



iShirt

CS5245 Project

By

**Cedric Manzoni
Wang Zhiyu
Zhang Xin**

13 Nov 2005



Outline

- Introduction
 - Scenario
 - Special Effect
- Special Effect
 - Several Approaches
 - Square Corner Detection
 - Corner Coordinates Re-organization
 - 2D-warping & Anti-aliasing
- Conclusion
 - Techniques
 - Further Improvements

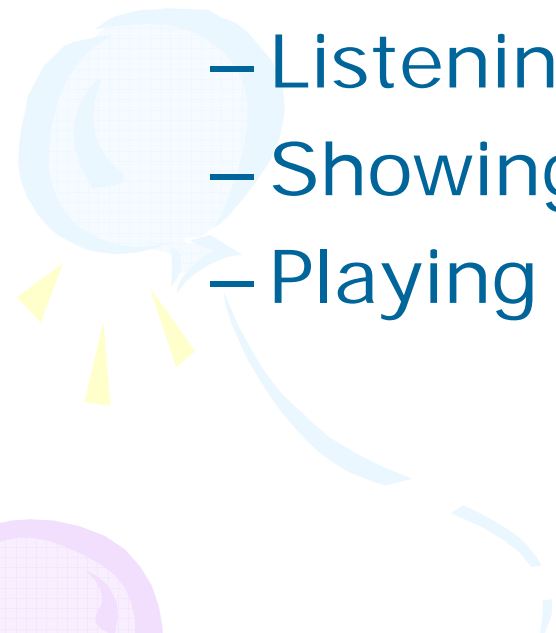



Introduction

- Scenario
- Special Effect



Scenario

- A parody of an Apple iPod ad
 - Releasing iShirt
 - Listening to mp3
 - Showing some photos
 - Playing a video
- 
- 

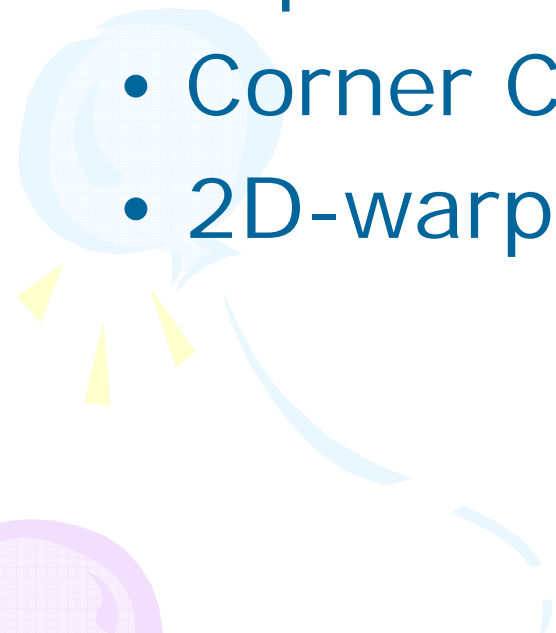
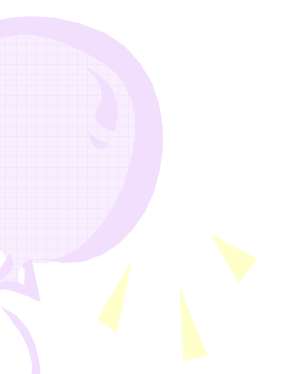
A decorative graphic on the left side of the slide features three balloons in green, blue, and purple. Each balloon has a grid-like texture and is surrounded by several yellow triangular rays, giving them a sun-like appearance. The balloons are connected by thin, curved lines.

Special Effect

- Video Mapping
on a non-flat surface



Special Effect

- **Several Approaches**
 - Square Corners Detection
 - Corner Coordinates Re-organization
 - 2D-warping & Anti-aliasing
- 
- 



