Rapid In-Line Detection of Macro Defects in Semiconductor Manufacturing

Dick de Roover, Gwen W. van der Linden, Sarbajit Ghosal,
Robert L. Kosut, Jon L. Ebert, Abbas Emami-Naeini

SC Solutions, Inc.
1261 Oakmead Pkwy
Sunnyvale, CA 94085
Email: roover@scsolutions.com

APC Conference XXVI 2014
Ann Arbor, MI
10/01/2014

SC SOLUTIONS

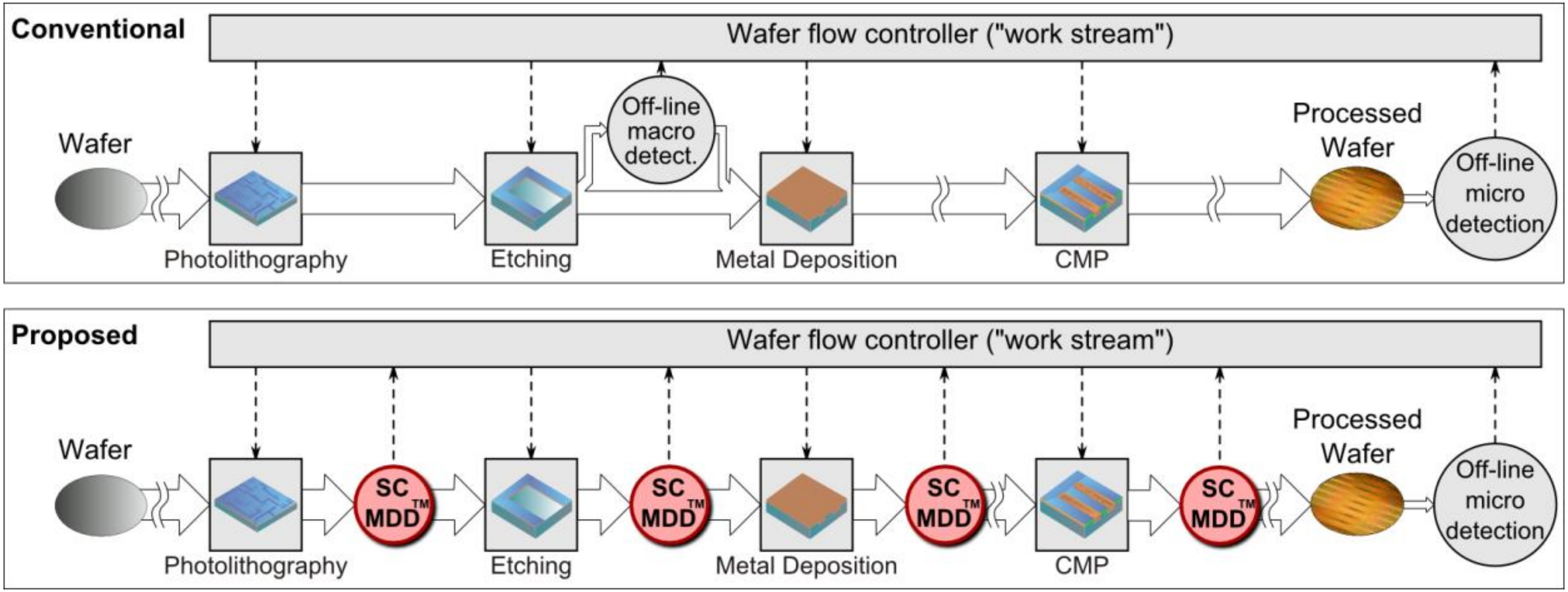
Content

- **Background & Motivation**
- **Macro Defect Detection Problem**
- **The Idea**
- **Development of the Hardware/Software Solution:**
  - ***Concept feasibility***
    - *Use of a commercial scanner & results*
    - *Development of required algorithms*
    - *Sample die estimation & defect detection results on process wafers*
  - ***Custom hardware design & performance***
    - *Custom hardware design*
    - *Sample results on process wafers*
  - ***Defect clustering***
    - *Clustering methods*
    - *K-means algorithm*
    - *Sample results*
- **Summary & Conclusions**

Background & Motivation

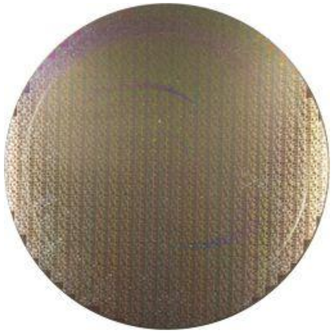
- The manufacturing of Integrated Circuits (ICs) on semiconductor wafers involves hundreds of complex and expensive process steps
- Defects can occur during any of several steps such as etch, resist removal, and can be caused by particle contamination, incomplete process, process variations, dislocations, scratches, cracks, etc.
- Defects range in size from submicron to visually-detectable "macro" defects that may be as large as several inches long and span multiple dies
- Defects reduce yield
- With increase in number of steps in semiconductor manufacturing, defect detection becomes more critical, however, current Defect Detection tools are slow and expensive, and not every wafer is inspected
- There is a critical need to **quickly** and **inexpensively** inspect each and every wafer without affecting throughput

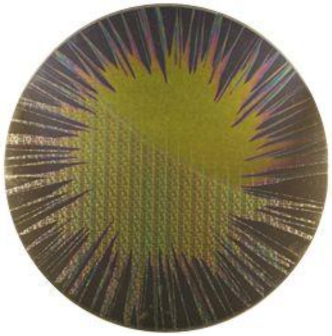
Conventional way: off-line detection of only a sample of wafers, expensive equipment (> $M)



Needed Solution: in-line detection of every wafer with relatively inexpensive equipment (< $50k)

