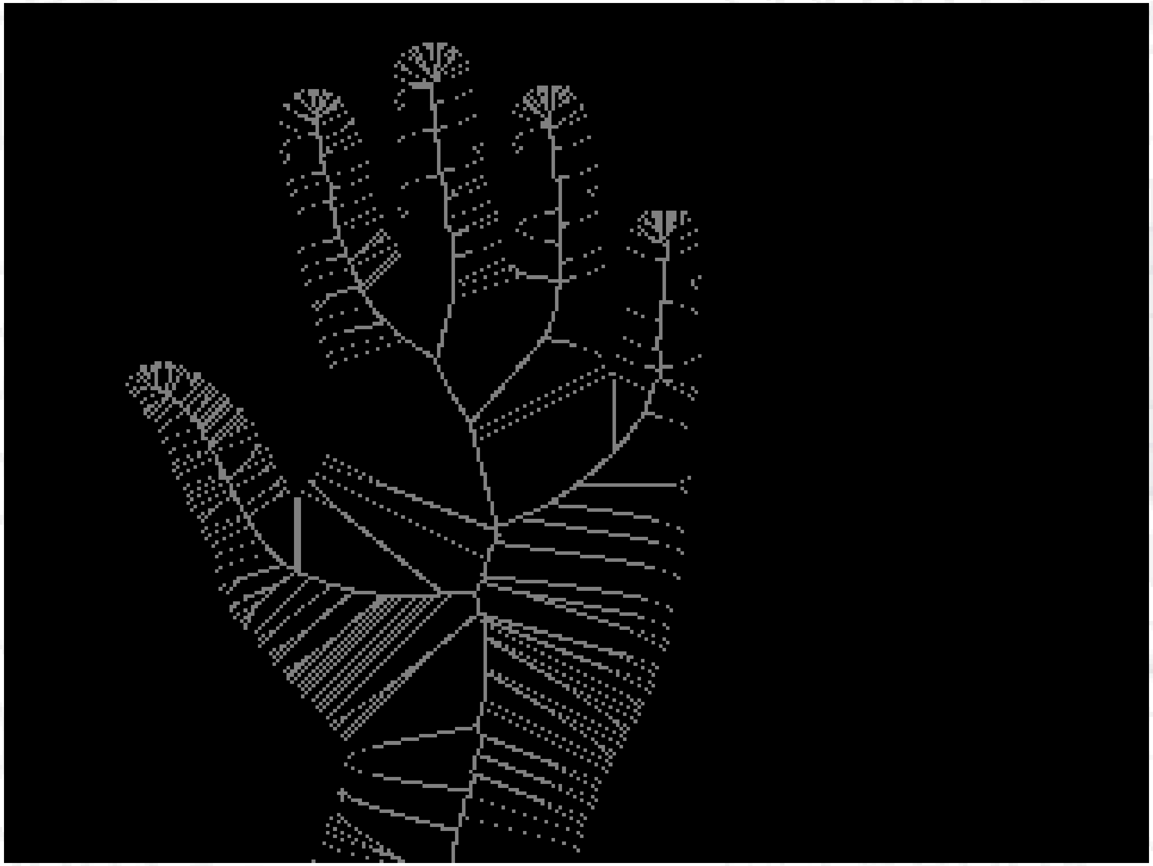


Fig. 3.5: Raw skeleton of a hand shape computed using a DT based skeletonization algorithm.

3. Theoretical Background

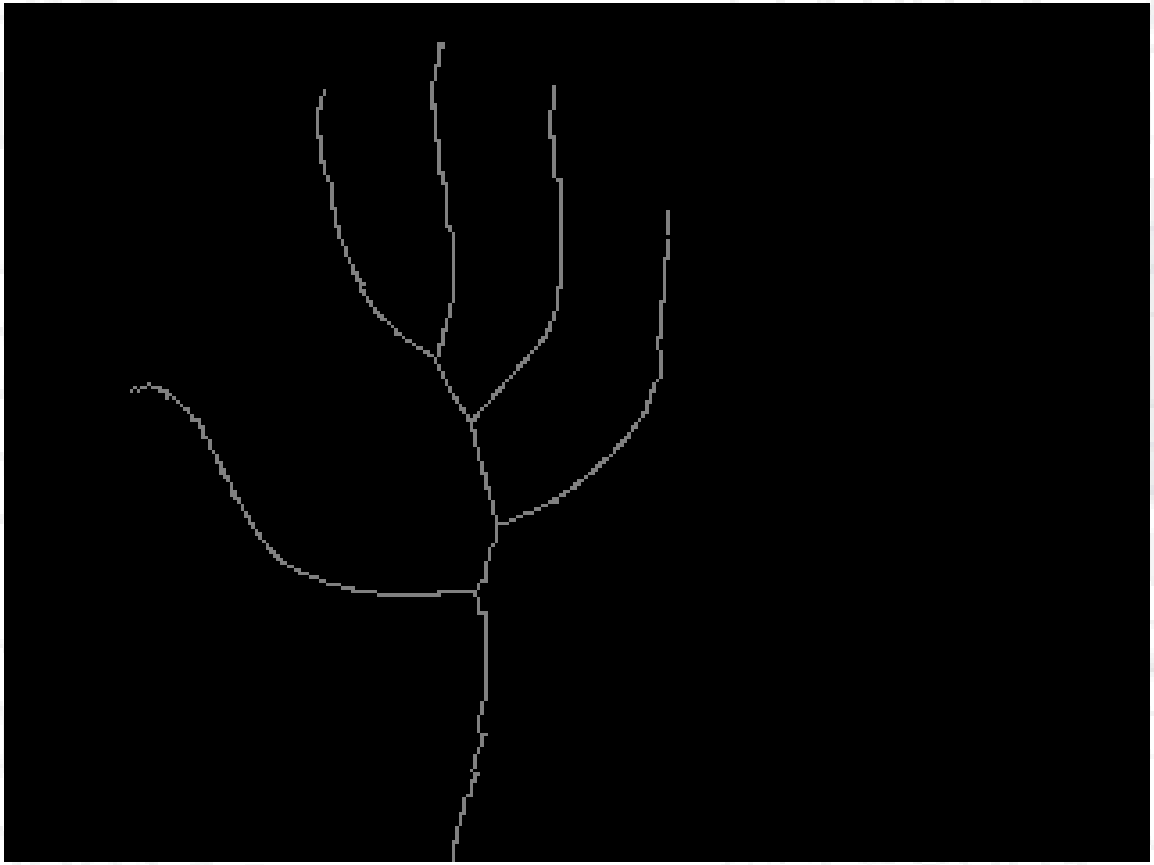


Fig. 3.6: Processed skeleton of an open hand shape.

1. Introduction
2. Motivation
3. Theoretical Background
4. **Theoretical Results**
5. Experimental Results
6. Conclusions

1. Introduction
2. Motivation
3. Theoretical Background
4. **Theoretical Results**
    1. *Efficient and Stable Voronoi Skeleton.*
    2. *Formal statement of some properties of the algorithm.*
5. Experimental Results
6. Conclusions

1. Introduction
2. Motivation
3. Theoretical Background
4. Theoretical Results
    1. ***Efficient and Stable Voronoi Skeleton.***
    2. *Formal statement of some properties of the algorithm.*
5. Experimental Results
6. Conclusions

4.1. Efficient and Stable Voronoi Skeleton

I. Subsample the shape boundary curve ***C*** of the shape ***F***, obtaining polygon ***P***.

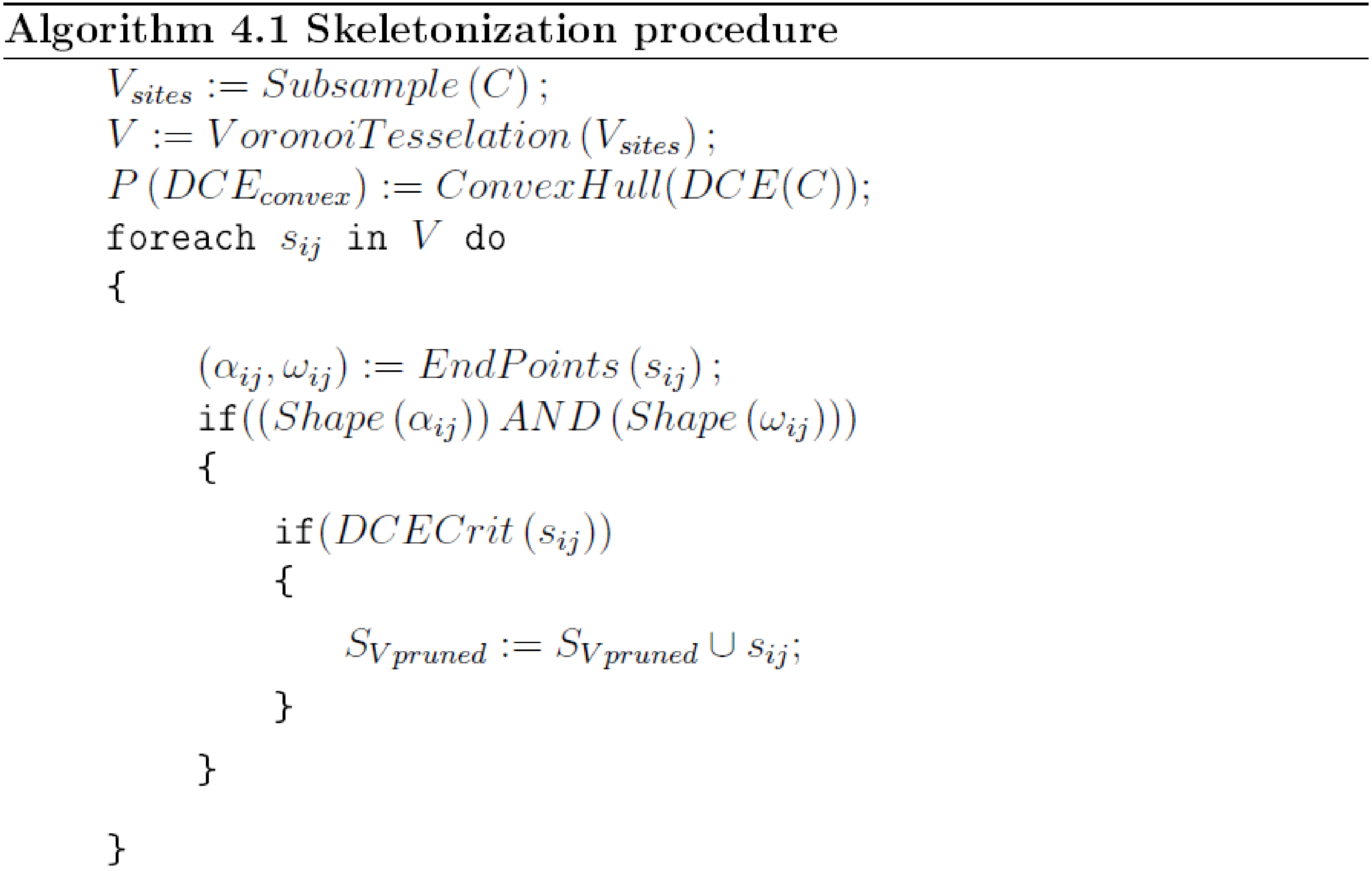
II. Compute the raw Voronoi Skeleton ***V*** using the vertices in ***P*** as Voronoi sites ($\boldsymbol{V_{sites}}$).

