Project Title

March 22, 2019

John Doe

Abstract

Lorem ipsum dolor sit amet, ea putent ceteros eam, duo ne lorem vidisse quaestio. Te dolor eleifend adolescens vis. Mea vivendum neglegentur cu. Vel ad nobis corrumpit percipitur, eam zril nonumy corpora id. Pro inani voluptua omittantur id, ne ipsum harum eos. Magna aeque consul at cum, ei melius definitiones usu.

CONTENTS

1 Algorithms that Solve TSP

1.1 Brute Force Search (Naive Algorithm)

1.1.1 Run-time Analysis

$O(n!)$

1.1.2 Pseudo-code

Algorithm 1 FindNextWaypoint

```
procedure FindNextWaypoint
    Input
        map:                    The current occupancy grid
        window_size:            Width of the sliding window used to generate candidate points
        stride_length:          Row much to shift the sliding window with each iteration
        visited_locations:      Grid with same size as map representing the locations we have been to
        robot_loc:              Currently location of the robot
        avoidance_radius:       Radius around candidate waypoint that should not contain walls
        empty_radius:           Radius around candidate waypoint that should be empty
        RESOLUTION:             How many meters per cell
    Output
        candidate               The best potential waypoint to go to next
    potential_candidates ← []
    for every center_cell in map that is greater than stride_length apart from each other do
        skip ← false
        for every neighbor_cell in a avoidance_radius away from center_cell do
            if alreadyVisited(visited_location, neighbor_cell) then
                skip ← true
                break
        if skip == true then
            continue
        for every neighbor_cell in a empty_radius away from center_cell do
            if getMapValue(visited_location) ≠ 0 then
                skip ← true
                break
        if skip == true then
            continue
        cell_sum ← sum(all cells within window_size/2 from center_cell)
        potential_candidates.append(center_cell)
    candidate ← sorted(potential_candidates)
    candidate ← potential_candidates.pop()
    while not reachableByAStar(robot_loc, candidate) do
        candidate ← potential_candidates.pop()
        best_loc = candidate
    return convert_row_col_to_coord(best_loc, RESOLUTION, map)
```

1.1.3 Description

Lorem ipsum dolor sit amet, ea putent ceteros eam, duo ne lorem vidisse quaestio. Te dolor eleifend adolescens vis. Mea vivendum neglegentur cu. Vel ad nobis corrumpit percipitur, eam zril nonumy corpora id. Pro inani voluptua omittantur id, ne ipsum harum eos. Magna aeque consul at cum, ei melius definitiones usu.

Nam ea odio tollit iuvaret, bonorum persecuti mel ex. Ancillae phaedrum erroribus vis ne. Prima ullum propriae et per, brute aliquam inermis ad nam. Nam unum veri nostrud ex, nemore fabellas conceptam cum ad. Vis ut discere ullamcorper.

Eu voluptua vituperata eum, no alii epicuri has. Vis magna utinam disputationi an. Probo feugait forensibus sea no, saperet perpetua splendide no pri, in nemore nominati vis. An sed aliquam iudicabit, harum feugait usu id, cu ius tota everti. Ne eripuit minimum pri, est te stet pertinax, aliquam noluisse iracundia et eos.

1.2 Held-Karp Algorithm (Dynamic Algorithm)

1.2.1 Run-time Analysis

$O(n^2 * 2^n)$

1.2.2 Code

```
struct zone {
    unsigned long watermark[NR_WMARK];
    unsigned long lowmem_reserve[MAX_NR_ZONES];
    struct per_cpu_pageset pageset[NR_CPUS];
    spinlock_t lock;
    struct free_area free_area[MAX_ORDER]
    spinlock_t lru_lock;
    struct zone_lru {
        struct list_head list;
        unsigned long nr_saved_scan;
    } lru[NR_LRU_LISTS];
    struct zone_reclaim_stat reclaim_stat;
    unsigned long pages_scanned;
    unsigned long flags;
    atomic_long_t vm_stat[NR_VM_ZONE_STAT_ITEMS];
    int prev_priority;
    unsigned int inactive_ratio;
    wait_queue_head_t *wait_table;
    unsigned long wait_table_hash_nr_entries;
    unsigned long wait_table_bits;
    struct pglist_data *zone_pgdat;
    unsigned long zone_start_pfn;
    unsigned long spanned_pages;
    unsigned long present_pages;
    const char *name;
};
```

1.2.3 Description

Lorem ipsum dolor sit amet, ea putent ceteros eam, duo ne lorem vidisse quaestio. Te dolor eleifend adolescens vis. Mea vivendum neglegentur cu. Vel ad nobis corrumpit percipitur, eam zril nonumy corpora id. Pro inani voluptua omittantur id, ne ipsum harum eos. Magna aeque consul at cum, ei melius definitiones usu.

Nam ea odio tollit iuvaret, bonorum persecuti mel ex. Ancillae phaedrum erroribus vis ne. Prima ullum propriae et per, brute aliquam inermis ad nam. Nam unum veri nostrud ex, nemore fabellas conceptam cum ad. Vis ut discere ullamcorper.

Eu voluptua vituperata eum, no alii epicuri has. Vis magna utinam disputationi an. Probo feugait forensibus sea no, saperet perpetua splendide no pri, in nemore nominati vis. An sed aliquam iudicabit, harum feugait usu id, cu ius tota everti. Ne eripuit minimum pri, est te stet pertinax, aliquam noluisse iracundia et eos.

