

Bacteria “Colonalyzer”

Design Review

MIT 6.111 Final Project



Yaw Anku

MIT Department of Electrical
Engineering and Computer
Science

November 15, 2006

Outline

- Overview : problem & objective
- Design Architecture
- Discussion of major modules
- Timeline

Problem

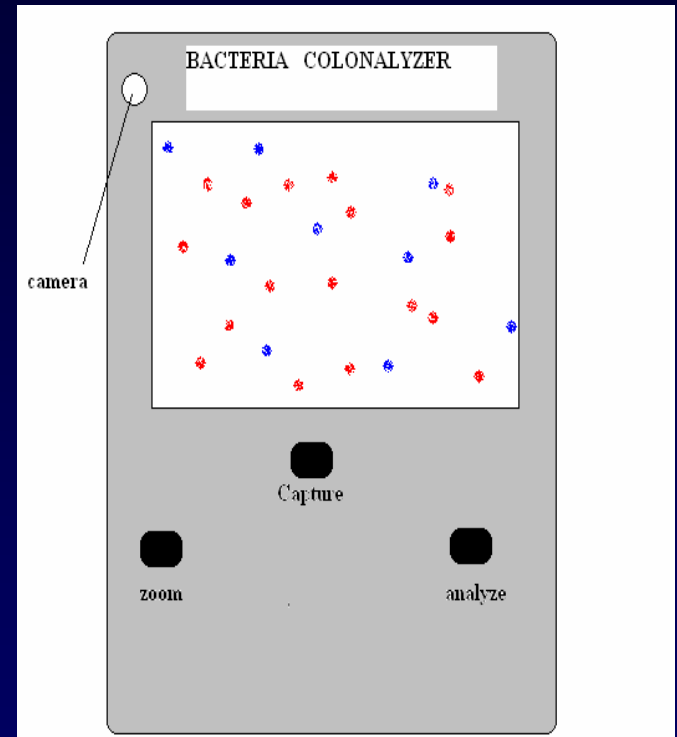
- Water testing in remote location
 - Create bacteria culture on filter membrane from water samples
 - Count bacteria colonies after specified period ~ 24hrs
 - E.coli = blue count
 - Coliforms (other bacteria) = red count.
 - Example analysis for drinking water
 - No bacteria → truly safe to drink
 - E. coli < 10 → okay
 - E. coli < 50 → acceptable, but not good



Current process involves manual counting – slow and prone to error

Objective

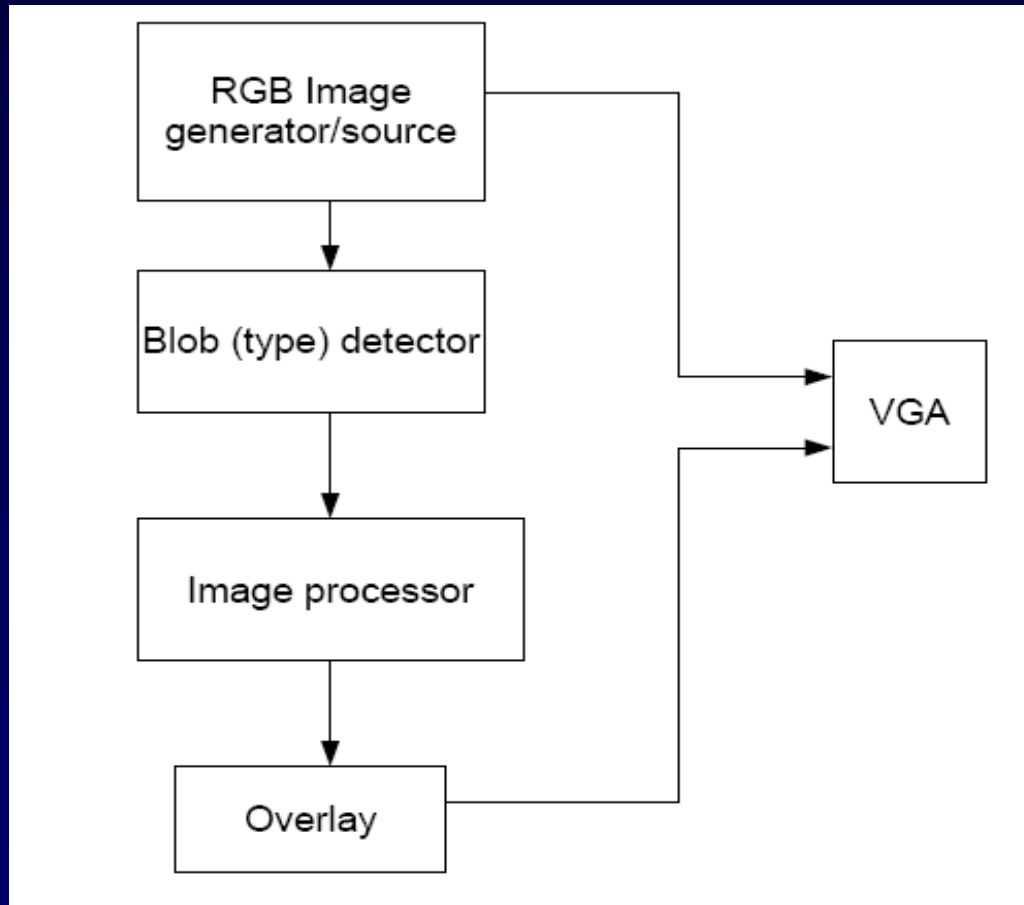
- Automate counting process by digital analysis of scanned culture membrane image
 - Increase throughput and hence time to providing water purification advice/solution
 - Provide more accurate bacteria concentration results