III. Prune the raw Voronoi Skeleton in two stages.

   i. Remove the Voronoi edges not completely contained in ***F***.

   ii. Remove the Voronoi edges not fulfilling the Discrete Curve Evolution criterion.

Efficient and Stable Voronoi Skeleton: The Algorithm



Algorithm 4.1 Skeletonization procedure

```
$V_{sites} := Subsample\,(C)\,;$
$V := VoronoiTesselation\,(V_{sites})\,;$
$P\,(DCE_{convex}) := ConvexHull(DCE(C));$
foreach $s_{ij}$ in $V$ do
{
    $(\alpha_{ij}, \omega_{ij}) := EndPoints\,(s_{ij})\,;$
    if$((Shape\,(\alpha_{ij}))\;AND\;(Shape\,(\omega_{ij})))$
    {
        if$(DCECrit\,(s_{ij}))$
        {
            $S_{V_{pruned}} := S_{V_{pruned}} \cup s_{ij};$
        }
    }
}
```

I. Shape boundary curve subsampling (*Subsample (C)*)

- Reduces the Voronoi Tessellation computation cost.
- Improves the stability of the skeletonization procedure reducing the effect of minor noise in the shape boundary curve.
- Optional step.
- The sampling criterion must guarantee that the Voronoi skeleton obtained from this subsampling produces a close approximation of the original shape’s Voronoi skeleton.

I. Shape boundary curve subsampling (*Subsample (C)*)

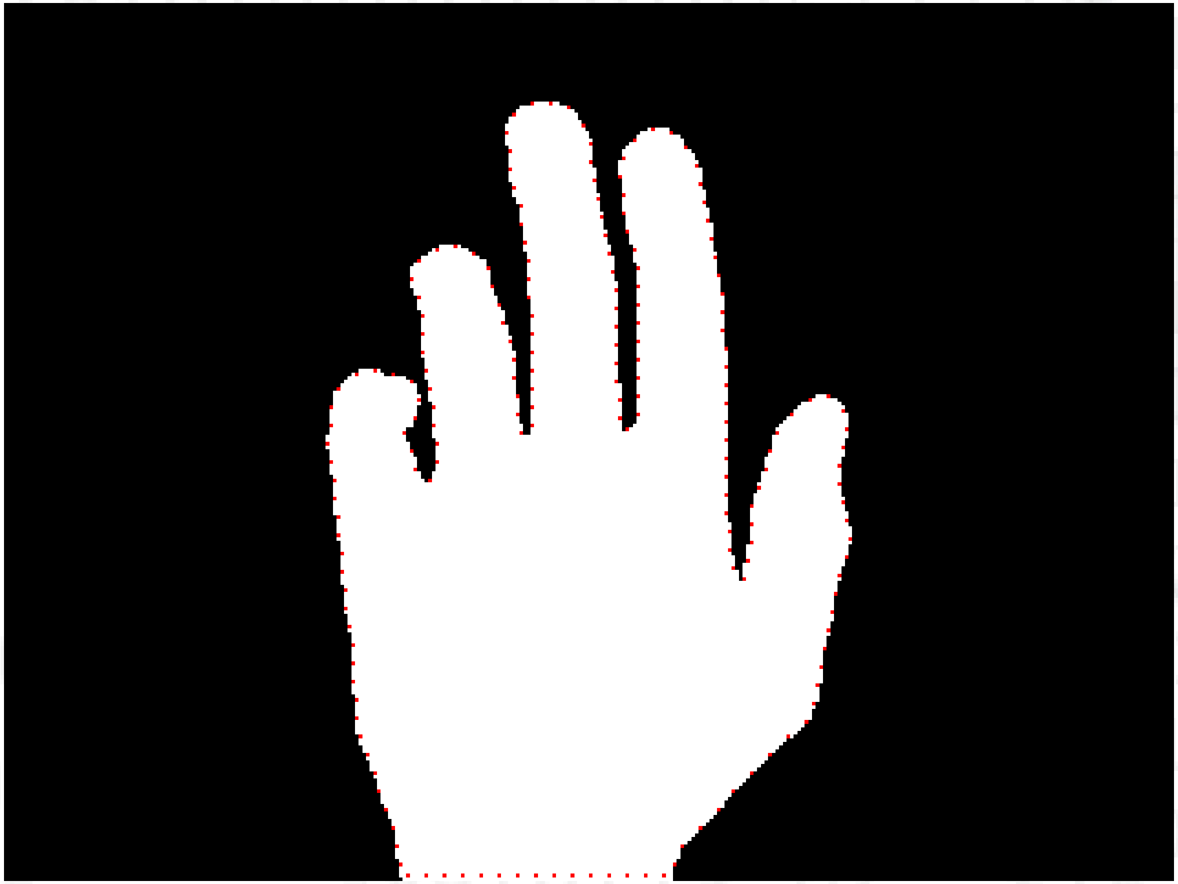


Fig. 4.1.1: Uniform shape boundary sampling.

II. Raw Voronoi Skeleton computation (*VoronoiTessellation* (V_{sites}))

- Computational complexity is *O(nlogn)*, with n the number of Voronoi sites.
- Any conventional algorithm can be used.
- Only the Voronoi edges at least partially contained in the shape constitute the raw Voronoi Skeleton.

II. Raw Voronoi Skeleton computation

($VoronoiTessellation\ (V_{sites})$)

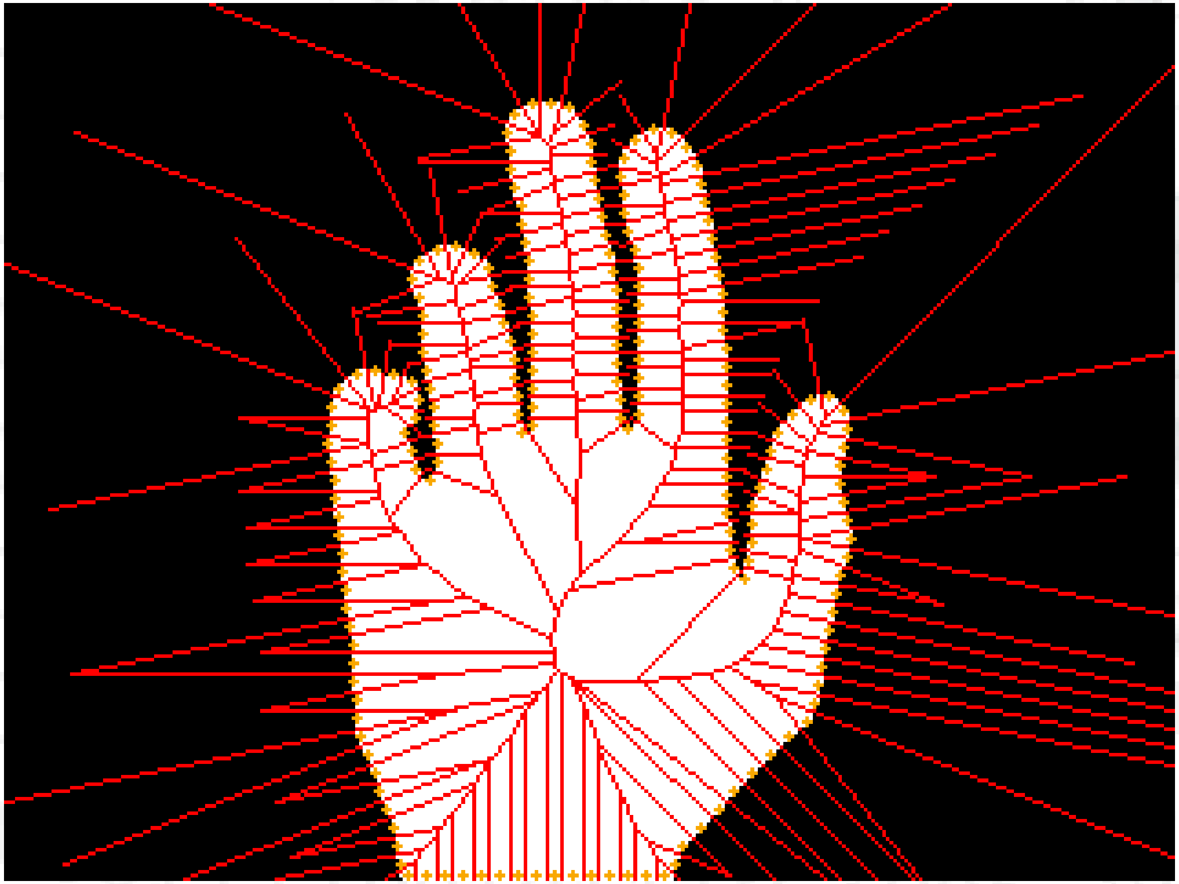


Fig. 4.1.2: Raw Voronoi **Skeleton.**

III.i. First pruning stage ($Shape(\alpha_{ij})$)

- Voronoi edges not completely contained inside the shape are removed.
- Only endpoints need to be checked to decide if the whole Voronoi edge is contained in the shape or not ($Shape(\alpha_{ij})$).
  - Mathematical proof provided.
- Extremely fast to check.

