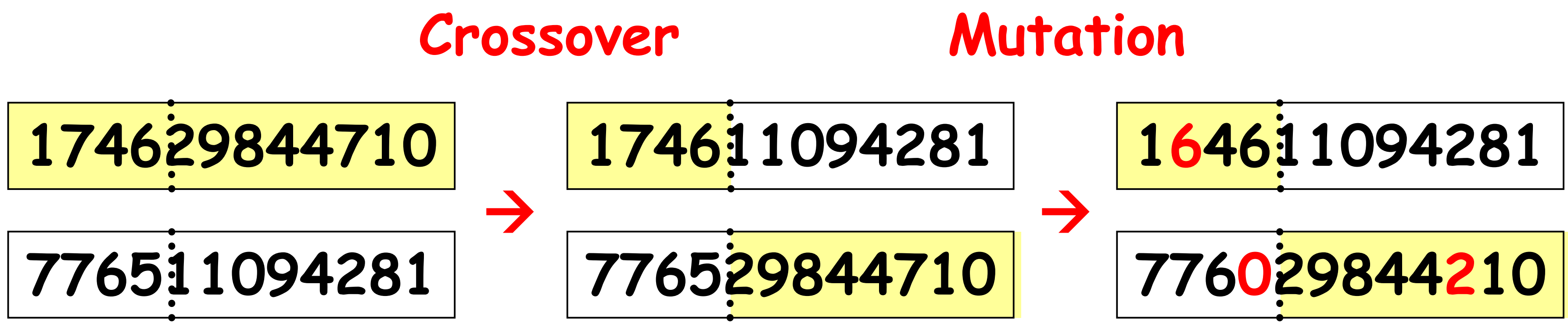
Genetic Algorithms

- Start with random population of states
  - Representation serialized (ie. strings of characters or bits)
  - States are ranked with "fitness function"
- Produce new generation
  - Select random pair(s) using probability:
    - probability ~ fitness
  - Randomly choose "crossover point"
    - Offspring mix halves
  - Randomly mutate bits



Genetic Algorithm

- Given: population P and fitness-function f
- repeat
  - newP ← empty set
  - for *i* = 1 to size(P)

    x ← RandomSelection(P,f)

    y ← RandomSelection(P,f)

    child ← Reproduce(*x*,*y*)

    if (small random probability) then child ← Mutate(child)

    add *child* to newP
  - P ← newP
- until some individual is fit enough or enough time has elapsed
- return the best individual in P according to f

Using Genetic Algorithms

- 2 important aspects to using them
  - 1. How to encode your real-life problem
  - 2. Choice of fitness function
- Research Example
  - I have N variables V1, V2, ... VN
  - I want to produce a single number from them that best satisfies my fitness function F
  - I tried linear combinations, but that didn't work
  - A guy named Stan (Matwin) I met at a workshop in Italy told me to try Genetic Programming

Genetic Programming

- Like genetic algorithm, but instead of finding the best character string, we want to find the best arithmetic expression tree
- The leaves will be the variables and the non-terminals will be arithmetic operators
- It uses the same ideas of crossover and mutation to produce the arithmetic expression tree that maximizes the fitness function.

Example: Classification and Quantification of Facial Abnormalities

- Input is 3D meshes of faces
- Disease is 22q11.2 Deletion Syndrome.
- Multiple different facial abnormalities
- We'd like to assign severity scores to the different abnormalities, so need a single number to represent our analysis of a portion of the face.

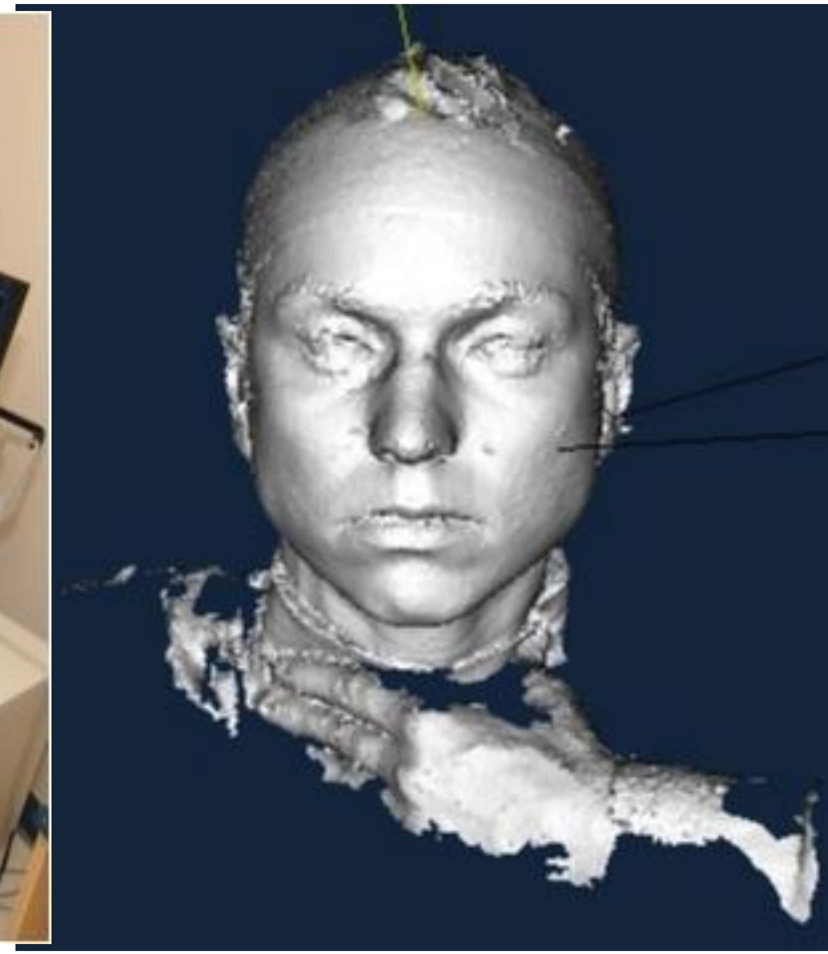
22q11.2 Deletion Syndrome (22q11.2DS)