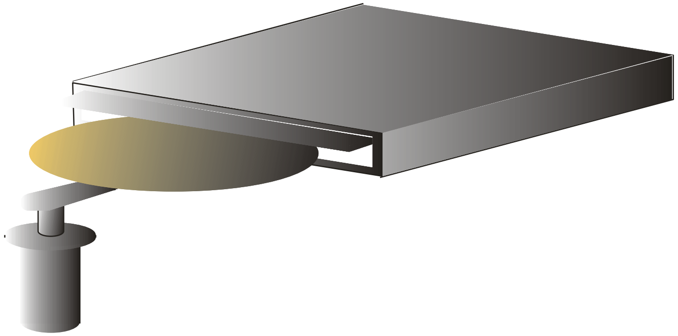
The Idea



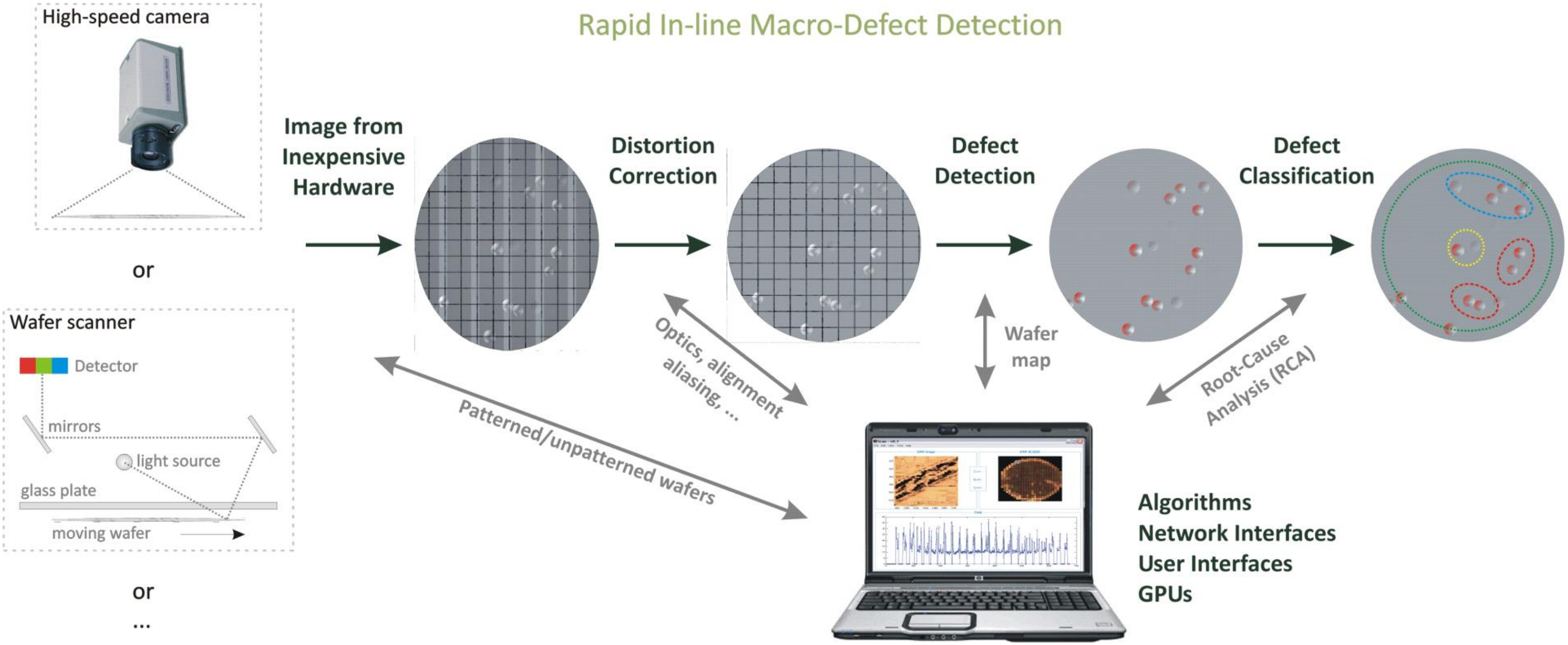


Detect large fraction of macro defects right at the process step by in-situ auto-inspection of every wafer using:

1. **inexpensive scanning hardware**
2. **sophisticated algorithms**

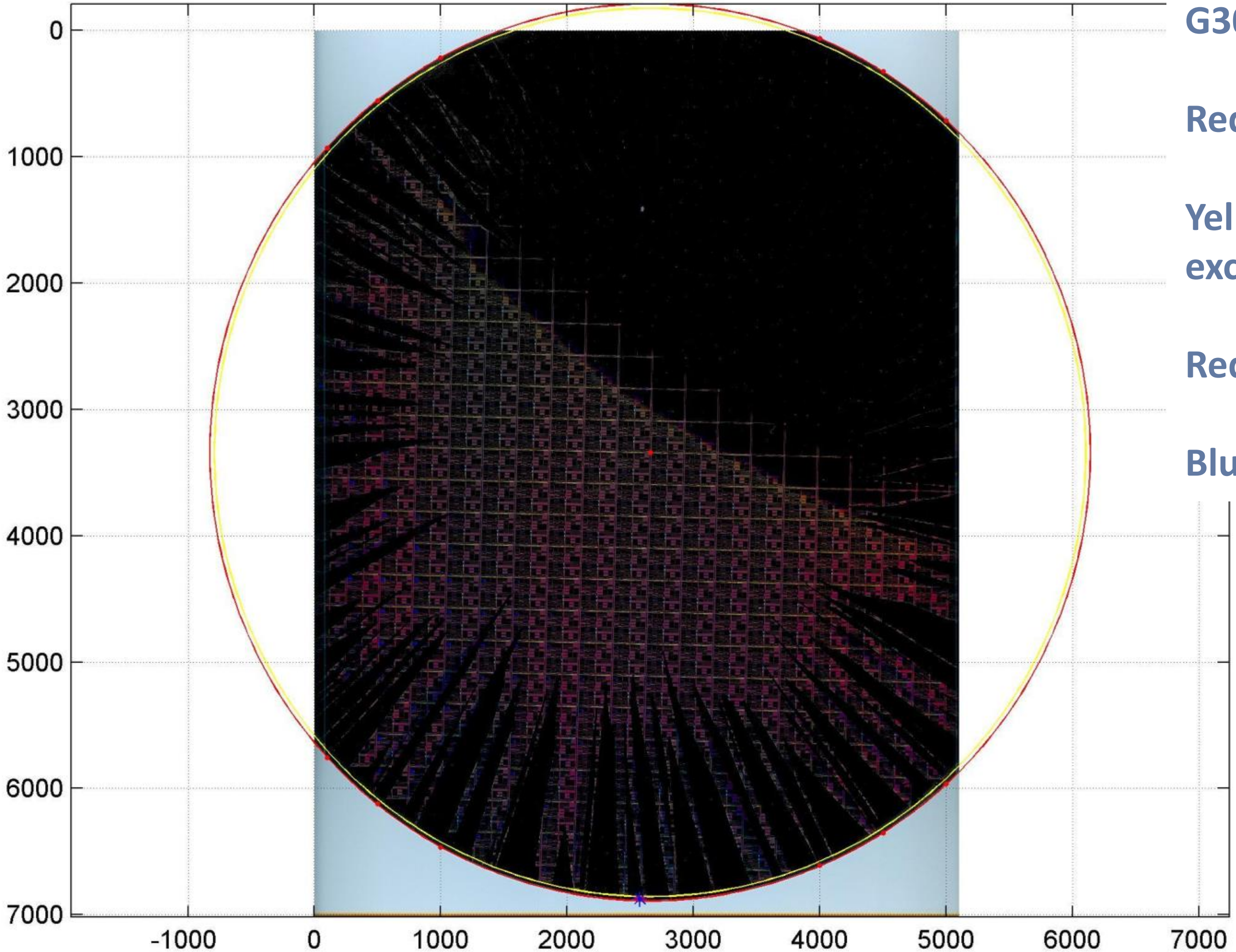


Details of the Idea: Defect Detection & Classification



Concept Feasibility: Use of Commercial Scanners

Our First Scan of a Wafer Using a Commercial Scanner



Scanned using HP Scanjet G3010

Red circle is wafer edge.