III.i. First pruning stage ($Shape(\alpha_{ij})$)

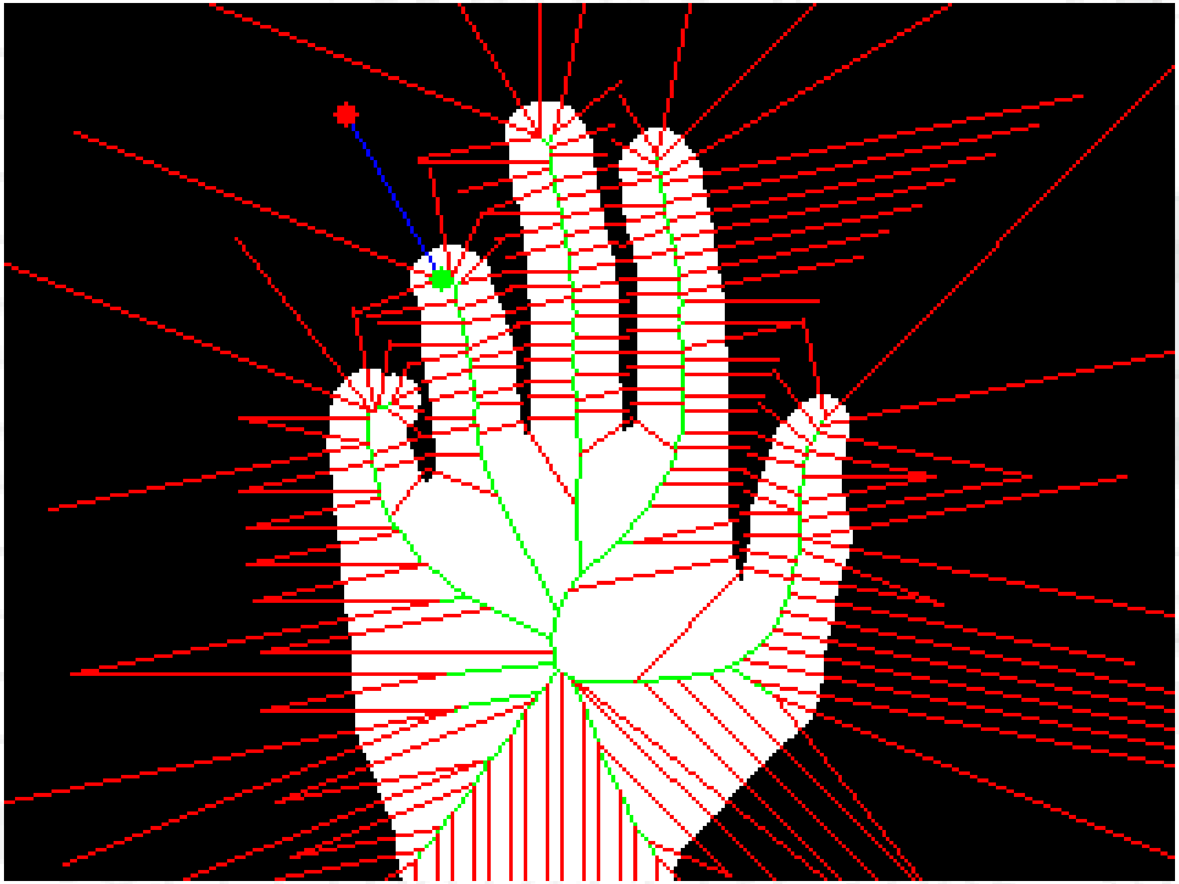


Fig. 4.1.3: Example of the first pruning stage of Beris algorithm.

III.ii. Second pruning stage ($DCECric(s_{ij})$)

Def. Discrete Curve Evolution (DCE):

- Shape boundary curve subsampling method.
- Produces a polygon which keeps the most prominent shape features.
- Iterative procedure which removes the point from the original shape boundary curve with less contribution to the shape at each step.
- The DCE polygon ($DCE(C)$) is composed of points from the original shape boundary curve.

III.ii. Second pruning stage ($DCECric(s_{ij})$)

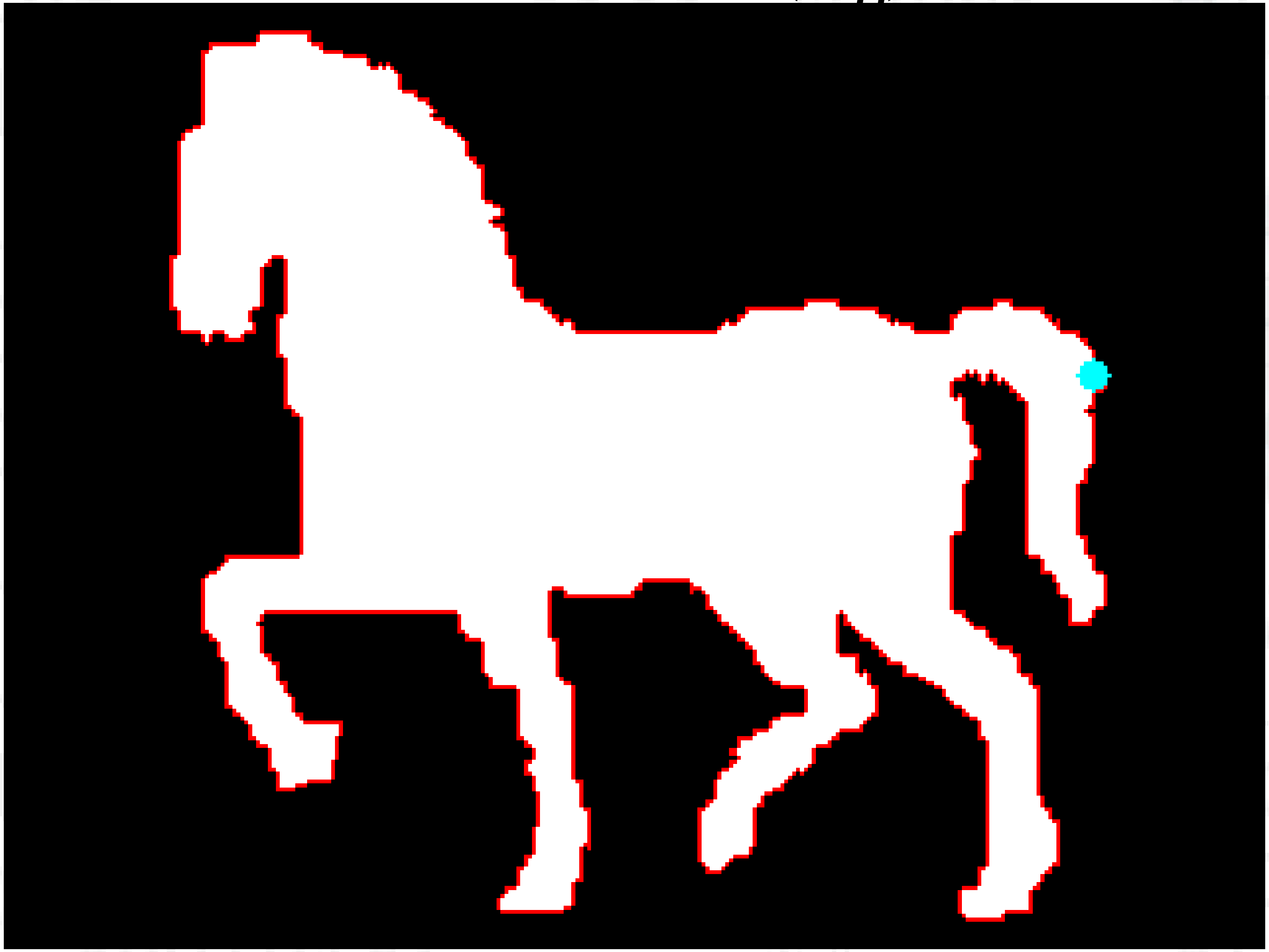


Fig. 4.1.4: DCE computation iterative procedure.