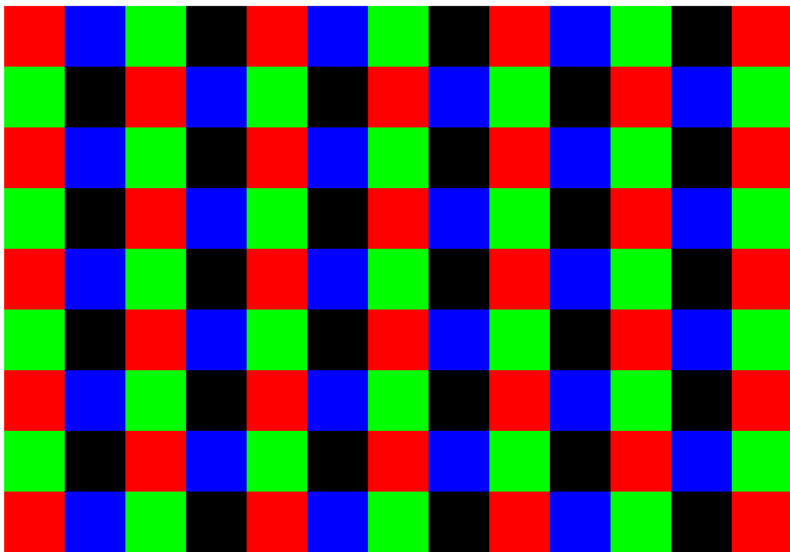
Several Approaches

- 3D Camera ☹️
- 2D Sub-Image Mapping
 - Tracking points/patches

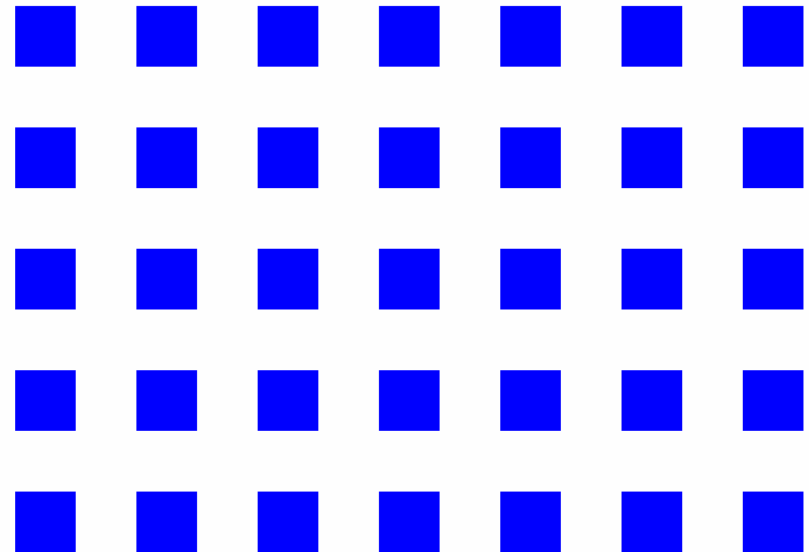
Several Approaches

- 2D Sub-Image Mappig

4-color pattern ☹️

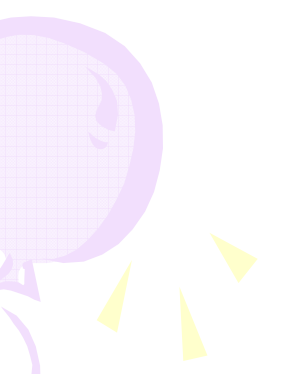


1-color pattern 😊



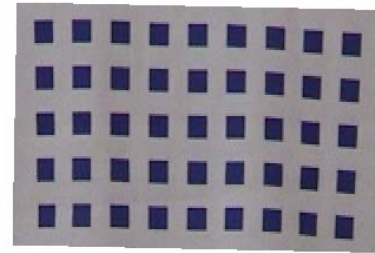
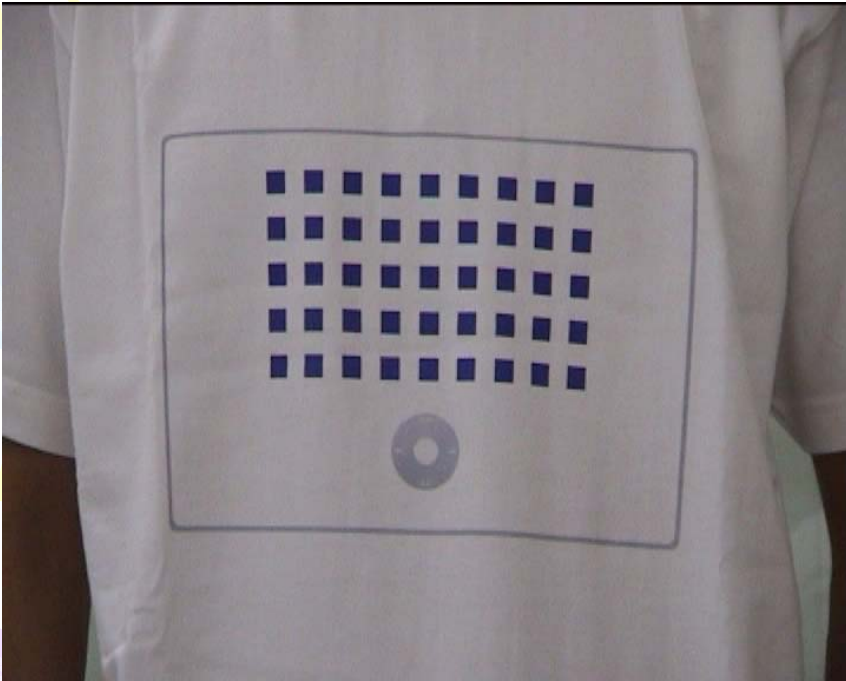