2 Testing

Lorem ipsum dolor sit amet, ea putent ceteros eam, duo ne lorem vidisse quaestio. Te dolor eleifend adolescens vis. Mea vivendum neglegentur cu. Vel ad nobis corrumpit percipitur, eam zril nonumy corpora id. Pro inani voluptua omittantur id, ne ipsum harum eos. Magna aeque consul at cum, ei melius definitiones usu.

Nam ea odio tollit iuvaret, bonorum persecuti mel ex. Ancillae phaedrum erroribus vis ne. Prima ullum propriae et per, brute aliquam inermis ad nam. Nam unum veri nostrud ex, nemore fabellas conceptam cum ad. Vis ut discere ullamcorper.

Eu voluptua vituperata eum, no alii epicuri has. Vis magna utinam disputationi an. Probo feugait forensibus sea no, saperet perpetua splendide no pri, in nemore nominati vis. An sed aliquam iudicabit, harum feugait usu id, cu ius tota everti. Ne eripuit minimum pri, est te stet pertinax, aliquam noluisse iracundia et eos [1].

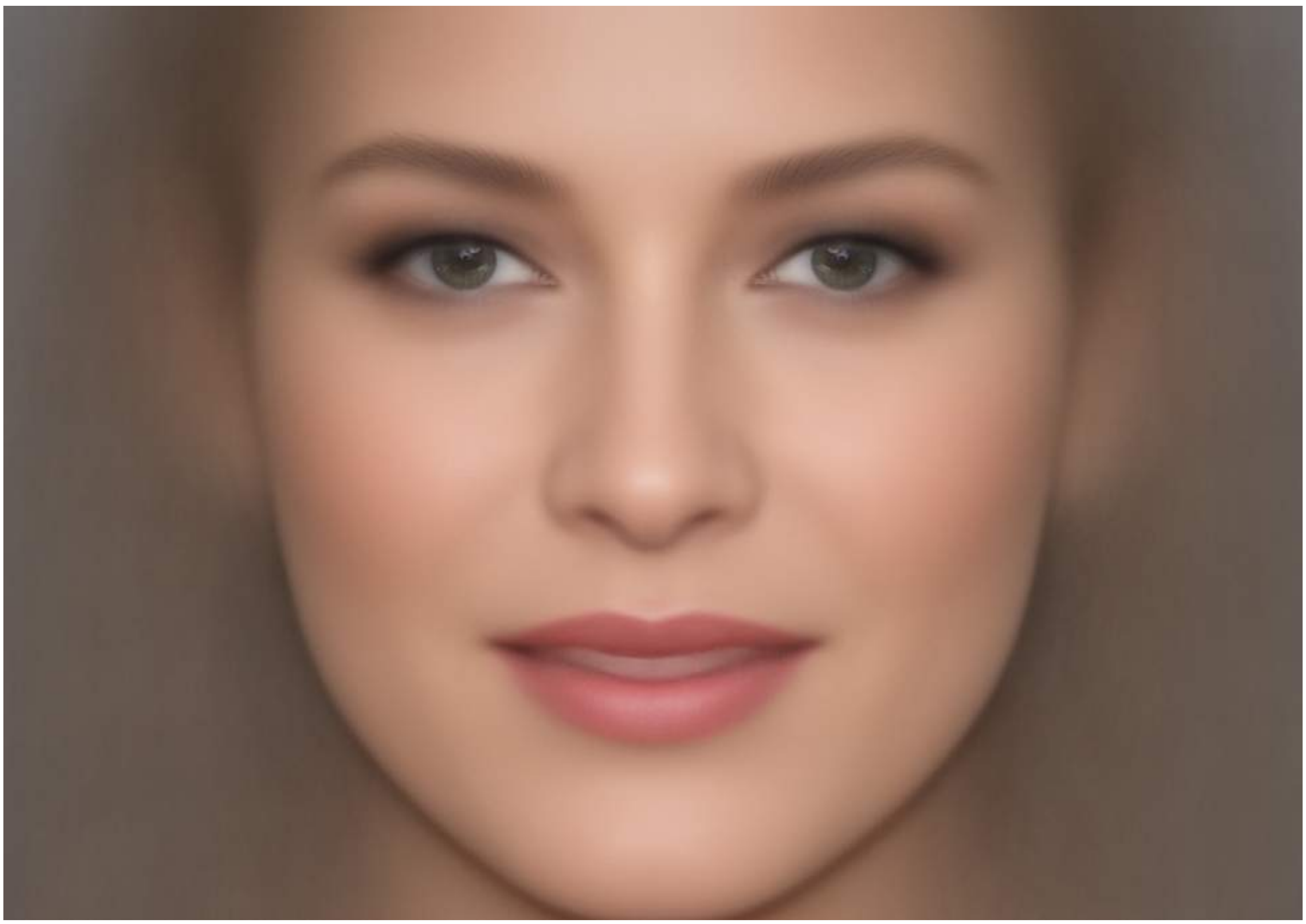

Figure 1: capture faces

References

[1] "Operating system - processes," Available at https://www.tutorialspoint.com/operating_system/os_processes.htm (2018/10/19).