III.ii. Second pruning stage ($DCECric(s_{ij})$)

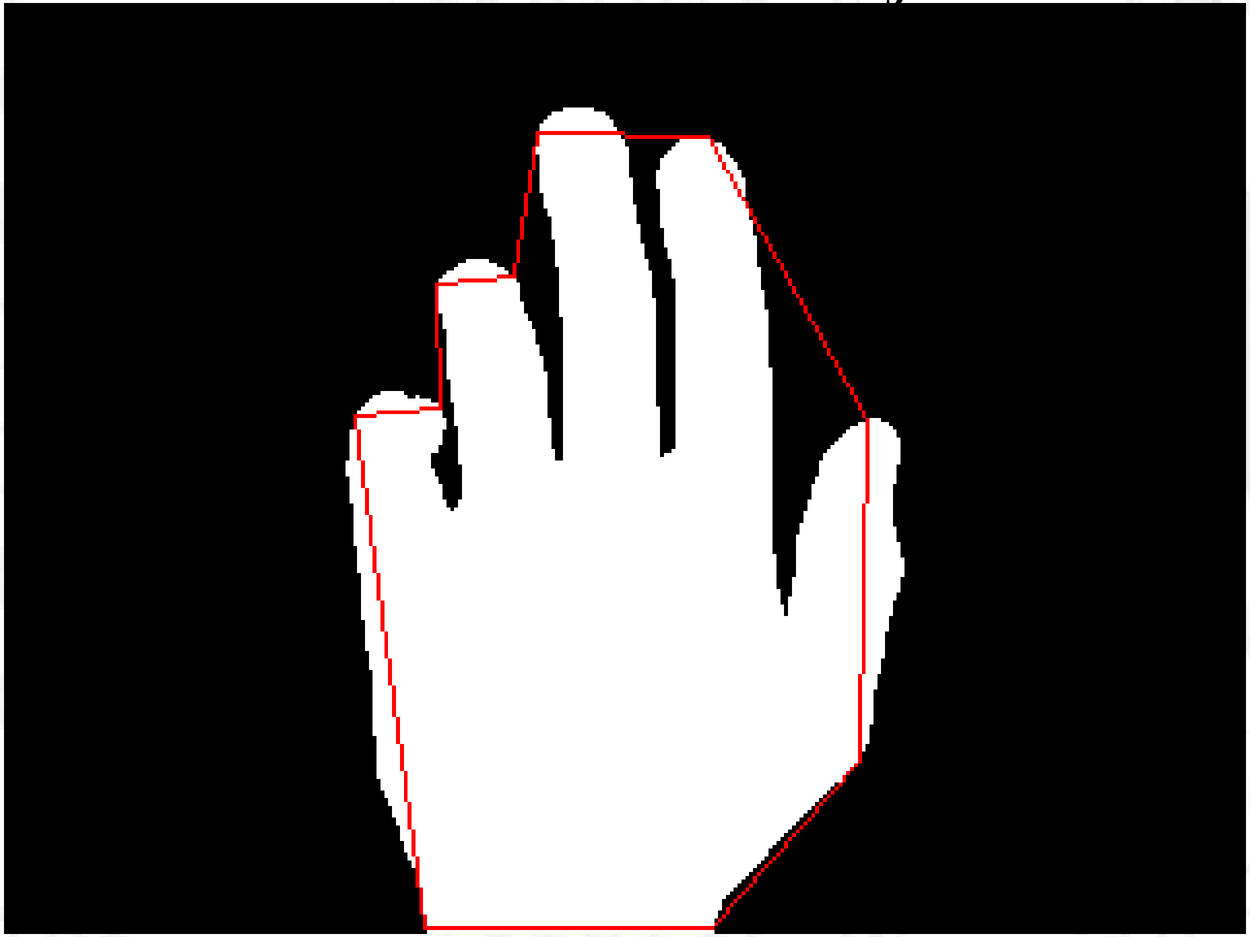


Fig. 4.1.5: Convex hull of the DCE polygon.

III.ii. Second pruning stage ($DCECric(s_{ij})$)

1. The convex hull of the DCE polygon ($ConvexHull(DCE(C))$) is obtained, with edges E_{DCE}.
2. Each e_i in E_{DCE} is associated to the subsequence of the original shape boundary curve points between the endpoints of e_i.
3. If the closest Voronoi sites to a Voronoi edge are assotiated to the same edge in E_{DCE} the edge is removed.

III.ii. Second pruning stage ($DCECric(s_{ij})$)

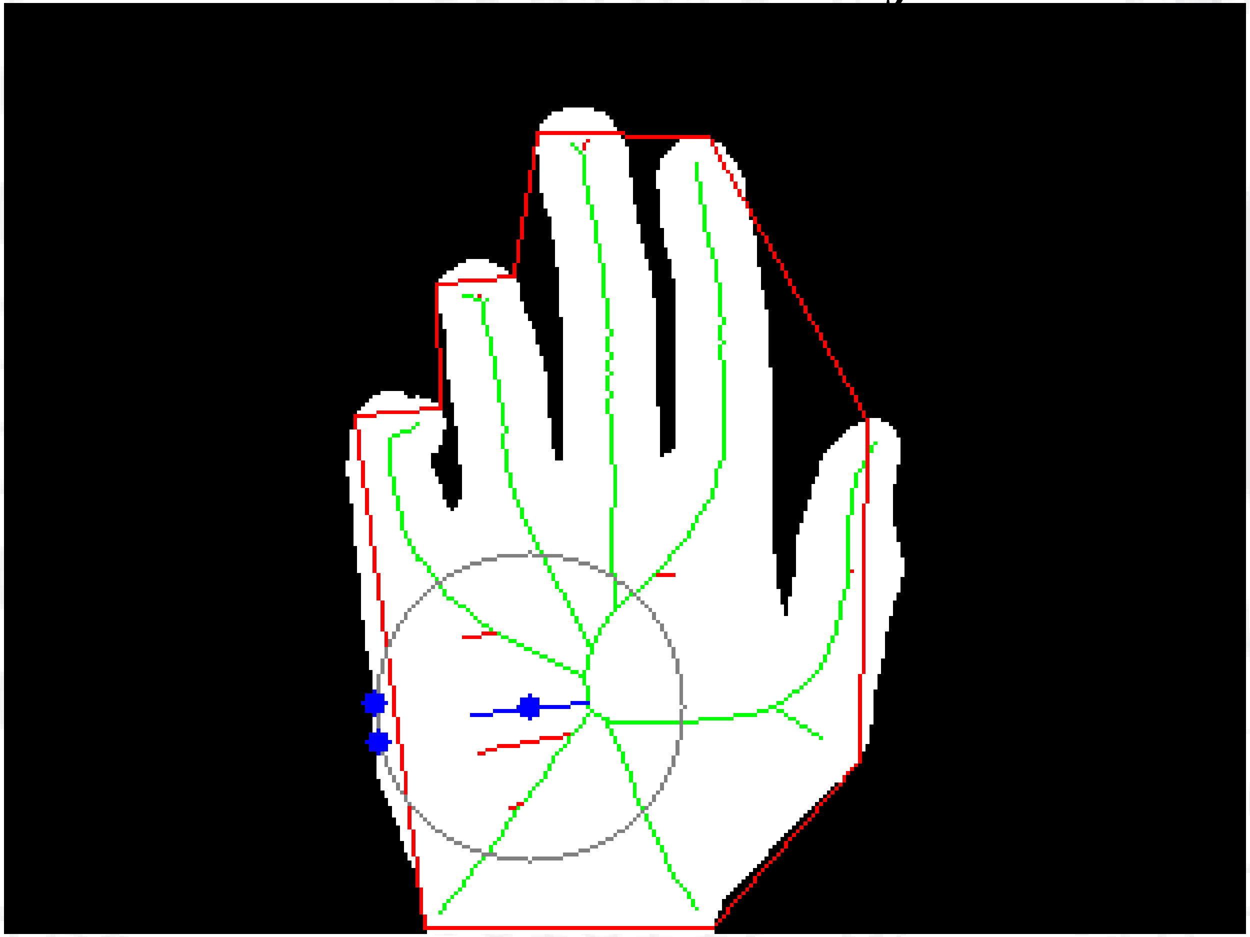


Fig. 4.1.6: DCE skeleton pruning result.

