Using Graph Layout to Generalise Focus+Context Image Magnification

Shawn Martin, Vijay Subramanya,
and Steven Mills

10th October, 2012

Image Magnification

- There are three main methods used for image magnification.

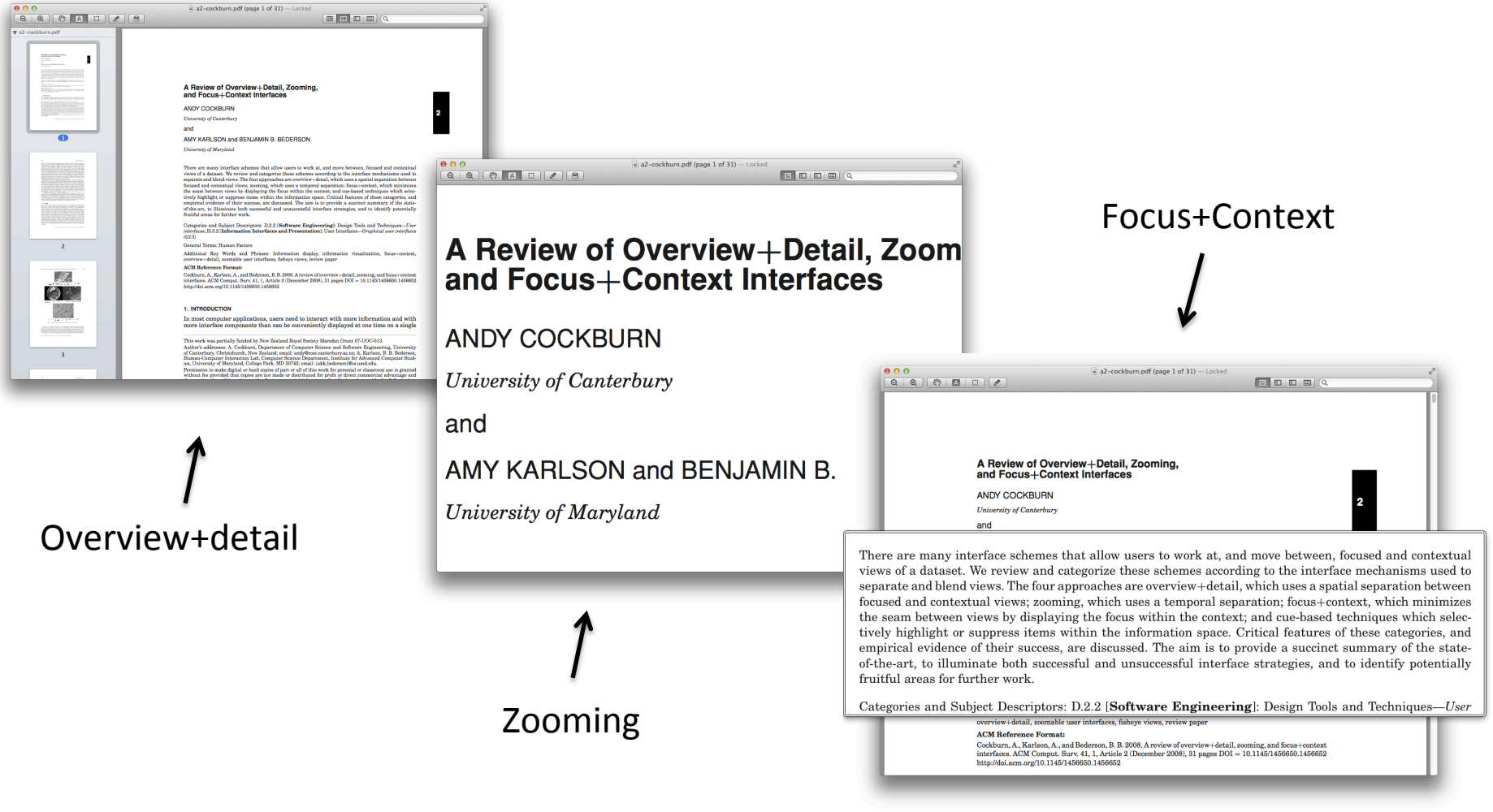
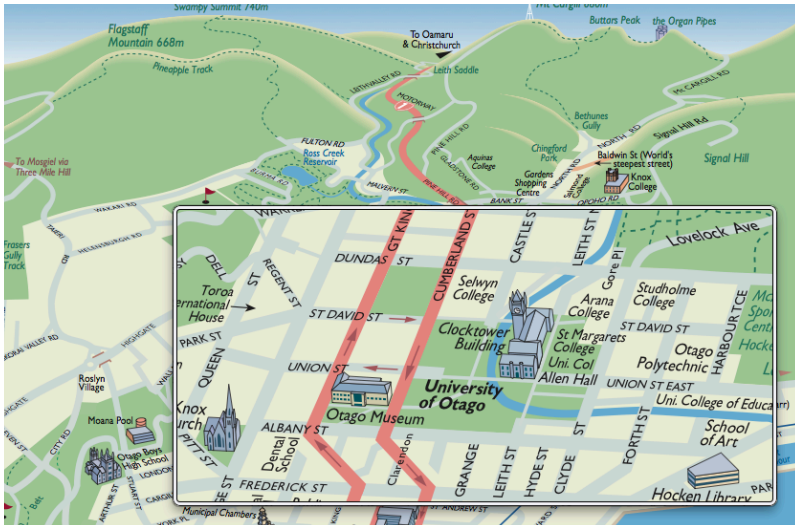