Special Effect

- Several Approaches
 - **Square Corners Detection**
 - Corner Coordinates Re-organization
 - 2D-warping & Anti-aliasing
- 
- 

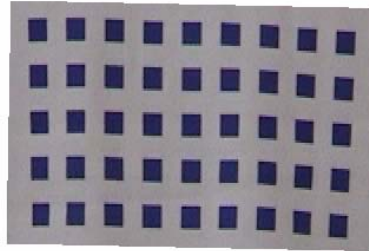
Square Corner Detection

1. Extract Pattern from video



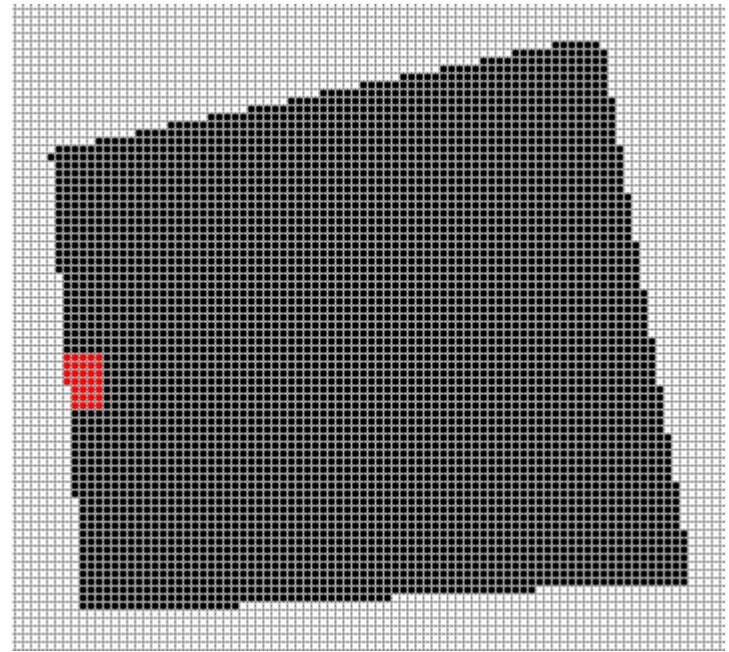
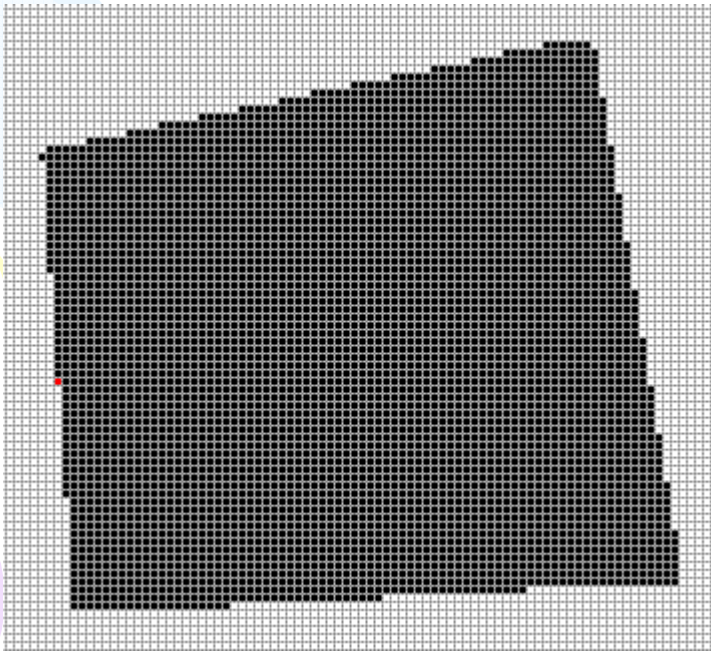
Square Corner Detection

2. Set a color threshold to differentiate black squares and background
3. Apply Scan-line Checking method to detect each square