Visualization of the algorithm

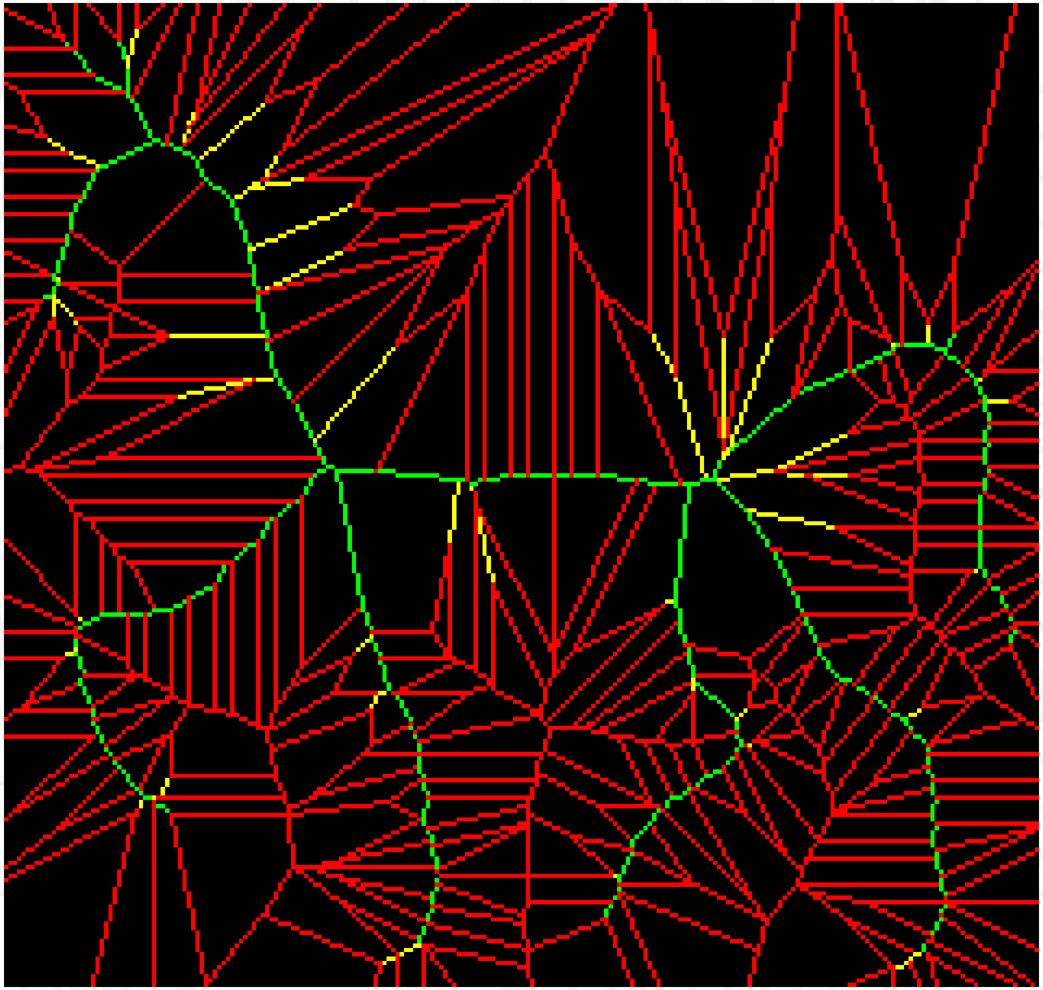
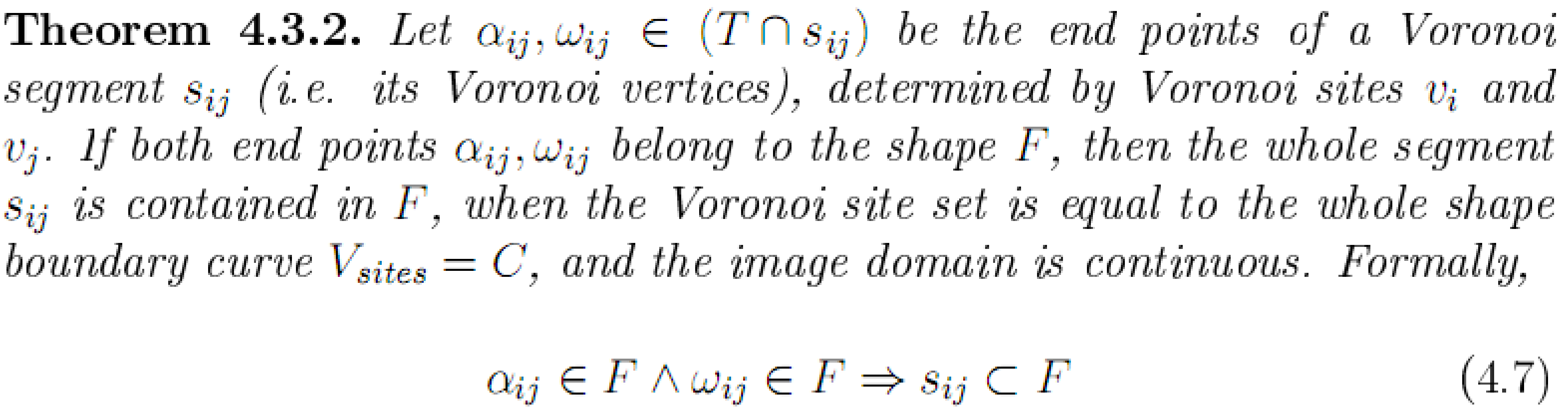


Fig. 4.1.7: Skeleton pruning steps.

- ❑Red: Voronoi edges removed by first pruning.
- ❑Yellow: Voronoi edges removed by the DCE pruning.
- ❑Green: Final skeleton.

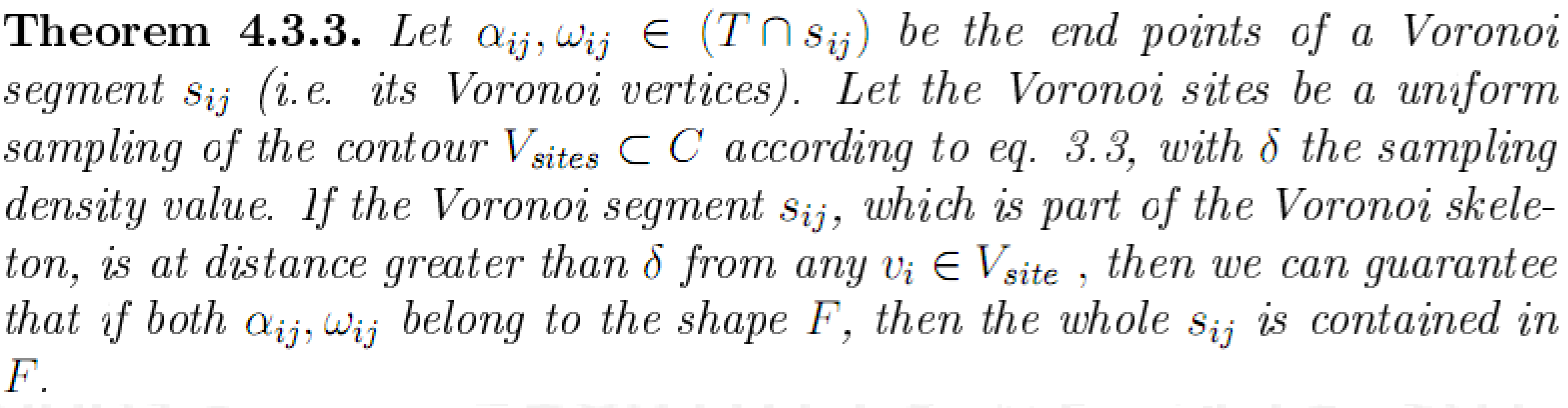
1. Introduction
2. Motivation
3. Theoretical Background
4. Theoretical Results
    1. *Efficient and Stable Voronoi Skeleton.*
    2. ***Formal statement of some properties of the algorithm.***
5. Experimental Results
6. Conclusions

4.2. Formal statement of some properties of the algorithm



Theorem 4.3.2. *Let $\alpha_{ij}, \omega_{ij} \in (T \cap s_{ij})$ be the end points of a Voronoi segment s_{ij} (i.e. its Voronoi vertices), determined by Voronoi sites v_i and v_j. If both end points α_{ij}, ω_{ij} belong to the shape F, then the whole segment s_{ij} is contained in F, when the Voronoi site set is equal to the whole shape boundary curve $V_{sites} = C$, and the image domain is continuous. Formally,*

$$\alpha_{ij} \in F \wedge \omega_{ij} \in F \Rightarrow s_{ij} \subset F \tag{4.7}$$



Theorem 4.3.3. *Let $\alpha_{ij}, \omega_{ij} \in (T \cap s_{ij})$ be the end points of a Voronoi segment s_{ij} (i.e. its Voronoi vertices). Let the Voronoi sites be a uniform sampling of the contour $V_{sites} \subset C$ according to eq. 3.3, with δ the sampling density value. If the Voronoi segment s_{ij}, which is part of the Voronoi skeleton, is at distance greater than δ from any $v_i \in V_{site}$, then we can guarantee that if both α_{ij}, ω_{ij} belong to the shape F, then the whole s_{ij} is contained in F.*

1. Introduction
2. Motivation
3. Theoretical Background
4. Theoretical Results
5. **Experimental Results**
6. Conclusions

1. Introduction
2. Motivation
3. Theoretical Background
4. Theoretical Results
5. **Experimental Results**
    1. *Real-time performance.*
    2. *Stability.*
    3. *Feature vector for shape recognition.*
6. Conclusions

5. Experimental Results

Three kind of results:

1. Check the **real-time performance** of the algorithm. The efficiency of the algorithm.
2. The **stability** of the our skeletons.
3. The effectiveness of our approach as a feature extraction for **shape classification**.

1. Introduction
2. Motivation
3. Theoretical Background
4. Theoretical Results
5. Experimental Results
   1. ***<u>Real-time performance.</u>***
   2. *Stability.*
   3. *Feature vector for shape recognition.*
6. Conclusions

5.1. Real-time Performance

- Real-time implementation details :
  - Written in C++, using Microsoft DirectShow and Open Computer Vision (OpenCV) libraries.
  - Non optimized prototypic code.