Yellow “circle” is 3mm exclusion

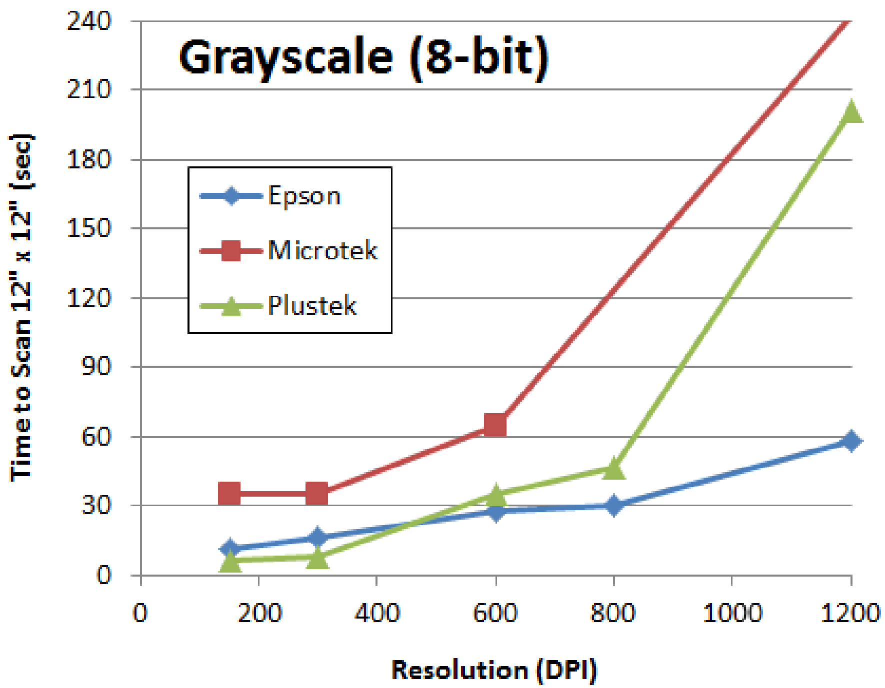
Red dot is wafer center

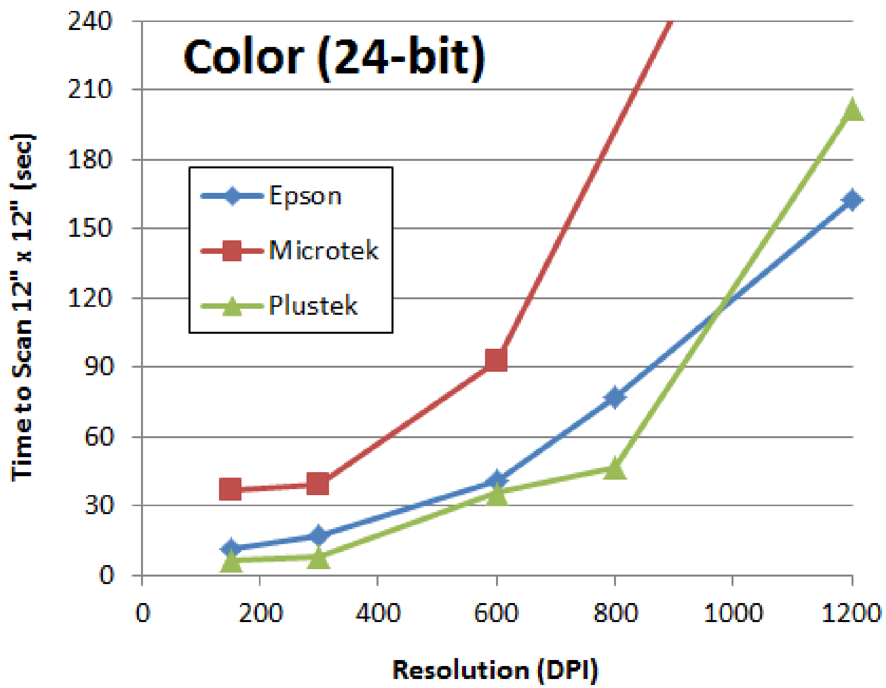
Blue dot is notch

Commercial Scanners: Resolution vs. Scanning Time

- Resolution determines maximum detectable defect size
- Scanning time increases rapidly with increasing resolution

Resolution (DPI)	Pixel Size (µm)
300	84
600	42
800	32
1200	21



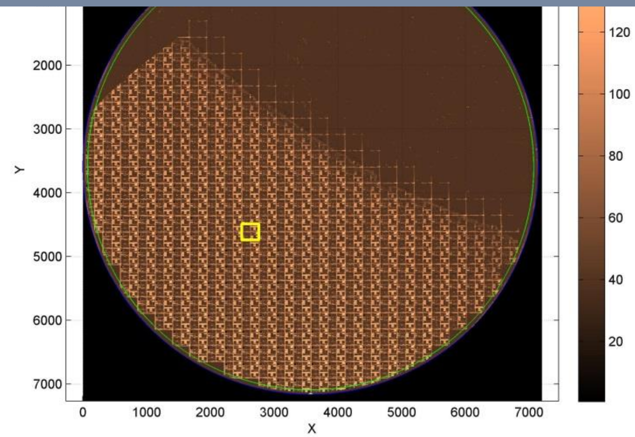


Development of Required Algorithms

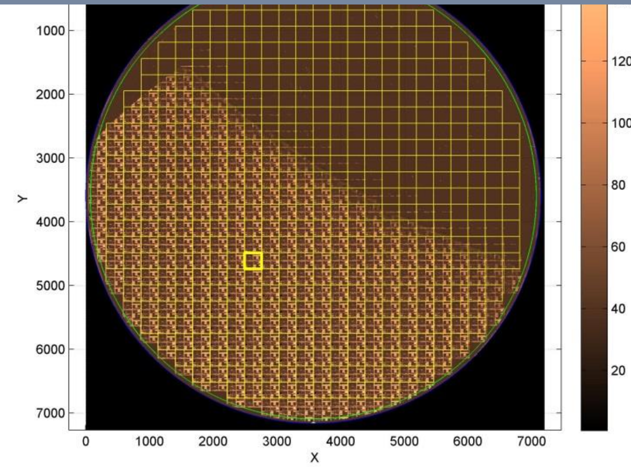
- **No *a priori* information regarding the wafer, die-size, edge, etc. is required**
- **Algorithms to detect wafer orientation**
  - ***Detect edge of wafer***
  - ***Find center of wafer***
  - ***Find the notch***
  - ***Rotate wafer to align image with center-notch***
- **Algorithms to detect die size & locations**
- **Algorithms to detect defects**
  - ***Estimate reference die (no defects)***
  - ***Use statistics to compare dies to reference die (zero false-positives)***
- **Algorithms to classify defects**
  - ***Identify clusters***
  - ***Pattern recognition***
  - ***Root-cause analysis***

Wafer Orientation & Die Detection

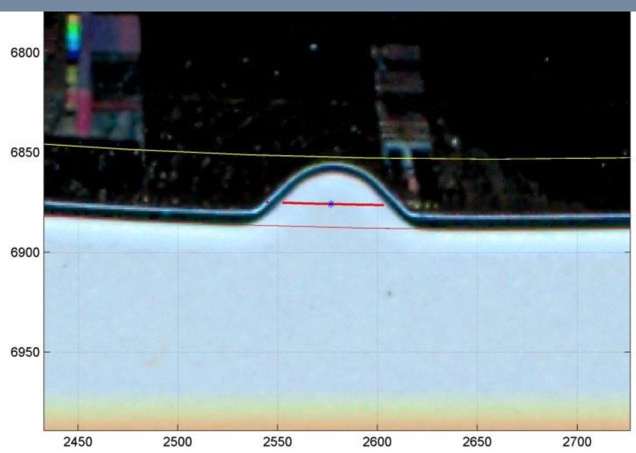
Scanned wafer – scaled, rotated, aligned to notch