Image Distortion

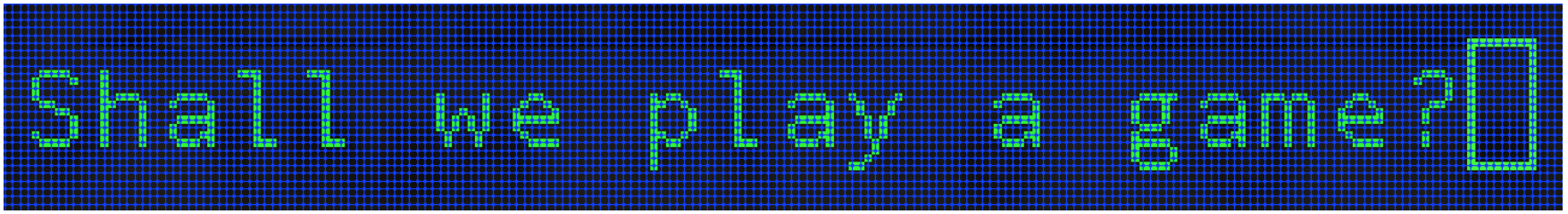
- Because focus+context image magnification can obscure the context, distortion-based methods have been proposed, e.g. fish-eye lenses.



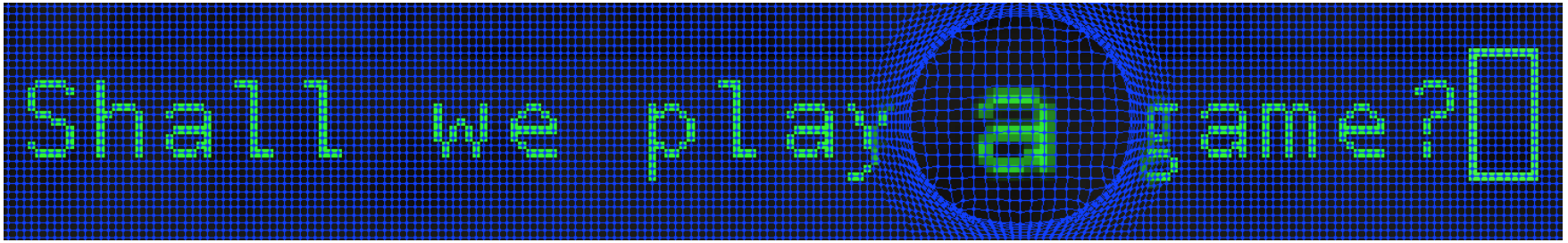
- Our goal is to improve image distortion for magnification.
  - Can we create an *image-aware* magnifier?

The Image Graph

- Underlying every image is a graph.



- A fish-eye lens is a distortion of the graph.



Spectral Graph Layout

- Spectral graph layout can be used to distort the image graph.
- Given a graph $G = (V, E)$ with edge weights w_{ij}, define the *Laplacian* of G:

$$L_{ij} = \begin{cases} \sum_k w_{ik} & i = j \\ -w_{ij} & i \neq j \end{cases}$$