5.1. Real-time Performance

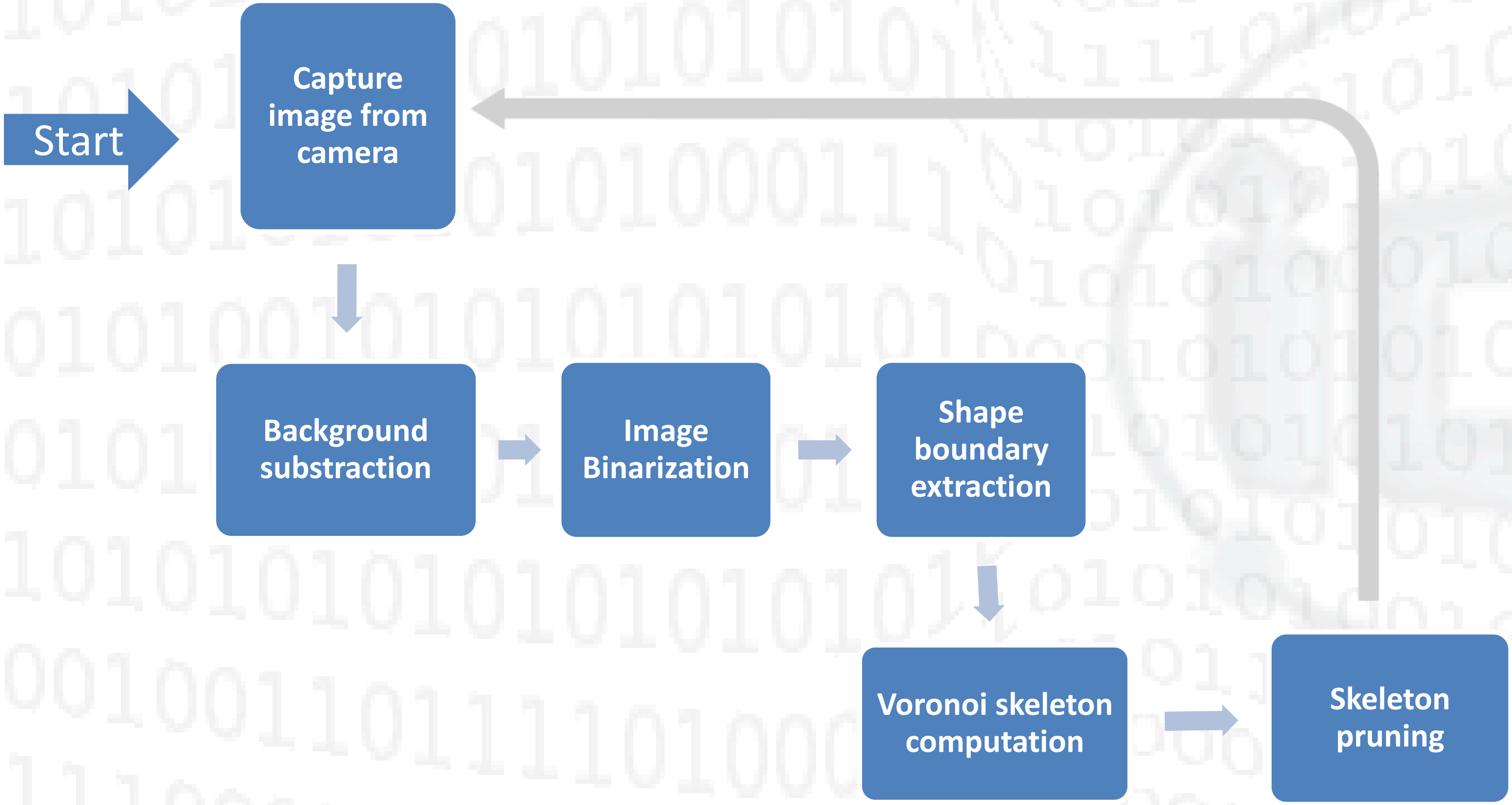

Start
Capture image from camera
Background substraction
Image Binarization
Shape boundary extraction
Voronoi skeleton computation
Skeleton pruning

Fig. 5.1.1: Real-time prototype flow schema.

5.1. Real-time Performance

- Test environment:
  - – Intel Pentium IV processor (3.0 GHz).
  - – 1GB memory.
  - – Windows XP SP2 32 bit.
- Performance of the implementation:
  - – 60 FPS for a 320x240 video capture resolution.
  - – 24 FPS for a 640x480 video capture resolution.

5.1. Real-time Performance

Efficient Voronoi Skeleton Computation and Pruning

Grupo de Inteligencia Computacional (GIC)

Fig. 5.1.2: Real-time prototype in use.

Experimental image database

- Hand gesture image sequences to test the recognition performance of different algorithms in a Tabletop hand gesture interaction context.
- 3 dynamic gestures:
  - Grab an object.
  - Point an object.
  - Turn a page.

Experimental image database

- 15 frames for each gesture sample sequence.
- 200 gesture sequence samples for each of the 3 gestures.
- 9000 images in total (15x200x3).
- Synthetic image database obtained using Poser® and modifying randomly translation, rotation, scale and deformation parameters from a basic prototypic gesture.

Database image samples

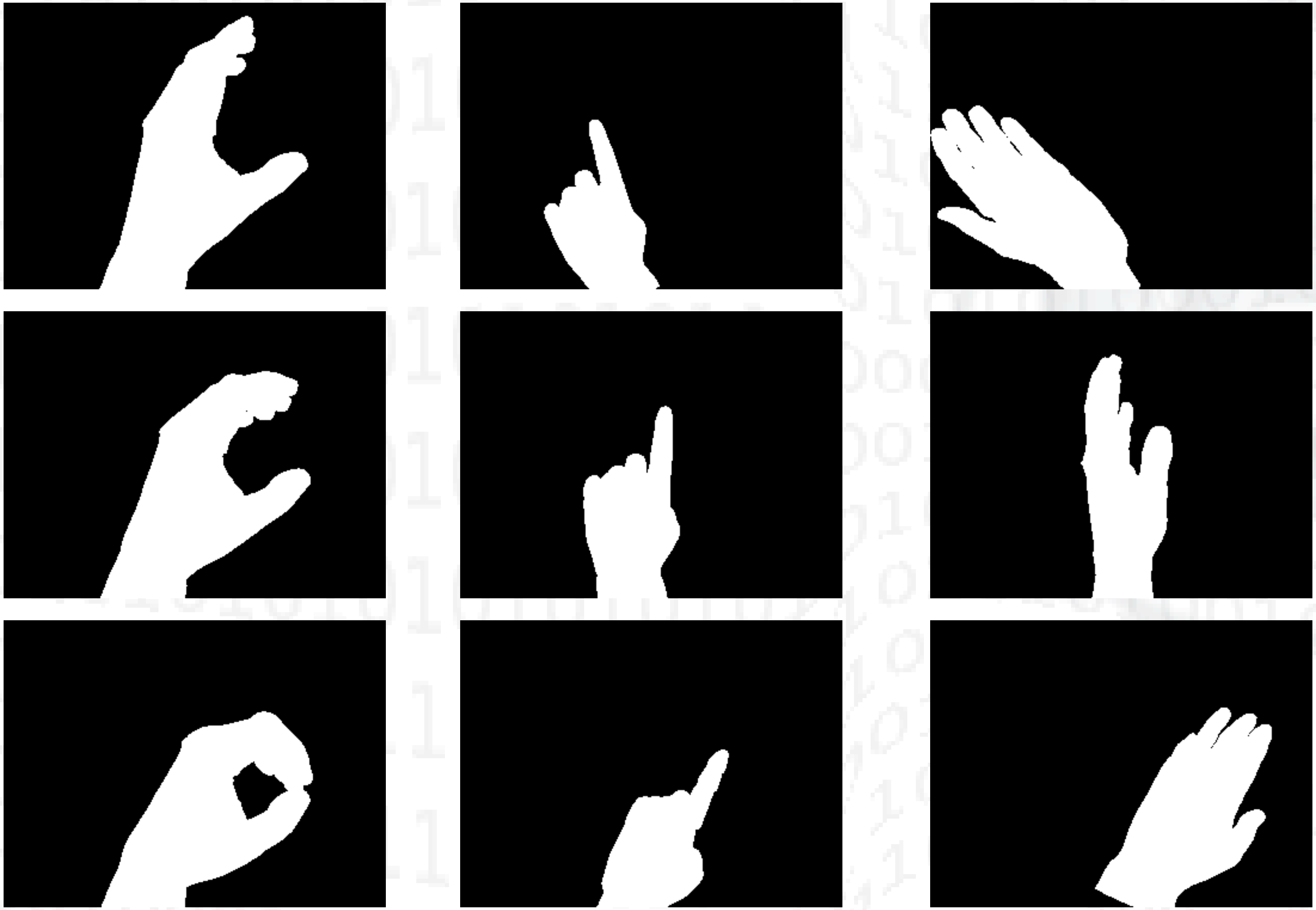


Fig. 5.2.1: Experimental image database examples.

