Micro Perceptual Human Computation for Visual Tasks

Yotam Gingold
George Mason University*

Ariel Shamir
Herzliya IDC

Daniel Cohen-Or
Tel-Aviv University

*Research performed while affiliated with Tel-Aviv University/Herzliya IDC/Rutgers/Columbia.

Computation

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.
```

```python
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.
    '''
    # generate_surface_from_normals(rows, cols, locations2normals)
```

Human Computation

user  application  code  electronic processors  human processors
The Human Advantage

Visual Perception
The Human Advantage

Visual Perception

- What is in this photo?
The Human Advantage

Visual Perception

- What is in this photo?
- Which object is farther away?
The Human Advantage

Visual Perception

- What is in this photo?
- Which object is farther away?
- Is this shape symmetric?
The Human Advantage

Visual Perception

- What is in this photo?
- Which object is farther away?
- Is this shape symmetric?
- What is the surface orientation (normal)?
Range of Solutions

How much human and how much computer is involved?

- More computer “cycles”
- Completely Automatic (no human)
- Human Computation
- Interactive Application
- Completely Manual (only human)
- More human “cycles”
Range of Solutions

How much human and how much computer is involved?

- More computer “cycles”
- More human “cycles”

- Completely Automatic (no human)
- Human Computation
- Interactive Application
- Completely Manual (only human)
Range of Solutions

How much human and how much computer is involved?

- More computer “cycles”
  - Completely Automatic (no human)
  - Human Computation
  - Interactive Application

- More human “cycles”
  - Completely Manual (only human)
Range of Solutions

How much human and how much computer is involved?

- More computer “cycles”
- Completely Automatic (no human)
- Human Computation
- Interactive Application
- Completely Manual (only human)
- More human “cycles”
Range of Solutions

How much human and how much computer is involved?

More computer “cycles”

Completely Automatic (no human)

Human Computation

Interactive Application

Completely Manual (only human)

More human “cycles”
Key Question

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

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.
Key Question

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.

• Use only as much human computation as necessary, and no more than is sufficient.
Type of Human Cycles

- Low level, fine grained
  - Our model
  - ESP Game [von Ahn & Dabbish 2004]

- High level, complex
  - Interactive Application
  - Foldit [Cooper et al. 2010]
  - LabelMe [Russel et al. 2005/2008]
  - Soylent [Bernstein et al. 2010]
Type of Human Cycles

Our model

Low level, fine grained

High level, complex

ESP Game [von Ahn & Dabbish 2004]

Soylent [Bernstein et al. 2010]

LabelMe [Russel et al. 2005/2008]

FoldIt [Cooper et al. 2010]

Interactive Application
Type of Human Cycles

- Low level, fine grained
  - No training
  - ESP Game [von Ahn & Dabbish 2004]
  - Soylent [Bernstein et al. 2010]
  - LabelMe [Russel et al. 2005/2008]
  - Foldit [Cooper et al. 2010]

- High level, complex
  - Interactive Application
Type of Human Cycles

- Low level, fine grained
  - No training
  - No dependency

Our model

High level, complex

Interactive Application

ESP Game [von Ahn & Dabbish 2004]
Soylent [Bernstein et al. 2010]
LabelMe [Russel et al. 2005/2008]
