Micro Perceptual Human Computation for Visual Tasks

ACM Transactions on Graphics

Yotam Gingold
Columbia/Rutgers

Ariel Shamir
Herzliya IDC

Daniel Cohen-Or
Tel-Aviv University

Historical Digression

[David Alan Grier 2005]

1700's

Historical Digression

[David Alan Grier 2005]

1700's

Clairaut

Historical Digression

[David Alan Grier 2005]

1700's

Clairaut

Halley's Comet

Historical Digression

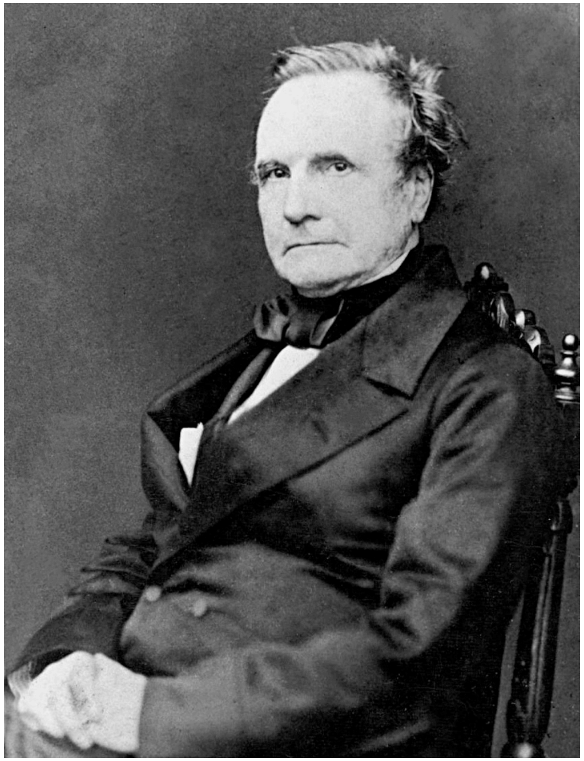
[David Alan Grier 2005]

1800's

Historical Digression

[David Alan Grier 2005]

1800's

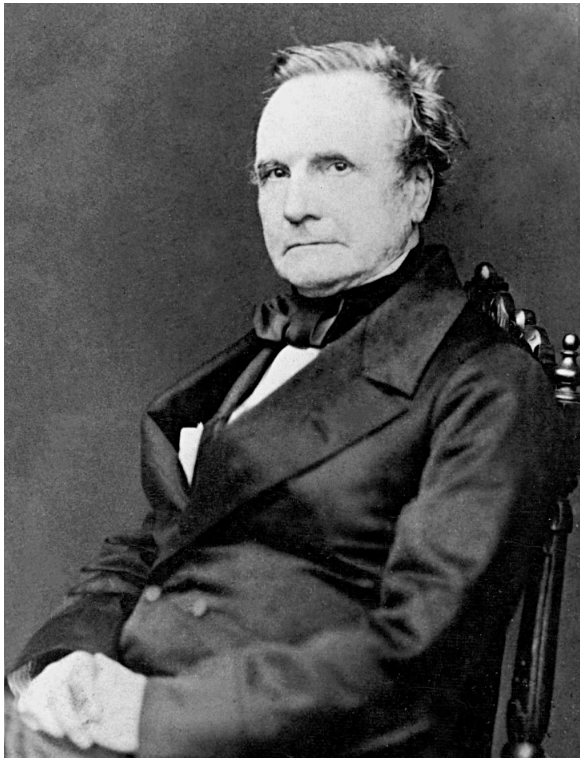


Babbage

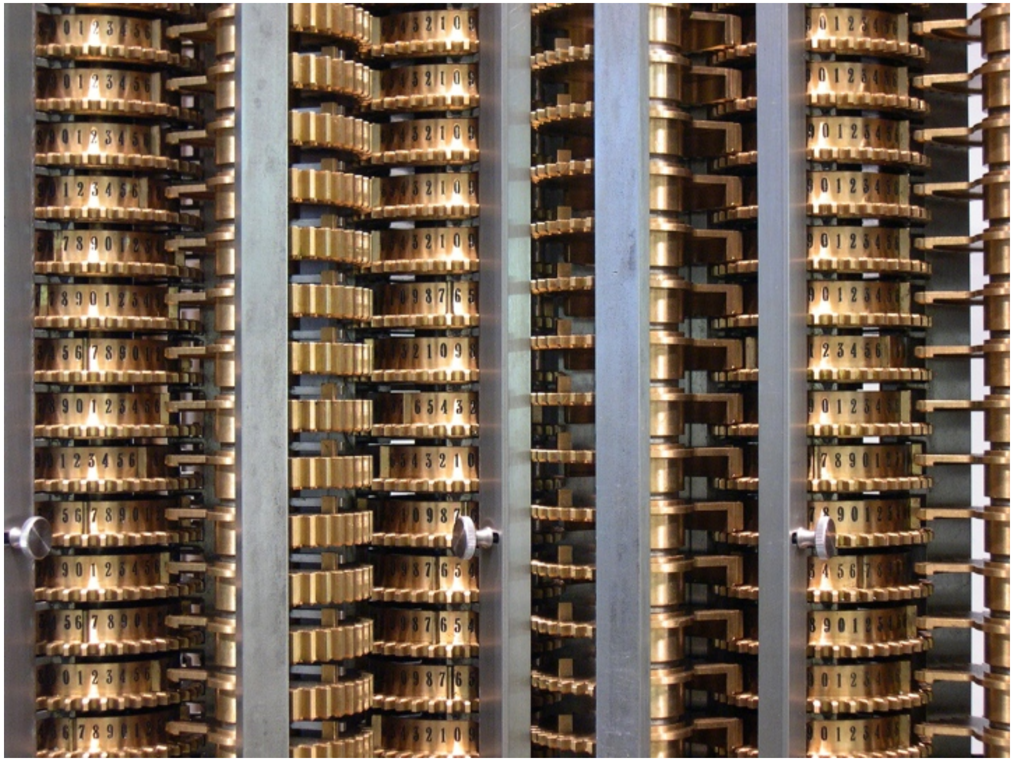
Historical Digression

[David Alan Grier 2005]

1800's



Babbage



[Carsten Ullrich]

Difference Engine

Historical Digression

[David Alan Grier 2005]

1900's

Historical Digression

[David Alan Grier 2005]

1900's

WPA/war effort/NACA

Historical Digression

[David Alan Grier 2005]

1900's

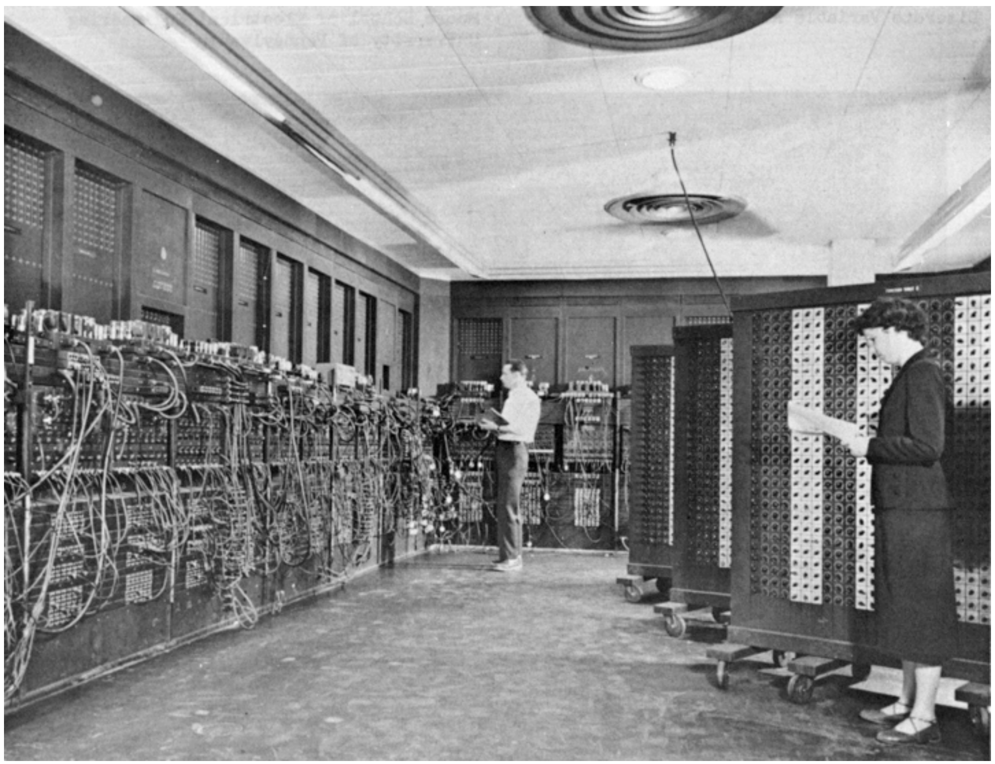
WPA/war effort/NACA

Trinity

Historical Digression

[David Alan Grier 2005]

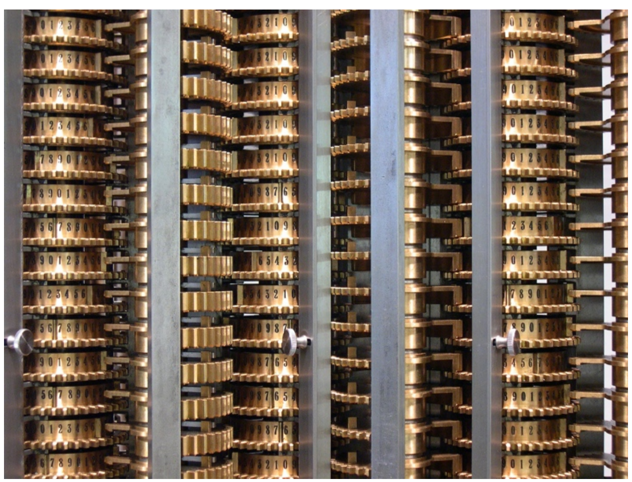
1900's



ENIAC

Electronic

- Fast
- Deterministic
- Arithmetic



[Carsten Ullrich]

Human

- Slow
- Inconsistent & noisy
- ???

The Human Advantage

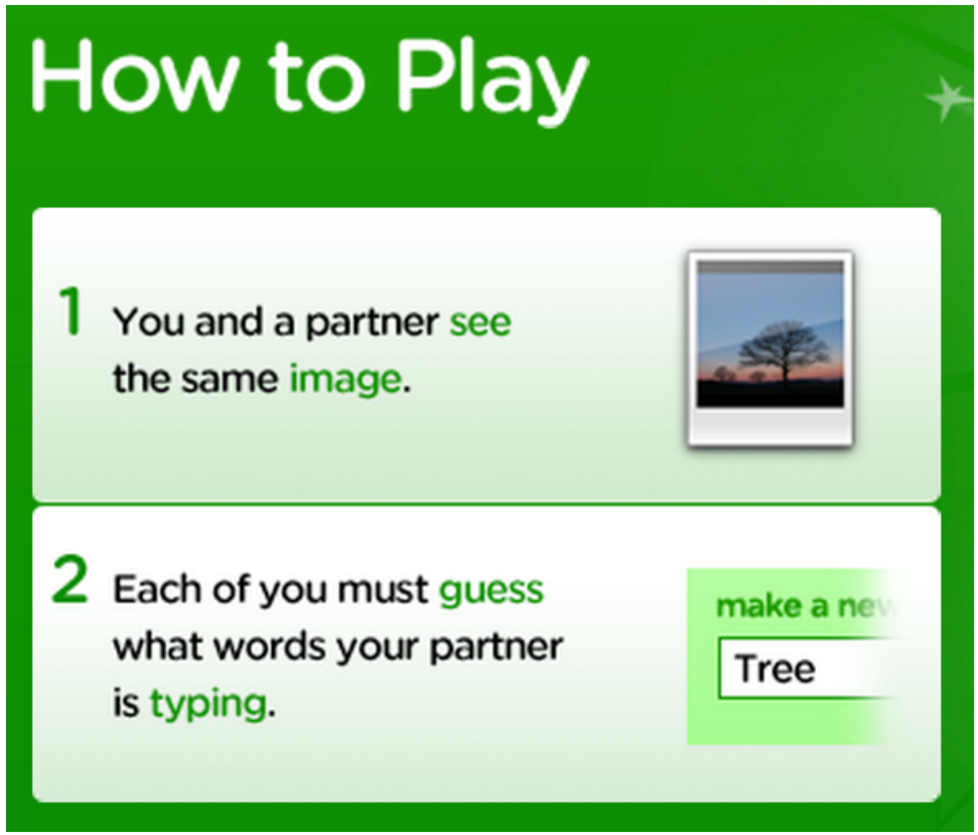
- Perception
- Preference
- Creativity

Human Computation

- Luis von Ahn’s 2005 PhD thesis:
  - “We treat human brains as processors in a distributed system, each performing a small part of a massive computation.”
  - “We argue that humans provide a viable, under-tapped resource that can aid in the solution of several important problems in practice.”

Example 1:

ESP Game
Concentrate...