- Caused by genetic deletion
- Cardiac anomalies, learning disabilities
- Multiple subtle physical manifestations
- Assessment is subjective



Data Collection

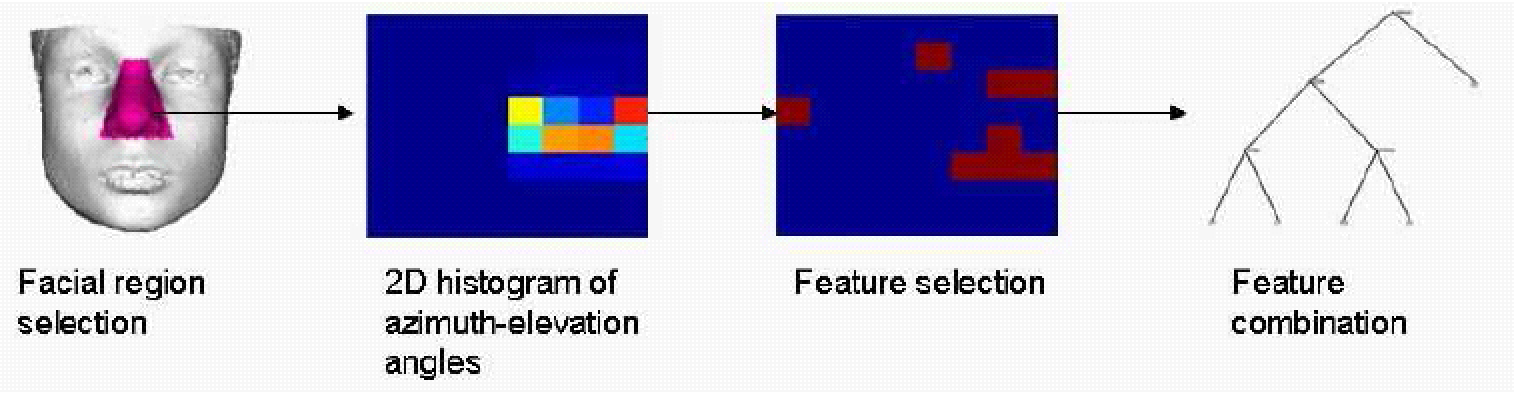
3dMD multi-camera stereo system



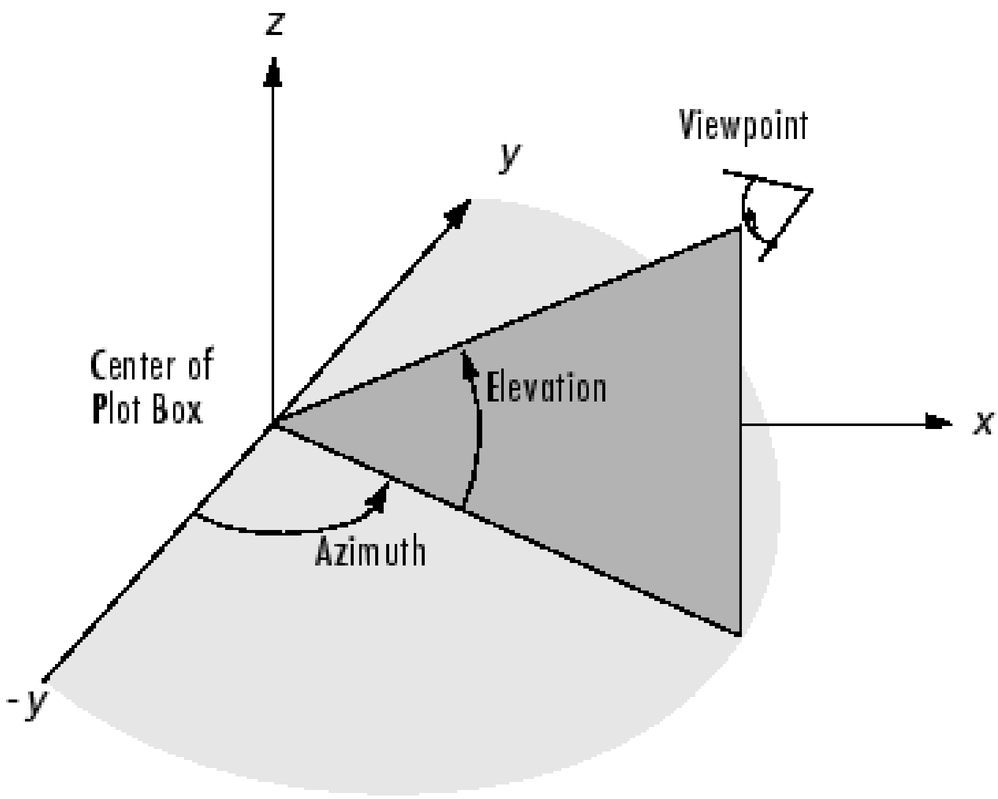
Reconstructed 3D mesh

Learning 3D Shape Quantification

- Analyze 22q11.2DS and 9 associated facial features
- Goal: quantify different shape variations in different facial abnormalities

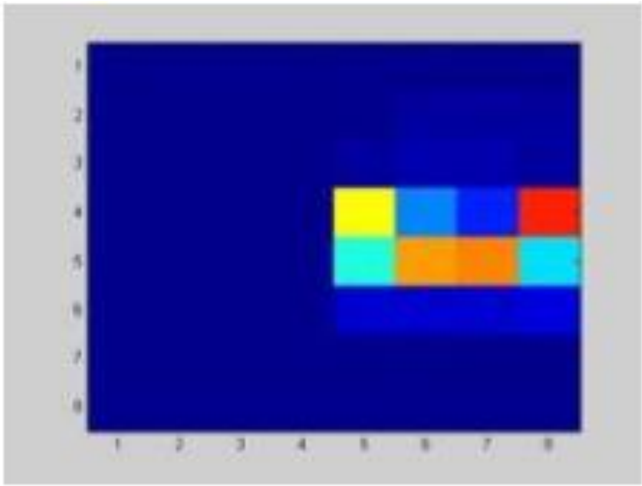


Azimuth and Elevation Angles



Learning 3D Shape Quantification - 2D Histogram Azimuth Elevation

- Using azimuth and elevation angles of surface normal vectors of points in selected region

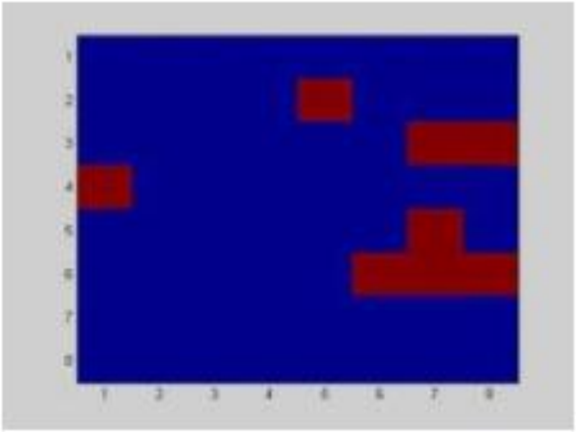


elevation

azimuth

Learning 3D Shape Quantification - Feature Selection

- Determine most discriminative bins
- Use Adaboost learning
- Obtain positional information of important region on face



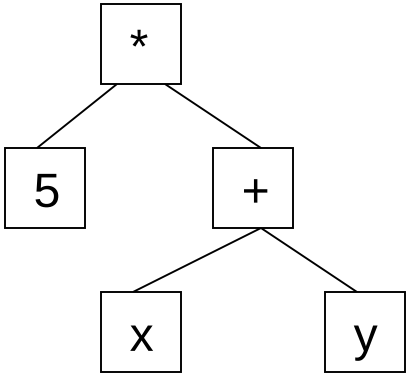
Learning 3D Shape Quantification - Feature Combination

- Use Genetic Programming (GP) to evolve mathematical expression

- Start with random population
  - Individuals are evaluated with fitness measure
  - Best individuals reproduce to form new population

Learning 3D Shape Quantification - Genetic Programming

- Individual:
  - – Tree structure
  - – Terminals e.g variables eg. 3, 5, x, y, …
  - – Function set e.g +, -, *, …
  - – Fitness measure e.g sum of square …



5*(x+y)

