Energetic Pathfinding and Perceptual Heuristics

Introduction

Subject Area

The work operates within **Artificial Intelligence Search and Reasoning Systems**, focusing on heuristic search, knowledge representation and perception-based reasoning. It links foundational AI theory from Russell and Norvig's textbook [1] to sensory modeling using computer vision and urban data streams.

Classical pathfinding algorithms (A*, weighted A*) are extended by incorporating perceptually informed heuristics derived from live camera data. This connects abstract state-space search to embodied decision-making in dynamic environments.

Project Type and Rationale

An **Applied Project** applies known methods to realistic situations, blending heuristic search (A*, weighted A*) with machine learning (Ridge regression, feature engineering) on real urban data to evaluate perceptually informed navigation.

Rationale

The work applies course concepts (uninformed search, informed search, constraint satisfaction, knowledge representation) to sensory data from a dynamic environment. Cost and optimality are extended to include perceptual load and behavioral friction.

Problem Statement

Traditional pathfinding algorithms optimize for geometric distance or travel time but fail to capture the **energetic and perceptual dynamics** of movement in dense urban environments. In Manhattan, the cost of traversing a route is not defined by distance alone but by **sensory and behavioral complexity**: crowding, visual noise, infrastructure gaps and movement unpredictability.

Figure 1: Carnegie Hall parable: the project asks how an agent can learn to navigate from Grand Central to Carnegie Hall while accounting for perceptual effort instead of pure geometric distance.

This work develops a **perceptually informed heuristic** for A* search using NYC traffic camera data. The heuristic models environmental resistance by extracting features from visual data. Human pedestrians naturally avoid stressful routes based on crowded intersections, poor lighting and aggressive traffic patterns; this approach encodes similar avoidance behavior.

Key Research Question: Can learned heuristics from vision-based environmental features outperform standard Manhattan-distance baselines in predicting route stress and improving path interpretability?

Aim

Implement and evaluate a heuristic search system that learns energetic pathfinding behavior by integrating sensory data into classical A* reasoning. Test whether perception-driven cost modeling improves interpretability and environmental realism in AI navigation.

This project will achieve the following objectives:

Objective	**Description**
Navigation Environment	Assemble the Grand Central–Carnegie Hall corridor graph in EPSG:2263 using NYC DCM street centerlines; snap 161 intersections to the projected grid, overlay official hydrography and park polygons, and link each node to the nearest DOT camera through Voronoi zone IDs.
Knowledge Base	Engineer a data pipeline that captures a one-week window of traffic camera imagery, runs computer vision analysis, normalizes feature vectors and stores symbolic predicates such as `high_pedestrian` and `bike_violation` for search-time reasoning.
Perceptual Heuristic	Fit a Ridge regression model on annotated camera data to learn energetic weights, calibrate cost scaling with stratified cross-validation and expose a reusable function returning $h(n) = \mathbf{w}^T \mathbf{f}_{zone(n)}$ with interpretable coefficients.
A* Implementation	Implement baseline A*, learned A* and adaptive weighted A* ($W \in \{1.0, 1.2, 1.5\}$); instrument node expansions, path costs and admissibility checks for the learned heuristic.
Evaluation	Compare systems via Monte Carlo route sampling, Mean Absolute Error on stress predictions, search effort (nodes expanded, runtime), path diversity and permutation importance to surface dominant perceptual factors.
Implications	Deliver a lightweight test-harness application that supports collaborator exploration of alternate corridors, crowdsourced qualitative feedback and follow-on embodied AI studies aligned with PEAS and constraint satisfaction themes.

Methodology

Agent Reasoning Framework

Component	Description
Performance Measure	Minimize energetic cost (stress score) along navigation path.
Environment	Manhattan street network with 108 camera coverage zones in the corridor (Voronoi tessellation); dynamic traffic conditions; partially observable via camera sensors.
Actuators	Directional movements (N, S, E, W) along street intersections.
Sensors	Computer vision analysis of traffic camera imagery; multi-dimensional feature vectors encoding violations, density, infrastructure quality.

The state space is constructed by combining three data sources:

1. Intersection Nodes (States): Extract intersection coordinates from NYC Planning Digital City Map (DCM) street centerlines. Each intersection i is defined by geographic coordinates (lat_i, lon_i). States are encoded sequentially (n0 to n160) or by street names (45th_Park, 51st_Madison).

