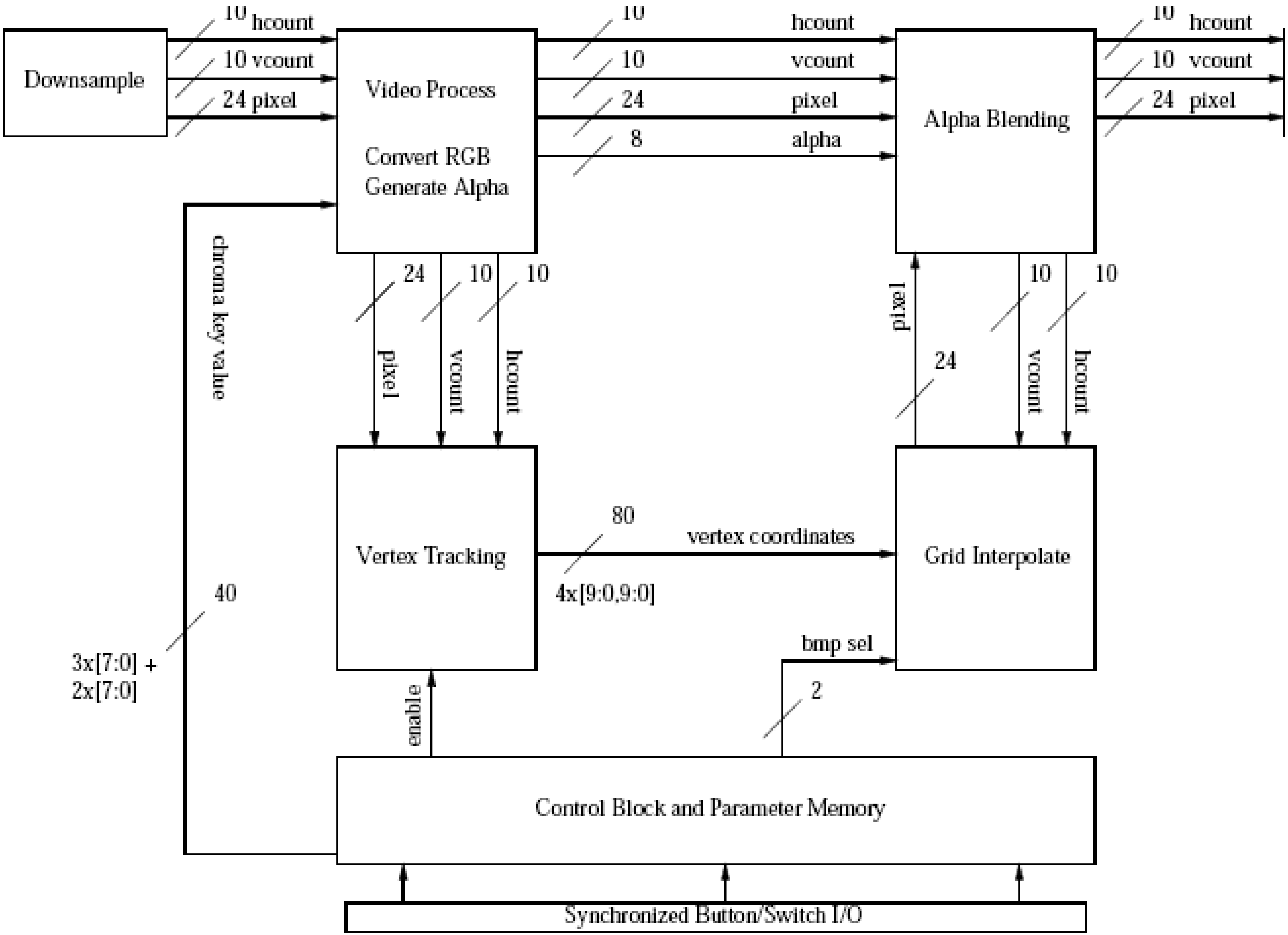
Virtual Postcard System

Jess Barber and AJ Meyer
6.111 Final Project Fall 2007

Overview:

- An augmented reality system for creating "virtual postcards"
- Camera takes in video of blank postcards
- System tracks position and motion of postcards by detecting corners
- System transforms a saved image to "fit on" the postcard
- Video output shows postcards with saved image apparently printed on its surface

System Overview: Block Diagram



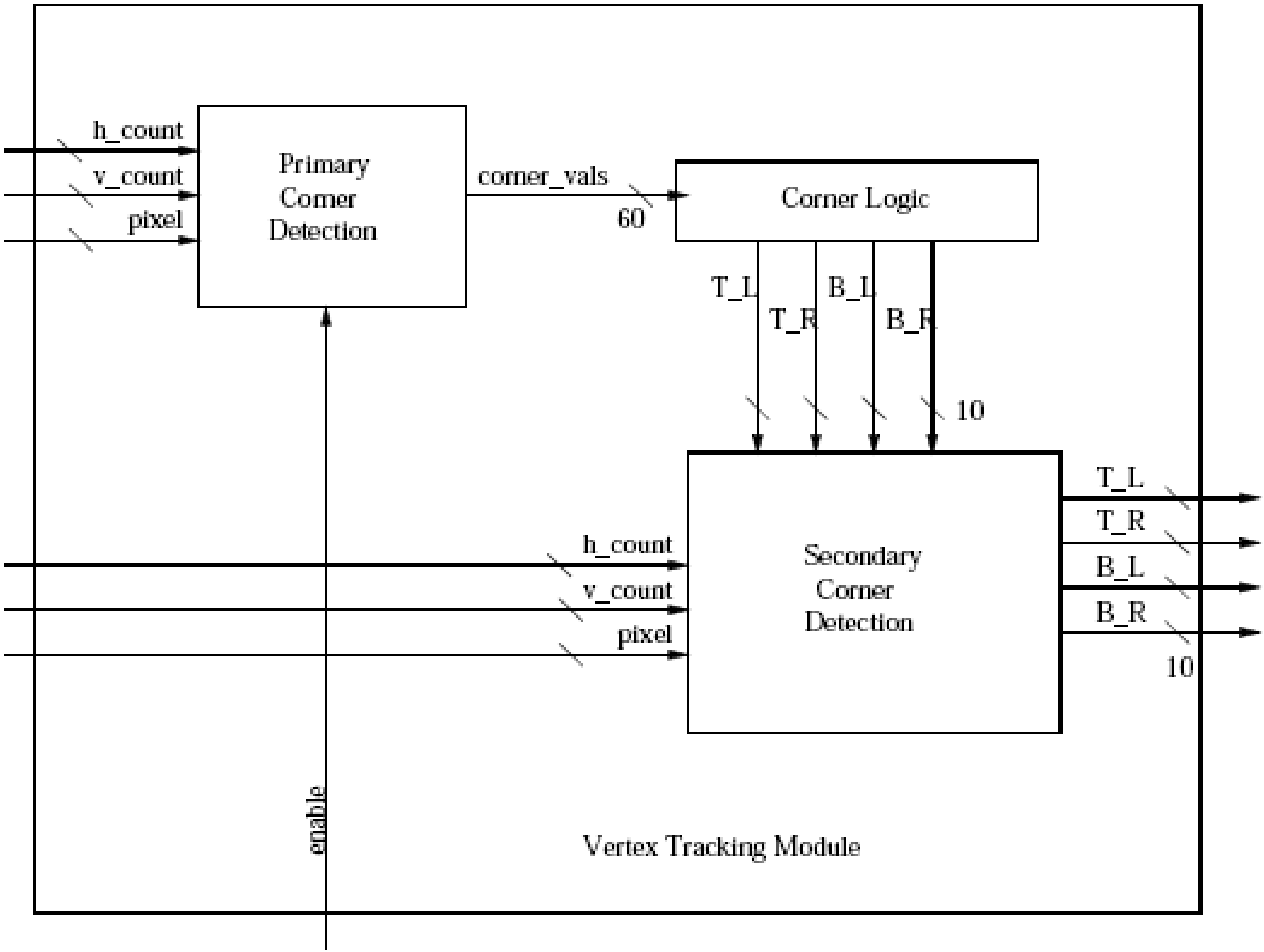
Video Processing

- Input is hsync, vsync, field, and downsampled pixel (Y'CbCr) values from the camera
- Converts to RGB values (as in Color Space Converter document)
- Produces alpha blending values (based on chroma key value set manually on labkit)