Learning 3D Shape Quantification - Feature Combination

- 22q11.2DS dataset
  - Assessed by craniofacial experts
  - Groundtruth is union of expert scores

- Goal: classify individual according to given facial abnormality

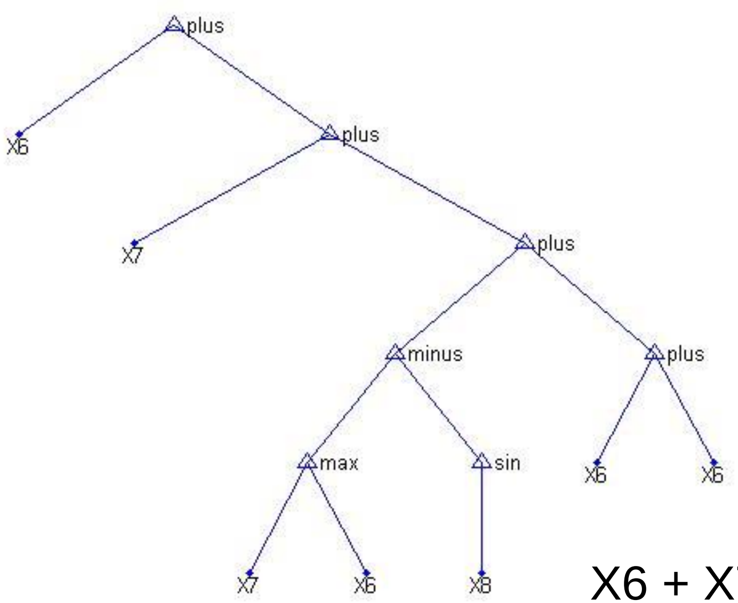
Learning 3D Shape Quantification - Feature Combination

- Individual
  – Terminal: selected histogram bins
  – Function set: +,-,*,min,max,sqrt,log,2x,5x,10x
  – Fitness measure: F1-measure

$$F(prec, rec) = \frac{2 \times (prec \times rec)}{prec + rec}$$

precision = TP/(TP + FP)

recall = TP/all positives



X6 + X7 + (max(X7,X6)-sin(X8) + (X6+X6))

Learning 3D Shape Quantification - Experiment 1

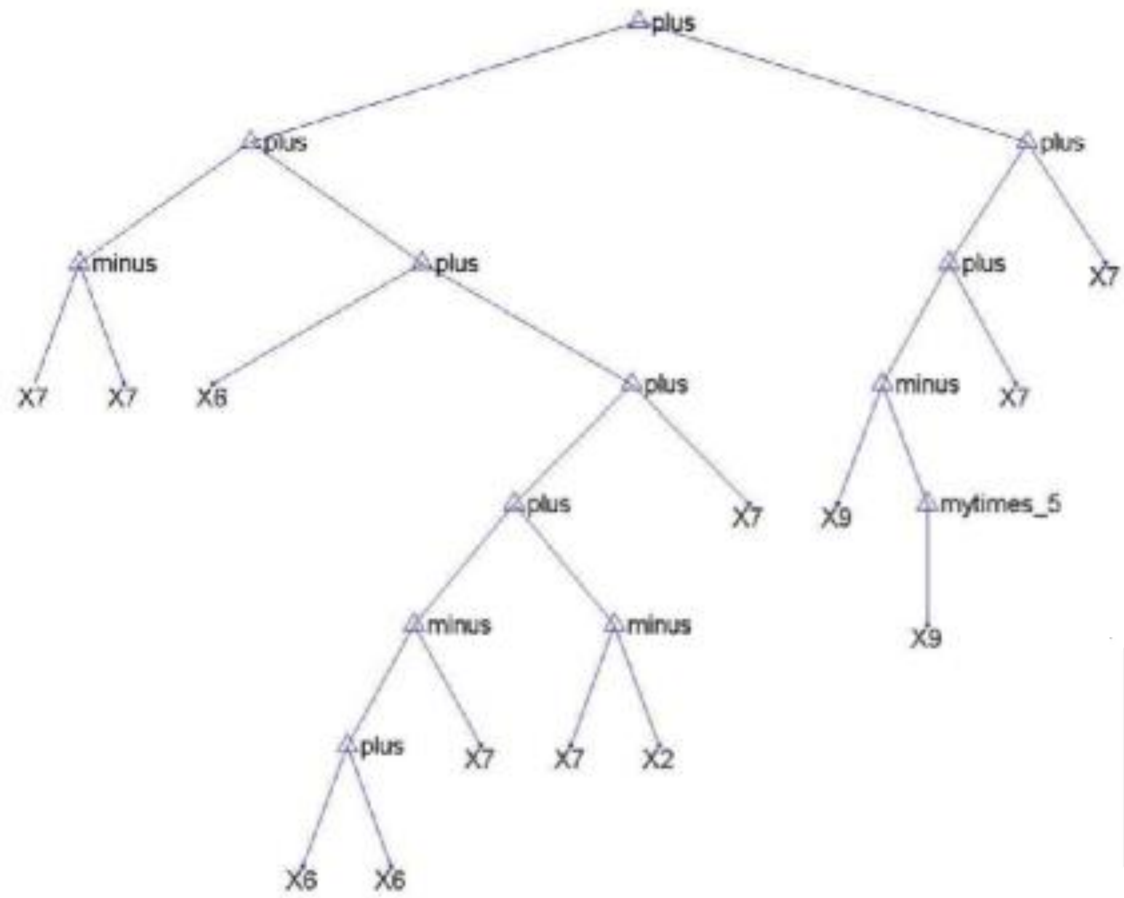
- Objective: investigate function sets
  - Combo1 = {+,-,*,*min*,*max*}
  - Combo2 = {+,-,*,*min*,*max*,*sqrt*,*log2*,*log10*}
  - Combo3 = {+,-,*,*min*,*max*,
    2x,*5x*,*10x*,*20x*,*50x*,*100x*}
  - Combo4 = {+,-,*,*min*,*max*,*sqrt*,*log2*,*log10*,
    2x,*5x*,*10x*,*20x*,*50x*,*100x*}