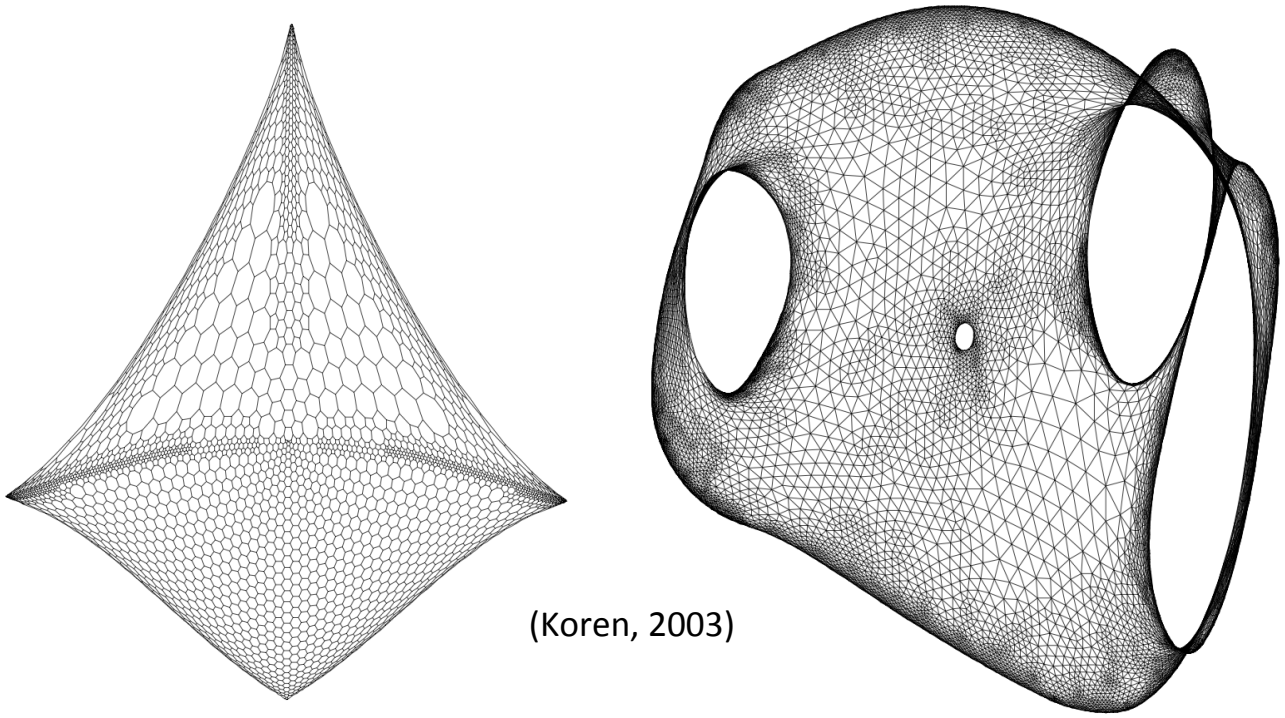
- A one-dimensional layout of G can be obtained by solving:

$$\begin{aligned} \min_{\mathbf{x}} \quad & E(\mathbf{x}) = \sum_{ij} w_{ij}(x_i - x_j)^2 = \mathbf{x}^T L \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x}^T \mathbf{x} = 1 \\ & \mathbf{x}^T \mathbf{1}_n = 0 \end{aligned}$$

- This problem is solved by taking the 2nd smallest eigenvector of L.
- A two-dimensional layout is obtained by taking the 2nd and 3rd smallest eigenvectors of L.

Spectral Graph Layout Example

- Spectral graph layout is fast and yields aesthetically pleasing drawings …



(Koren, 2003)

- … however, the drawings are not (in general) rectangular, so they don't work well for images.

Constrained Graph Layout

- We can modify the standard spectral layout algorithm to fix the edges of the graph to lie on a rectangle.

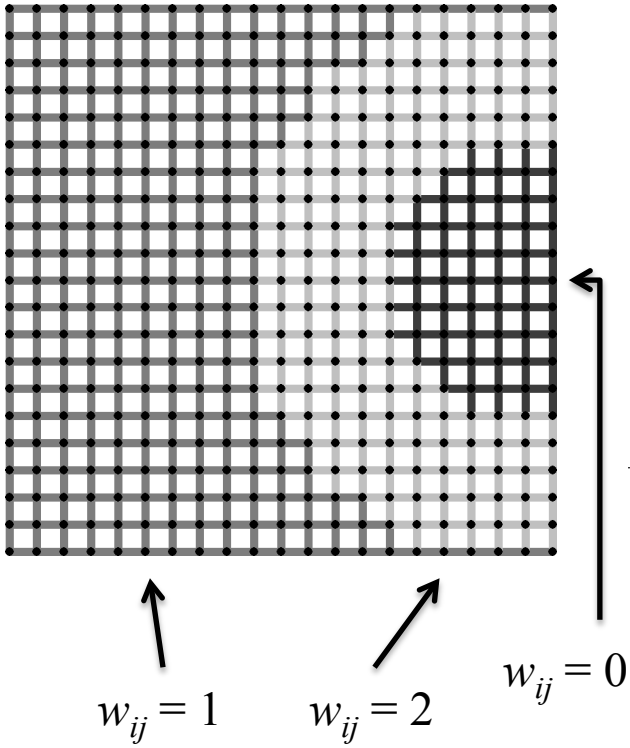
$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^T L \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x}^T \mathbf{x} = 1 \\ & \mathbf{x}^T \mathbf{1}_n = 0 \end{aligned} \quad \Longrightarrow \quad \begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^T L \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b} \end{aligned}$$