Handheld scanner and analyzer tool

Design Architecture

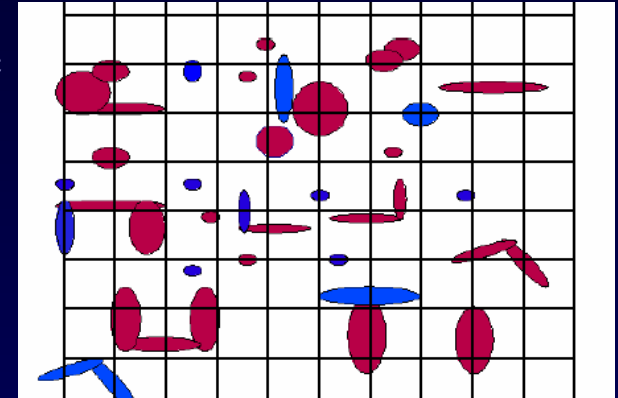
- High Level System Block Diagram



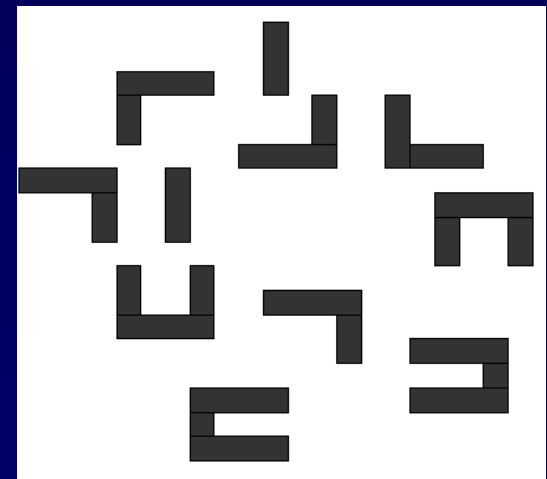
Discussion of major modules

- **RGB image generator/source**
 - Input image from scanner/camera
 - Dummy image for purposes of testing blob detection algorithm on different shapes and orientations

More realistic image



Dummy image



Discussion of major modules

- **Blob (type) detector**

- Implementation iteration one : generates black and white binary image from input image
- Implementation iteration two: distinguish between different image **types/colors** – **blue** vs. **red**
- Operation

- Sets threshold, T

- Binary \rightarrow

$$F_T[i, j] = \begin{cases} 1 & \text{if } F[i, j] \geq T \\ 0 & \text{otherwise.} \end{cases}$$