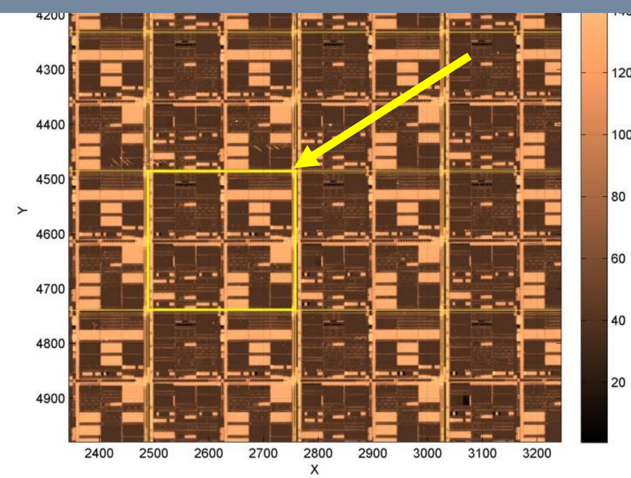
Die Location detection



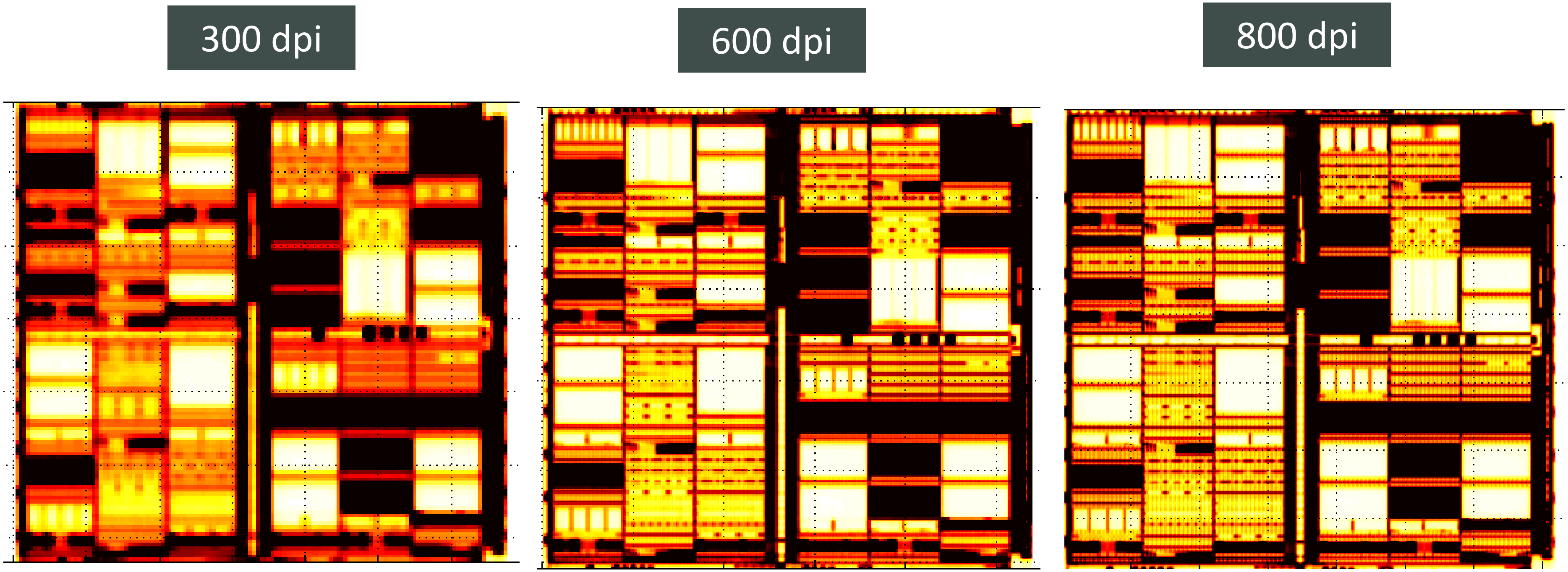
Notch detection



Die Size detection



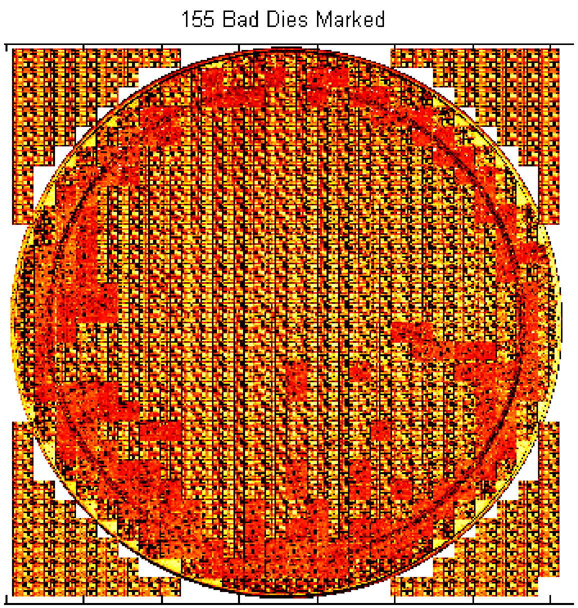
Reference Die Estimation



- ❑ Resolution impacts quality of 'reference die'
- ❑ Algorithms needed to compensate for image distortion such as 'aliasing', line curvature, etc.
- ❑ Algorithms are time consuming

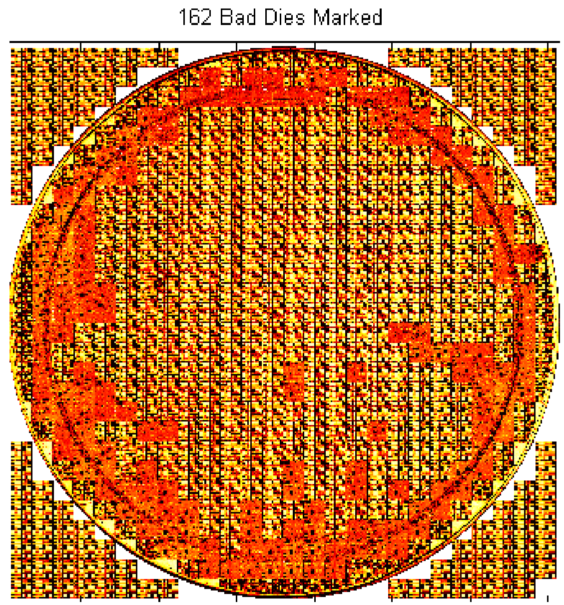
Sample Results: Die Estimation & Defect Detection

300 dpi



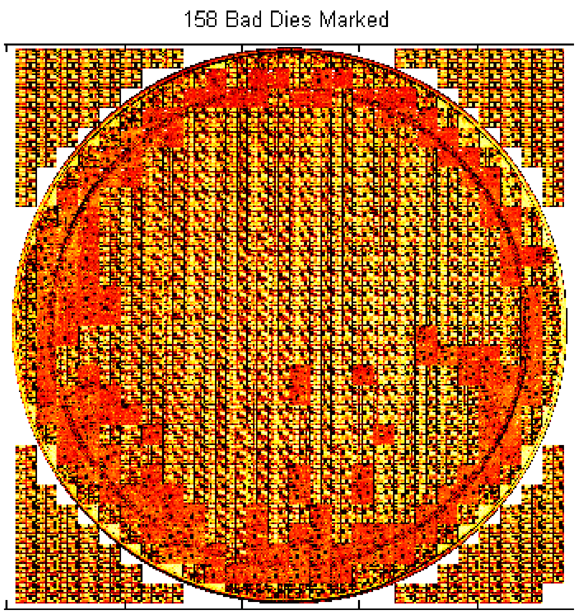
155 bad dies

600 dpi



162 bad dies

800 dpi



158 bad dies

- ❑ **Aim for 0 'false positives'**
- ❑ **Use reference wafer (no defects) to determine thresholds**

Custom Hardware Design & Performance

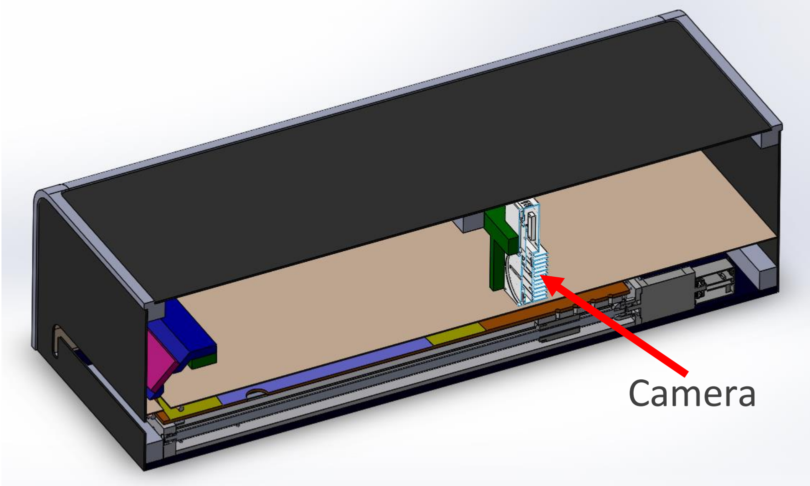
Improved Performance with Custom Hardware Design

- Having demonstrated concept feasibility with commercially available scanners, we can speed up the process and achieve better performance using custom hardware/software design
- To not affect throughput, scanning needs to occur in the order of seconds, while wafer is moving in or out of a tool, subsequent processing < 30sec.

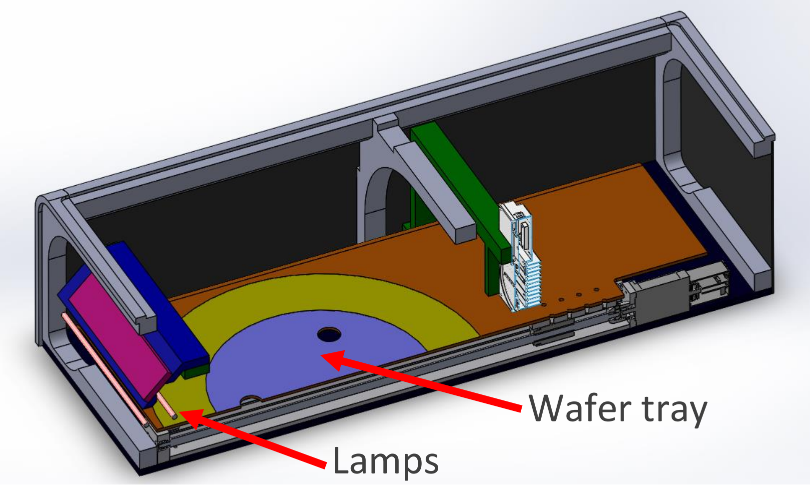
DEVELOPMENT TARGET	
Max. Wafer Diameter	300mm
Max. Carrier Size	450mm
Max. Optical Resolution	1365dpi
Min. Image Pixel	18.6µm
Min. Pixel Depth	8bit X 1
Max. Scan Time	4sec
Max. Process Time	25sec
False Positive Defect	0

- Target: hardware that detects defects as small as 10µm at scanning speed in the order of seconds, but still at a favorable price compared to off-line inspection equipment (<$50k)

