

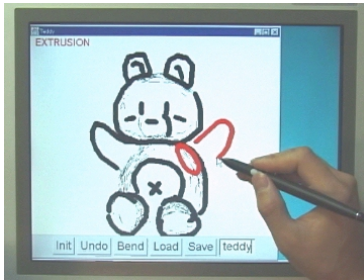
Smart Graphics: Milestones: Text and UIs

Vorlesung „Smart Graphics“
Andreas Butz, Otmar Hilliges
Mittwoch, 25. Januar 2006

Topics today

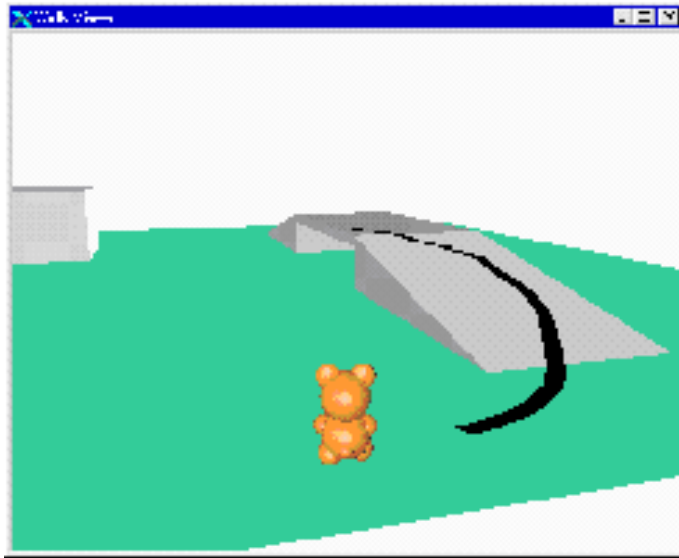
- Freeform UIs, Igarashi 03
- Smart Text Arvo 00
- Fluid Documents, Chang 00
- SCWM, Badros 00
- Diagrams, Anderson 00+01

Some more Freeform UIs



[Takeo Igarashi, SG03]

Path drawing for VR



- Interpretation of pen strokes by projection onto the ground in 3D
- Observation of additional constraints in 3D world
- „smart“ interpretation filters out unwanted input
- [Video](#)

Flatland

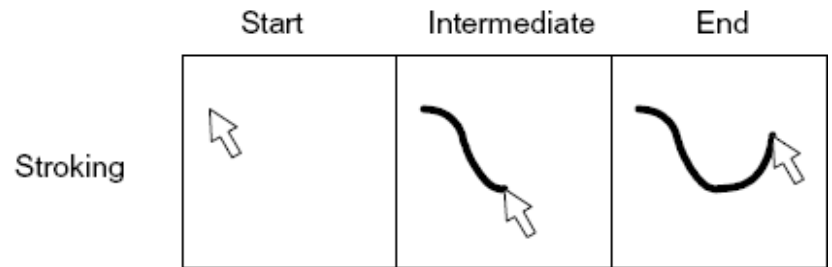


- Whiteboard application for informal activities
- Interpretation of strokes
- Spatial constraints between objects ([Video](#))

Characteristics of Freeform UIs

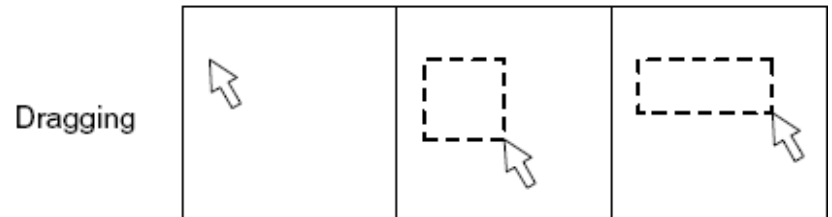
- Stroke-based Input

- Richer than mouse-based dragging, since intermediate states make a difference



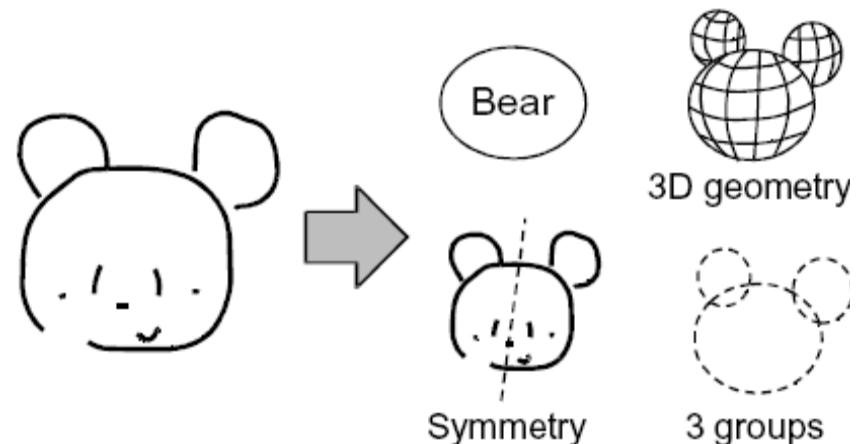
- Perceptual Processing


- Humans perceive much richer information in drawings



- Informal Presentation

- „imprecise“ visual representation suggests that also the computation is not precise