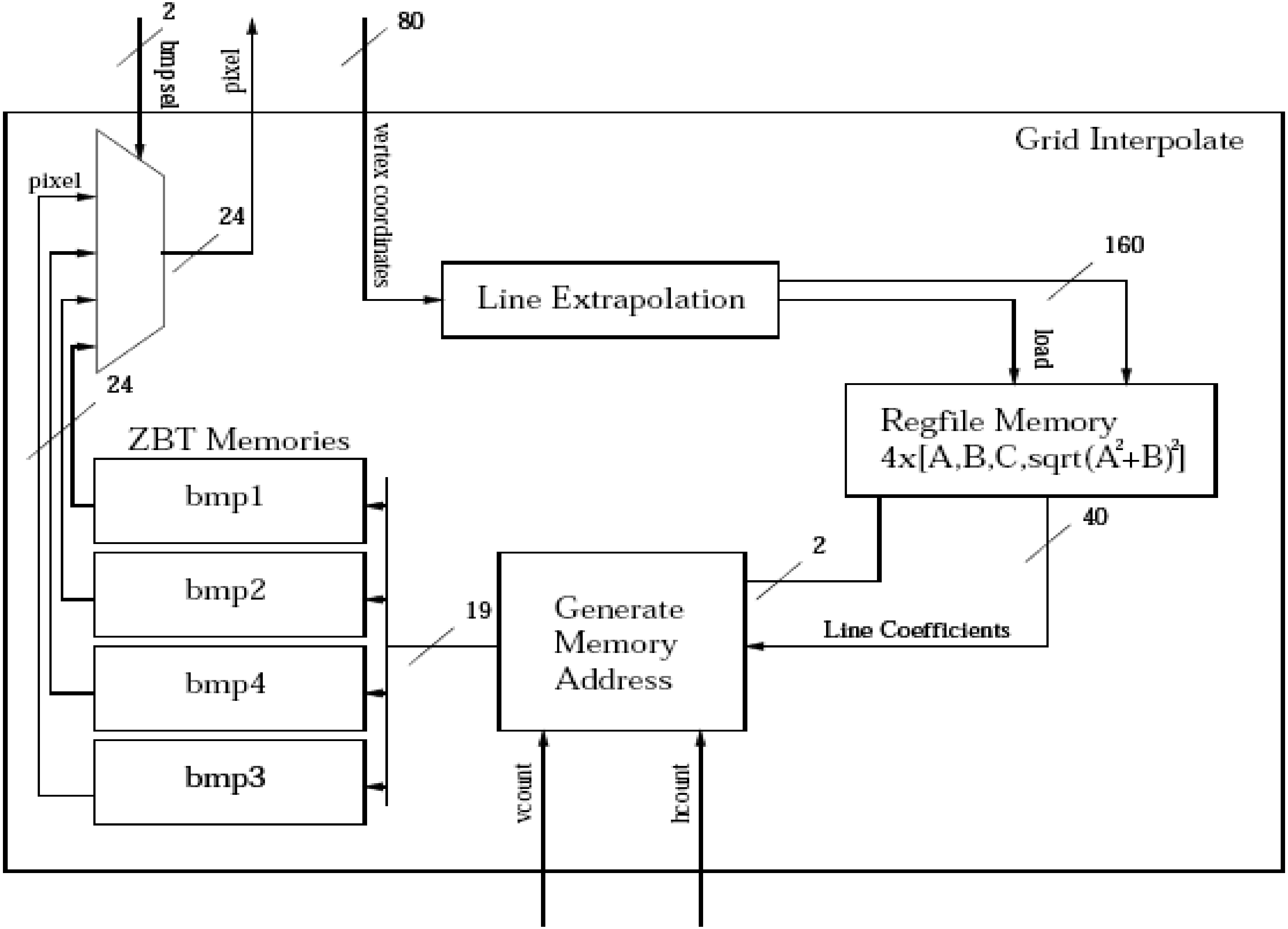
Vertex Tracking

• On enable, primary corner detection scans image and locates at least three vertices, sends these coordinates to corner logic
• Corner logic determines which corner is top right, top left, bottom right, bottom left vertex (extrapolates position of fourth corner if necessary)
• Secondary corner detection keeps vertex positions up-to-date and keeps track of which vertex is which (taking into account previous coordinates of vertices)

Vertex Tracking Diagram



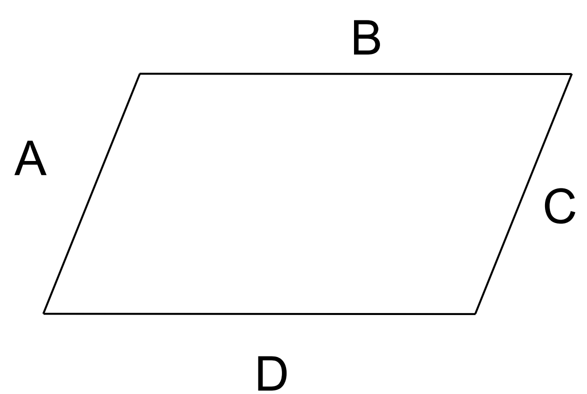
Grid Interpolate Diagram



Transforming the Bitmap

From vertices, generate 4 lines of the form Ax+By+C=0
- For each hcount,vcount:
  - Check if the point is interior to quadrilateral
  - If no, output [255,255,255]
  - If yes, find the point distance from bounding lines:
    - di = (Aix0+Biy0+Ci)/sqrt(Ai^2+Bi^2)
    - X = da*xmax/(da+dc)
    - Y = db*ymax(db+dd)

What about that sqrt term...?



Note on Interpolation

Usually, it would be necessary to interpolate over skewed regions

By storing a rectangular bitmap whose minimum dimension is greater than the maximum screen dimension (diagonal), it is never necessary to interpolate to a higher resolution

Geometric transformations such as scale, rotation, and perspective are done in the REAL WORLD and are reflected by the change in vertices!

ITS NOT NECESSARY TO PERFORM MATRIX OPERATIONS

Alpha Blending

Naiive Chroma key can look choppy and jagged along edges

Chroma key generates an 8 bit value for every pixel

Target Color = [Rt,Gt,Bt] = Pt   Input Pixel = [R,G,B] = Pi

1 - Pt dot Pi = E   gives a heuristic for how close we are to our target

alpha is piecewise:

255, E < Tlow

k*E, Tlow < E < Thigh

0, E > Thigh

Were effectively Mux'ing between the original image and the transformed bitmap, but interpolating along some threshold!

Possible Augmentations

- Multiple index cards
- Overlapping cards
- Selecting between multiple bitmaps

Timeline

- Before Thanksgiving – Jess: accurately detect four vertices (debug w/video out; AJ: generate transformed image “grid” given four vertices
- Nov. 30: Coherent system: alpha blending, timing issues allow a functional system
- Dec. 7: All bugs worked out (i.e. Output looks good, system is robust) and if time, augmentations added