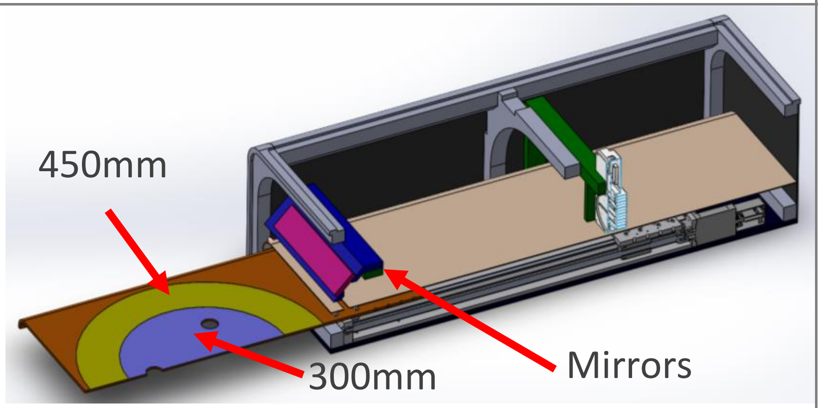
Custom Hardware Design



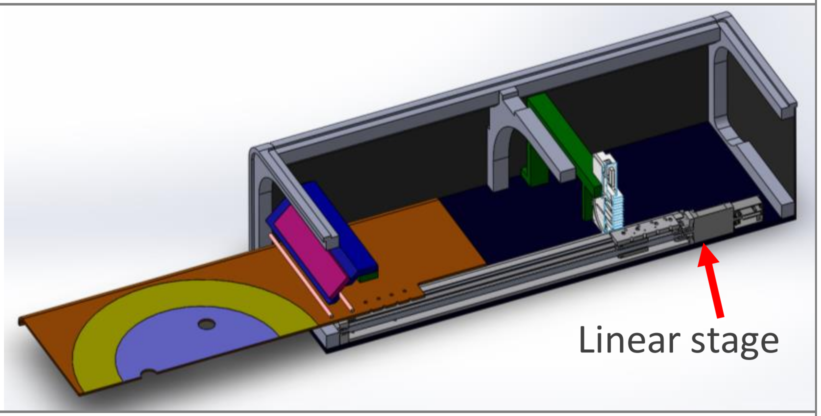
Scanner cross-section view, with top and front cover. Wafer tray fully retracted.



Same view, but with top and front covers removed, showing the wafer tray.



Scanner cross-section view with covers removed and wafer tray fully extended.



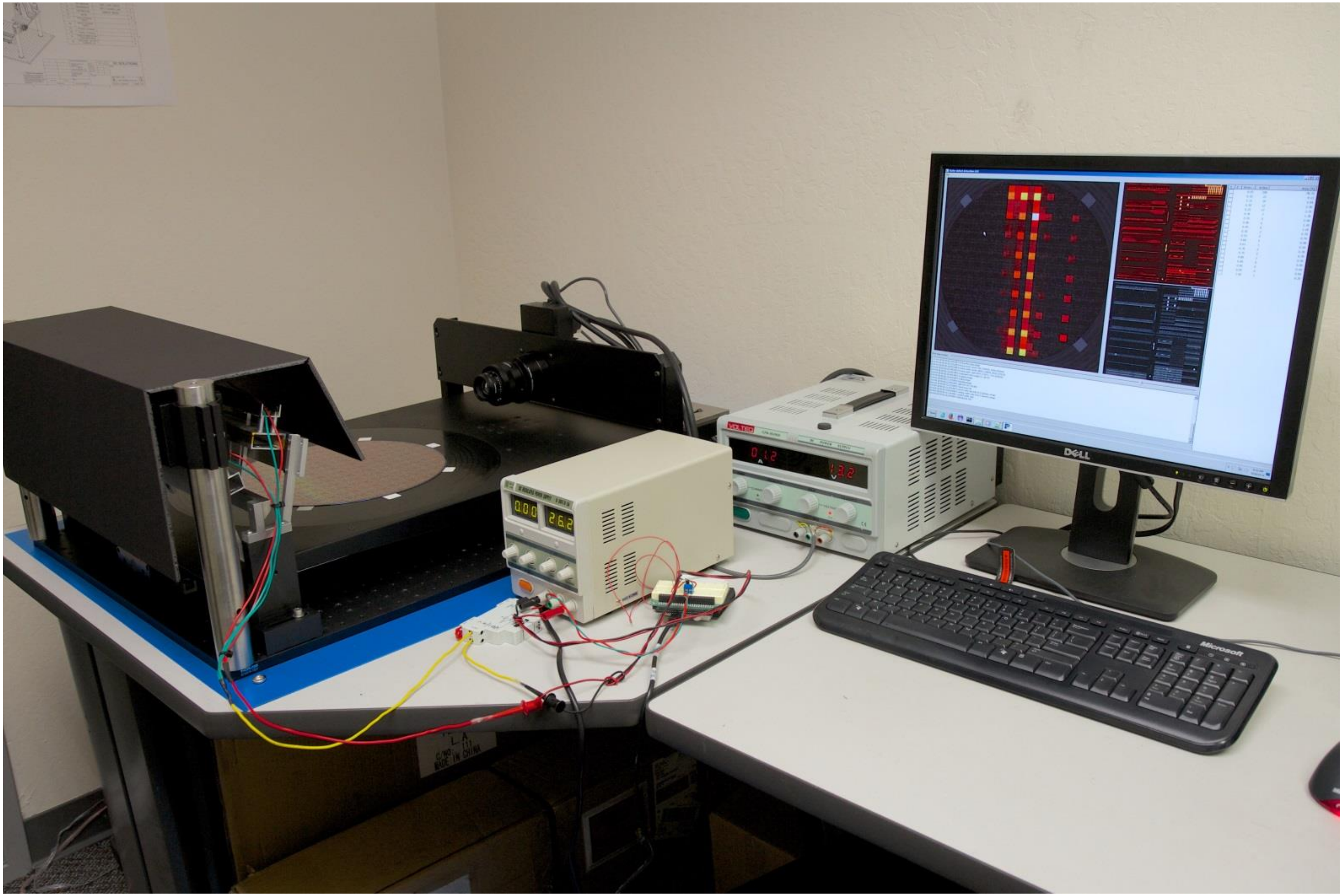
Same view, with inner cover removed, showing the linear stage.

System Components and Performance for 300mm wafer

- **System components:**
  - ***High-resolution line camera (16000 pixels per line ~ pixel size of 18.7µm for 300mm wafer)***
  - ***Light sources, Lenses, Mirrors***
  - ***High-speed linear motor***
  - ***Software***