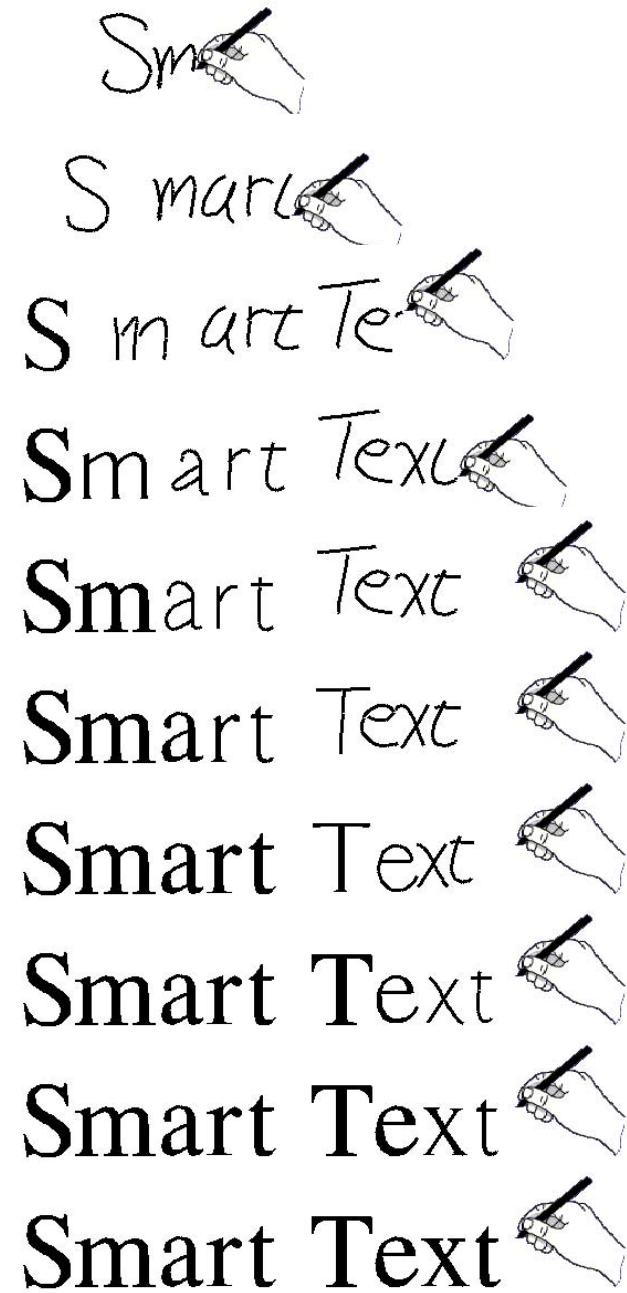
Smart Text 



[James Arvo, SG 2000]

Motivation

- Character input on pen tablets and tablet PCs visually unsatisfying:
 - Pen strokes show jittery hand writing
 - Recognized characters are displayed abruptly and in different location
 - Visual interruption of the process
- Wanted: smooth transformation between pen strokes and recognized characters
 - No visual interruption in the process
 - Touching up recognized chars for correction
- Metaphor: self-organizing virtual ink



Recognition process

- Input: Sequences of strokes
 - Temporally distinct
 - Spatially overlapping
- Find groups of strokes
 - Queue all strokes and lag behind
 - Find most promising matches from the queue
 - Remove the corresponding strokes and continue
- Recognize stroke groups
 - Nearest neighbor matching
 - 50-dimensional feature space
 - Features: stroke length, slope, curvature, gap between strokes, scale invariant!

Morphing process

- Assign recognized strokes to lines in the recognized symbol
- Morph strokes to lines (stroke font)
 - Move if necessary
 - Scale if necessary
- Morph lines to character shapes (outline font)

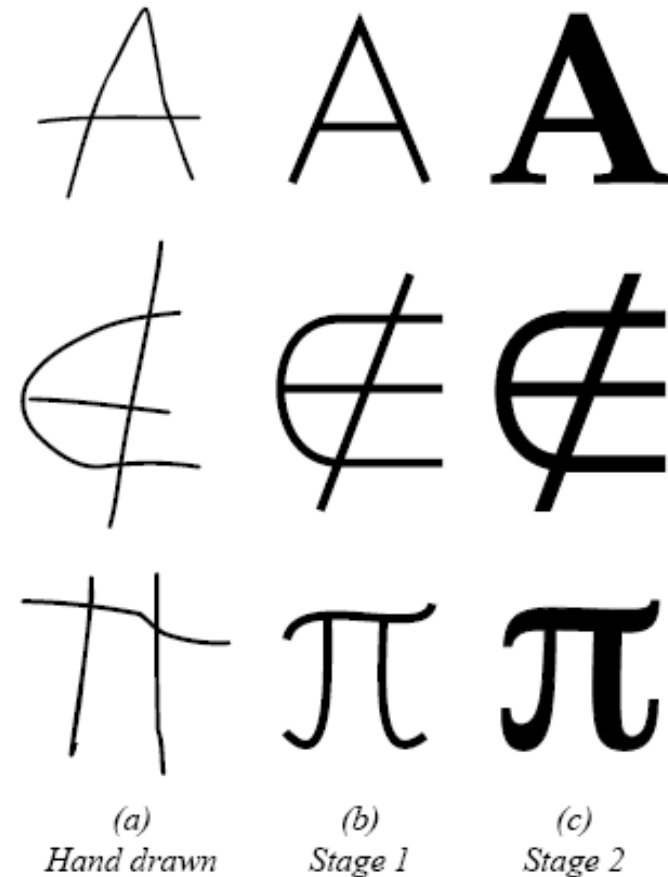
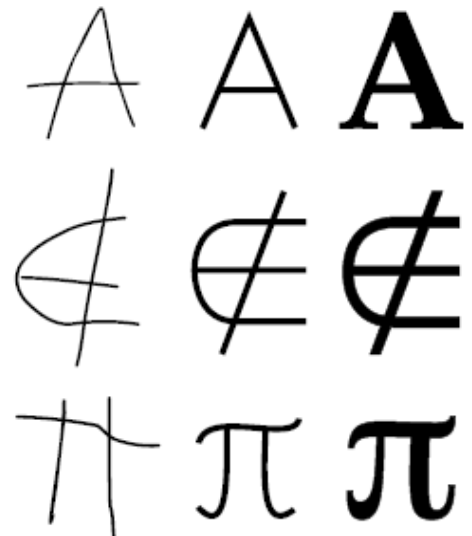