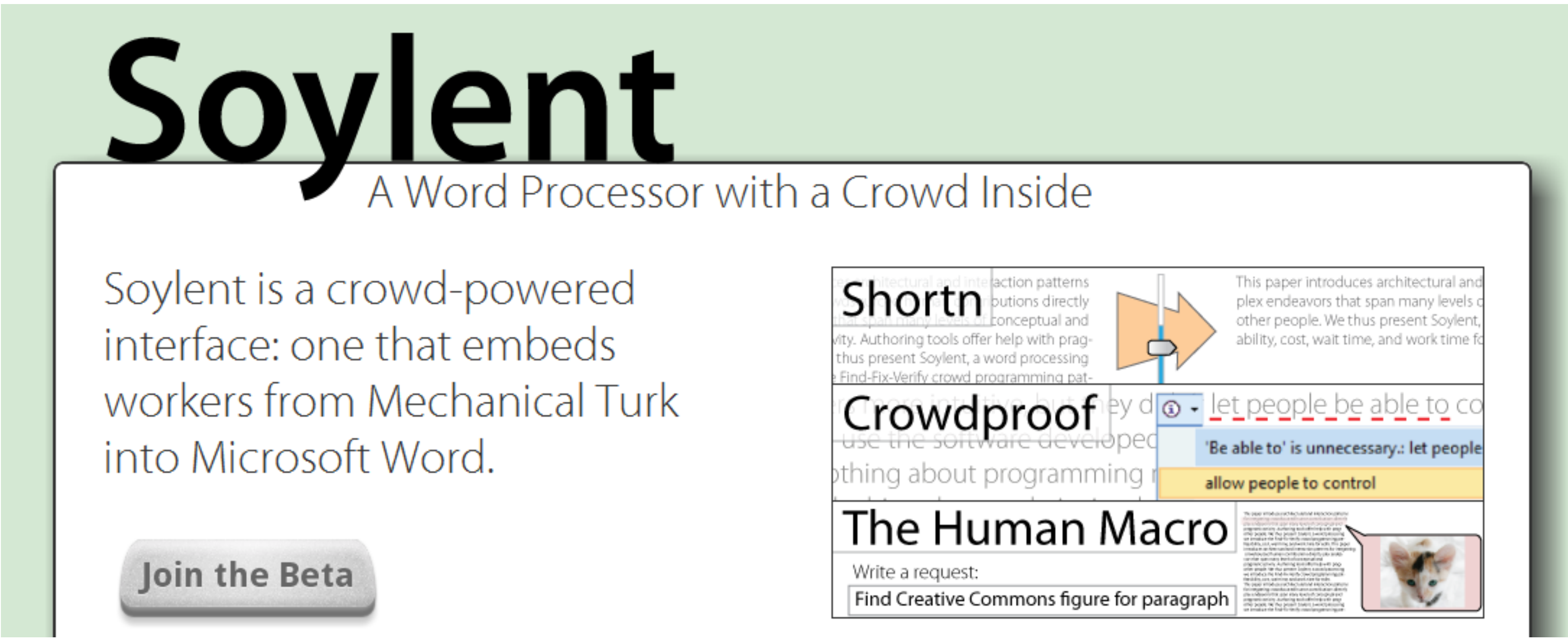


[von Ahn and Dabbish 2004]

Example 2: LabelMe

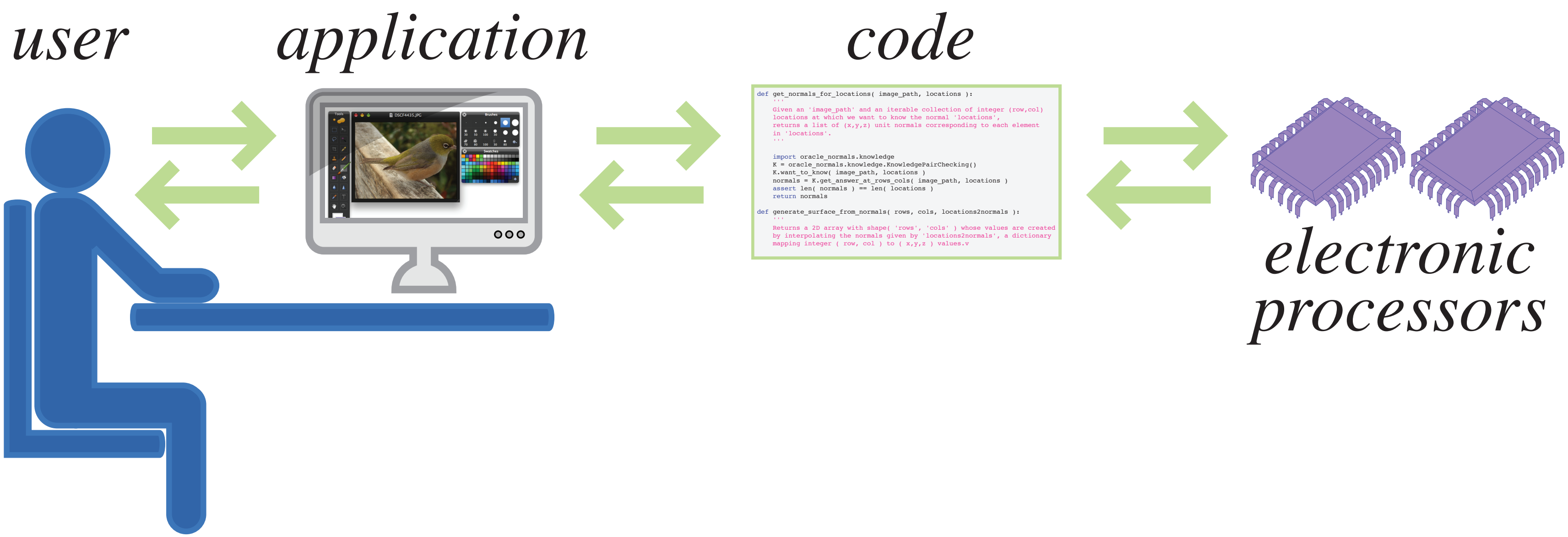
[Russel et al. 2005/2008]

Example 3:

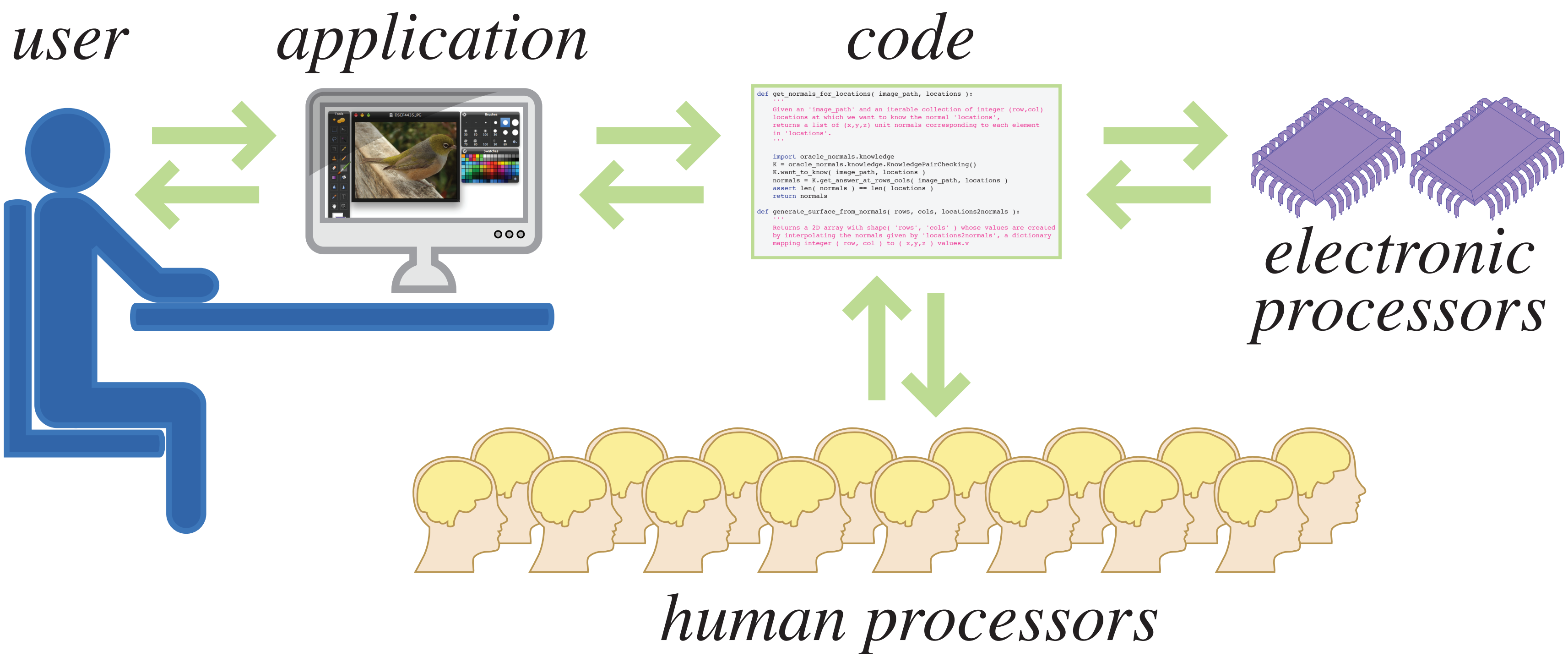


[Bernstein et al. 2010]

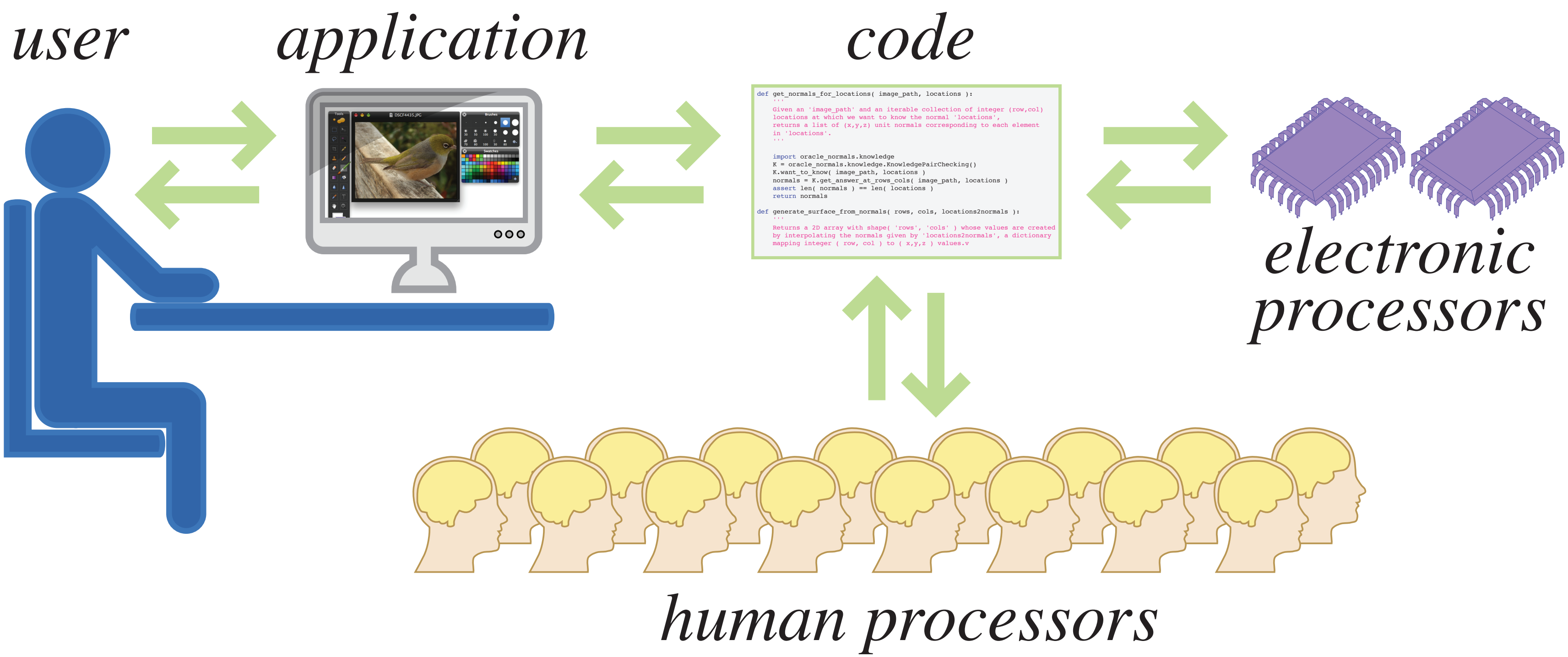
Computation



Human Computation



Why?