FoldIt [Cooper et al. 2010]
Type of Human Cycles

- No training
- No dependency
- Highly parallel

Low level, fine grained

Our model

High level, complex

ESP Game [von Ahn & Dabbish 2004]
Soylent [Bernstein et al. 2010]
LabelMe [Russel et al. 2005/2008]
Foldit [Cooper et al. 2010]

Interactive Application
Guidelines for Designing Micro-Tasks
Guidelines for Designing Micro-Tasks

Task must be simple (instantaneous)
Guidelines for Designing Micro-Tasks

Task must be simple (instantaneous)
Task must be specific (well-defined)
Guidelines for Designing Micro-Tasks

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

Large Task -> Partition -> Micro-task -> Verify -> Compose
Issues
Motivation
Issues

Motivation  money (via Amazon Mechanical Turk)
Issues

Motivation

money (via Amazon Mechanical Turk)

Efficiency
Issues

Motivation
money (via Amazon Mechanical Turk)

Efficiency
Massive parallelism
Extremely simple visual queries
Issues

Motivation
money (via Amazon Mechanical Turk)

Efficiency
Massive parallelism
Extremely simple visual queries

Quality Control
Issues

Motivation
money (via Amazon Mechanical Turk)

Efficiency
Massive parallelism
Extremely simple visual queries

Quality Control
Batches:

1 2 3 4 5 6

queries
Issues

Motivation
money (via Amazon Mechanical Turk)

Efficiency
Massive parallelism
Extremely simple visual queries

Quality Control
Batches:

1 2 3 4 5 6 1 2 3 4

queries queries with known answers
Issues

Motivation  money (via Amazon Mechanical Turk)

Efficiency  Massive parallelism
            Extremely simple visual queries

Quality Control  Batches:
                 1 2 3 4 5 6 1 2 3 4 1 2 3 4 5 6 1 2 3 4
Issues

Motivation money (via Amazon Mechanical Turk)

Efficiency Massive parallelism
Extremely simple visual queries

Quality Control Batches:

[4 3 2 6 4 1 2 6 2 4 1 3 3 5 1 3 4 1 5 2]
Three Example Algorithms

Given an image, create
Three Example Algorithms

Given an image, create

- depth layers
Three Example Algorithms

Given an image, create

- depth layers
- a normal map
Three Example Algorithms

Given an image, create

- depth layers
- a normal map
- a bilateral symmetry map
Algorithm 1: Depth Layers
Automatic methods

e.g. [Hoiem et al. 2005; Assa and Wolf 2007; Saxena et al. 2009]
Automatic methods

e.g. [Hoiem et al. 2005; Assa and Wolf 2007; Saxena et al. 2009]

Depth increases in the up direction
Automatic methods

e.g. [Hoiem et al. 2005; Assa and Wolf 2007; Saxena et al. 2009]

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

e.g. [Hoiem et al. 2005; Assa and Wolf 2007; Saxena et al. 2009]

Depth increases in the up direction

Color similarity implies depth similarity

Not always correct
Automatic methods

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

Ask “What is the depth of a pixel?”
Micro-Task

Ask “What is the depth of a pixel?”

- Too fine, can be ambiguous
Micro-Task

Ask “What is the depth of a pixel?”
  · Too fine, can be ambiguous

Ask “What is the depth of an object?”
Micro-Task

Ask “What is the depth of a pixel?”
  · Too fine, can be ambiguous

Ask “What is the depth of an object?”
  · Segmentation is too difficult
Micro-Task

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

Ask “What is the depth of an object?”
- Segmentation is too difficult

Ask “What is the depth of a patch in the image?”
Micro-Task

Ask “What is the depth of a pixel?”
  · Too fine, can be ambiguous

Ask “What is the depth of an object?”
  · Segmentation is too difficult