Figure 1: *Each morph has two conceptual stages. In the first stage, lines are blended into lines. In the second stage, lines are expanded to fill in all the font features.*

Properties of the stroke font



- **Containment:** Each stroke character is entirely contained within its corresponding outline character.
- **Clarity:** Each character of the stroke font should be aesthetically pleasing as well as easily recognizable.
- **Correspondence:** The strokes comprising each character in the stroke font should correspond as closely as possible to the strokes of a typical hand-drawn rendition.

Design of the stroke font

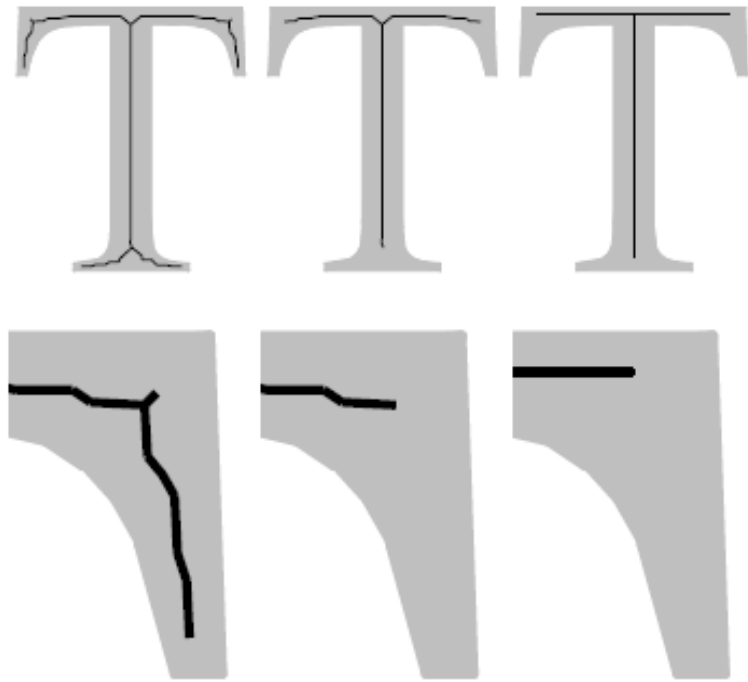


Figure 2: A skeleton versus a hand-designed stroke character for the symbol “T” (top row) and a close-up of a serif region (bottom row). The raw CAT skeleton on the left is pruned, resulting in the automatically generated stroke character shown at center. On the right, a hand-designed stroke character is shown. (Lines are overlaid on the filled outline character for reference.)

- Skeletonization of the outline font
 - Containment OK!
 - recognizable
 - Contains unwanted details from serifs etc.
- Pruning of the resulting skeleton
 - Better, but...
- Hand-drawn font
 - Aesthetically most pleasing

Post-processing strokes



Figure 3: To create a one-to-one correspondence between strokes, the strokes of a character may be (a) split at the sharpest bend, or (b) joined where they are most nearly collinear.

Matching strokes to stroke font lines



- Connect points on stroke to points on the line
- Evaluate energy function on connecting lines
 - Sum of the squares of distances
- Typically only few strokes \rightarrow try all combinations
- Retain the match with minimal energy