Square Corner Detection

4. Use Color Filling algorithm to find the 4 corners of each square



Square Corner Detection

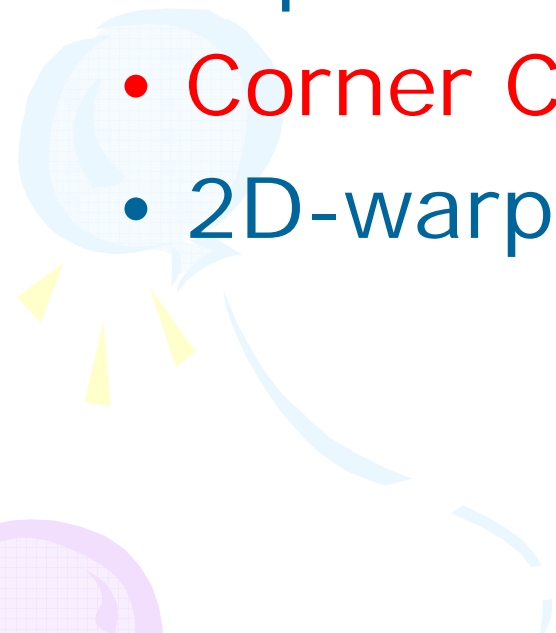
4. Use Color Filling algorithm to find the 4 corners of each square:

Corner	Characteristic
Top-Left	$-x-y$ is max
Top-Right	$x-y$ is max
Bottom-Left	$-x+y$ is max
Bottom-Right	$x+y$ is max

Assumption: the pattern can not rotate more than 45 degrees.


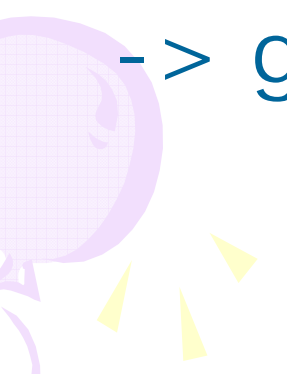


Special Effect

- Several Approaches
 - Square Corners Detection
 - **Corner Coordinates Re-organization**
 - 2D-warping & Anti-aliasing
- 
- 

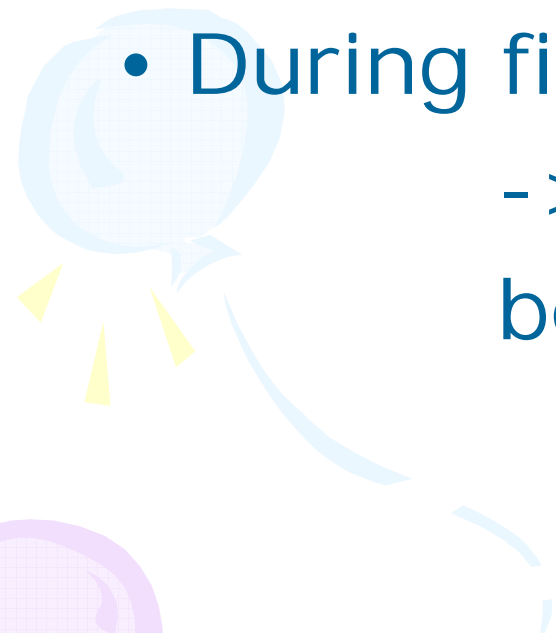



Corner Coordinates Re-organization

- During experiment :
 - > tracking of the corners rather than sorting.
 - > good results.
- 
- 



Corner Coordinates Re-organization

- During final work :
 - > sorting :
better results than tracking.
- 
- 



Special Effect

- Several Approaches
 - Square Corners Detection
 - Corner Coordinates Re-organization
 - 2D-warping & Anti-aliasing
- 
- 

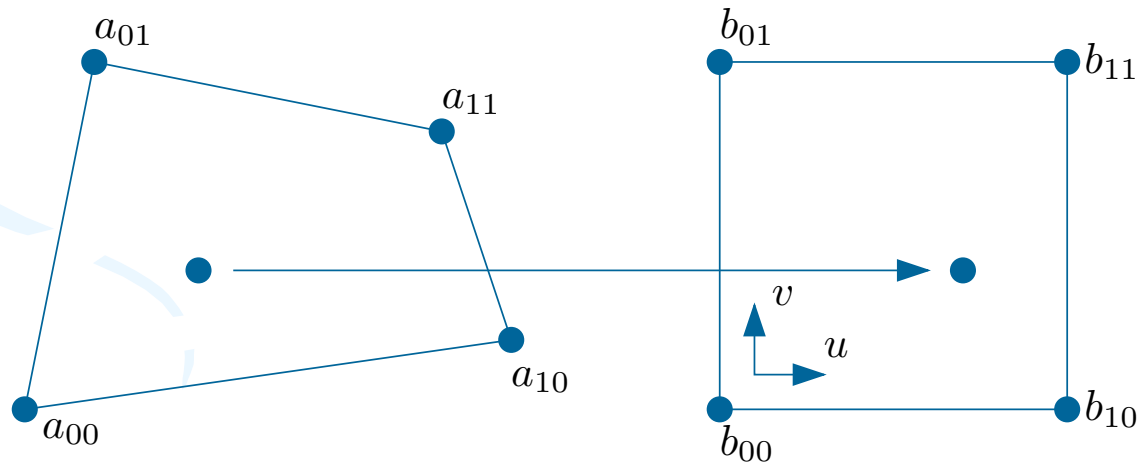
A decorative graphic on the left side of the slide features three balloons in green, blue, and purple. Each balloon has a grid-like texture and is attached to a string with several yellow triangular flags. The balloons are positioned vertically, with the green one at the top, the blue one in the middle, and the purple one at the bottom.