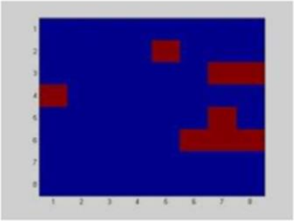
Learning 3D Shape Quantification - Experiment 1

- Best F-measure out of 10 runs

Facial anomaly	Combo1	Combo2	Combo3	Combo4
Midface Hypoplasia	0.8393	0.8364	0.8527	0.80
Tubular Nose	0.8571	0.875	0.8667	0.8813
Bulbous Nasal Tip	0.8545	0.8099	0.8103	0.7544
Prominent Nasal Root	0.8667	0.8430	0.8571	0.8335
Small Nasal Alae	0.8846	0.8454	0.8454	0.8571
Retrusive Chin	0.7952	0.8000	0.7342	0.7586
Open Mouth	0.9444	0.9714	0.9189	0.9189
Small Mouth	0.6849	0.7568	0.6829	0.7750
Downturned mouth	0.8000	0.7797	0.8000	0.8000

Tree structure for quantifying midface hypoplasia





Xi are the selected histogram bins from an azimuth-elevation histogram of the surface normals of the face.

Learning 3D Shape Quantification - Experiment 2

- Objective: compare local facial shape descriptors

Facial abnormality	Region Histogram	Selected Bins	GP
Midface hypoplasia	0.697	0.721	0.853
Tubular nose	0.701	0.776	0.881
Bulbous nasal tip	0.617	0.641	0.855
Prominent nasal root	0.704	0.748	0.867
Small nasal alae	0.733	0.801	0.885
Retrusive chin	0.658	0.713	0.800
Open mouth	0.875	0.889	0.971
Small mouth	0.694	0.725	0.775
Downturned mouth	0.506	0.613	0.800

Learning 3D Shape Quantification - Experiment 3

- Objective: predict 22q11.2DS

Method	F-measure
Quantification vector with SVM	0.709
Quantification vector with Adaboost	0.721
Quantification vector with GP	0.821
Global saliency map	0.764
Selected bins of global saliency map	0.9
Global 2D histogram	0.79
Selected bins of global 2D histogram	0.9
Selected bins of global saliency map with GP	0.96
Selected bins of global 2D histogram with GP	0.92
Expert's median	0.68

Local Search in Continuous Spaces

- Given a continuous state space
  $S = \{(x_1,x_2,\ldots,x_N) \mid x_i \varepsilon \mathbb{R}\}$
- Given a continuous objective function
  $f(x_1,x_2,\ldots,x_N)$
- The gradient of the objective function is a vector $\nabla f = (\partial f/\partial x_1,\partial f/\partial x_2,\ldots,\partial f/\partial x_N)$
- The gradient gives the magnitude and direction of the steepest slope at a point.

Local Search in Continuous Spaces

- To find a maximum, the basic idea is to set $\nabla f = 0$
- Then updating of the current state becomes

  $x \leftarrow x + \alpha \nabla f(x)$

  where α is a small constant.
- Theory behind this is taught in numerical methods classes.
- Your book suggests the Newton-Raphson method. Luckily there are packages…..

Computer Vision Pose Estimation Example

I have a
3D model
of an object
and an image
of that object.

I want to find
the pose: the
position and
orientation
of the camera.

Figure 18: The pose computed from six point correspondence using the algorithm described in Section 5.2.