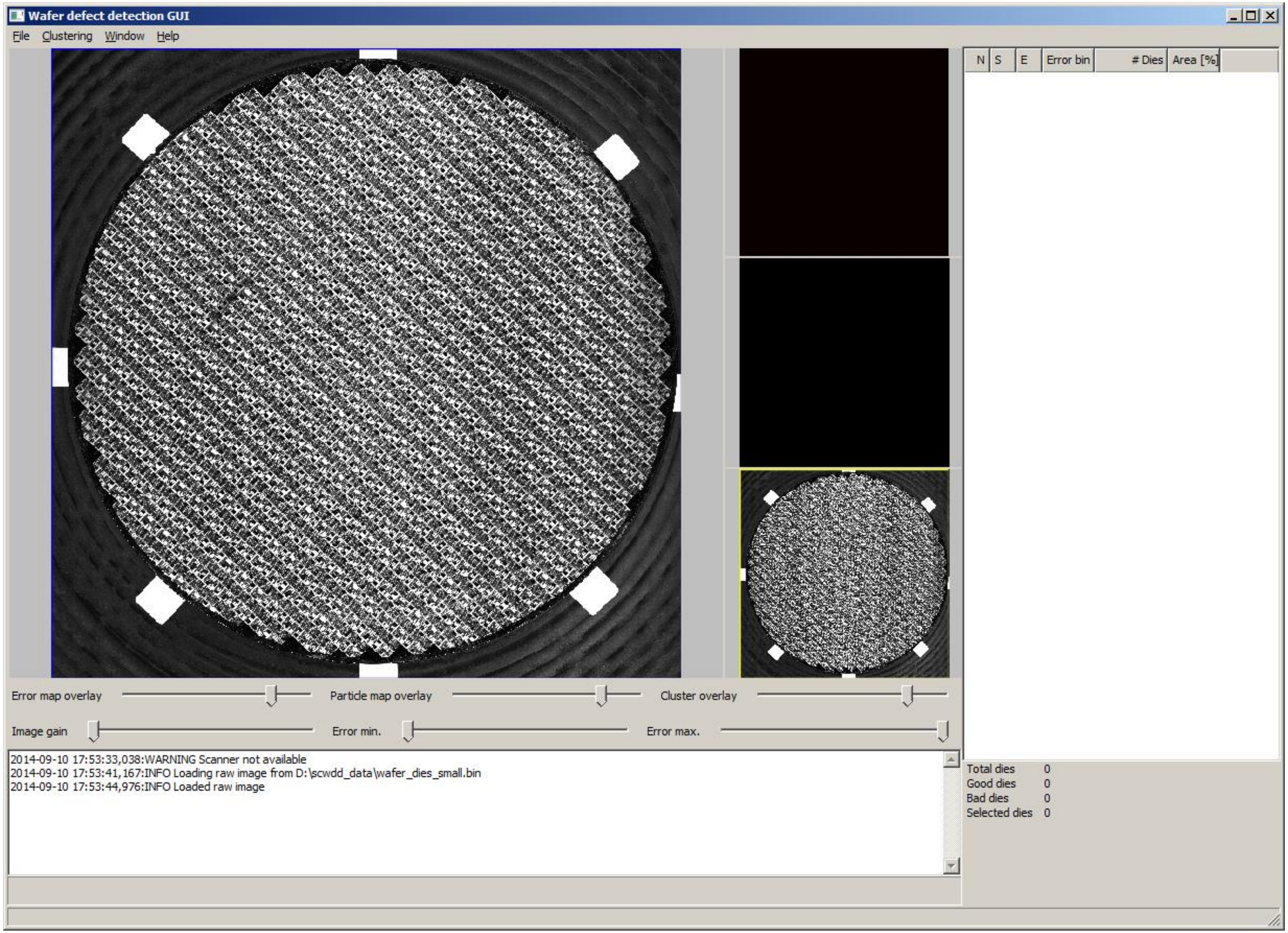
- **Achieved performance with custom hardware:**
  - ***A 300mm wafer was scanned at 1280dpi in 3.5 seconds***
  - ***Subsequent processing for defect detection in 8.5 seconds***
  - ***Defects as small as 5.6µm were detected***