2D Warping

- Assumption: approximate each cell on T-Shirt as a quadrilateral

2D Warping

- Approach: Inverse Mapping
 - 17×9 cells on the T-Shirt
 - Do inverse mapping for 153 cells on each frame



2D Warping

- Quadratic Bezier Parameterization

- Forward mapping:

$$Q(u, v) = u \cdot v \cdot a_{00} + u \cdot (1-v) \cdot a_{01} + (1-u) \cdot v \cdot a_{10} + (1-u) \cdot (1-v) \cdot a_{11}$$

- Inverse mapping:

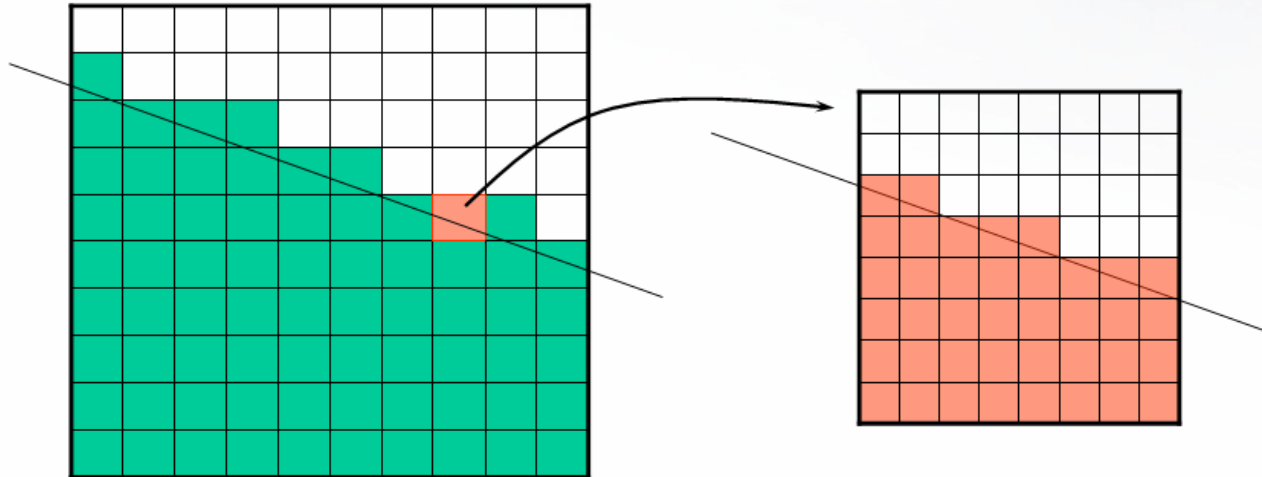
- Given $Q(u, v)$, which is a vector of length 2 specifying the pixel coordinates in the frame, we can also compute the (u, v) pair by solving the forward mapping equation.

2D Warping



Anti-aliasing

- Aliasing - *Jigsaw* effect
- Algorithm
 - Divide pixel into sub-pixels
 - Count sub-pixels





Conclusion

- Techniques
- Further Improvements


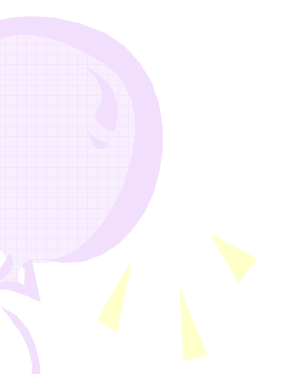


Techniques

- Image Masking
- Threshold
- Color Filling
- Bubble Sort
- 2D Warping
- Anti-aliasing (& Alpha-blending)



Further Improvements

- 2D → 3D
 - Get the depth info
 - Add ambient light effect
 - New Pattern
 - Handle overlaps
- 
- 



Thank you!