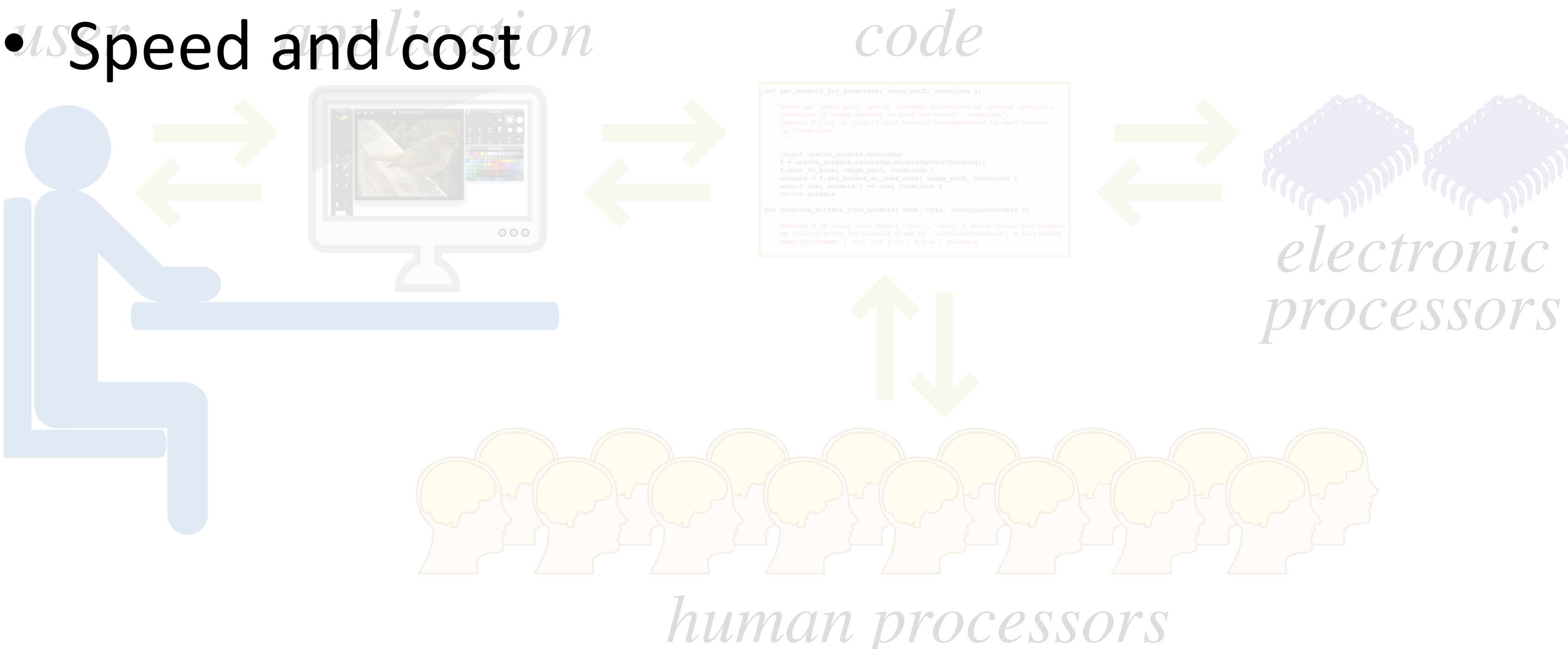

user
application
code
def get_normals_for_locations(image_path, locations):
'''
Given an 'image_path' and an iterable collection of integer (row,col)
locations at which we want to know the normal 'locations',
returns a list of (x,y,z) unit normals corresponding to each element
in 'locations'.
'''
import oracle_normals.knowledge
K = oracle_normals.knowledge.KnowledgePairChecking()
K.want_to_know(image_path, locations)
normals = K.get_answer_at_rows_cols(image_path, locations)
assert len(normals) == len(locations)
return normals
def generate_surface_from_normals(rows, cols, locations2normals):
'''
Returns a 2D array with shape('rows', 'cols') whose values are created
by interpolating the normals given by 'locations2normals', a dictionary
mapping integer (row, col) to (x,y,z) values.v
electronic processors
human processors

Why?

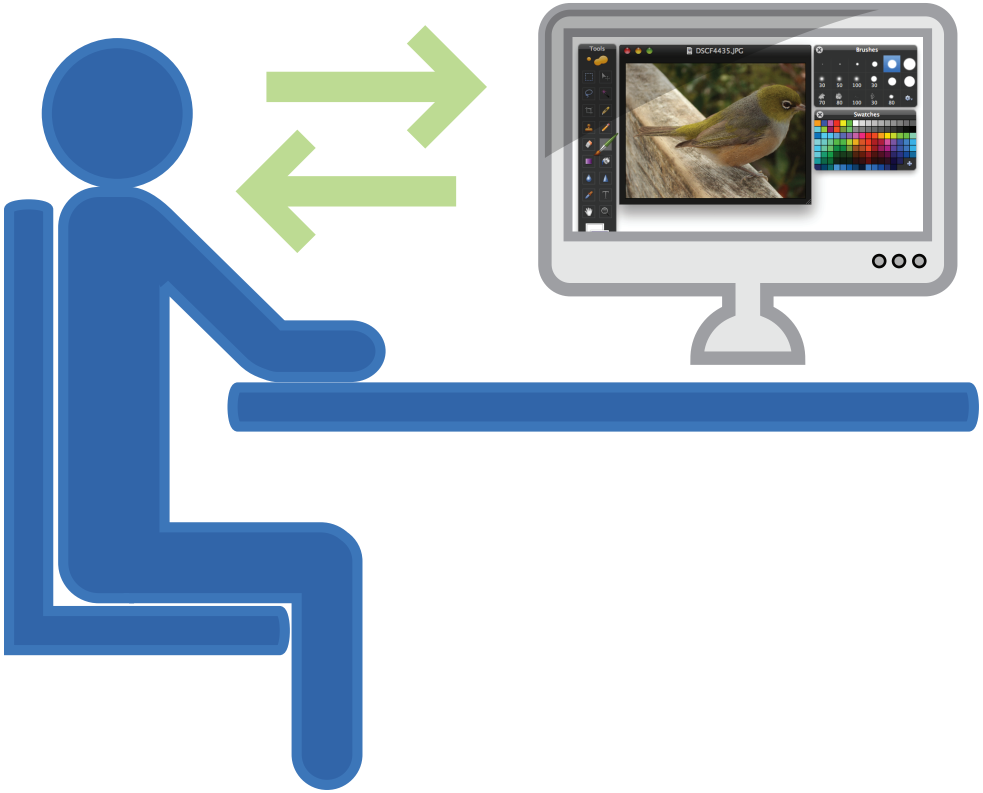
- Make the impossible possible

Why?

- Make the impossible possible
- Speed and cost

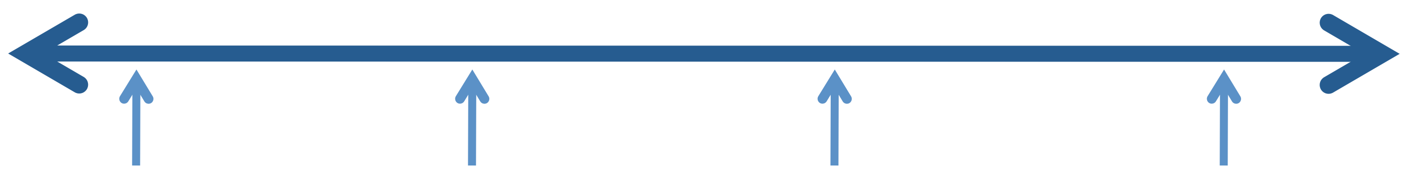


Humans using Computers



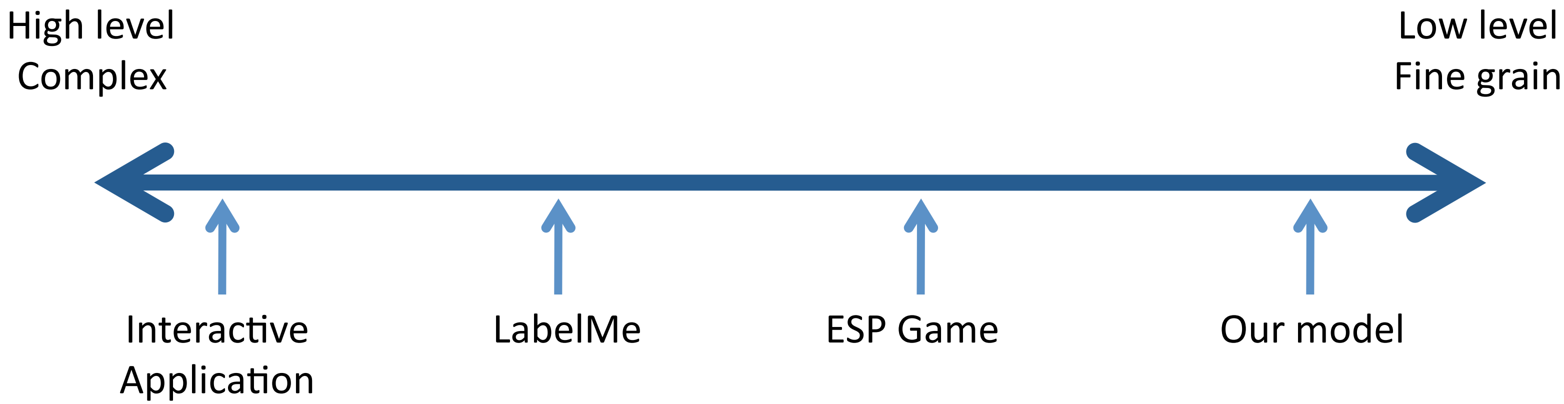
Range of Solutions

- How much human and how much computer is involved?

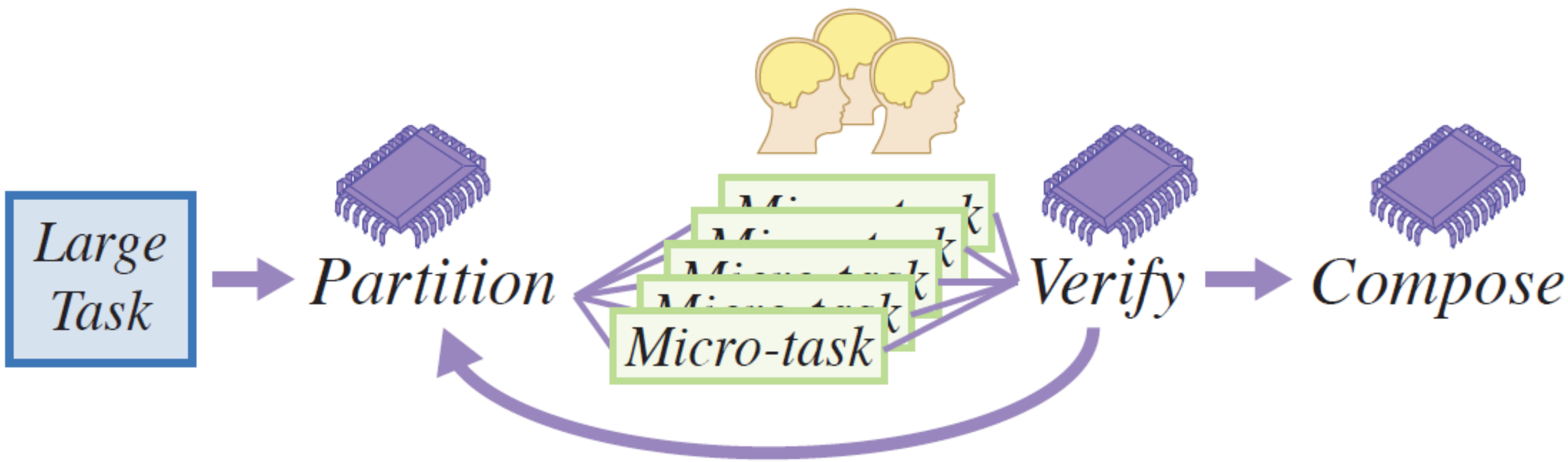


Type of Human Cycles

- You can also think of the type of activity the human does.



Algorithm Design Pattern



The Question We Ask

- What is the minimum amount of information a human could provide in order to solve the original problem?

- Rephrase the algorithm in terms of the smallest piece of information that without it the problem could not be solved.