Hardware Setup

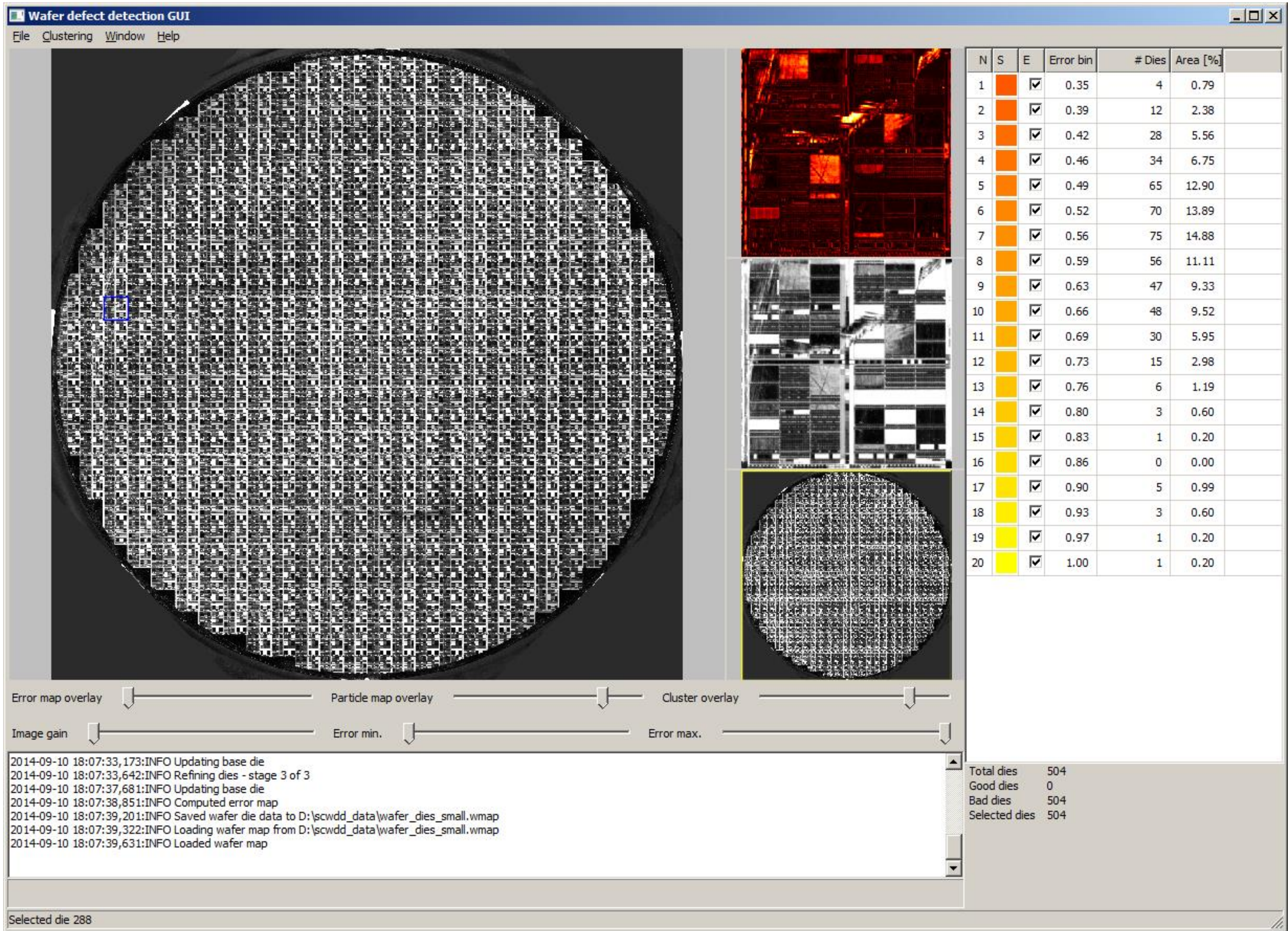


Hardware in Action

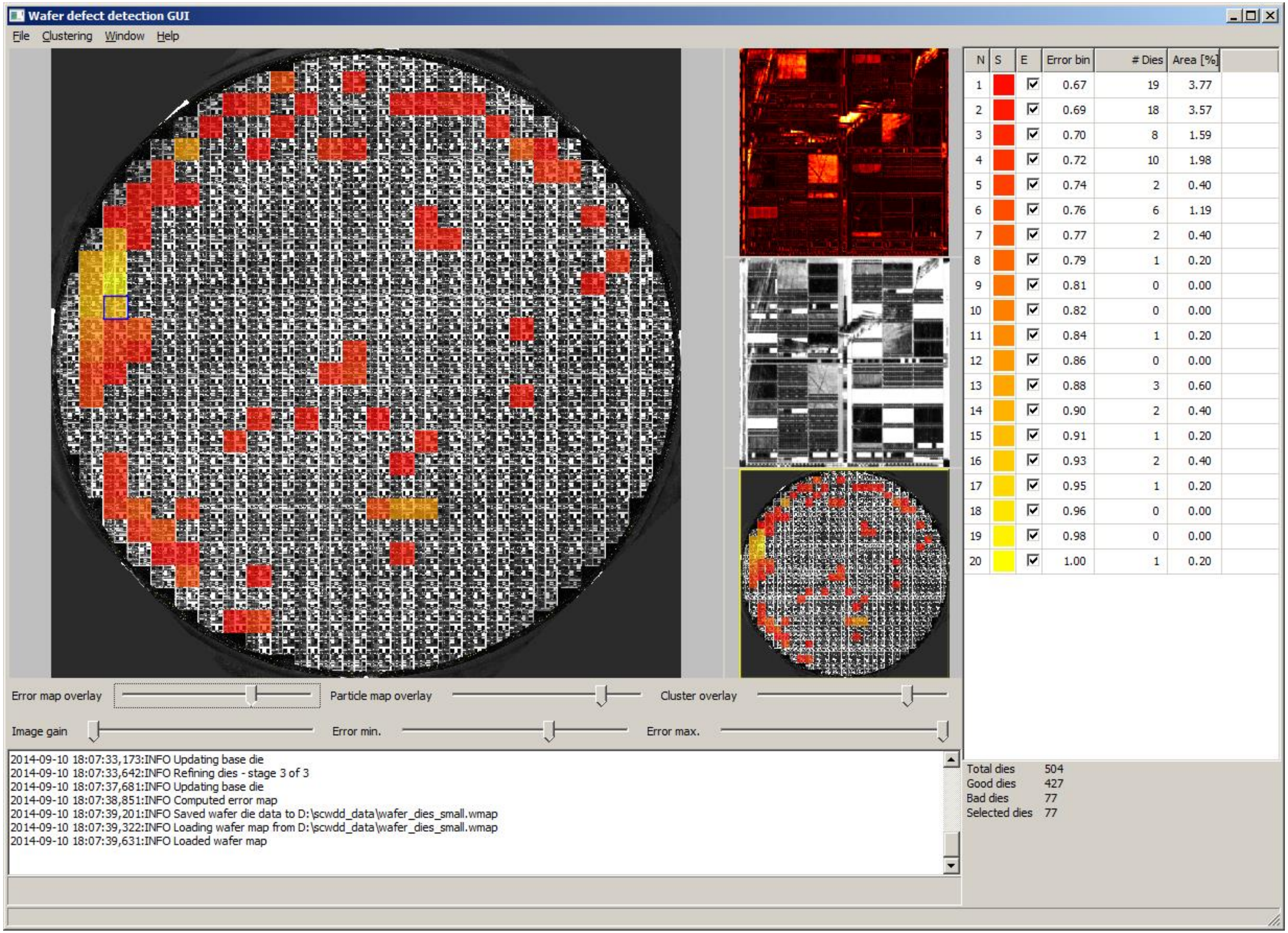
Sample Result: 300 mm process wafer – raw image



Sample Result: edge, notch & dies detected



Sample Result: die error map



SC SOLUTIONS

Defect Clustering

Clustering Methods

- Two classes of clustering techniques:

  1. *Hierarchical or agglomerative:* These algorithms start with each point in its own cluster. Clusters are combined based on their proximity to each other, using one of many possible definitions of closeness.
  2. *Point assignment: Points are considered in some specified order, and each point is assigned to the cluster into which it best fits.*

- We have used the k-means algorithm, a widely used point assignment method for classifying

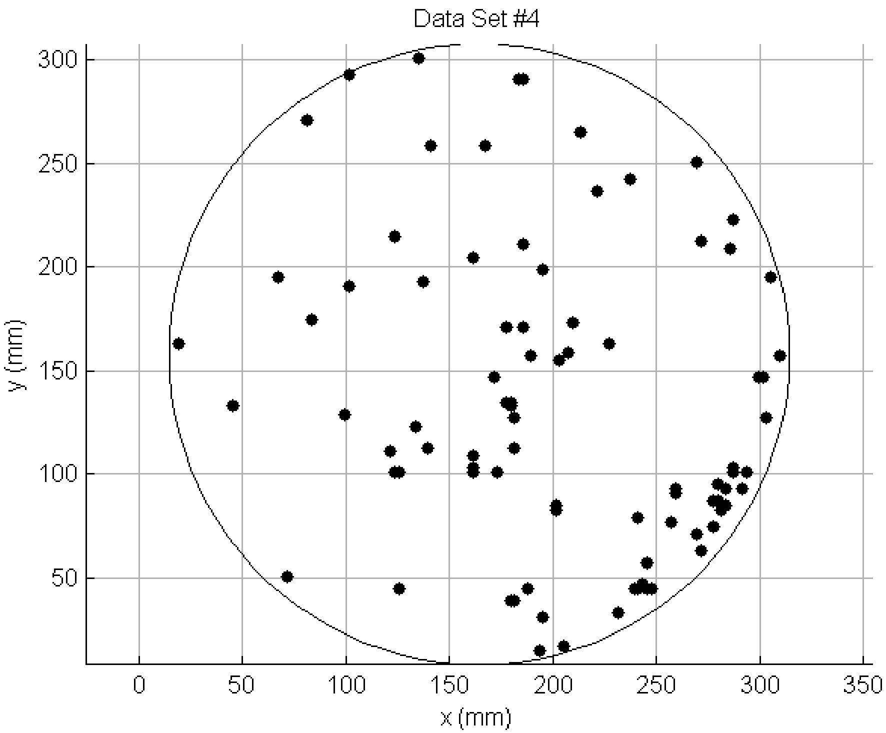
K-means algorithm

Pseudo-code* of k-mean clustering shown below:

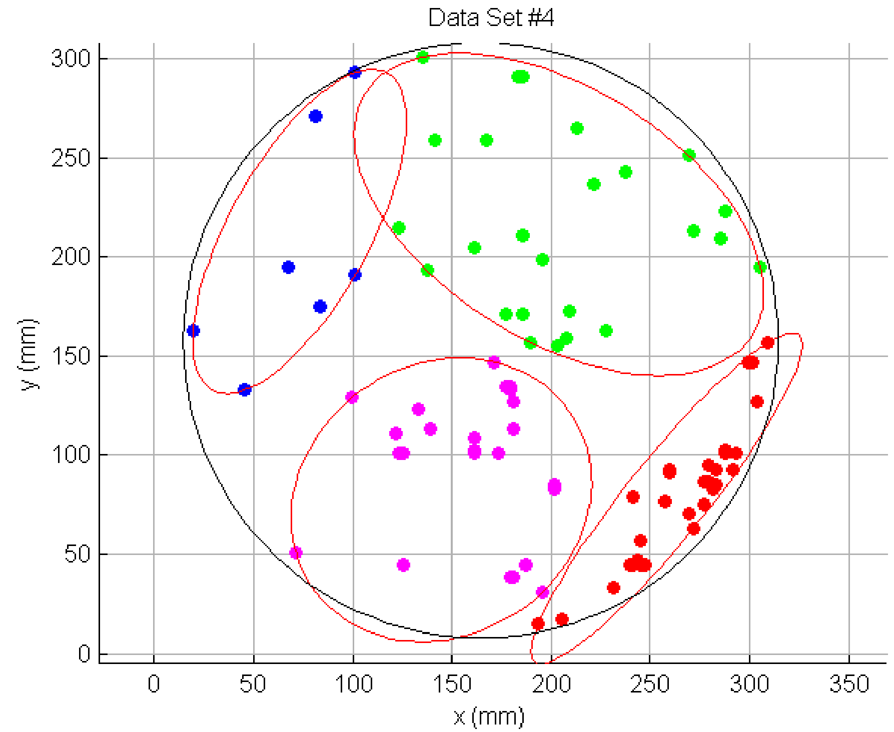
```
Initially choose k points that are likely to be in
    different clusters;
Make these points the centroids of their clusters;
FOR each remaining point p DO
    find the centroid to which p is closest;
    Add p to the cluster of that centroid;
    Adjust the centroid of that cluster to account for p;
END;
```

- **Iterative loop was added to above algorithm:**
  - **In each iterative step, starting cluster centers were assigned value of the final center of previous iteration until convergence was reached (i.e., no change in cluster composition).**

* J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of Massive Datasets*, 2014.