Stage 1 morph

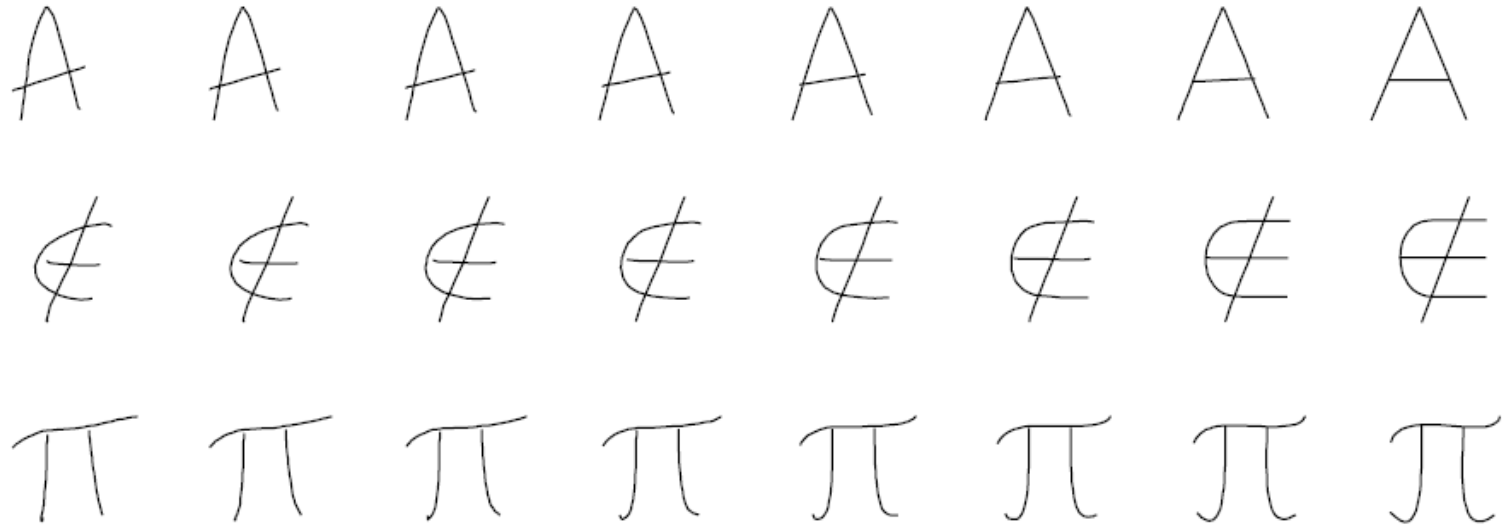


Figure 6: *Three examples of the Stage-1 metamorphosis, which maps user-drawn characters to corresponding characters of a stroke font.*

Stage 2 morph

- For each point on the outline font, find closest point on the stroke font
- Linearly interpolate points from stroke font to the outline font
- Intermediate shapes seem to gradually „grow“ from the stroke font to the outline font

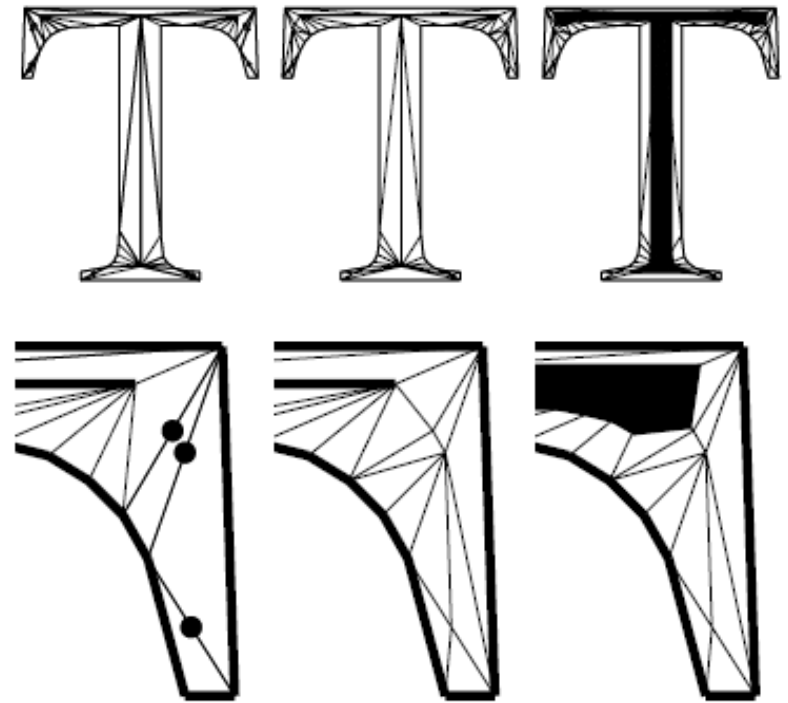


Figure 7: *Creating a Stage-2 morph for the letter “T” (top row) and a close-up in a serif region (bottom row). The triangulation at left respects the constraint edges indicated by thick lines. The points indicated in the left panel are added to the mesh shown at center. At right, an interpolated level set is shown.*



Figure 8: *A few steps in a Stage-2 morph.*

Fluid Documents

Bay-Wei Chang

Motivation

- Annotate primary text with secondary information
 - Footnotes
 - Hyperlinks
 - Literature references
 - Index references
- Problem with current techniques
 - Disrupt the reading flow
- Solution in dynamic media
 - Place the secondary material close to the primary text
 - Different techniques to do so...
- Transitions are animated

Callout

When in the Course of human Events, it becomes necessary for one People to dissolve the Political Bands which have connected them with another, and to assume among the Powers of the Earth, the separate and equal Station to which the Laws of Nature and of Nature's God entitle them, a decent Respect to the Opinions of Mankind requires that they should declare the causes which impel them to the Separation.

We hold these Truths to be self-evident, that all Men are created equal, that they are endowed by their Creator with certain unalienable Rights, that among these are Life, Liberty and the Pursuit of Happiness -- That to secure these Rights, Governments are instituted among Men, deriving their just Powers from the Consent of the Governed, that whenever any Form of Government becomes destructive of these Ends, it is the Right of the People to alter or to abolish it, and to institute new Government, laying its

As unequal in many ways as humans may be, no one human or class of humans is superior to another human or class of humans.

Compress

When in the Course of human Events, it becomes necessary for one People to dissolve the Political Bands which have connected them with another, and to assume among the Powers of the Earth, the separate and equal Station to which the Laws of Nature and of Nature's God entitle them, a decent Respect to the Opinions of Mankind requires that they should declare the causes which impel them to the Separation.