- Replace the scaling ($\mathbf{x}^T\mathbf{x} = 1$) and translation ($\mathbf{x}^T\mathbf{1}_n = 0$) constraints by constraints which directly restrict the edge coordinates ($A\mathbf{x} = \mathrm{b}$).
- This is a quadratic programming problem with linear constraints, so it's a linear problem.

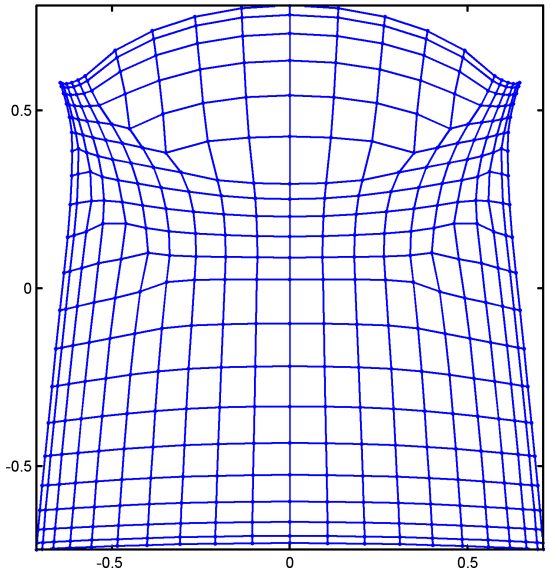
Constrained Graph Layout Example

- By supplying appropriate weights, we can use our layout algorithm to obtain a fish eye lens.

Weighted Graph



Standard Layout



Constrained Layout

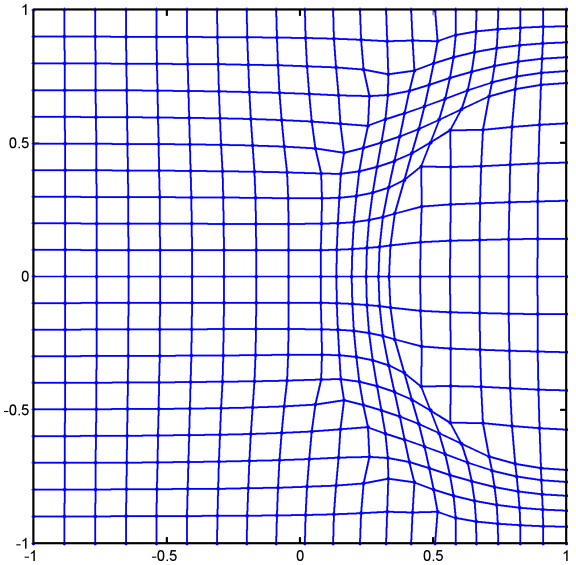


Image Magnification Example

- An example of a fish-eye lens using our constraint layout algorithm.

- New Zealand House of Representatives, circa 1900-1902.