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:

1. ![Diagram 1]
2. ![Diagram 2]
3. ![Diagram 3]
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.

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

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

Patch Segmentation

Laplace equation
\[ \Delta f = 0 \]
with constraints

Micro-tasks (human)
Algorithm

Partition

Patch Segmentation

Micro-tasks (human)

Laplace equation

\[ \Delta f = 0 \]

with constraints
Algorithm

Quality Control Setup

Patch Segmentation

Laplace equation

$\Delta f = 0$

with constraints
Algorithm

Patch Segmentation

Micro-tasks (human)

Laplace equation

\[ \Delta f = 0 \]

with constraints
Algorithm

Patch Segmentation

Dispatch

Micro-tasks (human)

Laplace equation

\[ \Delta f = 0 \]

with constraints
Algorithm

Patch Segmentation

Laplace equation
\[ \Delta f = 0 \]
with constraints
Algorithm

Patch Segmentation

Laplace equation
\[ \Delta f = 0 \]
with constraints

Combine
Algorithm

Patch Segmentation

Laplace equation

$\Delta f = 0$

with constraints
Algorithm

\begin{algorithm}
\textbf{DEPTH-LAYERS}(image }I\text{, sentinel queries }S\text{)
\begin{enumerate}
\item Segment }I\text{ into regions (using mean-shift and SLIC)
\item Insert all pairs of neighboring regions into }Q\text{
\item \textbf{loop in parallel until} each pair has been visited }N\text{ times
\item Gather }K\text{ random pairs from }Q\text{
\item Gather }M\text{ random pairs from }S\text{
\item \textbf{for} each pair: Build the visual query & Duplicate it
\item Mix the }2K + 2M\text{ queries
\item }results = \text{ send all queries to an HP
\item \textbf{if} average\((\text{consistent}(results))\) \geq 0.75
\item \textbf{for} each pair
\item \hspace{1em} Add consistent results to the list of votes
\item \hspace{1em} Increment \#visited
\item \textbf{for} each pair of neighboring regions
\item \hspace{1em} \textit{final result} = \textit{majority}(list of votes)
\item Solve the Laplace equation to construct a depth map
\end{enumerate}
\end{algorithm}
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.
Algorithm

Select normal locations

Micro-tasks (human)

Bi-Laplace equation

\[ \Delta^2 f = 0 \]
with constraints

Make3D

Shape-from-Shading
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.
## Statistics

<table>
<thead>
<tr>
<th>example</th>
<th>micro-tasks used</th>
<th>total $ cost</th>
<th>successful micro-task duration</th>
<th>algorithm delay until 50% complete</th>
<th>algorithm delay until 100% complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>normal map</td>
<td>1620–4340</td>
<td>$5.04–10.76</td>
<td>8.8 s 8.1 s</td>
<td>1.1–5.0 hrs</td>
<td>2.8–15.1 hrs</td>
</tr>
<tr>
<td>depth layers</td>
<td>2669–7620</td>
<td>$6.41–17.15</td>
<td>6.2 s 5.5 s</td>
<td>0.95–1.6 hrs</td>
<td>3.7–8.0 hrs</td>
</tr>
<tr>
<td>symmetry map</td>
<td>1020–1740</td>
<td>$3.24–3.92</td>
<td>9.0 s 8.5 s</td>
<td>0.4–1.6 hrs</td>
<td>0.7–4.9 hrs</td>
</tr>
<tr>
<td>example</td>
<td>micro-tasks used</td>
<td>total $ cost</td>
<td>successful micro-task duration</td>
<td>algorithm delay until % complete</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------</td>
<td>--------------</td>
<td>--------------------------------</td>
<td>---------------------------------</td>
<td></td>
</tr>
<tr>
<td>normal map</td>
<td>1620–4340</td>
<td>$5.04–10.76</td>
<td>avg 8.8 s外媒</td>
<td>median 8.1 s</td>
<td>50% 1.1–5.0 hrs</td>
</tr>
<tr>
<td>depth layers</td>