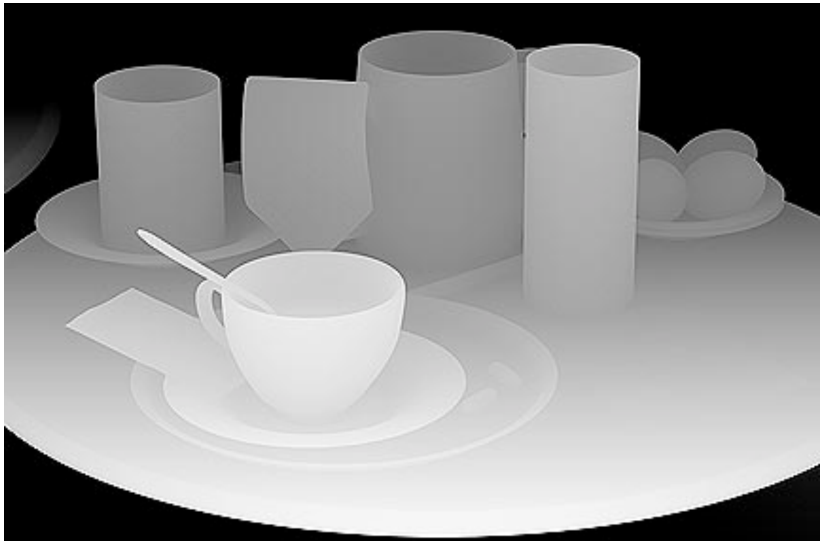
Three Example Algorithms

- Given an image, create
  - depth layers
  - a normal map
  - a bilateral symmetry map

Issues

- Motivation:
  - Money: Amazon's Mechanical Turk
  - Fun: Games with a Purpose (GWAP)
- Efficiency
- Quality Control:
  - Duplication
  - Sentinel Operations
  - Self-Refereeing

Algorithm 1: Depth Layers



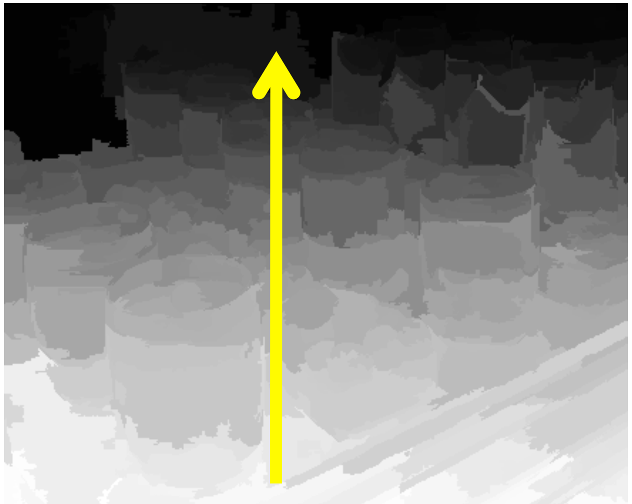
Calculate Depth of a Given Image?

Calculate Depth of a Given Image?

- Automatic methods:
  - Depth increases in the up direction
  - Color similarity implies depth similarity

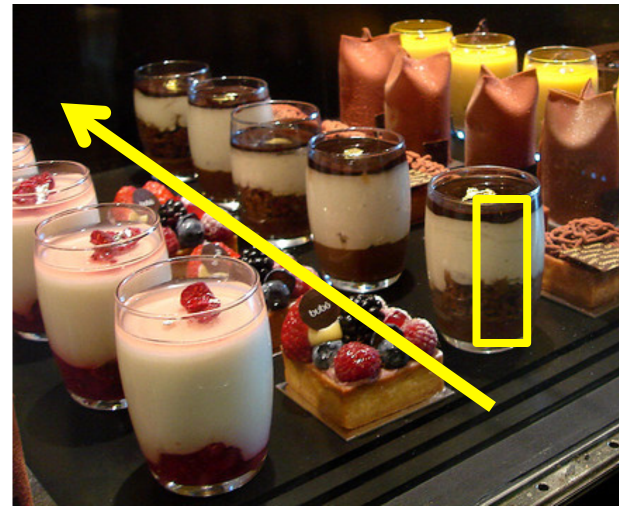
Calculate Depth of a Given Image?

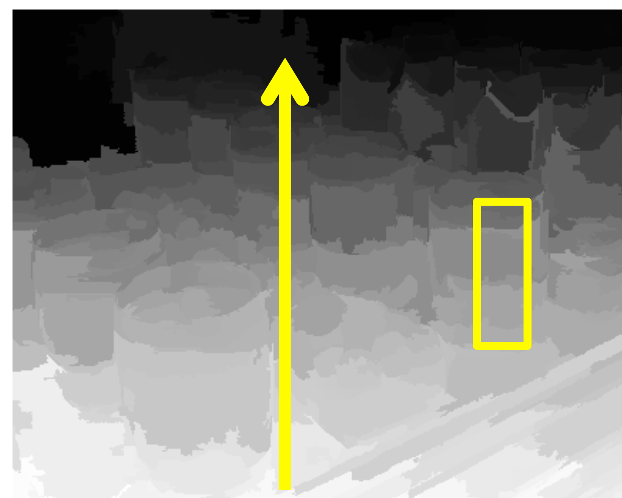
- Automatic methods:
  - Depth increases in the up direction
  - Color similarity implies depth similarity
- Not always correct



Calculate Depth of a Given Image?

- Automatic methods:
  - Depth increases in the up direction
  - Color similarity implies depth similarity
- Not always correct





Calculate Depth of a Given Image?

- Automatic methods:
  - Depth increases in the up direction
  - Color similarity implies depth similarity
- Not always correct
- Some images are very challenging (art)

[Hiroshige]

Micro Task?

Micro Task?

- Ask “what is the depth of the pixel?”
  - Too fine, can be ambiguous

Micro Task?

- Ask "what is the depth of the pixel?"
  - Too fine, can be ambiguous
- Ask "what is the depth of an object?"
  - Segmentation is too complex

Micro Task?

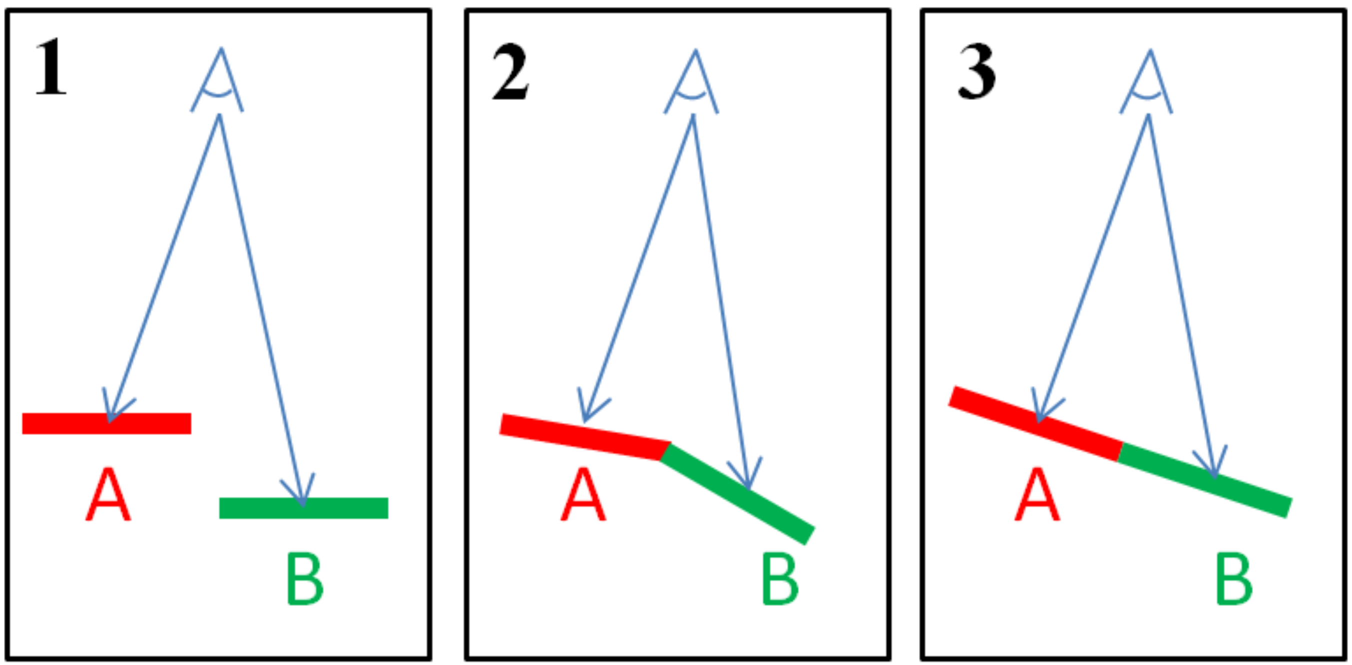
- Ask "what is the depth of the pixel?"
  - Too fine, can be ambiguous
- Ask "what is the depth of an object?"
  - Segmentation is too complex
- Ask "what is the depth of a patch in the image?"
  - Getting better... but humans are not good at assessing absolute depth

Relative Ordering

- Ask "which is closer" on neighboring patches?

Relative Ordering

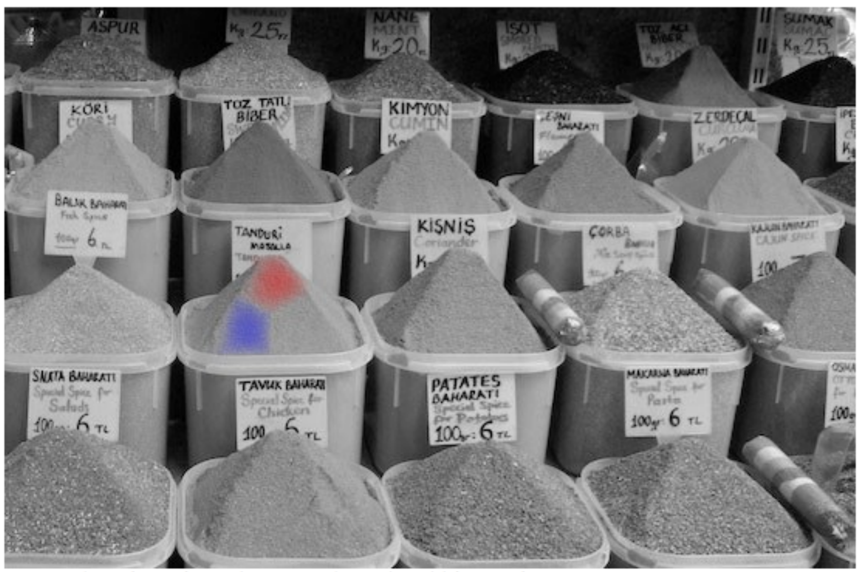
- Ask "which is closer" on neighboring patches?
  - Reliable, but not well-defined. A is closer than B:



Our Micro Task

Is there a jump between the red region and the blue region, in terms of distance from the camera?

Place the mouse over an image to hide the highlighted regions.



◯ **No**, there is no jump between the red and blue regions.
◯ Yes, and the **blue** region is farther from the camera.
◯ Yes, and the **red** region is farther from the camera.

[-] Example

Yes, and the blue region is father from the camera.

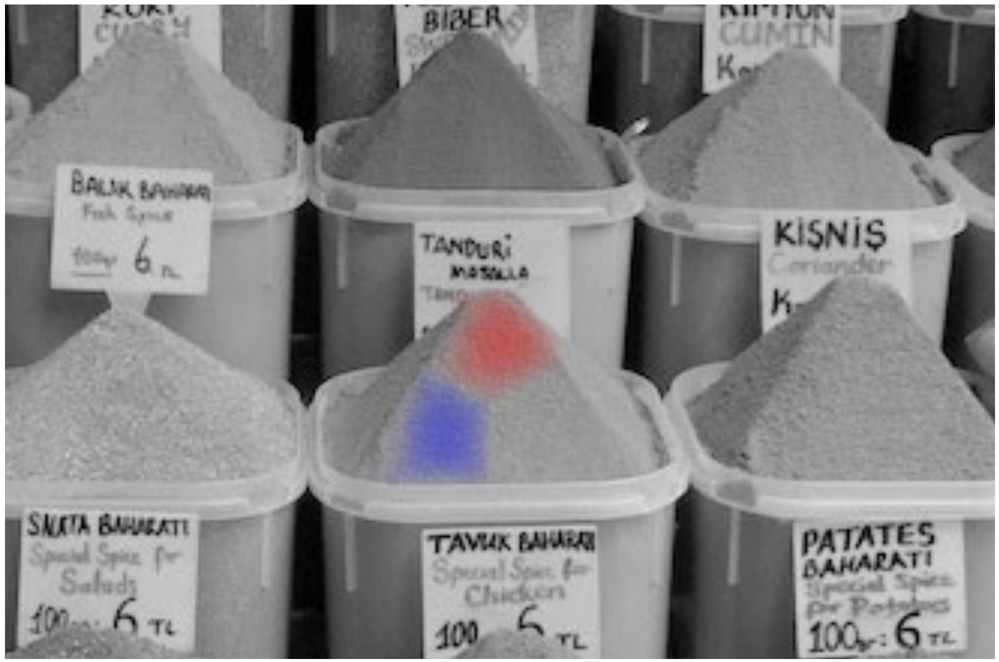
Yes, and the red region is father from the camera.

No, there is no jump between the red region and the blue region, in terms of distance from the camera.