State Space Representation

Component	Definition
States S	Set of 161 intersection nodes (where streets cross) spanning a corridor bounded by extending ± 2 streets and ± 1 avenue from each endpoint: 40th to 59th St, Lexington to 8th Ave
Initial State s_0	Grand Central (42nd & Park)
Goal State s_g	Carnegie Hall (57th & 7th)
Actions $A(s)$	{North, South, East, West} ∩ {adjacent intersections via street connectivity}
Transition Model	Successor function returns neighboring intersections
Path Cost	$g(n) + h(n)$ where $h(n) = \mathbf{w}^T \mathbf{f}_{zone(n)}$ (learned heuristic)

2. Camera Coverage Zones (Sensors): Assign each intersection to its nearest camera using Voronoi tessellation of 108 camera positions from NYC DOT Traffic Management Center API. Each camera c provides a multi-dimensional feature vector $\mathbf{f}_c$ encoding perceptual attributes (violations, density, infrastructure). Intersections within camera c's Voronoi cell inherit feature vector $\mathbf{f}_c$.

State Space: Intersection Nodes and Camera Coverage Zones

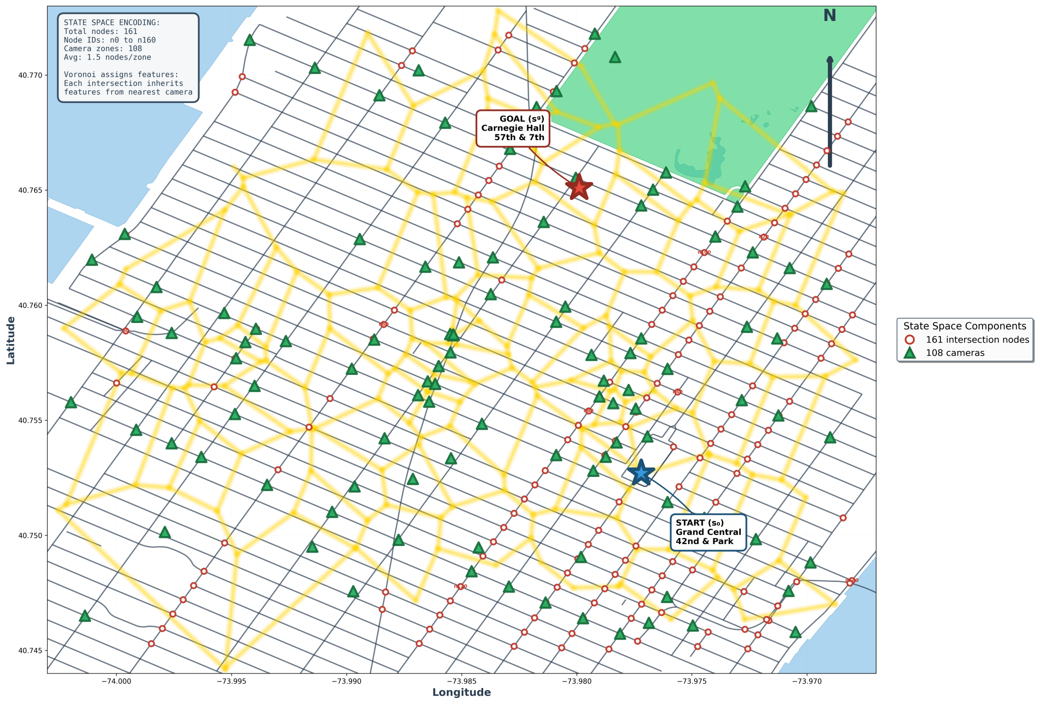

Figure 2: State space showing 161 intersection nodes (white circles with red borders), 108 camera positions (green triangles) and Voronoi tessellation (yellow) overlaying the Manhattan corridor map.

3. Graph Structure: Build an adjacency graph where edges (i, j) exist if intersections i and j are connected by a street segment. Each edge has cost $w_{ij} = d_{Manhattan}(i, j) + \mathbf{w}^T \mathbf{f}_{zone(i)}$ combining geometric distance with learned perceptual weights.

Graph Model with Energy Cost Matrix

	GC	**I1**	**I2**	**I3**	**I4**	**CH**
GC	0	2	∞	3	∞	∞
I1	2	0	2	∞	4	∞
I2	∞	2	0	∞	4	∞
I3	3	∞	∞	0	3	∞
I4	∞	4	4	3	0	5
CH	∞	∞	∞	∞	5	0

Table 1: Energy cost matrix showing weighted costs between intersections based on multi-dimensional feature vectors. Transition costs combine Manhattan distance with perceptual stress factors (traffic density, obstructions, enforcement signals).