We hold these Truths to be self-evident, that all Men are created

As unequal in many ways as humans may be, no one human or class of humans is superior to another human or class of humans.

equal, that they are endowed by their Creator with certain unalienable Rights, that among these are Life, Liberty and the Pursuit of Happiness -- That to secure these Rights, Governments are instituted among Men, deriving their just Powers from the Consent of the Governed, that whenever any Form of Government

Overlay

When in the Course of human Events, it becomes necessary for one People to dissolve the Political Bands which have connected them with another, and to assume among the Powers of the Earth, the separate and equal Station to which the Laws of Nature and of Nature's God entitle them, a decent Respect to the Opinions of Mankind requires that they should declare the causes which impel them to the Separation.

We hold these Truths to be self-evident, that all Men are created

equal, that they are endowed by their Creator with certain unalienable Rights, that among these are Life, Liberty and the Pursuit of Happiness -- That to secure these Rights, Governments are instituted among Men, deriving their just Powers from the

As unequal in many ways as humans may be, no one human or class of humans is superior to another human or class of humans.

Consent of the Governed, that whenever any Form of Government becomes destructive of these Ends, it is the Right of the People to alter or to abolish it, and to institute new Government, laying its

How is this achieved?

- In normal reading mode, just show a small visual cue that there is secondary material
- When selected (i.e., mouse over), material must become
 - Readable, i.e. bigger
 - Salient and distinguishable from primary text
- Primary text has different strategies for making space
 - Moving, shrinking, overlay, margin
- Supporting material has different display strategies
 - Text reflow, text & image layout
- Space and salience are negotiated between primary text and secondary material
 - Common protocol for characteristics

SpecificGChar

size	170x28 pixels
font face	Times
font size	12pt
color	black

RangedGChar

size	>= 115x10 pixels
font face	<any>
font size	>= 8pt
color	<any>

Negotiation process

1. Supporting object makes initial proposition
2. Primary text checks available space and characteristics and provides guideline
3. Supporting object chooses presentation strategy
 - May differ from original proposal
 - Must fit guideline
4. Primary object chooses space-making strategy

Examples of fluid documents ([Video](#))

But soft! What light through yonder window breaks?
It is the East, and Juliet is the sun!
Arise, fair sun, and kill the envious moon,
Who is already sick and pale with grief
That thou her maid art far more fair than she.
Be not her maid, since she is envious.
Her vestal livery is but sick and green,
And none but fools do wear it. Cast it off.

Figure 1. Excerpt of *Romeo and Juliet* with underline cues indicating annotations.

But soft! What light through yonder window breaks?
It is the East, and Juliet is the sun!
Arise, fair sun, and kill the envious moon,
Who is already sick and pale with grief
That thou her maid art far more fair than she.
the moon is here thought of as Diana
Be not her maid, since she is envious.
Her vestal livery is but sick and green,
And none but fools do wear it. Cast it off.

Figure 2. An animation expands the annotation fluidly, moving the surrounding text apart.

But soft! What light through yonder window breaks?
It is the East, and Juliet is the sun!
Arise, fair sun, and kill the envious moon,
Who is already sick and pale with grief
That thou her maid art far more fair than she.
Be not her maid, since she is envious.
Her vestal livery is but sick and green,
And none but fools do wear it. Cast it off.

The moon
is here
thought of
as Diana



Figure 3. Margins can also be used for supporting material.

But soft! What light through yonder window breaks?
It is the East, and Juliet is the sun!
Arise, fair sun, and kill the envious moon,
Who is already sick and pale with grief
That thou her maid art far more fair than she.
Be not her maid, since she is envious.
the moon is here thought of as Diana
Her vestal livery is but sick and green,
And none but fools do wear it. Cast it off.

Figure 4. When space is at a premium, supporting material can overlay the primary material.

SCWM - Scheme Constraints Window Manager



[Greg Badros, SG 2000]

Motivation

- Window managers allow window positioning only by direct manipulation.
- Users might want to specify window placements on a higher level
 - examples: tile, cascade
 - ..but more elaborate relations needed..
- Approach: specify spatial constraints between windows

```
upintheir
face.h          util.c
focus.c       util.h
focus.h       validate.h
font.c         virtual.c
font.h         virtual.h
fvwm2.man     window.c
generate_scm_init_funcs* window.h
getopt.c      window_fwd.h
getopt.h      winprop.c
getopt1.c     winprop.h
guile-compat.c xmisc.c
guile-compat.h xmisc.h
icons.c       xproperty.c
icons.h       xproperty.h
image.c       xrm.c
image.h       xrm.h
upintheir:~/scwm/srcZ
```

```
upintheir
cn-keep-bottoms-even.xpm  resize_h.xpm
cn-keep-lefts-even.xpm   resize_tr.xpm
cn-keep-rights-even.xpm  resize_v.xpm
cn-keep-to-left-of.xpm   scwm-logo-orange-bg.xpm
cn-keep-top-bottom-even.xpm scwm-logo.xpm
cn-keep-tops-even.xpm    scwm-name.xpm
cn-max-hsize.xpm         small-anchor.xpm
cn-max-vsize.xpm         win-pos-center.xpm
cn-min-hsize.xpm         win-pos-e.xpm
cn-min-vsize.xpm         win-pos-n.xpm
cn-relative-hsize.xpm    win-pos-ne.xpm
cn-relative-vsize.xpm    win-pos-nw.xpm
cn-right-left-even.xpm   win-pos-s.xpm
cn-strict-relative-pos.xpm win-pos-se.xpm
cn-window-height-sum.xpm win-pos-sw.xpm
cn-window-width-sum.xpm  win-pos-w.xpm
*upintheir:~/scwm/pixmapsZ
```

```
upintheir's xemacs:
File Edit Apps Options Buffers Tools Help
Open Dired Save Print Cut Copy Paste Undo Spell
(Text)----L1--All-----
```

```
man scwm
scwm(1)
scwm(1)
NAME
scwm, scwmexec, scwmrepl - Scheme Constraints Window Manager.
SYNOPSIS
scwm      [--display|-d          dpy]
           [--debug][--expression|-e      expression]
           [--file|-f rc_file] [--single-screen|-s]
           [--blackout|-b] [--version|-V] [--help|-h]
scwmexec expression
:
```