<td>2669–7620</td>
<td>$6.41–17.15</td>
<td>avg 6.2 s外媒</td>
<td>median 5.5 s</td>
<td>50% 0.95–1.6 hrs</td>
</tr>
<tr>
<td>symmetry map</td>
<td>1020–1740</td>
<td>$3.24–3.92</td>
<td>avg 9.0 s外媒</td>
<td>median 8.5 s</td>
<td>50% 0.4–1.6 hrs</td>
</tr>
</tbody>
</table>
## Statistics

<table>
<thead>
<tr>
<th>example</th>
<th>micro-tasks used</th>
<th>total $ cost</th>
<th>successful micro-task duration</th>
<th>algorithm delay until % complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>normal map</td>
<td>1620–4340</td>
<td>$5.04–10.76</td>
<td>avg 8.8 s  median 8.1 s</td>
<td>50% 1.1–5.0 hrs  100% 2.8–15.1 hrs</td>
</tr>
<tr>
<td>depth layers</td>
<td>2669–7620</td>
<td>$6.41–17.15</td>
<td>avg 6.2 s  median 5.5 s</td>
<td>50% 0.95–1.6 hrs  100% 3.7–8.0 hrs</td>
</tr>
<tr>
<td>symmetry map</td>
<td>1020–1740</td>
<td>$3.24–3.92</td>
<td>avg 9.0 s  median 8.5 s</td>
<td>50% 0.4–1.6 hrs  100% 0.7–4.9 hrs</td>
</tr>
</tbody>
</table>
## Statistics

<table>
<thead>
<tr>
<th>example</th>
<th>micro-tasks used</th>
<th>total $ cost</th>
<th>successful micro-task duration</th>
<th>algorithm delay until % complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>normal map</td>
<td>1620–4340</td>
<td>$5.04–10.76</td>
<td>avg 8.8 s median 8.1 s</td>
<td>50% 1.1–5.0 hrs 100% 2.8–15.1 hrs</td>
</tr>
<tr>
<td>depth layers</td>
<td>2669–7620</td>
<td>$6.41–17.15</td>
<td>avg 6.2 s median 5.5 s</td>
<td>50% 0.95–1.6 hrs 100% 3.7–8.0 hrs</td>
</tr>
<tr>
<td>symmetry map</td>
<td>1020–1740</td>
<td>$3.24–3.92</td>
<td>avg 9.0 s median 8.5 s</td>
<td>50% 0.4–1.6 hrs 100% 0.7–4.9 hrs</td>
</tr>
</tbody>
</table>
Summary

HC algorithms can work where automatic algorithms still cannot.

Identify the essential difficulty, and rephrase the algorithm in terms of micro human perception.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Micro-task</th>
<th>Combining Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>depth layers</td>
<td>identify depth jumps</td>
<td>laplace equation</td>
</tr>
<tr>
<td>normal map</td>
<td>orient thumbtacks</td>
<td>bi-laplace equation</td>
</tr>
<tr>
<td>symmetry map</td>
<td>position point pair</td>
<td>none</td>
</tr>
</tbody>
</table>

If this were a Photoshop plug-in, would you use it?
End
Accuracy
Accuracy
Accuracy
Cost and Reliability

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

Table 1: Micro-tasks

<table>
<thead>
<tr>
<th>example</th>
<th>total HPs</th>
<th>% completely unreliable</th>
<th>average reliability for reliable HPs</th>
<th>micro-tasks per HP</th>
</tr>
</thead>
<tbody>
<tr>
<td>normal map</td>
<td>61</td>
<td>42%</td>
<td>89%</td>
<td>123</td>
</tr>
<tr>
<td>depth layers</td>
<td>48</td>
<td>35%</td>
<td>87%</td>
<td>193</td>
</tr>
<tr>
<td>symmetry map</td>
<td>19</td>
<td>24%</td>
<td>99%</td>
<td>97</td>
</tr>
</tbody>
</table>

Table 2: Human Processors
## Timing

<table>
<thead>
<tr>
<th>example</th>
<th>successful micro-task duration</th>
<th>algorithm delay until % complete</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>avg</td>
<td>median</td>
</tr>
<tr>
<td>normal map</td>
<td>8.8 s</td>
<td>8.1 s</td>
</tr>
<tr>
<td>depth layers</td>
<td>6.2 s</td>
<td>5.5 s</td>
</tr>
<tr>
<td>symmetry map</td>
<td>9.0 s</td>
<td>8.5 s</td>
</tr>
</tbody>
</table>
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]
- [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]
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 ) \sim 1 billion perceptions per second