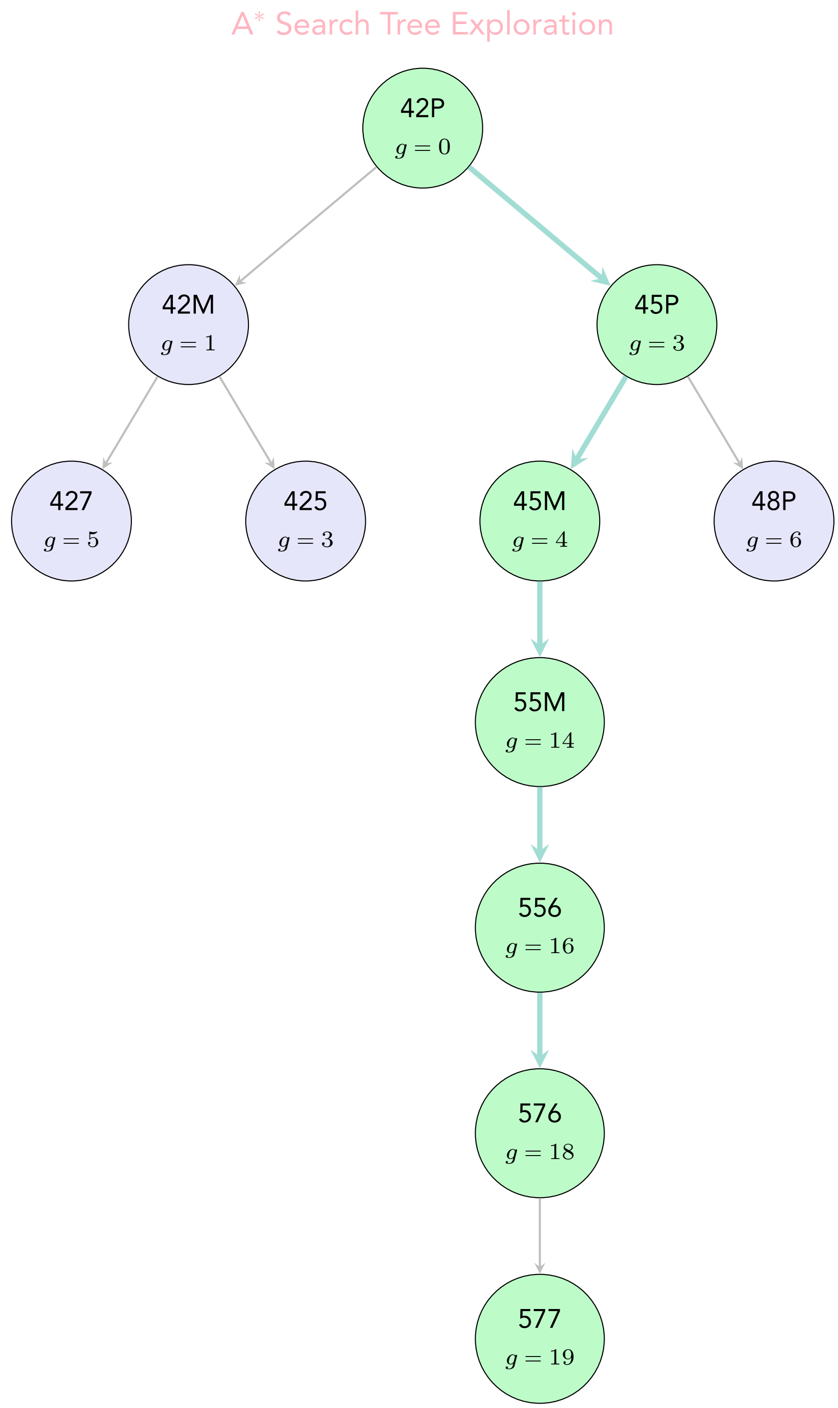

Figure 3: A* search tree with learned heuristic finding the optimal (lowest energetic cost) path. Node names encode intersections (42P = 42nd & Park, 45M = 45th & Madison, 577 = 57th & 7th). Values show $g(n)$ = cumulative blocks traveled. The optimal path (mint green): 42P ($g = 0$) $\rightarrow$ 45P ($g = 3$) $\rightarrow$ 45M ($g = 4$) $\rightarrow$ 55M ($g = 14$) $\rightarrow$ 556 ($g = 16$) $\rightarrow$ 576 ($g = 18$) $\rightarrow$ 577 ($g = 19$) goes north on Park to 45th, west to Madison, north on Madison to 55th, west to 6th, north to 57th, west to 7th. Gray branches show A* rejecting the baseline west-on-42nd route (42M) because the diagonal path has lower perceptual stress despite same total distance.