Types of constraints

- **Constant Height/Width Sum** Keep the total of the height/width of two windows constant.
- **Horizontal/Vertical Separation** Keep one window always to the left of or above another.
- **Strict Relative Position** Maintain the relative positions of two windows.
- **Vertical/Horizontal Maximum/Minimum Size** Keep the height/width of a window below/above a threshold.
- **Vertical/Horizontal Relative Size** Keep the change in heights/widths of two windows constant (i.e., resize them by the same amount, together).
- **Vertical/Horizontal Alignment** Align the edge or center of one window along a vertical/horizontal line with the edge or center of another window.
- **Anchor** Force a window position to stay in place.



Application of constraints

- Click on toolbar to select constraint type
- Click on windows to be constrained
 - Middle of the window: constrain middle (e.g., alignment)
 - Edge or corner: constrain edge or corner

NW 0	N 1	NE 3
W 3	C 4	E 5
SW 6	S 7	SE 8



XTerm A

```
clavicle:~/scwm/src$  
bck-i-search: _
```

XTerm B

```
/bin/sh ../mkinstalldirs /usr/include  
upintheir:~$
```

XTerm C

```
clavicle:~/scwm/src$
```

clavicle's xemacs:

File Edit Apps Options Buffers Tools Help

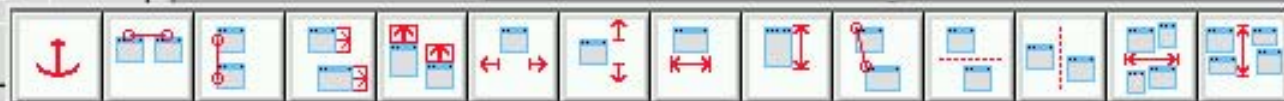
Open Dired Save Print Cut Copy Paste Undo Spell

Auto-saving... done

Constraint investigator

Vertical separation: 2 windows	<input type="checkbox"/>	Delete
Horizontal separation: 2 windows	<input type="checkbox"/>	Delete
Minimum width	<input type="checkbox"/>	Delete
Anchor:	<input type="checkbox"/>	Delete

Disable All Enable All



Managing constraints

- Constraint investigator window
- Visual overlay on the actual window
- Compositions (programming by example)
 - Record the composition of n windows
 - At playback, ask for n other windows
- Implicit constraints
 - Snap-dragging: near windows snap together and a constraint is added.
 - Scheme code can be written to automatically add constraints when a window is created

Implementation

- Based on fvwm2
- Extended by Scheme programming language
- Integration of a constraint solver written in C++, wrapped in scheme
- <http://scwm.sourceforge.net/>
- ...ends in 2000 with version 0.99.6.2

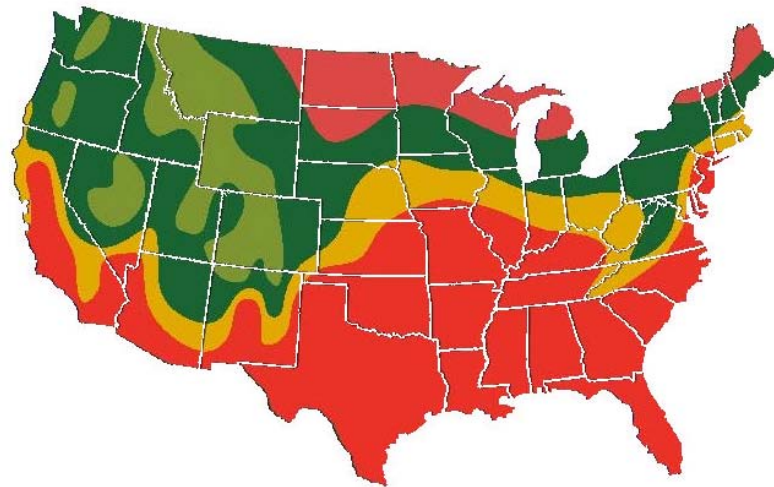
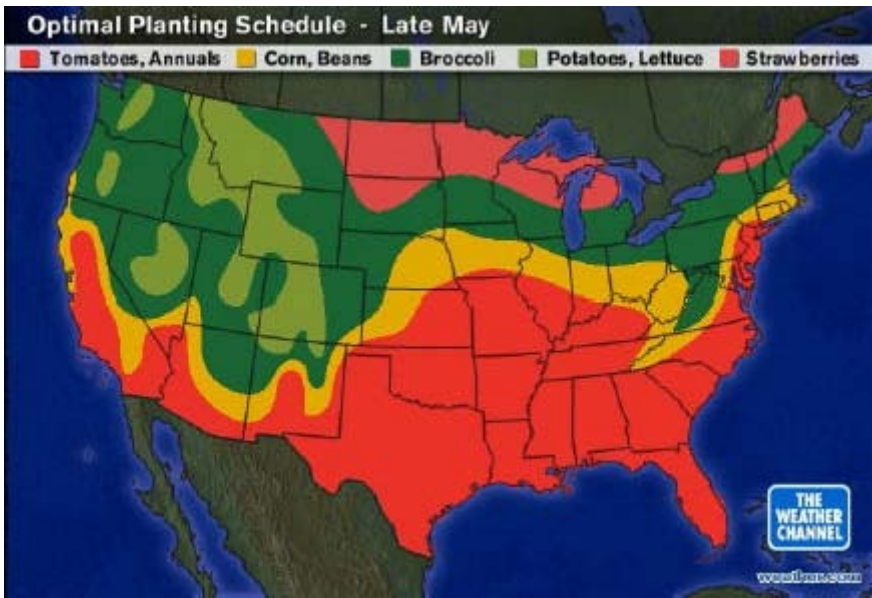
Diagrammatic reasoning