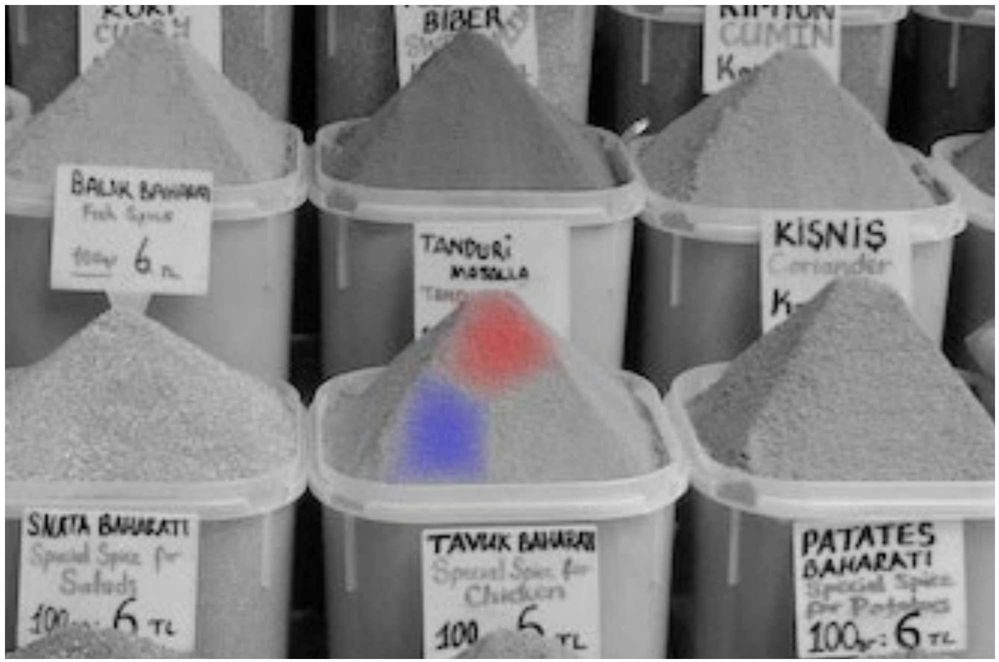
Guidelines for Choosing Tasks

- Task must be simple (instantaneous)
- Task must be specific (well-defined)
- Task must be reliable (humans can do it)

Combining


BIBER
CUMIN
KIŞNIŞ
Coriander
TANDURI
Fish Spice
Special Spice for Salads
Special Spice for Chicken
PATATES
BAHARATI

Combining



- Laplace equation $\Delta f = 0$ with constraints

Algorithm

```
DEPTH-LAYERS(image I, sentinel queries S)
Segment I into regions (using mean-shift and SLIC)
Insert all pairs of neighboring regions into Q
loop in parallel until each pair has been visited N times
    Gather K random pairs from Q
    Gather M random pairs from S
    for each pair: Build the visual query & Duplicate it
    Mix the 2K + 2M queries
    results = send all queries to an HP
    if average(consistent(results)) ≥ 0.75
        for each pair
            Add consistent results to the list of votes
            Increment #visited
for each pair of neighboring regions
    final_result = majority(list of votes)
Solve the Laplace equation to construct a depth map
```

Algorithm

Partition

DEPTH-LAYERS(image I, sentinel queries S)

Segment I into regions (using mean-shift and SLIC)
Insert all pairs of neighboring regions into Q
loop in parallel until each pair has been visited N times
    Gather K random pairs from Q
    Gather M random pairs from S
    for each pair: Build the visual query & Duplicate it
    Mix the $2K + 2M$ queries
    $results$ = send all queries to an HP
    if $average(consistent(results)) \geq 0.75$
        for each pair
            Add consistent results to the list of votes
            Increment #visited
for each pair of neighboring regions
    $final\_result = majority$(list of votes)
Solve the Laplace equation to construct a depth map

Algorithm

Partition

Quality Control Setup

```
DEPTH-LAYERS(image I, sentinel queries S)
Segment I into regions (using mean-shift and SLIC)
Insert all pairs of neighboring regions into Q
loop in parallel until each pair has been visited N times
    Gather K random pairs from Q
    Gather M random pairs from S
    for each pair: Build the visual query & Duplicate it
    Mix the 2K + 2M queries
    results = send all queries to an HP
    if average(consistent(results)) ≥ 0.75
        for each pair
            Add consistent results to the list of votes
            Increment #visited
for each pair of neighboring regions
    final_result = majority(list of votes)
Solve the Laplace equation to construct a depth map
```

Algorithm

DEPTH-LAYERS(image I, sentinel queries S)

Partition

Segment I into regions (using mean-shift and SLIC)
Insert all pairs of neighboring regions into Q
loop in parallel until each pair has been visited N times

Quality Control Setup

Gather K random pairs from Q
Gather M random pairs from S
for each pair: Build the visual query & Duplicate it

Dispatch

Mix the $2K + 2M$ queries
$results$ = send all queries to an HP
if $average(consistent(results)) \geq 0.75$
for each pair
Add consistent results to the list of votes
Increment #visited
for each pair of neighboring regions
$final\_result = majority$(list of votes)
Solve the Laplace equation to construct a depth map

Algorithm

```
DEPTH-LAYERS(image I, sentinel queries S)
Segment I into regions (using mean-shift and SLIC)
Insert all pairs of neighboring regions into Q
loop in parallel until each pair has been visited N times
    Gather K random pairs from Q
    Gather M random pairs from S
    for each pair: Build the visual query & Duplicate it
    Mix the 2K + 2M queries
    results = send all queries to an HP
    if average(consistent(results)) ≥ 0.75
        for each pair
            Add consistent results to the list of votes
            Increment #visited
for each pair of neighboring regions
    final_result = majority(list of votes)
Solve the Laplace equation to construct a depth map
```

Partition

Quality Control Setup

Dispatch

Verify

Algorithm

```
DEPTH-LAYERS(image I, sentinel queries S)
Segment I into regions (using mean-shift and SLIC)
Insert all pairs of neighboring regions into Q
loop in parallel until each pair has been visited N times
    Gather K random pairs from Q
    Gather M random pairs from S
    for each pair: Build the visual query & Duplicate it
    Mix the 2K + 2M queries
    results = send all queries to an HP
    if average(consistent(results)) ≥ 0.75
        for each pair
            Add consistent results to the list of votes
            Increment #visited
for each pair of neighboring regions
    final_result = majority(list of votes)
Solve the Laplace equation to construct a depth map
```

Partition

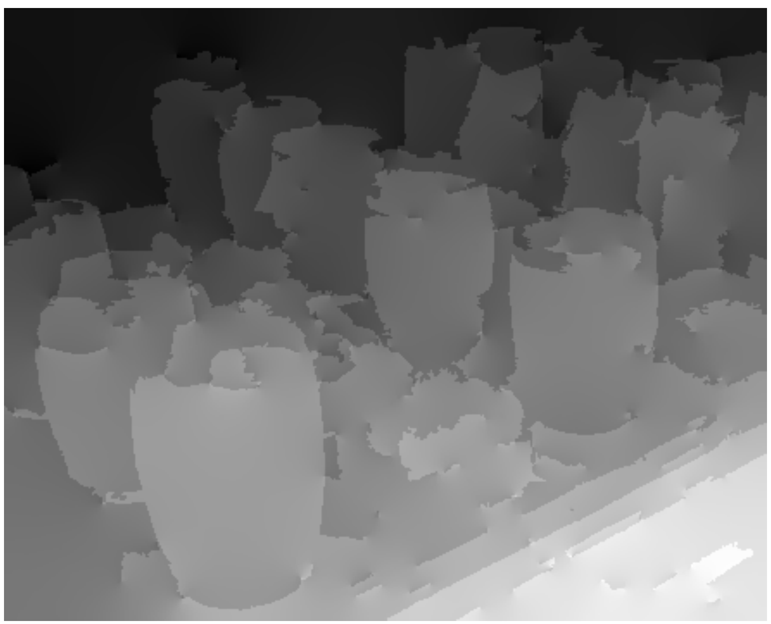
Quality Control Setup

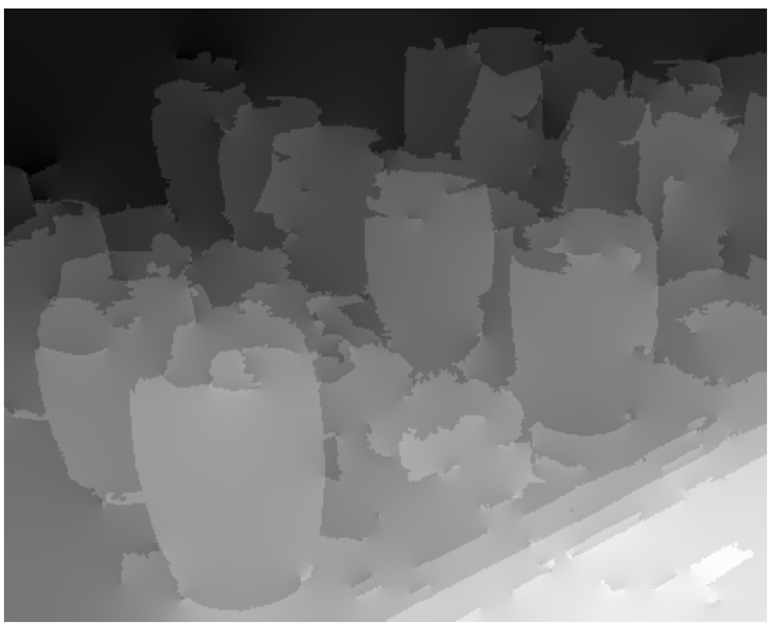
Dispatch

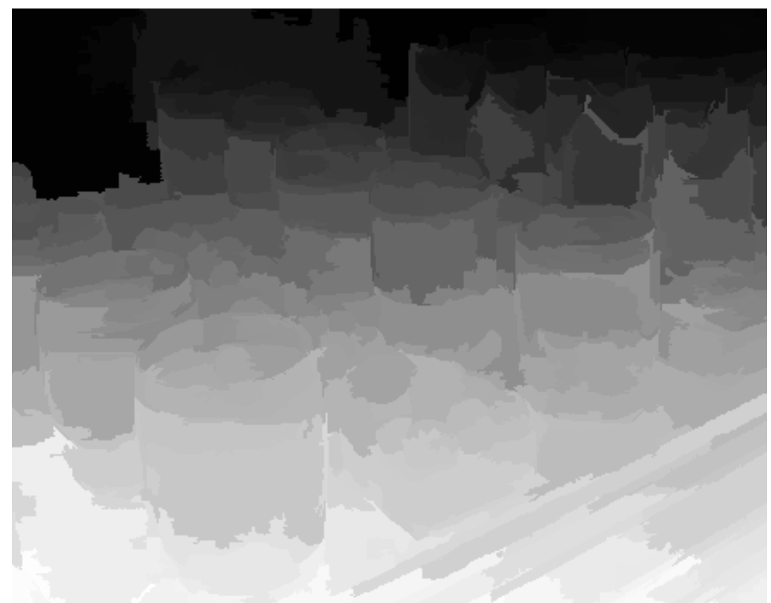
Verify

Combine

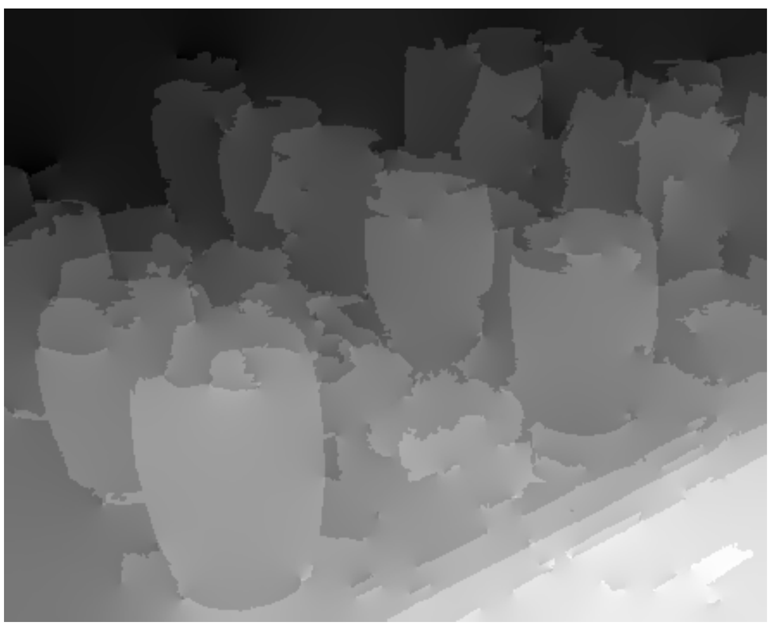

bubó



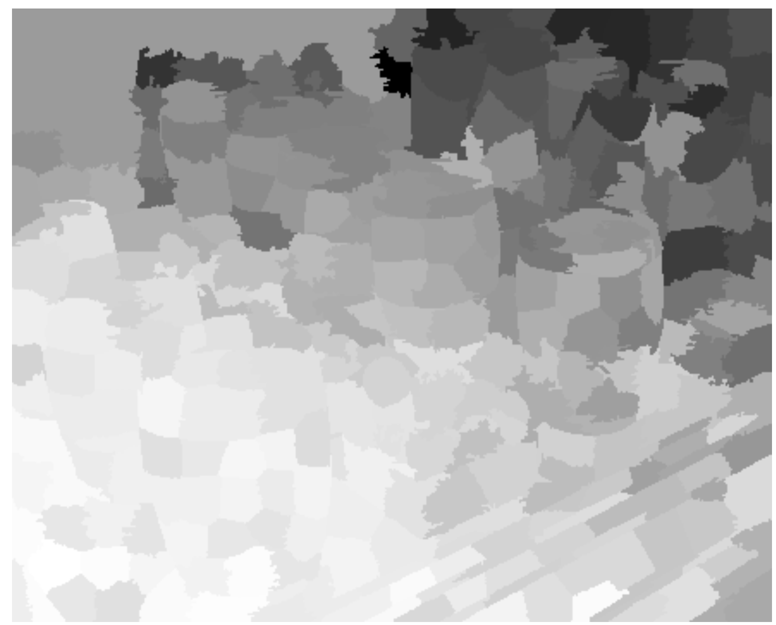




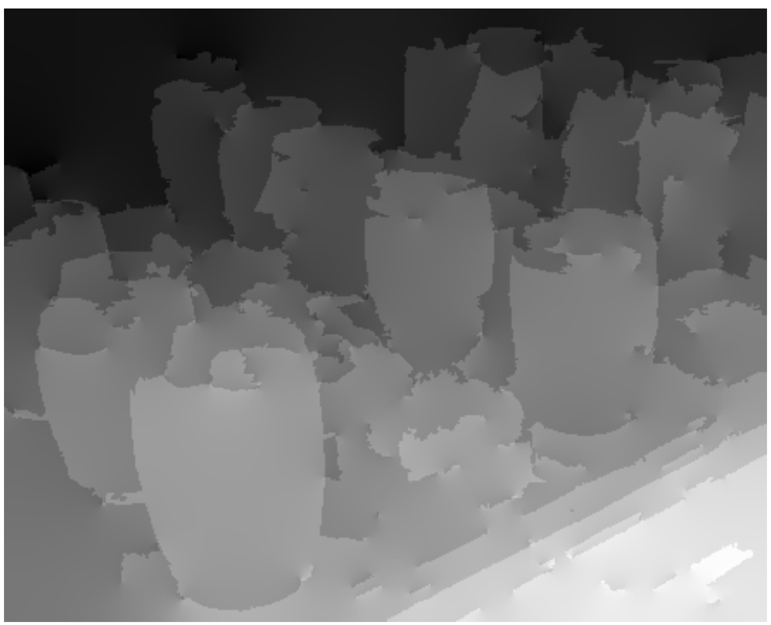
Automatic (Make3D)