Database animation sample

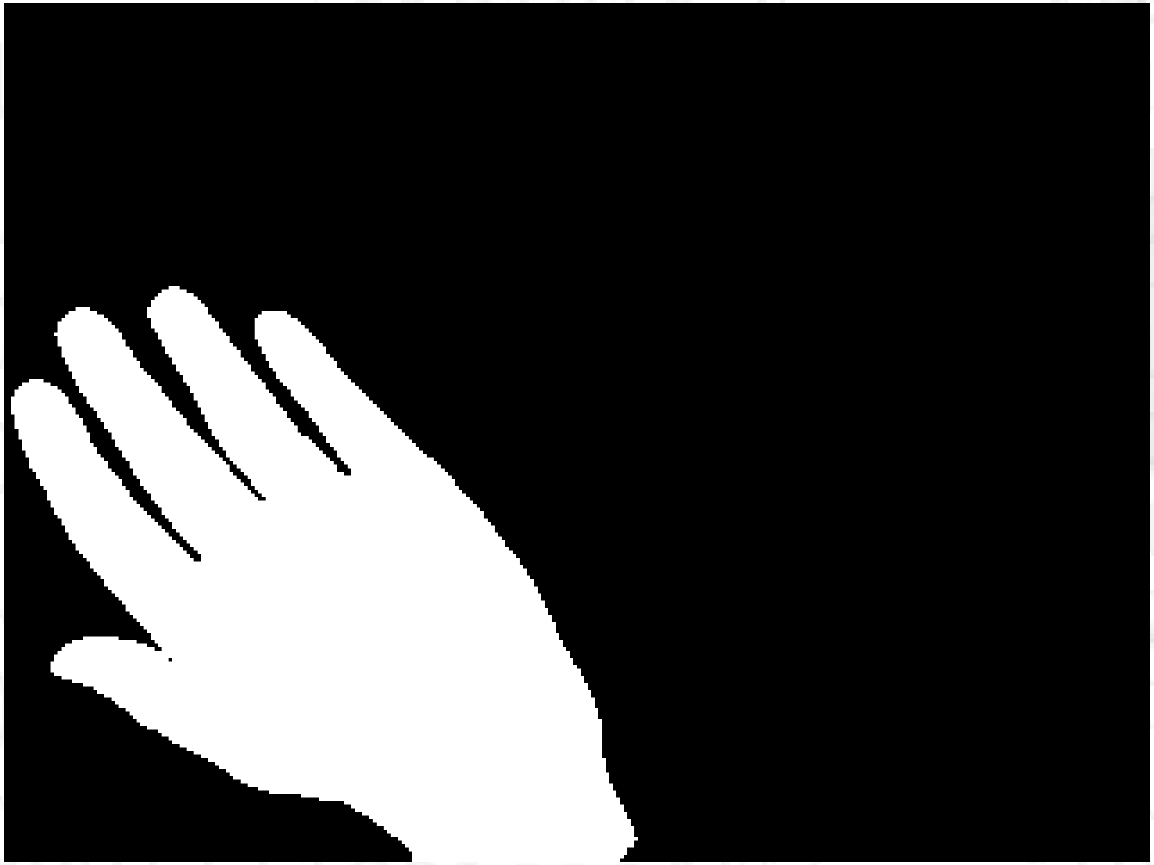


Fig. 5.2.2: Turn page gesture sample sequence.

State of the art algorithm for comparison (BAI)

- Uses a basic DCE pruning.
- Focused in efficiency and stability of the skeleton computation.
- In these tests we used the Matlab implementation of the algorithm provided by the authors.
- Our approach is denoted as **BERIS**, and the state of the art alternative is denoted as **BAI**.

1. Introduction
2. Motivation
3. Theoretical Background
4. Theoretical Results
5. Experimental Results
    1. *Real-time performance.*
    2. ***<u>Stability.</u>***
    3. *Feature vector for shape recognition.*
6. Conclusions

5.2. Stability

- Two measures:
  - Static: Number of skeleton branches.
  - Dynamic: Difference in the number of skeleton branches between consecutive images in a gesture sample.
- Test variables:
  - DCE pruning values: 15, 20, 25.
- Result grouping:
  - Average for each of the three gestures.
  - Global average.

Static measure: Average branch number

	BERIS		BAI	
	Average	Std. dev.	Average	Std. dev.
Grab	17.67	3.14	24.55	3.04
Point	12.61	2.52	22.73	3.03
Turn Page	14.92	3.45	24.33	4.55
Global	**15.07**	**3.03**	**23.87**	**3.54**

Table 5.2.1: Average branch number per gesture sample, for DCE pruning value 25 (table 5.3 in the Thesis Report).

Dynamic measure: Average branch difference between consecutive images

	BERIS		BAI	
	Average	Std. dev.	Average	Std. dev.
Grab	2.10	1.72	2.79	2.16
Point	1.75	1.47	3.24	2.50
Turn Page	2.58	2.20	3.89	3.06
Global	**2.15**	**1.80**	**3.30**	**2.57**

Table 5.2.2: Average branch number difference between consecutive images in each gesture sample, for DCE pruning value 25. (Table 5.6 in the Thesis Report).

Comments on stability

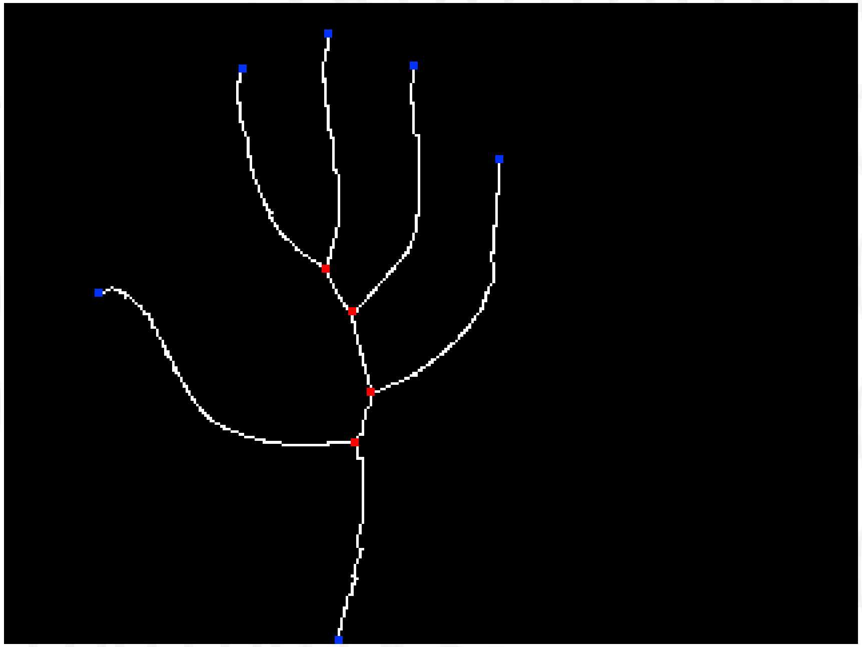
- BERIS produces skeletons with less number of branches than BAI.
  - Our approach produces more concise skeletons.
- Skeletons of similar shapes have less branch number variation for BERIS than BAI.
  - Our approach produces more stable skeletons.

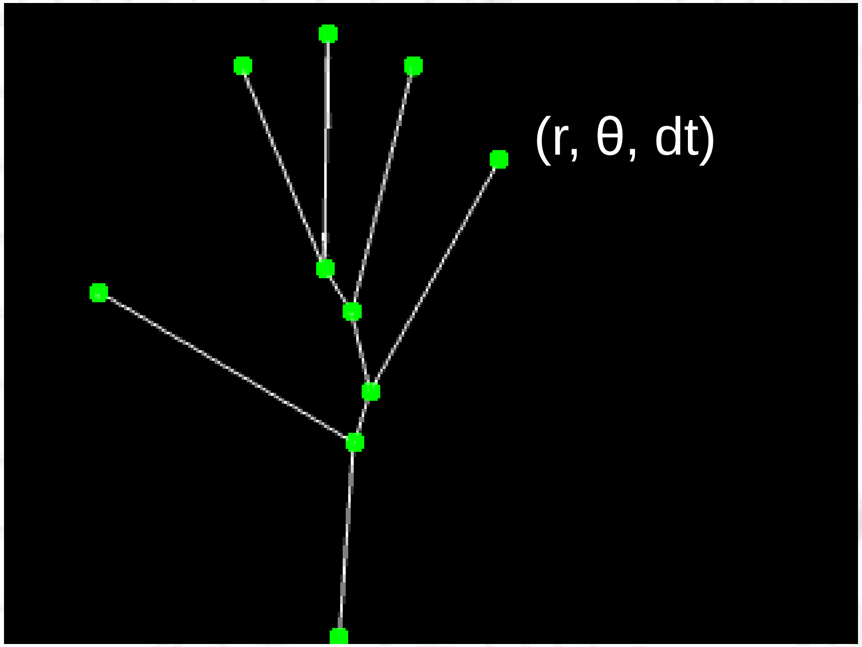
1. Introduction
2. Motivation
3. Theoretical Background
4. Theoretical Results
5. Experimental Results
    1. *Real-time performance.*
    2. *Stability.*
    3. ***Feature vector for shape recognition.***
6. Conclusions

5.3. Feature vector for shape recognition

- The skeleton is represented by an Skeleton Graph:
  - Graph nodes -> skeleton end points and joint points.
  - Graph links -> skeleton branches.
- The attributes of each node are its normalized position and normalized distance transform value.
- Shape recognition based in a greedy graph matching procedure, using only graph nodes.

Labeled Skeleton and Skeleton Graph





Labeled skeleton:

- Blue dots: End points.
- Red dots: Joint points.
- White pixels: branch points.

Skeleton graph:

- Green dots: Graph nodes (end and joint points).
- White segments: Graph links.

5.3. Feature vector for shape recognition

- Two kind of classifiers:
  - K-NN
    - 1-NN, 3-NN, 5-NN.
    - Whole database.
  - Probabilistic Neural Networks (PNN)
    - Two-fold crossvalidation on the 40% of the database.
    - Two values of sigma: 0.1 and 1.
- Three DCE pruning values: 15, 20, 25.

5.3. Feature vector for shape recognition

- Initially one class for each frame (15 per gesture, 45 in total).
- Then two class aggregations in order to improve effectiveness:
  - Consecutive frames of each gesture are grouped into 5 image packages, with three classes per gesture (9 classes in total)
  - One class for each gesture (3 gestures in total).

K-NN Results

- Our best results were obtained using a DCE pruning value 15, alpha value 0.75 and 5-NN.

Class number	BERIS	BAI
45	35.42 %	30.17 %
9	73.38 %	67.85 %
3	94.57 %	93.25 %

Table 5.3.1: Shape recognition results using a Greedy Graph matching algorithm (Extracted from tables 5.10, 5.11 and 5.12 of the Thesis Report).