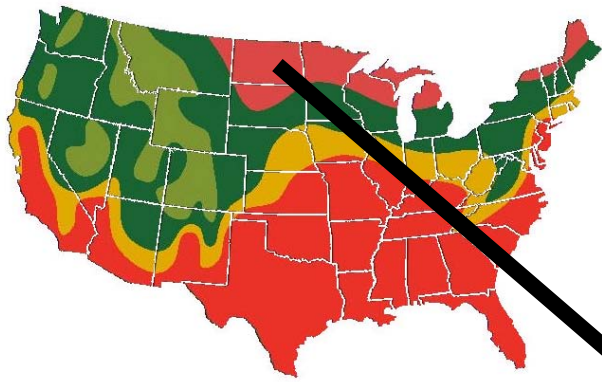
[Michael Anderson, SG00+01]

Motivation

- Information is often encoded in diagrams
 - Very direct and understandable
 - Very compact
- Problem: how to extract this information or reason with it
- Solution: computing with diagrams
- Definition of a query language for diagrams



Which states had Regions optimal for planting strawberries in late May?



Which states had Regions optimal for planting strawberries in late May?

Diagrammatic SQL

- Variation of SQL for diagrams
- Example:

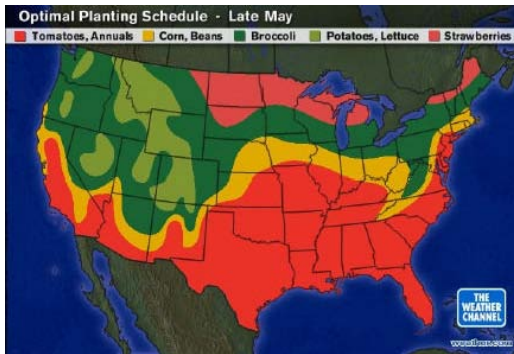
```
DSELECT    state
FROM      us
WHERE     crop = strawberries
AND       season = late_may
```

Which states had
Regions optimal
for planting straw-
berries in late May?

Logical operations (slightly simplified)

- **OR** returns the *maximum* of each pair of tiles
- **AND** returns the *minimum* of each pair of tiles
- **OVERLAY** returns the *sum* of each pair of tiles
- **PEEL** returns the *difference* of each pair of tiles
- **NOT** is a one place operator that returns the value of $-d$
- **NULL** is TRUE if $d =$ the *null diagram*, else FALSE.
- **ACCUMULATE** the accumulation of the results of successively applying of o to each diagram in ds .
- **MAP** application of f to each diagram in ds_1, \dots, ds_n .
- **FILTER** all diagrams in ds for which f returns TRUE.