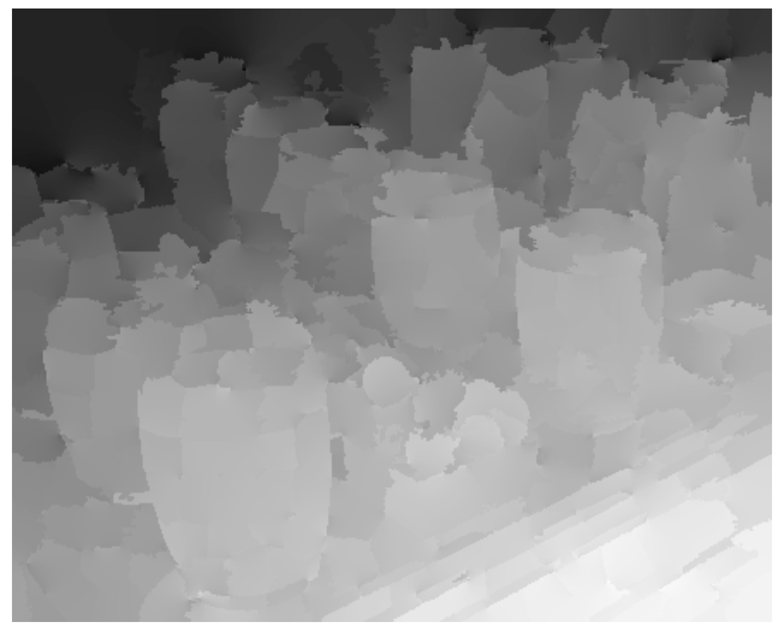
discrete depth



absolute depth

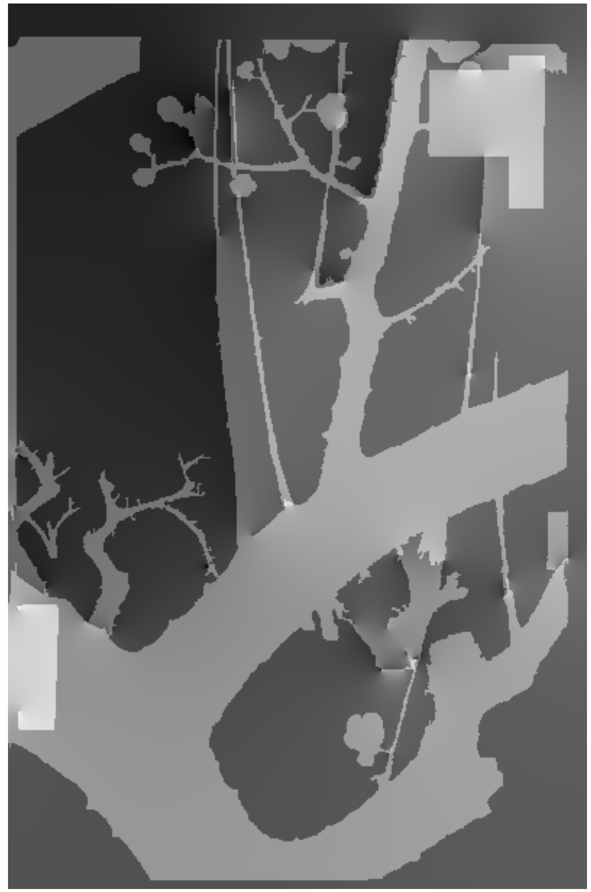


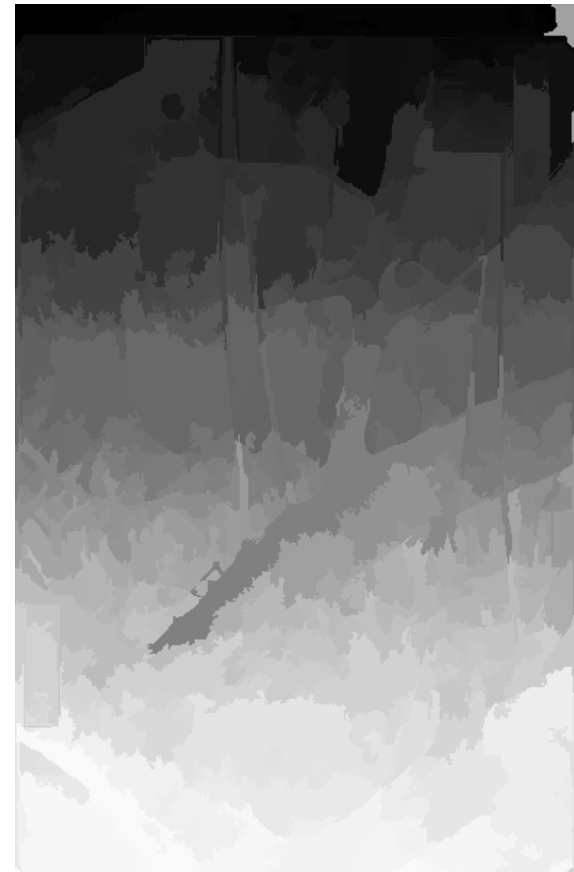
discrete depth



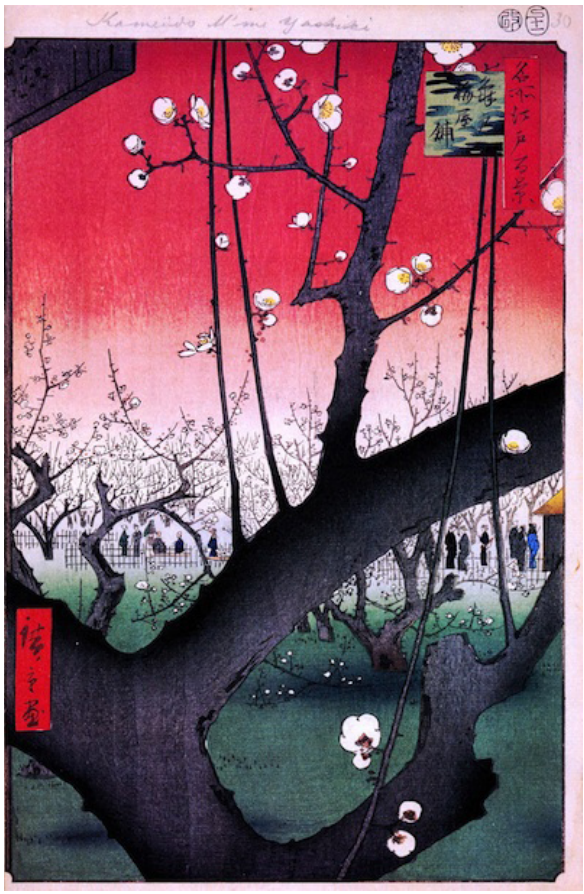
relative depth

[Hiroshige]





Automatic (Make3D)



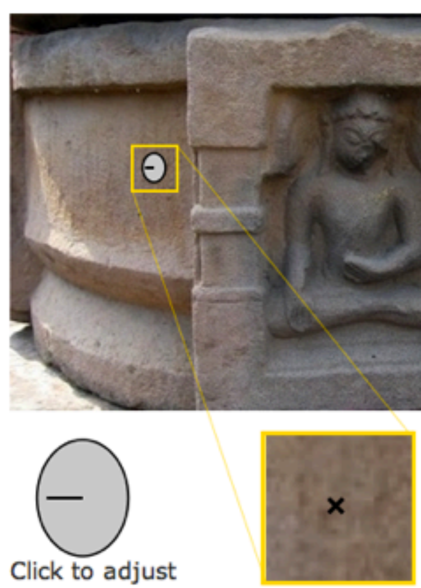
[Hiroshige]

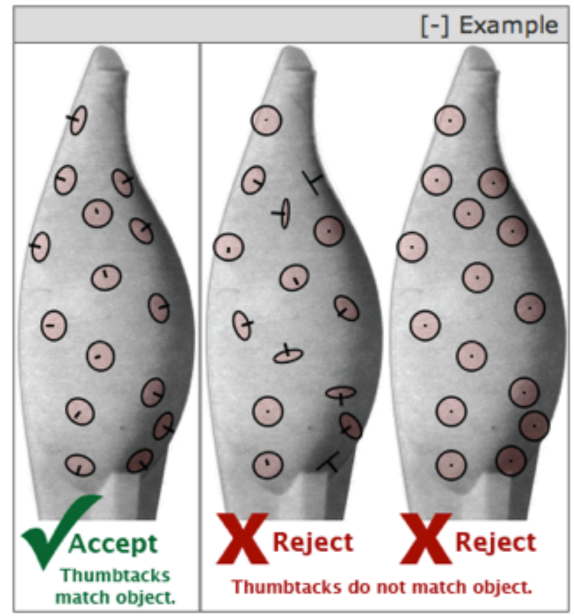
Algorithm 2: Normal Map

Orient the thumbtacks flush against the surface.

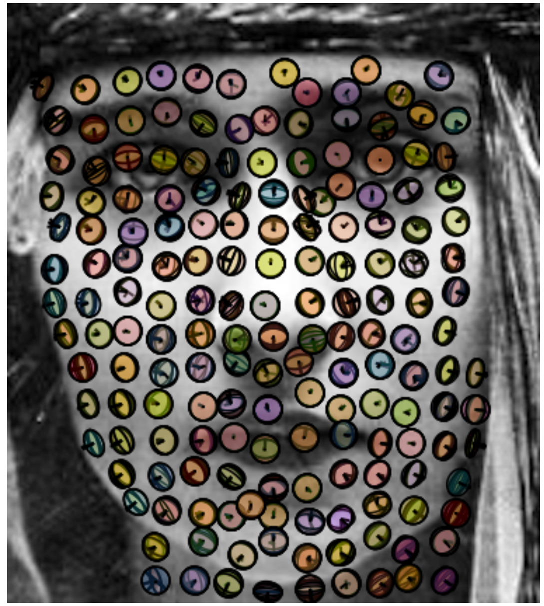
The thumbtack's pin should point away from the surface behind it. See the Example for good and bad examples.

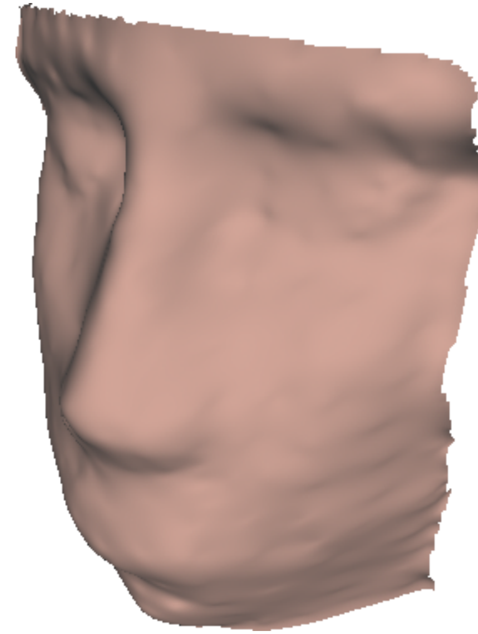
Thumbtacks may appear at the same location multiple times. We check for consistency and may reject inconsistent HITs.