Figure 19: The pose computed from an ellipse-circle correspondence using the algorithm described in Section 5.6.

and

$$R = \begin{pmatrix} s^2 + l^2 - m^2 - n^2 & 2(lm - sn) & 2(ln + sm) \\ 2(lm + sn) & s^2 - l^2 + m^2 - n^2 & 2(mn - sl) \\ 2(ln - sm) & 2(mn + sl) & s^2 - l^2 - m^2 + n^2 \end{pmatrix}. \quad (52)$$

Powell's method [19] in the seven-dimensional space of the pose solution (four quaternion parameters and the translation t) is used, along with the constraint that the sum of the squares of the quaternion parameters must equal 1, as seen in equation (51). Figure 21 shows an initial pose estimate for a single-object image as

Figure 20: The pose computed from six point correspondence and one ellipse-circle correspondence using the generalized methodology.

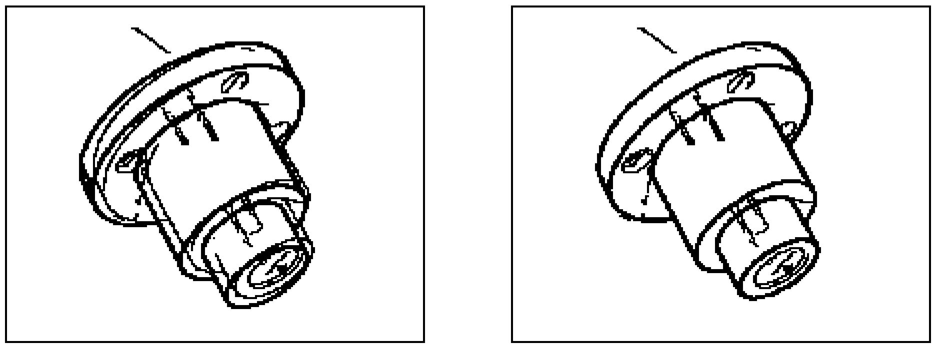
pose from
6 point
correspondences

pose from ellipse-
circle
correspondence

pose from both
6 points and
ellipse-circle
correspondences

Computer Vision Pose Estimation Example

Initial pose from points/ellipses and
final pose after optimization.



(a) Initial pose (b) Final pose

Figure 21: Example pose hypothesis and final pose after constrained optimization.

- The optimization was searching a 6D space: (x,y,z,θx,θy,θz)
- The fitness function was how well the projection of the 3D object lined up with the edges on the image.

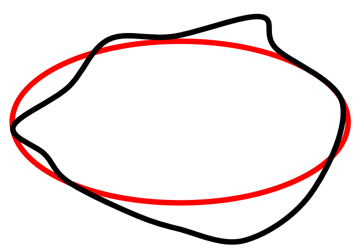
Fitness Function

- Modified Hausdorf Distance between the image of the projected model and the image of the detected edges

The directed distance d_6 [18] is used to quantitatively evaluate how well the projected model point set (A) overlays the edge image point set (B), and it is defined as

$$d_6(A, B) = \frac{1}{N_A} \sum_{a \in A} d(a, B), \tag{41}$$

where N_A is the number of points in set A.



Searching with Nondeterministic Actions

- Vacuum World (actions = {left, right, suck})

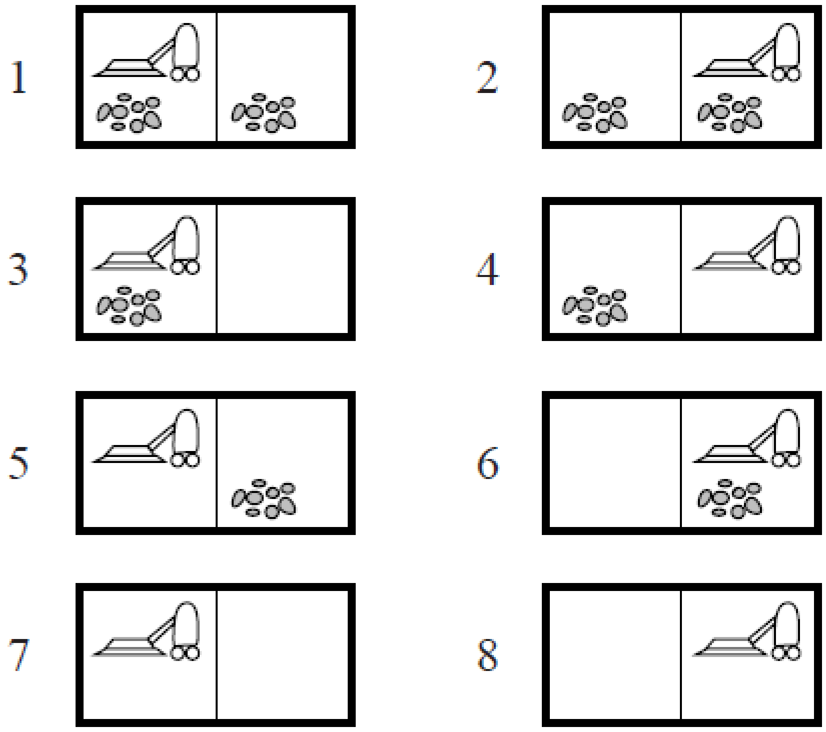


Figure 4.9 The eight possible states of the vacuum world; states 7 and 8 are goal states.