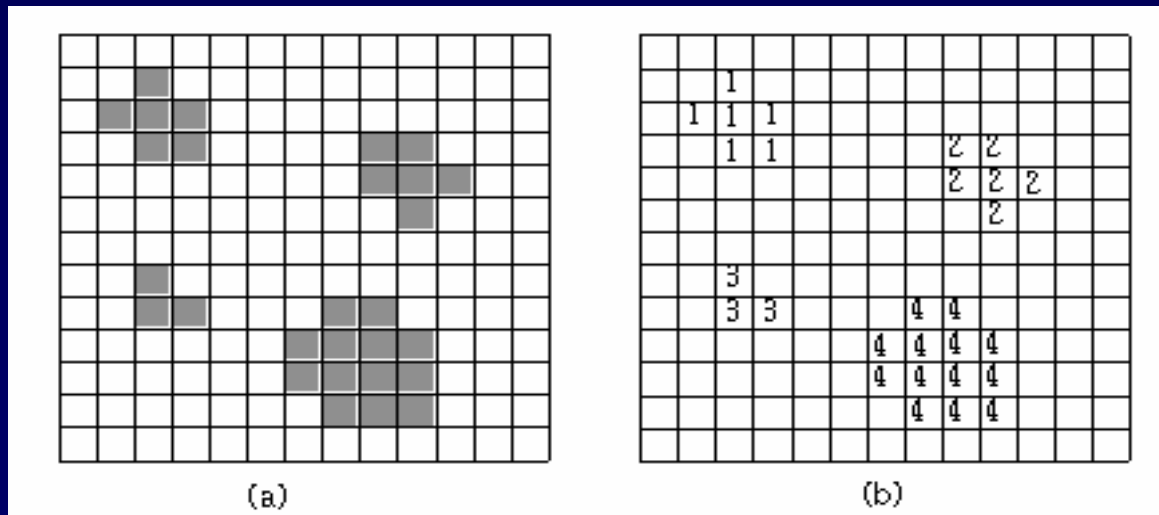
- Different colors \rightarrow

$$F_T[i, j] = \begin{cases} 1 & \text{if } T_1 \leq F[i, j] \leq T_2 \\ 0 & \text{otherwise.} \end{cases}$$

Discussion of major Modules

- **Image Processor**

- Utilizes a ‘connected component labelling algorithm to detect separate blobs – **all connected components assigned unique label (1, 2, 3...)**
- Count number of sets of labelled components

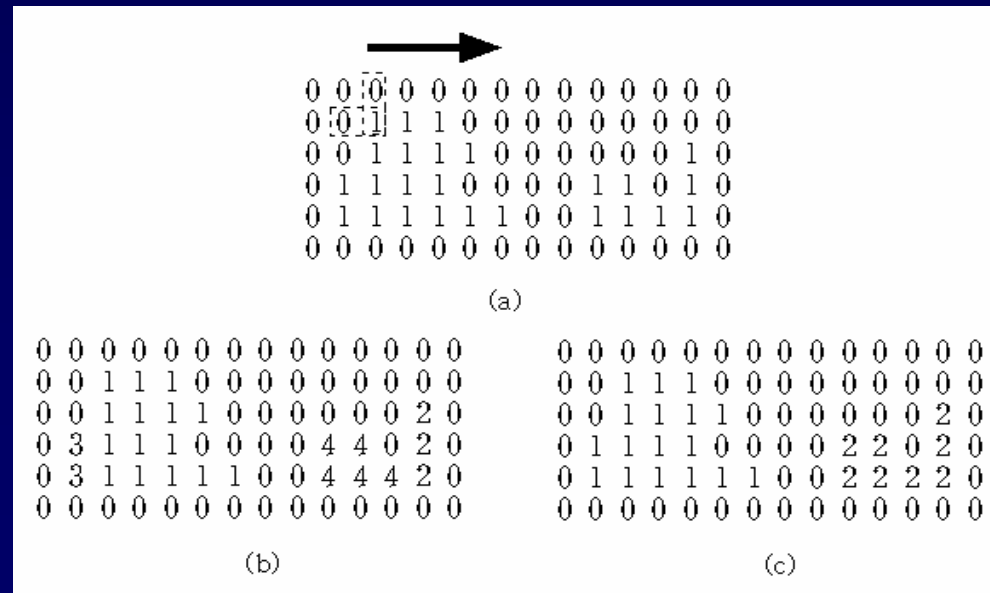


Sequential Component labeling algorithm

- 1. Scan the image from left to right and top to bottom.**
- 2. If the pixel is 1 , then**
 - (a) If only one of its upper or left neighbors has a label, then copy the label.
 - (b) If both have the same label, then copy the same label.
 - (c) If both have different labels, then copy the upper pixel's label and enter the labels in an equivalence table as equivalent labels.
 - (d) Otherwise assign a new label to this pixel and enter this label in the equivalence table.

Sequential Component labeling continued

- If there are more pixels to consider, then go to step 2.
- Find the lowest label for each equivalent set in the equivalence table.
- Scan the picture (**second scan**). Replace each label by the lowest label in its equivalent set.



Project Timeline

		PROJECT TIME SCHEDULE									
		November					December				
		3	8	13	18	22	28	5	8	11	
Construction	VGA Display		█	█	█						
	Image Overlay		█	█	█						
	RGB image generator			█	█	█					
	Image Processor				█	█	█				
	Image Processor FSM							█	█	█	
Testing & Debugging	Module Testing		█	█	█	█	█	█	█		
	System Testing					█	█	█	█		
	Interface ext. devices								█	█	
		Project Proposal	Block diagram conference	Design presentation		Project checkoff list	Oral report	Final Oral Report		Final Report and Project Checkoff	

Possible extensions If there's time

- Implement detection/distinction between different blob colors
- Allow user input to select region on image for analysis; using a stylus