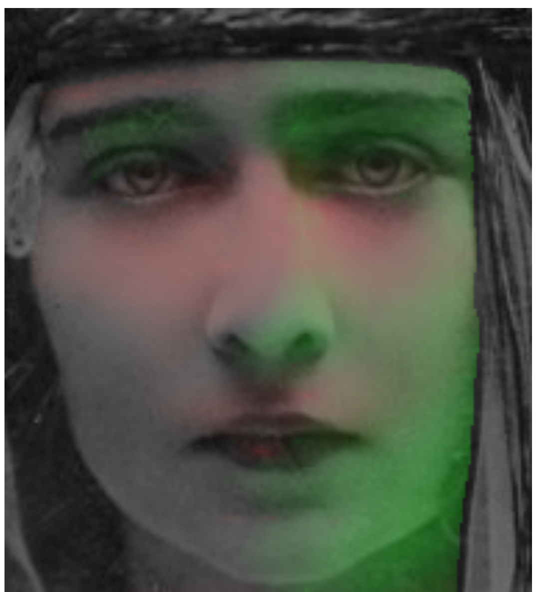


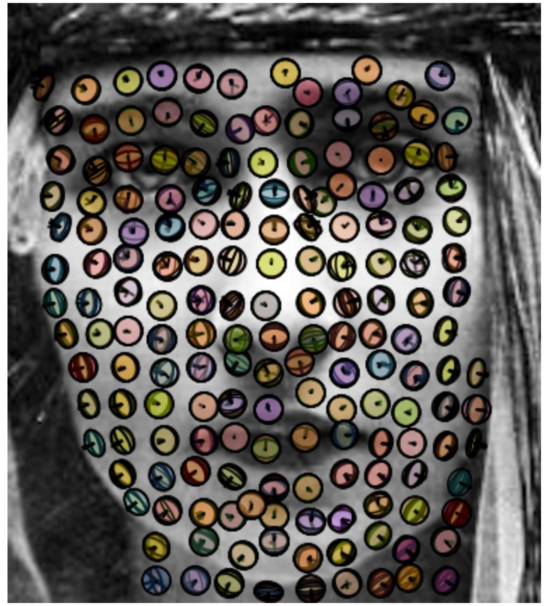


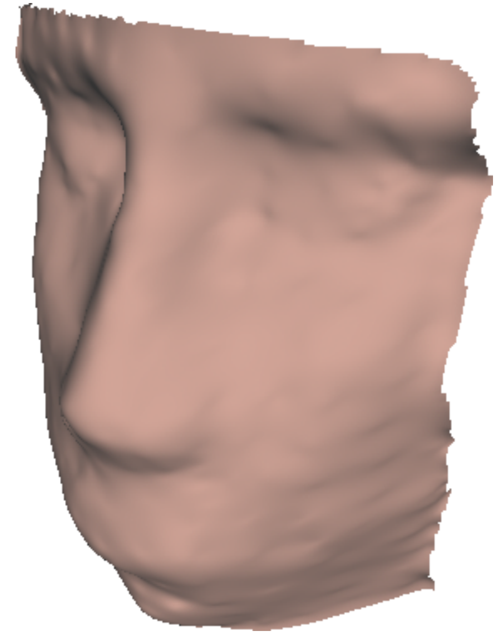
Hide thumbtack

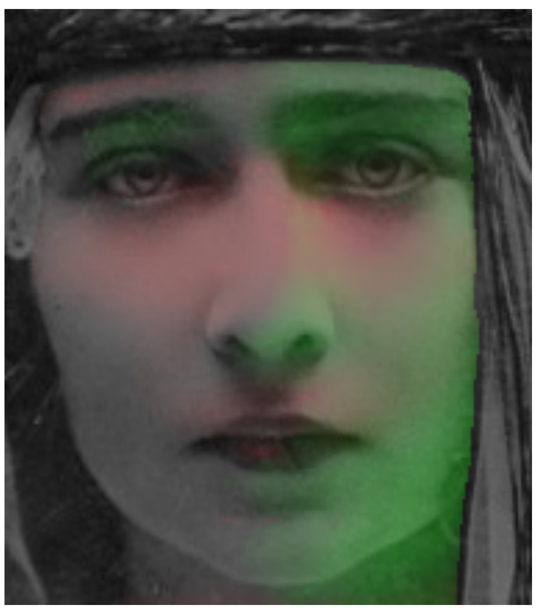


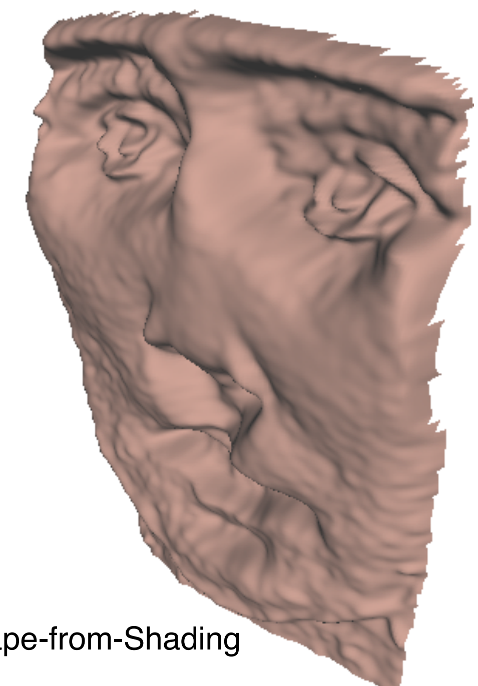






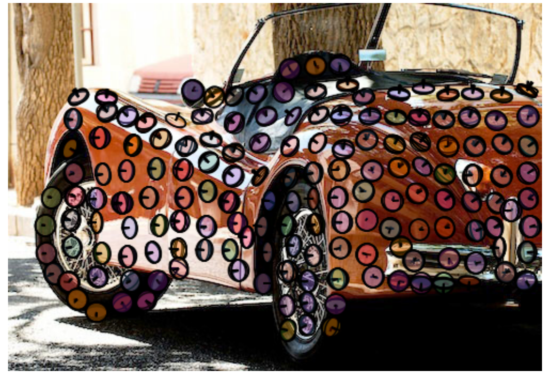




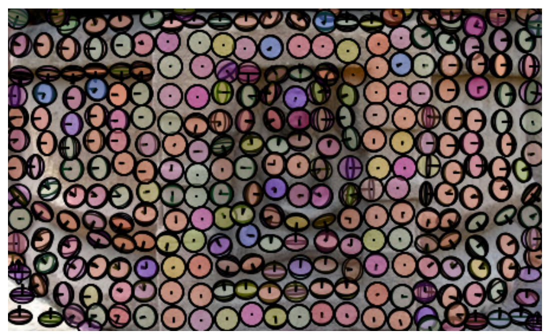


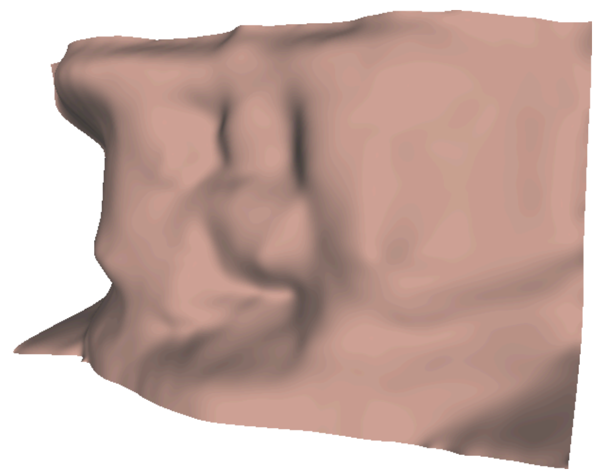
Shape-from-Shading

[Pedro Ribeiro Simões]



[Warren Apel]



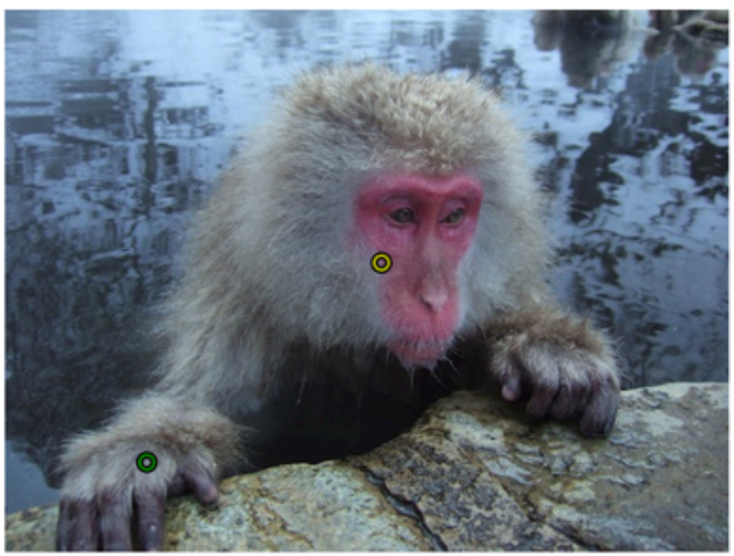


Algorithm 3: Bilateral Symmetry Map

Move the green circle so it is symmetric to the yellow circle.

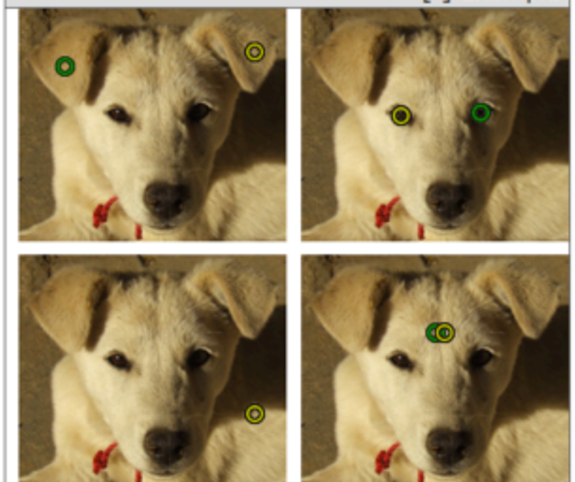
If the yellow circle is over a point on the left side of the body, place the green circle over the same point on the right side. See the Example for good and bad examples.

Dots may appear at the same location multiple times. We check for consistency and may reject inconsistent HITs.



☐ The point symmetric to the yellow circle is not visible in the image.

[-] Example



☑ The point symmetric to the yellow circle is not visible in the image.

Hide circles.

[flickr user dalbera]

Some Statistics

example	micro-tasks used	ratio of used per executed	$ per micro-task	total $ cost
normal map	1620–4340	0.60	.002–.003	$5.04–10.76
depth layers	2669–7620	0.76	.002	$6.41–17.15
symmetry map	1020–1740	0.93	.002	$3.24–3.92

Table 1: *Micro-tasks*

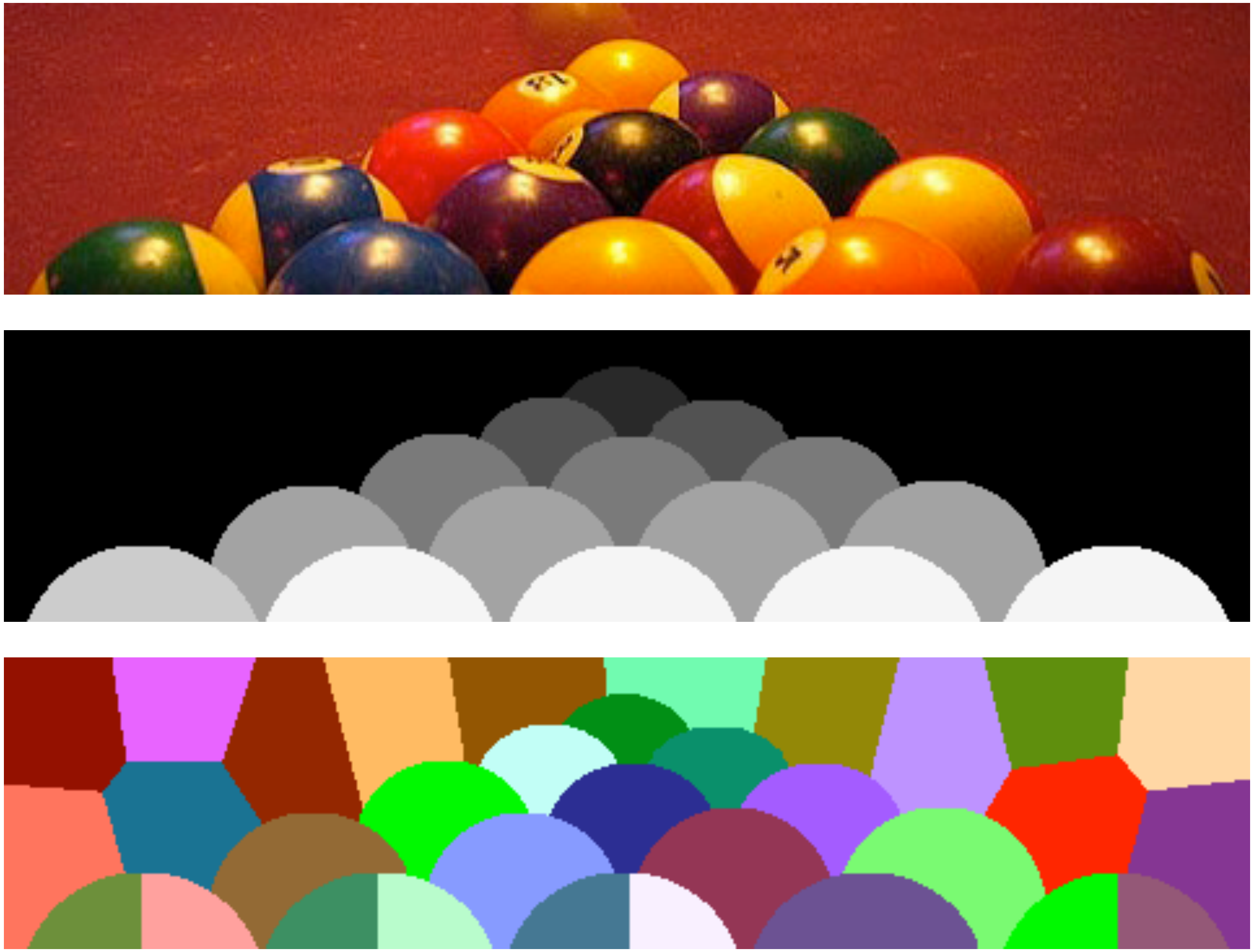
example	total HPs	% completely unreliable	average reliability for reliable HPs	micro-tasks per HP	
				avg	median
normal map	61	42%	89%	123	33
depth layers	48	35%	87%	193	63
symmetry map	19	24%	99%	97	20