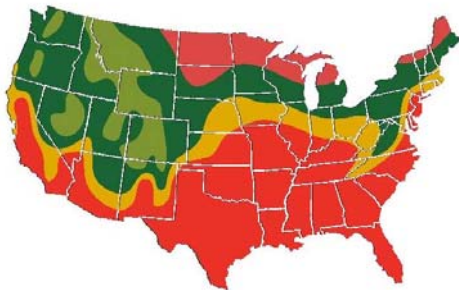
Computation of the result



AND



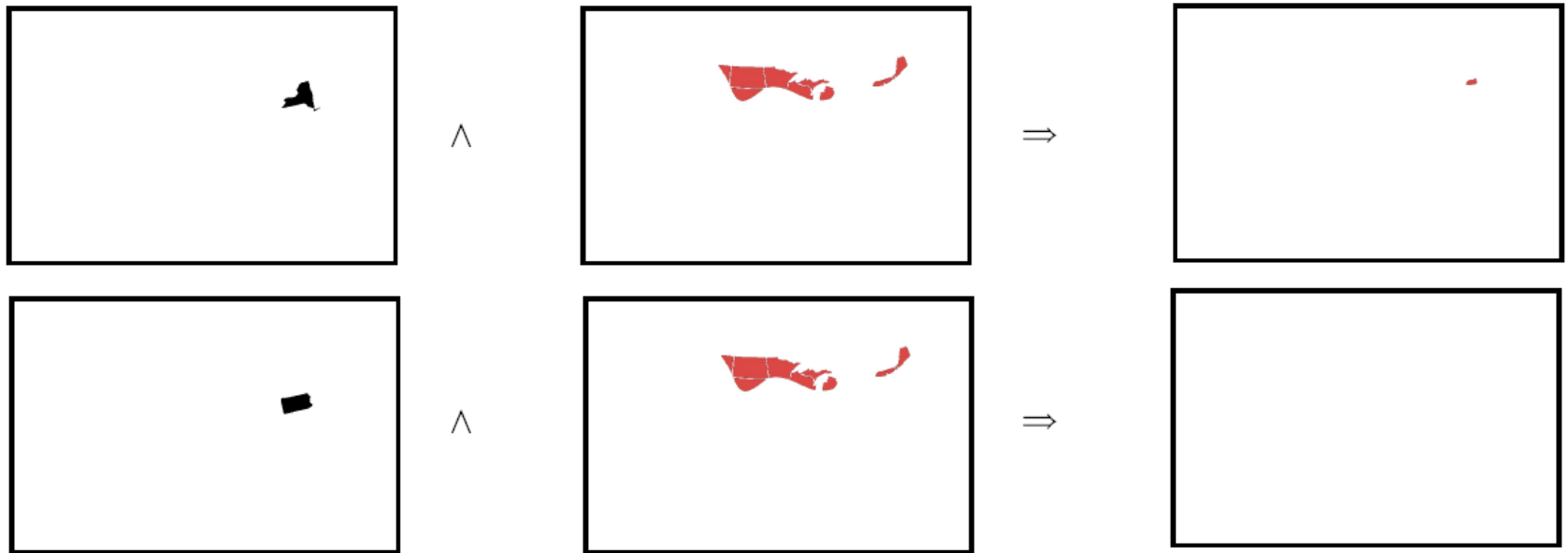
=



FILTER (red1)

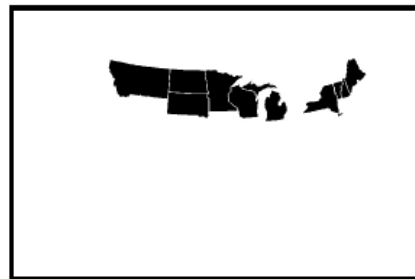


Computation of the result

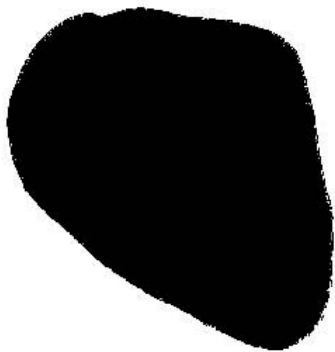


[ACCUMULATE State diagrams that give NONNULL result]

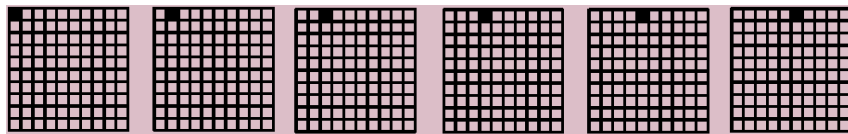
⋮



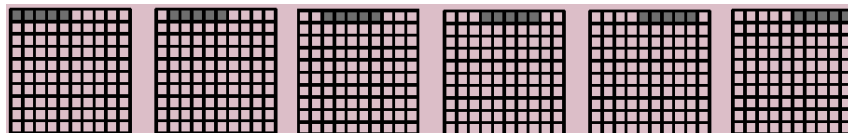
Diasketches



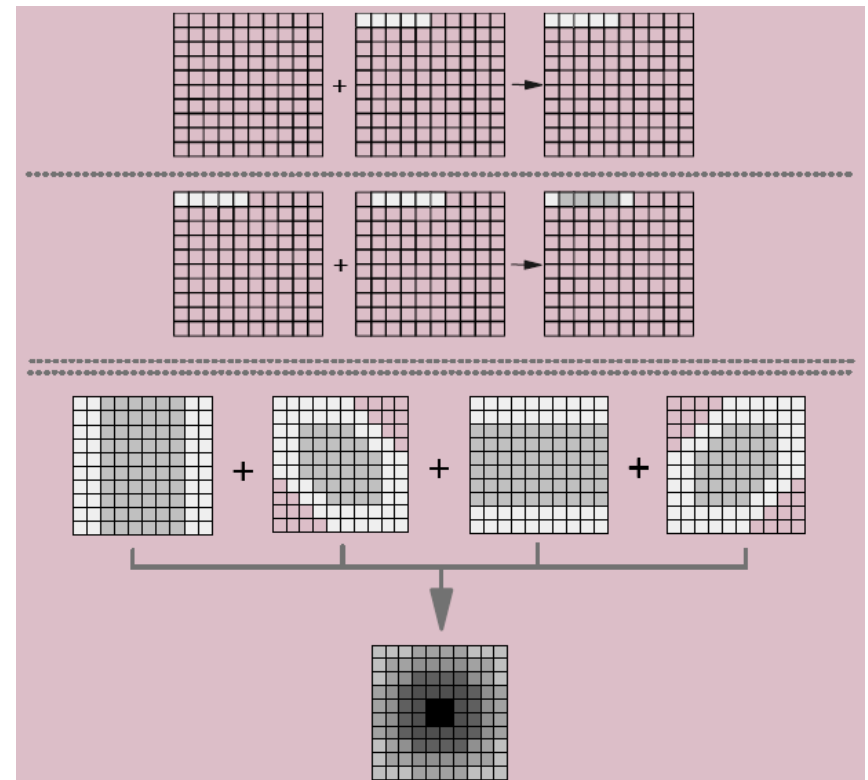
Diagrammatic reasoning for battleship game



Inspection Diagrams



Ship Position Diagrams



Diagrammatic reasoning for battleship game

