Search
Lecture 4

Jim Rehg

January 29, 2016

College of Computing
Georgia Tech

Where Was Prof. Rehg last Wed?

NSF I-Corps Kick-Off Meeting (Houston, TX)

Program to explore commercialization opportunities for research technology

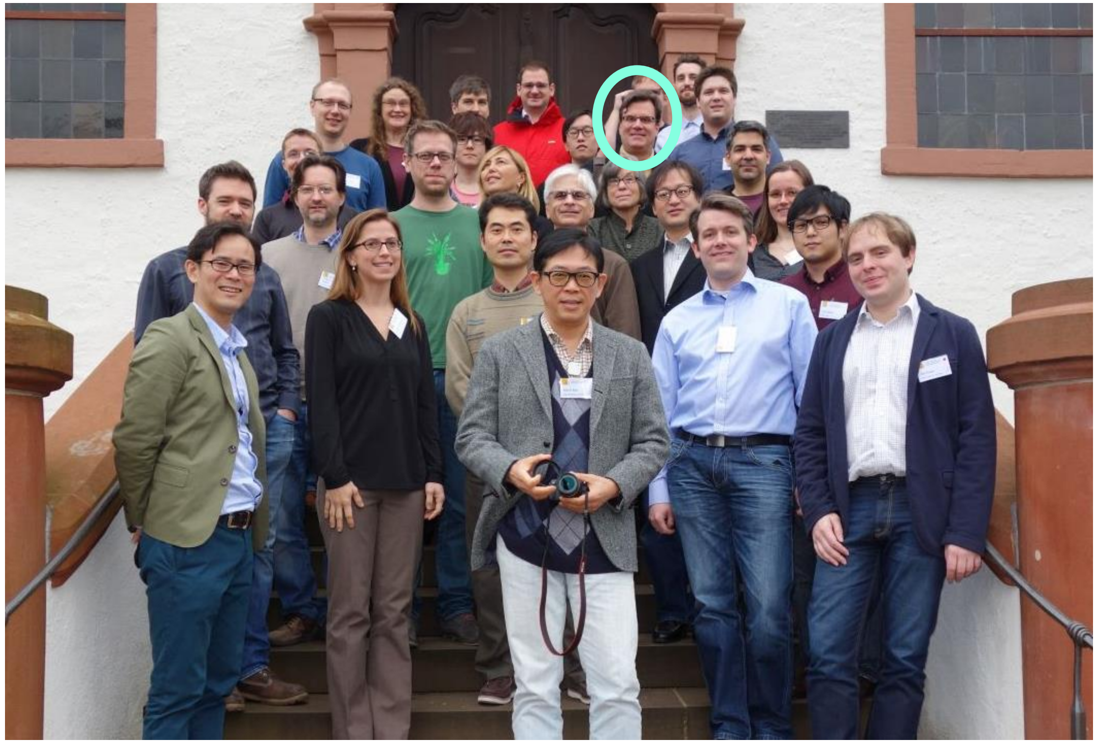
Participating with Agata Rozga (Res. Scientist), Yin Li (PhD student), and Ernesto Escobar (Mentor)

$50K to conduct interviews and purse customer discovery

Where Was Prof. Rehg on Mon and Wed?

Dagstuhl Seminar 16042

Eyewear Computing – Augmenting the Human with Head-Mounted Wearable Assistants



Well-known international forum for exchange of research ideas

I was co-organizer with researchers from Germany, Japan, and Google (CA)

Focus on “wearable cameras meet HCI”

A* Search – Additional Details

What is the relationship between

A* Search

Greedy Best-First Search

Uniform Cost Search?

Work Through Examples of Corner Cases

A* Algorithm

- Priority queue *PQ* begins empty.
- *V* (= set of previously visited (*state,f,backpointer*)-triples) begins empty.
- Put *S* into *PQ* and *V* with priority $f(s) = g(s) + h(s)$
- Is *PQ* empty?
  - Yes? Sadly admit there's no solution
  - No? Remove node with lowest $f(n)$ from queue. Call it n.
  - If n is a goal, stop and report success.
  - "expand" n : For each n' in **successors**(n)….
    - Let $f' = g(n') + h(n') = g(n) + cost(n,n') + h(n')$
    - **If** n' not seen before, or n' previously expanded with $f(n')>f'$, or n' currently in *PQ* with $f(n')>f'$
    - **Then** Place/promote n' on priority queue with priority f' and update *V* to include (*state*=n', f', *BackPtr*=n).
    - **Else** Ignore n'

= $h(s)$ because $g(\text{start}) = 0$

use sneaky trick to compute $g(n)$

8-Puzzle Example

Example State

1		5
2	6	3
7	4	8

Goal State

1	2	3
4	5	6
7	8	

Which of the following are admissible heuristics?

- $h(n)$ = Number of tiles in wrong position in state n
- $h(n) = 0$
- $h(n)$ = Sum of Manhattan distances between each tile and its goal location
- $h(n) = 1$

- $h(n) = \min(2, h^*[n])$
- $h(n) = h^*(n)$
- $h(n) = \max(2, h^*[n])$

Dominance

Reminder: Heuristic $h(n)$ is *admissible* if $h(n) <= h^*(n)$

If h1 and h2 *admissible* heuristics, and

$h2(n) >= h1(n)$ for all n,

Then h2 *dominates* h1, and

h2 is better for search

Why?

Note: $h'(n) = \max\{ h1(n), h2(n) \}$ is admissible and dominates h1 and h2

Dominant Heuristic is Better

In A* every node n with

$f(n) < C^*$ will be expanded

$h(n) < C^* - g(n)$ will be expanded

$h2(n) > h1(n)$ for all n, then

Set of n for which $h1(n) < C^* - g(n)$

Will be *larger* than set of n for which $h2(n) < C^* - g(n)$

Thus $h1$ will expand more nodes

Questions?