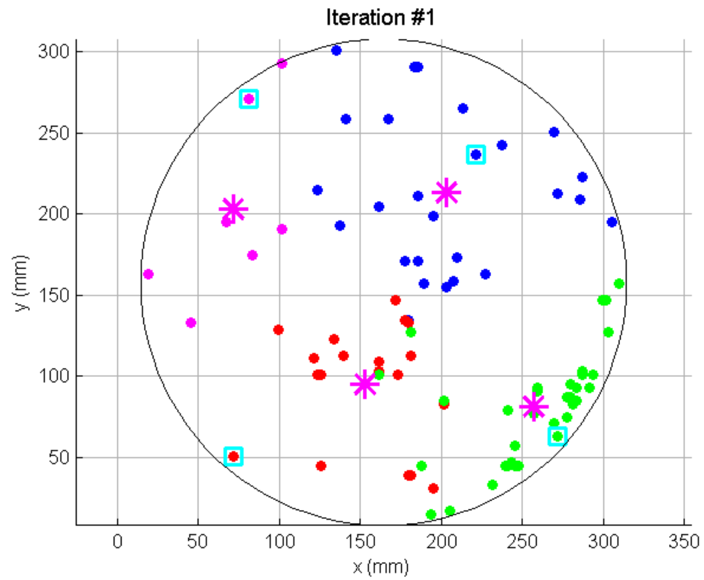


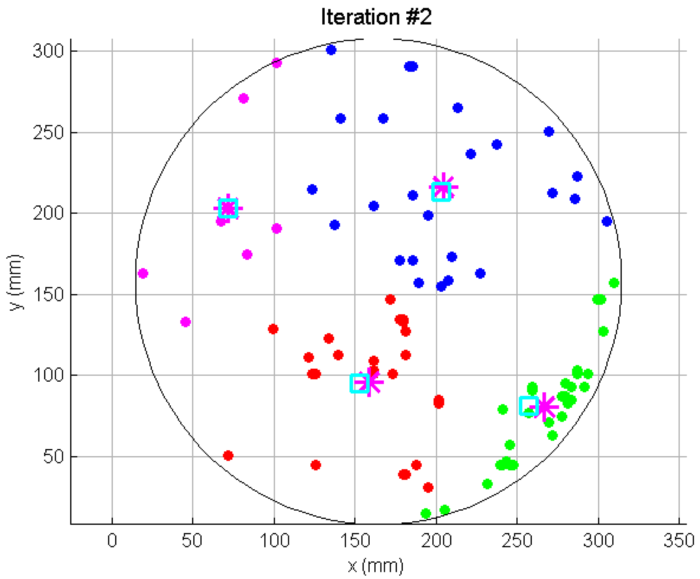
Sample defect data set

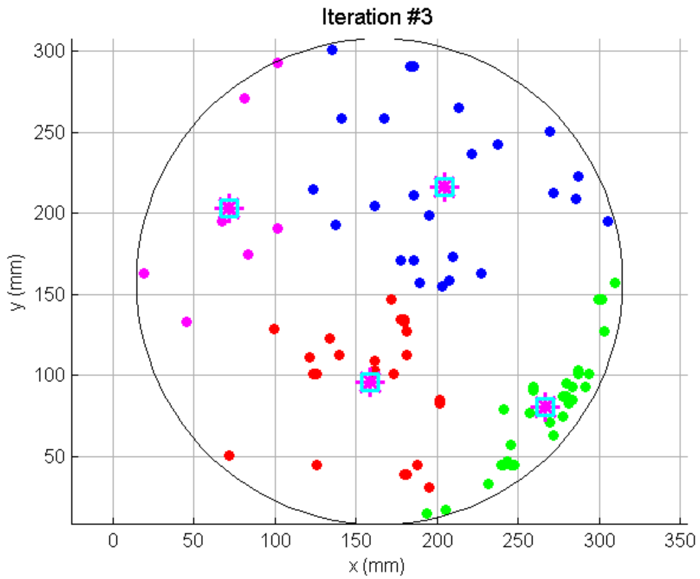


Data in four clusters with best-fit ellipses determined for each cluster for quantitative characterization

- **Iterative determination of clusters using the k-means algorithm:**
  - ***Defect within blue square is starting cluster center (centroid)***
  - ***Defect shown in purple star is the new cluster center***
  - ***Clustering converges in three iterations***





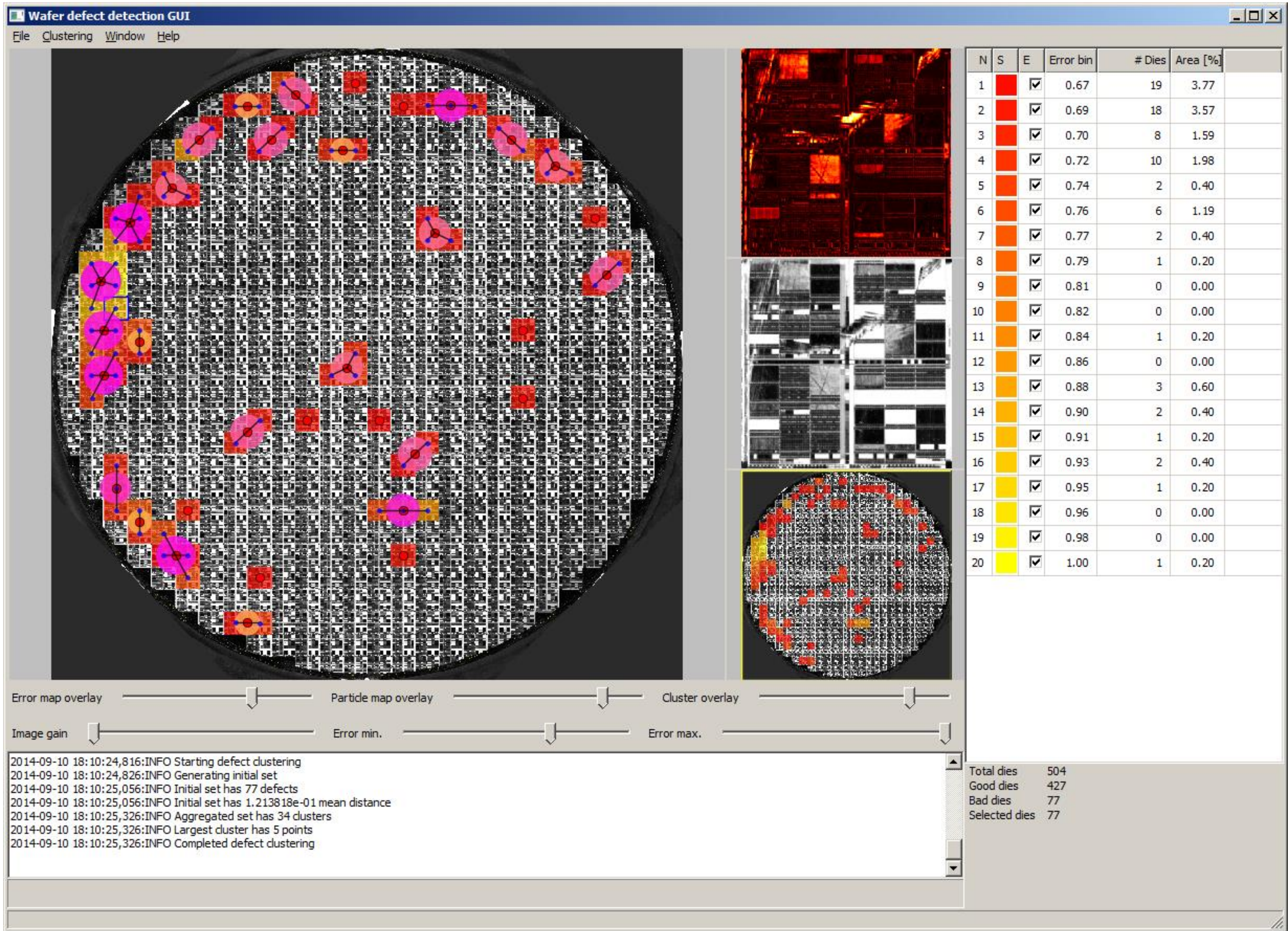


Summary of defect clustering

- Advantages of using the k-means algorithm for defect clustering:
  1. Simple implementation
  2. Fast for low-dimensional data like defect data

- Disadvantages of the k-means algorithm for defect analysis:
  1. Since k-Means is restricted to data which has some sort of a center (centroid), it cannot handle data of spatially-varying densities such as annuli
  2. Total number of clusters must be specified
  3. Does not identify outliers

Implementation of Defect Cluster Map



Summary & Conclusions

- Evaluated different hardware solutions to build an inexpensive industrial-grade scanner that detects defects in the order of 10 to 20 µm
- High-resolution images can be scanned in a few seconds; subsequent processing is in the order of 10's of seconds
- High-resolution imaging (1365dpi) allows for defect detection in the 10 to 20 µm range
- Effective defect clustering using k-means algorithm
- Addresses defect detection and clustering for both patterned and un-patterned wafers

Is it possible to quickly and inexpensively detect macro defects at *every step on every wafer* without affecting throughput?

We think yes!

The next question is: is the industry ready for it?