Heuristic Design

Baseline: Manhattan Distance

$$h_{\text{manhattan}}(n) = |x_n - x_{\text{goal}}| + |y_n - y_{\text{goal}}|$$

Admissible but ignores environmental context.

Learned Perceptual Heuristic

$$h_{\text{learned}}(n) = \mathbf{w}^T \mathbf{f}(n) + b$$

where $\mathbf{f}(n)$ represents the feature vector for node n, $\mathbf{w}$ denotes the learned weights from Ridge regression and b is the intercept term. Each intersection n inherits features from its nearest camera via Voronoi tessellation: $\mathbf{f}(n) = \mathbf{f}_c$ where $c = \arg\min_{c'} d(n, c')$. The feature vector encodes perceptual and behavioral attributes including violations (pedestrian walkway, dangerous bike positioning, red light, flow blocking, car-bike lane), density metrics (pedestrian, vulnerable populations, traffic volume), infrastructure quality (gaps, poor signage, signal malfunctions), behavioral factors (cyclist speed, aggressive behavior) and environmental conditions (visibility, weather impact, overall safety risk).

Manhattan Navigation: State Space Search Paths

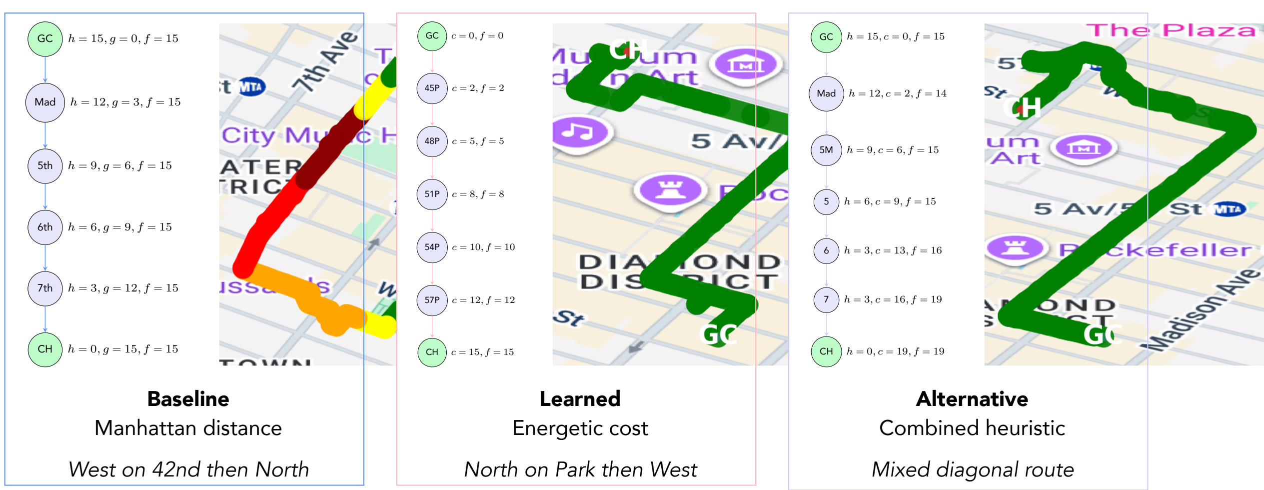

Figure 4: Six-panel comparison of A* search pathways. Each column shows tree and map side-by-side: baseline Manhattan distance, learned energetic cost and alternative mixed strategy.

Adaptive Weighted A*

$$f(n) = g(n) + W \cdot h_{\text{learned}}(n)$$

with $W \in \{1.0, 1.2, 1.5\}$ to balance the exploration-exploitation tradeoff: higher weights prioritize heuristic guidance (faster search) while lower weights preserve optimality guarantees.

Training Pipeline

Model: Ridge Regression

$$\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 + \alpha\|\mathbf{w}\|^2$$

Ridge regression offers several advantages: interpretable feature weights, regularization that prevents overfitting and fast training through closed-form solution. The model will be trained on the collected camera data (36,288 records from 108 corridor cameras) with temporal cross-validation splitting data by time periods to assess generalization.