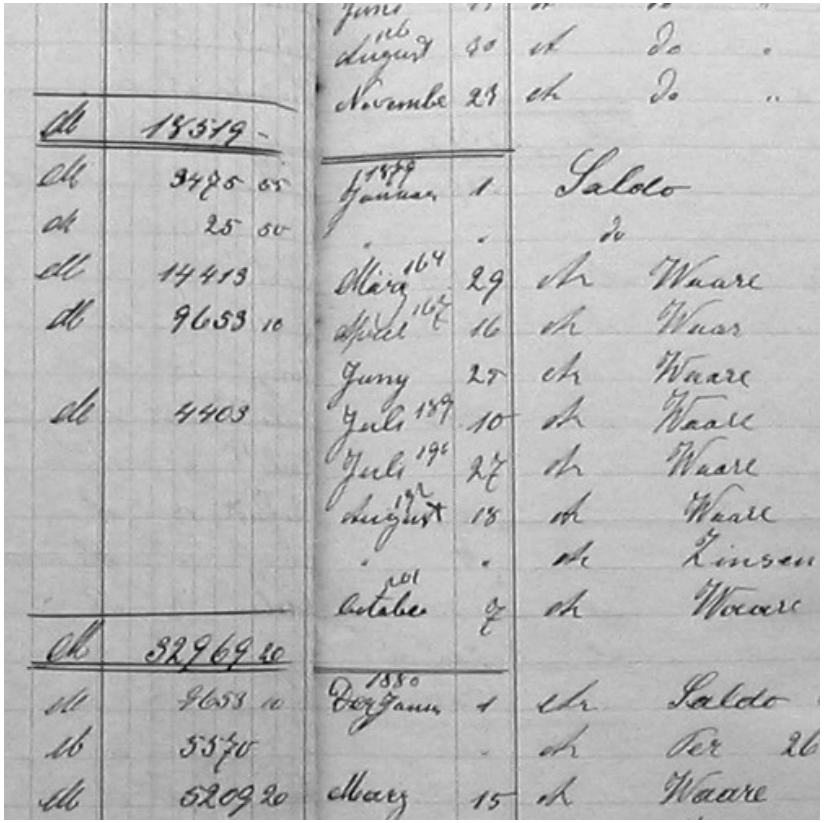
Image-aware Distortion

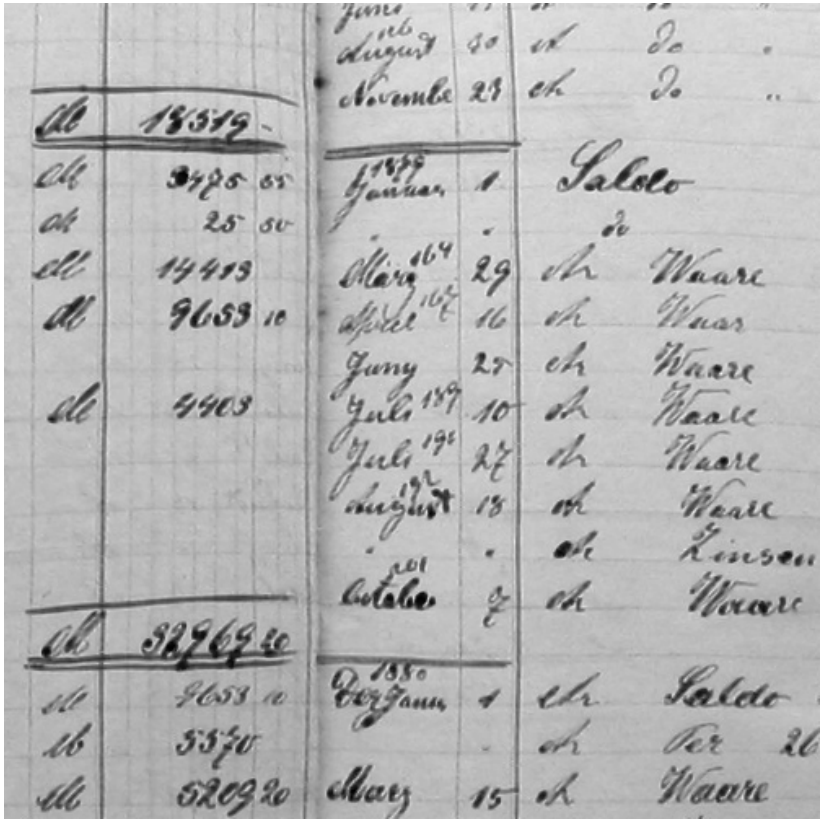
- We can modify the image graph weights according to pixel value to get image-aware magnification.

- Here we weight according to 1 - pixel intensity.

Image-aware Distortion

- Another example using pixel intensity.





- Fur trader's ledger, 1876-1885.

An Algebra of Distortions

- We can combine weight functions.

- Cook's map of Bank's island (actually a peninsula).

Conclusions

- Constrained graph layout can be used to magnify and distort images in useful ways.
- Future Work (Technical)
  - We need to investigate/invent useful weighting functions.
  - What is the best way to combine weighting functions?
  - A weighting function can produce a distortion. Can we solve for a weighting function if given a distortion?
  - More generally, can we use constrained graph layout as a means of universal image distortion?
- Future Work (User Interface)
  - Is this type of image magnification/distortion useful to an average person?
  - What is the best way to present this technology to a user?
  - How can users select/combine weighting functions in an intuitive manner?