Table 2: *Human Processors*

Timing

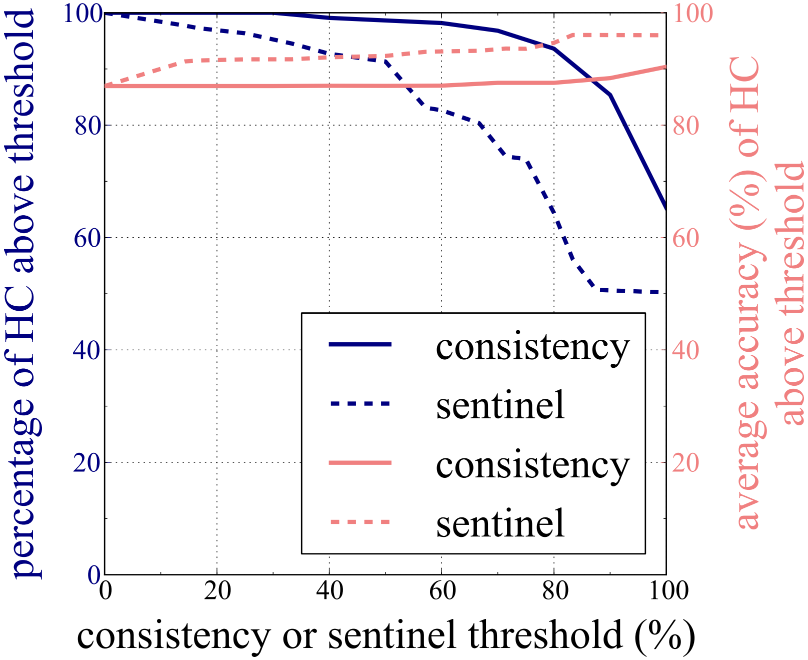
	successful micro-task duration		algorithm delay until % complete	
example	avg	median	50%	100%
normal map	8.8 s	8.1 s	1.1–5.0 hrs	2.8–15.1 hrs
depth layers	6.2 s	5.5 s	0.95–1.6 hrs	3.7–8.0 hrs
symmetry map	9.0 s	8.5 s	0.4–1.6 hrs	0.7–4.9 hrs

Accuracy

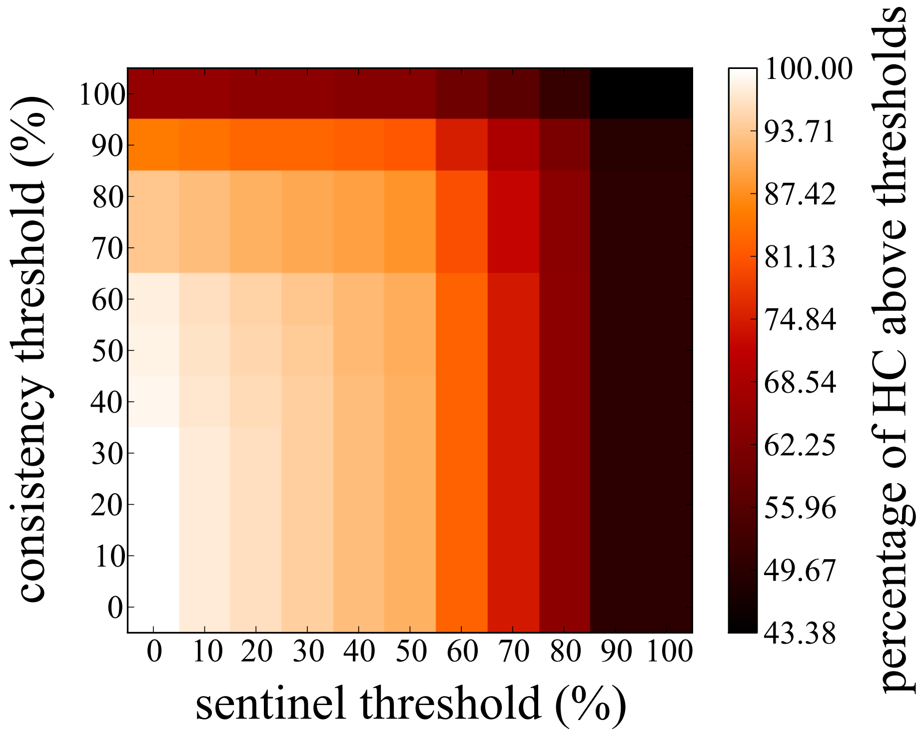


[flickr user fictures]

Accuracy

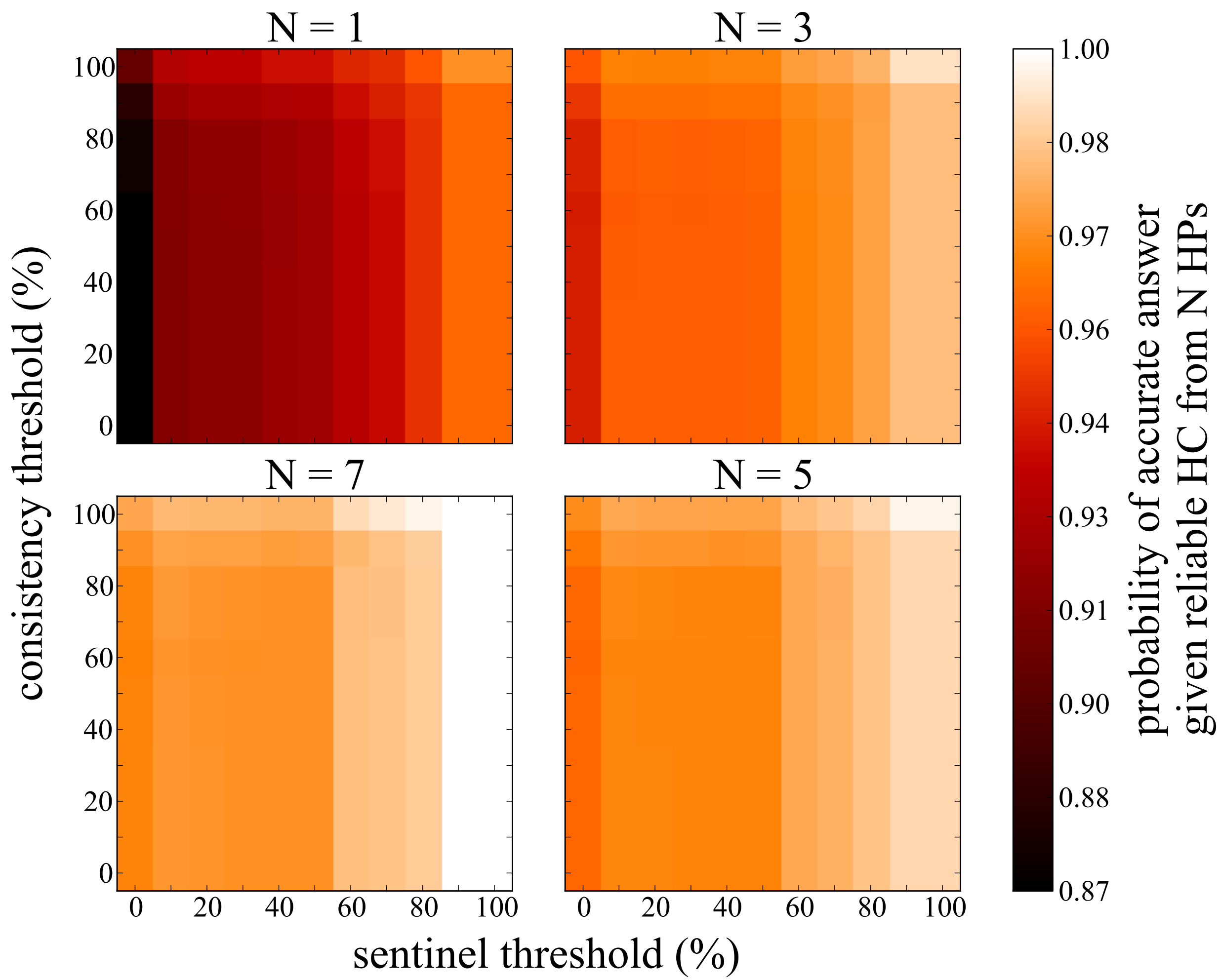


percentage of HC above threshold
average accuracy (%) of HC above threshold
consistency or sentinel threshold (%)
0
20
40
60
80
100
consistency
sentinel
consistency
sentinel



consistency threshold (%)
sentinel threshold (%)
percentage of HC above thresholds
0
10
20
30
40
50
60
70
80
90
100
100.00
93.71
87.42
81.13
74.84
68.54
62.25
55.96
49.67
43.38

Accuracy



N = 1
N = 3
N = 7
N = 5
consistency threshold (%)
0
20
40
60
80
100
sentinel threshold (%)
probability of accurate answer given reliable HC from N HPs
1.00
0.98
0.97
0.96
0.94
0.93
0.91
0.90
0.88
0.87

Conclusions

- For hard problems, HC algorithms can beat automatic algorithms.
- Rephrase your problem in terms of reliable human perception.

- How can we improve efficiency?
- If this were a Photoshop plug-in, how much would people pay to use it?

End

Related Work (1/6)

- Many kinds of collective intelligence
  - open-source software, Wikipedia, PageRank, supervised learning, elections?
- Modern assembly line (Ford Motor Company 1908–1915)
- Interchangeable parts:
  - Adam Smith on division of labor (1776)
  - Terracotta army (3rd century BC)
  - Venetian Arsenal (ship building)

Related Work (2/6)

- Online:
  - [von Ahn 2008]
  - [Little et al. 2010a,b] and [Bernstein 2010]
  - [Bigham et al. 2010] and [Bernstein 2011]
  - [Davis et al. 2010]
  - [Sorokin et al. 2010]
  - many more recent/contemporary applications
- Recast existing experiments
  - [Koenderink et al. 1992], [Cole et al. 2009]
  - [Chen et al. 2009]

Related Work (3/6)

- Training data:
  - ESP Game [von Ahn and Dabbish 2004], …
  - LabelMe [Russel et al. 2008; Yuen et al. 2009]
  - Hands by Hand [Spiro et al. 2010]

- Using HC data gathered offline:
  - [Talton et al. 2009]
  - [Kalogerakis et al. 2010] using [Chen et al. 2009]

Related Work (4/6)

- Depth Layer Algorithm
  - automatic: [Hoiem et al. 2005; Assa and Wolf 2007; Saxena et al. 2009]
  - manual: [Oh et al. 2001; Ventura et al. 2009; Sykora et al. 2010]
- Normal Map Algorithm
  - manual: [Wu et al. 2008]
- Symmetry Map Algorithm
  - automatic: [Chen et al. 2007]

Related Work (5/6)

- History
  - “When Computers Were Human” [Grier 2005]
  - Genetic Algorithms
    - [Sims 1991]
    - Interactive Genetic Algorithm [Takagi 2001]
    - Human-Based Genetic Algorithms [Kosorukoff 2001]
    - Electric Sheep
  - Open Mind Initiative
  - collaborative filtering: [Goldberg et al. 1992; Adomavicius and Tuzhilin 2005]
- “Human Computation” [von Ahn 2005]

Related Work (6/6)

- Recent survey: [Quinn and Bederson 2011]
- Market properties:
  - [Ipeirotis 2010; Chilton et al. 2010; Faridani et al. 2011; Mason and Suri 2011; Mason and Watts 2010]
- Surface perception:
  - [Koenderink et al. 1992; Belheumer et al. 1997; Koenderink et al. 2001]
- Shape-from-Shading:
  - [Durou et al. 2008]

Theoretical Limits

- 125–180 seconds (median) / 20 questions = 6.25–9 seconds per perception for our tasks
- 7 billion humans (does not include other animals capable of similar tasks)
- (number of humans) / (seconds per perception) ~= 1 billion perceptions per second