Feature Importance Analysis

The Ridge model produces a weight w_i for each feature category (violations, pedestrian density, infrastructure gaps). Rank features by absolute coefficient value: $|w_{\text{violations}}| > |w_{\text{density}}| > |w_{\text{infrastructure}}|$ reveals which factors dominate stress predictions. This shows whether pedestrians avoid routes due to enforcement violations, crowding or infrastructure quality.

Data Pipeline

Data Collection Plan

Feature Vector Structure: Each camera zone record contains a robust multi-dimensional feature vector extracted via computer vision analysis of traffic camera footage. The features comprehensively encode violation types (pedestrian walkway, dangerous bike positioning, red light, flow blocking, car-bike lane), density metrics (pedestrian, vulnerable populations, traffic volume), infrastructure assessments (gaps, poor signage, signal malfunctions), behavioral indicators (cyclist speed, aggressive behavior) and environmental factors (visibility, weather impact, infrastructure quality, overall safety risk). Each record also includes a temperature/stress score (range: 0 to 30) as the target variable for supervised learning.

Sampling Plan: Data collection will run during the first week of November 2025 with a sampling frequency of every 30 minutes across 108 cameras in the corridor from the NYC DOT Traffic Management Center. This will yield approximately 336 samples per camera (48 samples/day × 7 days) for a total dataset of 36,288 records. This sampling frequency balances data volume with API rate limits and captures traffic patterns across different times of day (morning rush, midday, evening rush, overnight). ## Experimental Design

Experiments

Experiment	Description
Baseline comparison	A with Manhattan distance vs. learned heuristic
Weight sensitivity (trade-off: optimality vs. speed)	Test $W \in \{1.0, 1.2, 1.5\}$
Path diversity	Monte Carlo sampling of routes, measure variance in total cost
Ablation study	Remove feature categories (violations, density, infrastructure) to assess individual impact

Metrics

Metric	Description
Mean Absolute Error (MAE)	Predicted stress vs. ground truth temperature scores.
Path Length	Number of nodes in solution path.
Nodes Expanded	Computational efficiency measure.
Stress Score Distribution	Variance across sampled paths.
Feature Importance Rankings	Top contributing factors.

Success Criteria

Success will be demonstrated by achieving interpretable feature weights and showing statistical significance in paired t-tests ($p < 0.05$) when comparing baseline versus learned heuristic performance.

Work Plan

Timeline (2 weeks)

Week 1: Knowledge Base Setup and Algorithm Implementation

Days 1 to 2: Knowledge Base

Parse camera zone data structures to extract features. Design symbolic knowledge representation mapping features to predicates (pedestrian density, bike violations, infrastructure quality). Build graph with nodes as intersections and edges as street segments using NetworkX. Integrate Manhattan street topology via OSMnx. Serialize knowledge base for search queries.

Days 3 to 4: Analytical Visualizations

Design visualizations to guide implementation and analysis: route comparison maps showing baseline versus learned heuristic paths overlaid on Manhattan street network, search tree diagrams showing A* node expansion patterns, feature importance heatmaps to identify which perceptual attributes drive routing decisions and stress score distributions across different path alternatives.

Days 5 to 7: A* Implementation

Implement three A* variants: baseline with Manhattan distance heuristic, learned heuristic using Ridge weights and adaptive weighted A* with $W \in \{1.0, 1.2, 1.5\}$ to explore speed-optimality tradeoffs in informed search. Build Monte Carlo path sampling framework.

Week 2: Model Training, Evaluation, and Visualization

Days 8 to 9: Model Training

Train Ridge regression on collected camera data using temperature/stress scores as target variable. Integrate learned model with graph structure to compute perceptual heuristic values. Validate heuristic admissibility bounds for A* optimality. Identify top predictive features.

Days 10 to 11: Evaluation

Run comparative experiments: baseline (Manhattan distance) versus learned heuristic. Collect performance metrics: MAE on predicted stress scores, path length, nodes expanded and total energetic cost. Perform statistical analysis using paired t-tests. Conduct feature ablation studies.

Days 12 to 14: Documentation

Generate final visualizations. Draft project report with embedded figures. Prepare presentation materials.

References

[1] S. Russell and P. Norvig, *Artificial intelligence: A modern approach*, 4th ed. Pearson, 2022.