K-NN Results

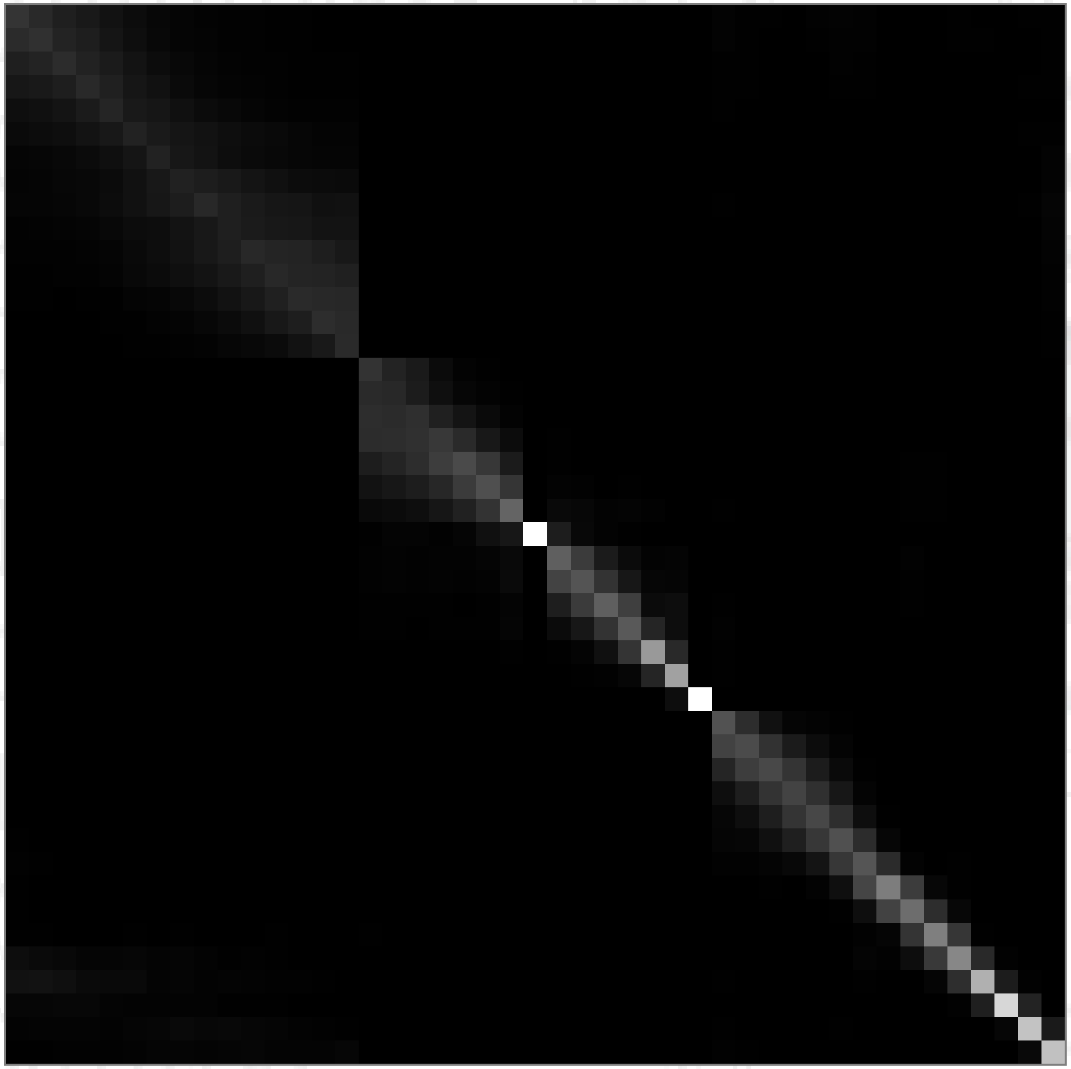


Fig. 5.3.1: Confusion Matrix for 45 Classes.

K-NN Results

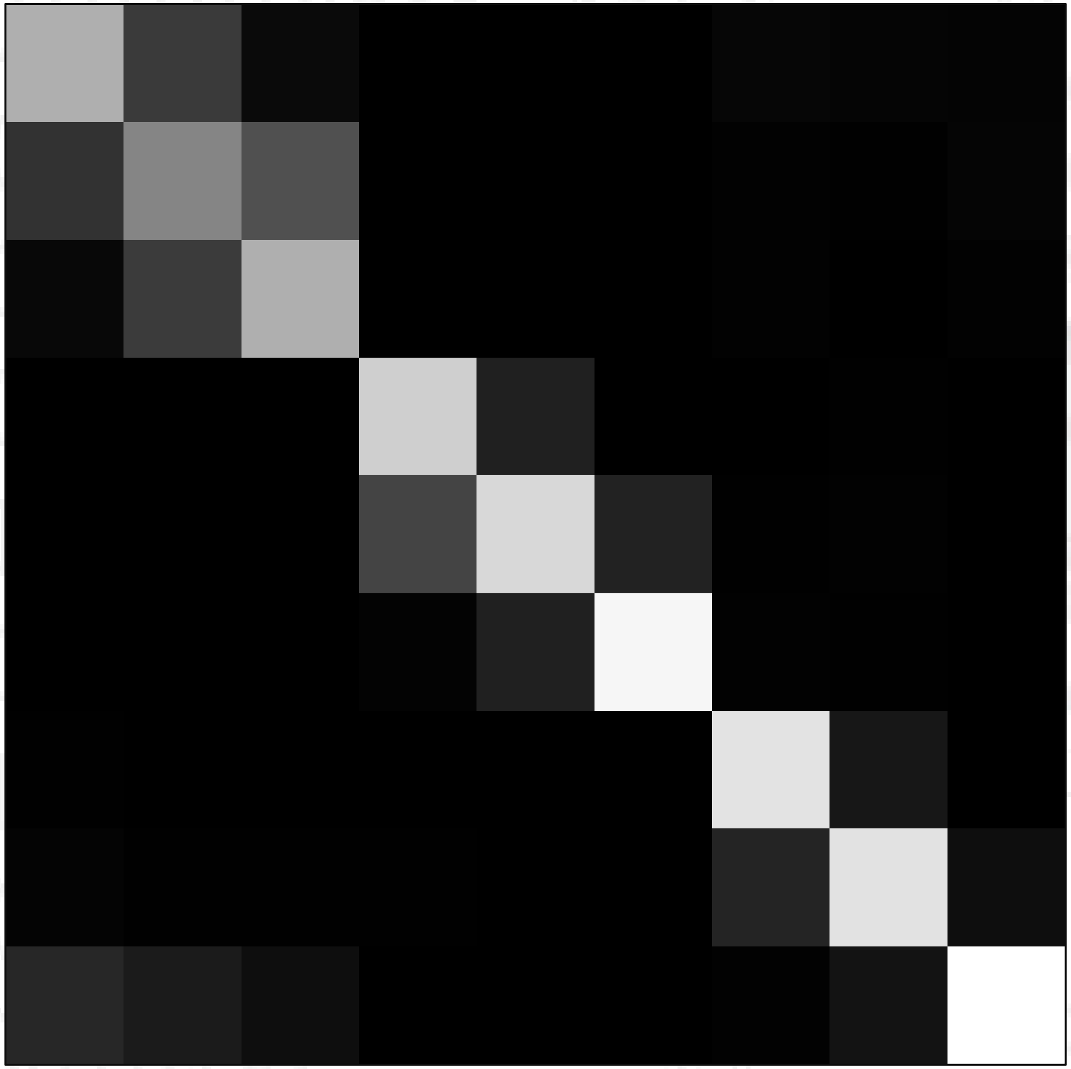


Fig 5.3.2: Confusion Matrix for 9 Classes.

PNN-Results

- Our best results were obtained using a DCE pruning value of 15, alpha value 0.75 and sigma value 0.1.

Class number	BERIS	BAI
45	30.72%	27.44%
9	72.00%	58.06%
3	87.94%	80.11%

Table 5.3.2: Shape recognition results using a Greedy Graph matching algorithm (Extracted from tables 5.23, 5.24 and 5.25 of the Thesis Report).

Some comments on the classification results

- Shape classification using the skeletons produced by our approach (BERIS) as feature vector outperforms using BAI.
- Skeleton graph node position is more important for shape recognition than the DT value (alpha value 0.75).
- As the number of training examples grows, the results improve, but the computational cost is also higher (5-NN).

Some comments on the classification results

- A strong DCE pruning improves over a gentle DCE pruning.
- Dynamic gesture recognition would improve the recognition rate, because of the high similarity between several poses of different gestures (i.e., grab and turn page).

1. Introduction
2. Motivation
3. Theoretical Background
4. Theoretical Results
5. Experimental Results
6. **Conclusions**

6. Conclusions

- Natural interaction and Tabletops is the context of this PhD Thesis.
- Hand gesture recognition in tabletops is the explored topic.
- Efficient and stable skeleton computation algorithm is the main contribution of the present work.

6. Conclusions

- Our skeleton computation and pruning algorithm is efficient and stable under noise in the shape boundary.
- Our approach improves the current state of the art.
- Our approach permits the implementation of hand gesture recognition in real-time for tabletops.

6. Conclusions

- Further research topics include:
  - Exploring different pattern recognition approaches using skeletons.
  - Combining hand gestures with multi-touch interaction.
  - The development of an abstraction layer to manage multiple user interaction in tabletops.

Thank you for your attention

Computational Intelligence Group

INNOVAE VISION
Comunicación Interactiva

Innovae Vision

University of The Basque Country

informatika fakultatea
facultad de informática

Computer Science Faculty

- PhD Candidate: **Andoni Beristain Iraola**
- PhD Advisor: **Dr. Manuel Graña Romay**