Searching with Nondeterministic Actions

In the nondeterministic case, the result of an action can vary.

Erratic Vacuum World:

- When sucking a dirty square, it cleans it and sometimes cleans up dirt in an adjacent square.
- When sucking a clean square, it sometimes deposits dirt on the carpet.

Generalization of State-Space Model

1. Generalize the transition function to return a set of possible outcomes.

   oldf: $S \times A \rightarrow S$ newf: $S \times A \rightarrow 2^S$

2. Generalize the solution to a contingency plan.

   if state=s then action-set-1 else action-set-2

3. Generalize the search tree to an AND-OR tree.

AND-OR Search Tree

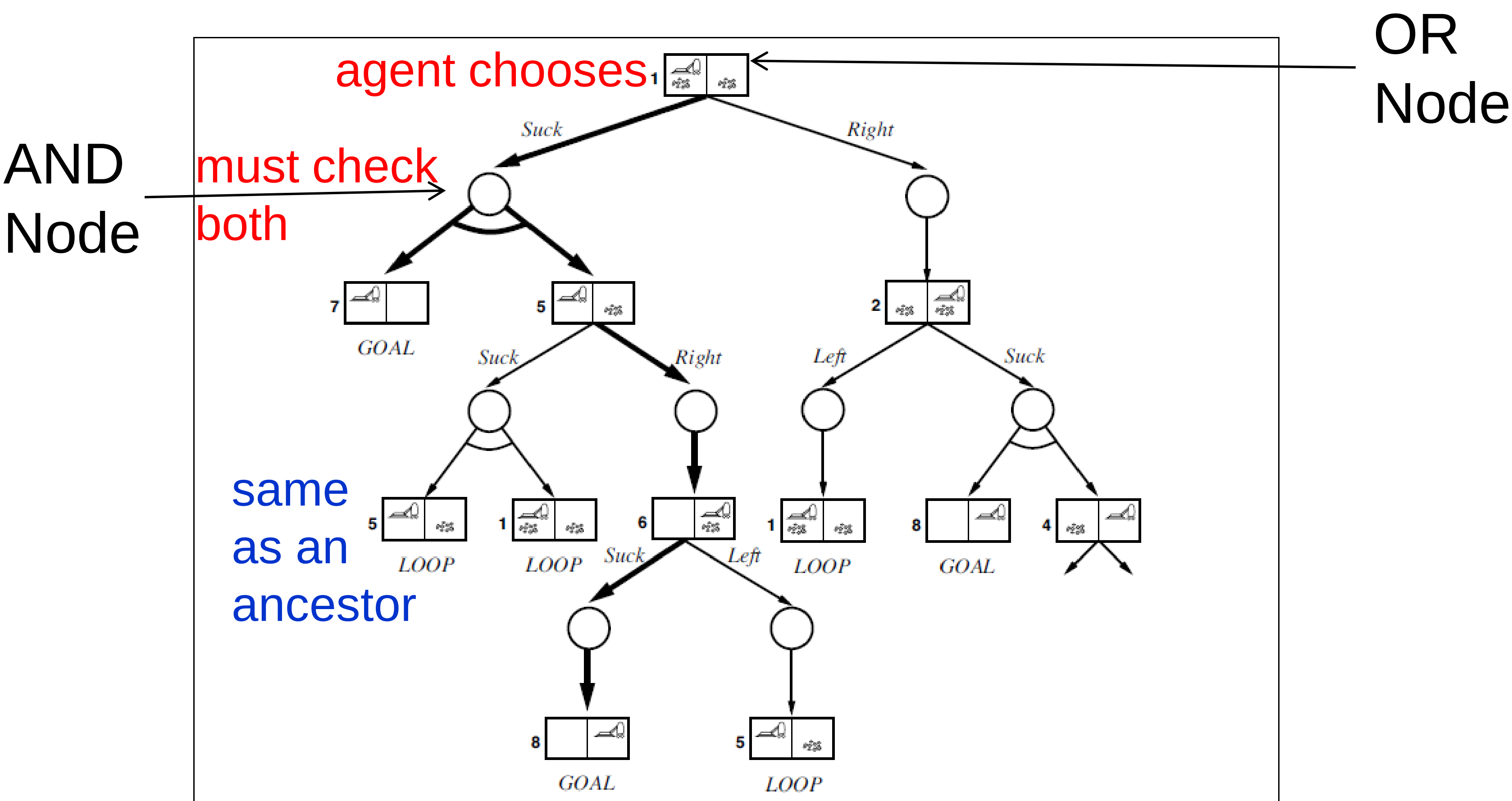


Figure 4.10 The first two levels of the search tree for the erratic vacuum world. State nodes are OR nodes where some action must be chosen. At the AND nodes, shown as circles, every outcome must be handled, as indicated by the arc linking the outgoing branches. The solution found is shown in bold lines.

Searching with Partial Observations

- The agent does not always know its state!
- Instead, it maintains a belief state: a set of possible states it might be in.
- Example: a robot can be used to build a map of a hostile environment. It will have sensors that allow it to "see" the world.

Belief State Space for Sensorless Agent

Knows it's
on the left

Knows it's
on the right.

Knows left
side clean

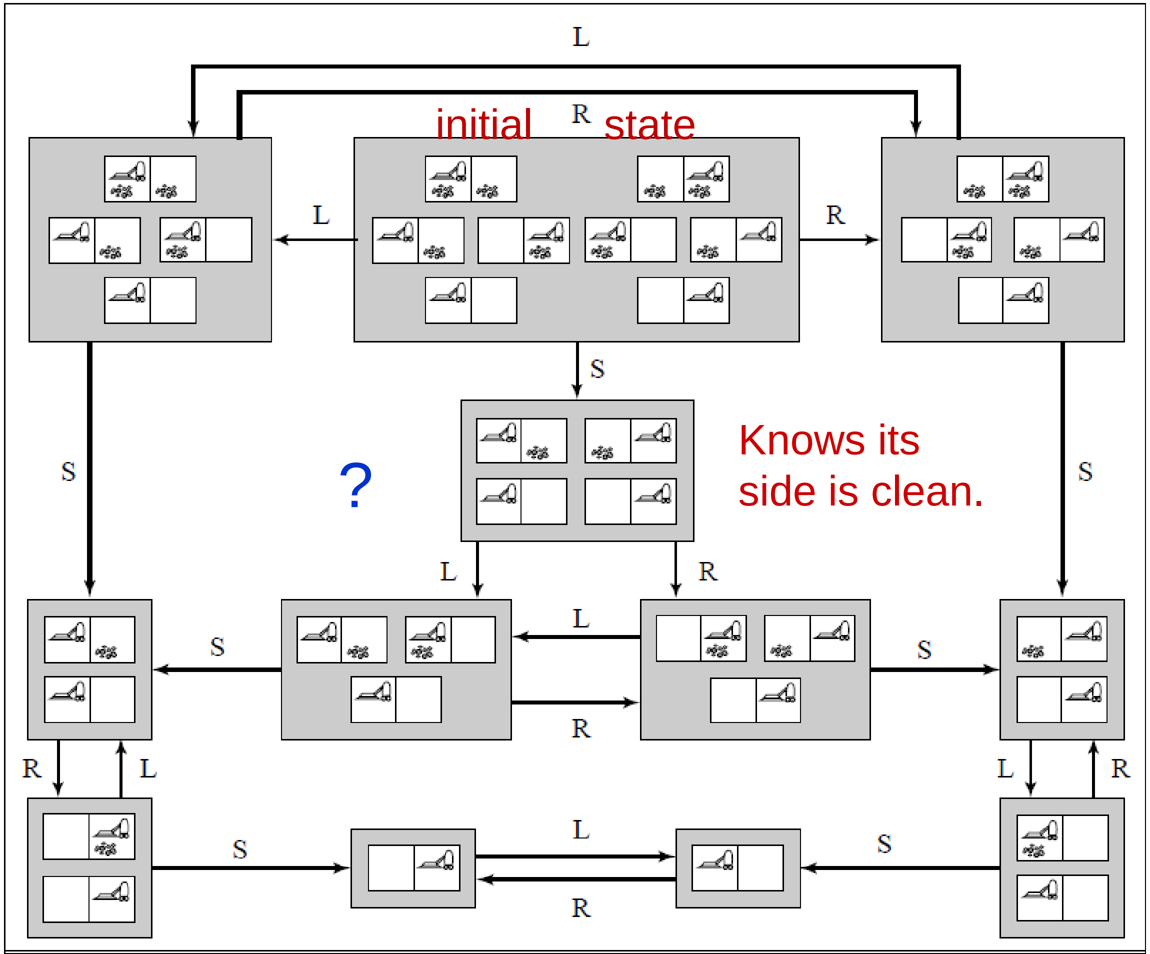


Figure 4.14 The reachable portion of the belief-state space for the deterministic, sensorless vacuum world. Each shaded box corresponds to a single belief state. At any given point, the agent is in a particular belief state but does not know which physical state it is in. The initial belief state (complete ignorance) is the top center box. Actions are represented by labeled links. Self-loops are omitted for clarity.

Online Search Problems

- Active agent
  - executes actions
  - acquires percepts from sensors
  - deterministic and fully observable
  - has to perform an action to know the outcome
- Examples
  - Web search
  - Autonomous vehicle