COMP3411/9814 Revision

Exam

- 2 May 2023
- Online – Moodle
- Reading time starts 1:45pm
- Open for answers 2pm to 4pm (ELS students have extra time)
- 20 multiple choice questions (very similar to quiz questions)
- 6 multi-part written questions (max 16 sub-parts)
  - Short answer, some calculation
  - Not more complex than multiple choice, so don’t panic.
- Supplementary exams in week 22-26 May (exact date not set yet)

Search

- What are the different types of search algorithms?

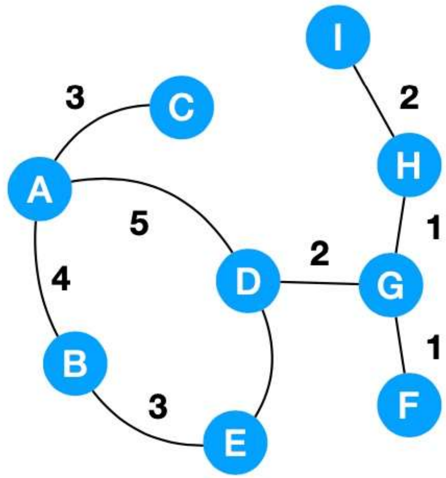
- What are their properties?
  - Order of search
  - Optimality
  - Complexity

- Can you demonstrate different search algorithms on a simple example?

Example Question

- Given Start (A) and Goal (I) nodes,
  can you work out the order of nodes visited in:
- A) Depth-first search
- B) Breadth-first search
- C) Iterative Deepening
- D) Uniform Cost Search
- E) A* Search
- F) Greedy Search

Node	H
A	10
B	13
C	5
D	7
E	15
F	6
G	5
H	2
I	0



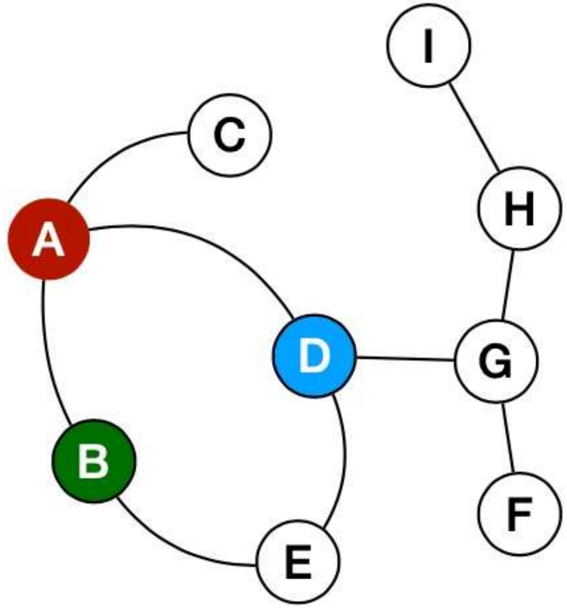
CSP

- What are the different CSP strategies?
- e.g. forward checking, arc consistency

- Can you apply them on a small example?

Example Question

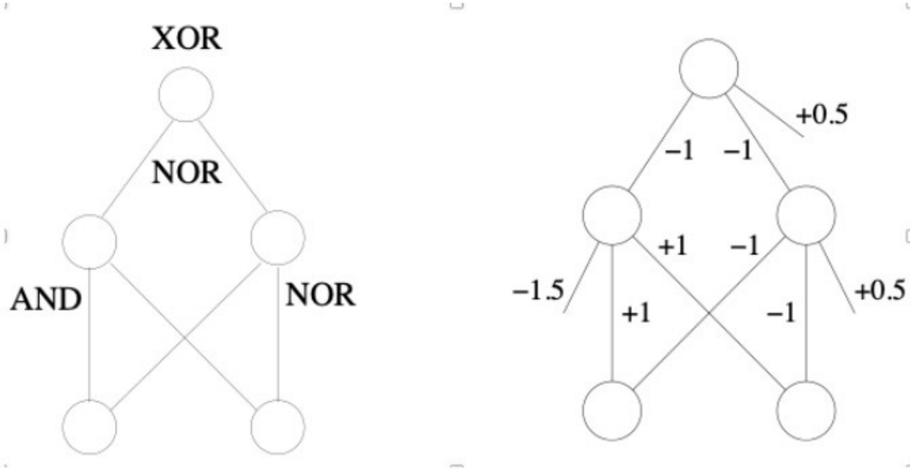
- In map colouring, can you apply forward checking to a graph?
- If you are given a partial assignment of variables, can you work out the remaining legal values by forward checking or arc consistency?



Neural Networks

- Given a network and an activation function, can you propagate value through a network to get an output (i.e. forward pass)?
- If you are given an incomplete network and an activation function, can you find the value for the missing weights so the network returns the expected output?
- Can you apply the perceptron training rule?

Example question



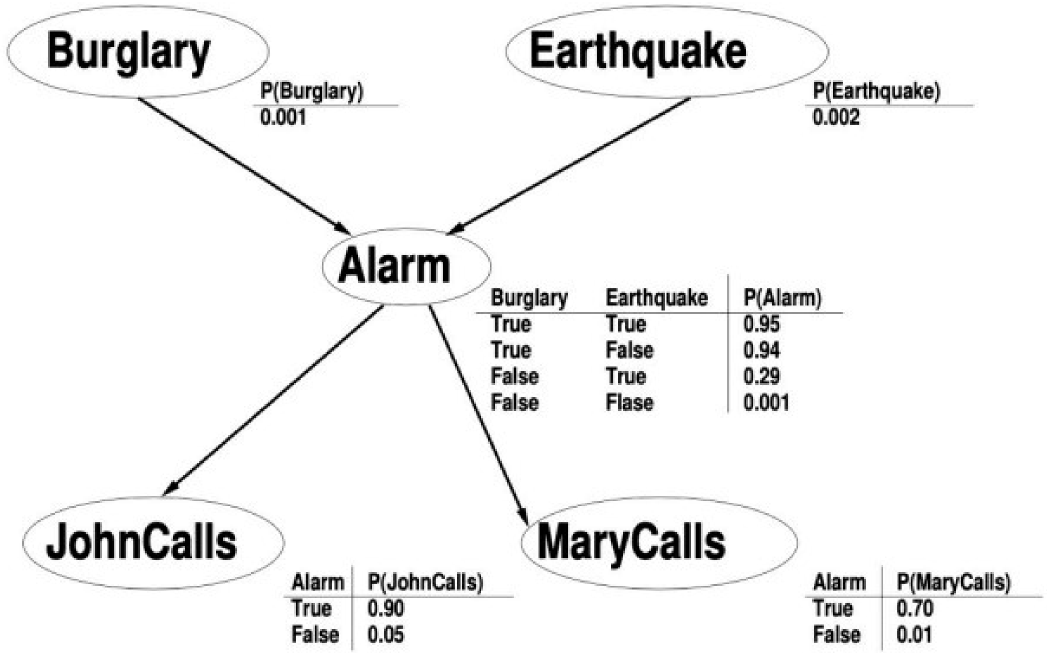
- Can you compute the output of a multilayer perceptron, given a simple activation function?

Uncertainty

- Given a joint probability distribution table, can you find the the probability of a proposition?
- Given, P(A), P(B) and P(A|B), can you calculate P(B|A)?
- Given a Bayesian network, can you calculate different probabilities using causal inference or diagnostic inference?

Example question

- Can you calculate
  P(E|~A)?



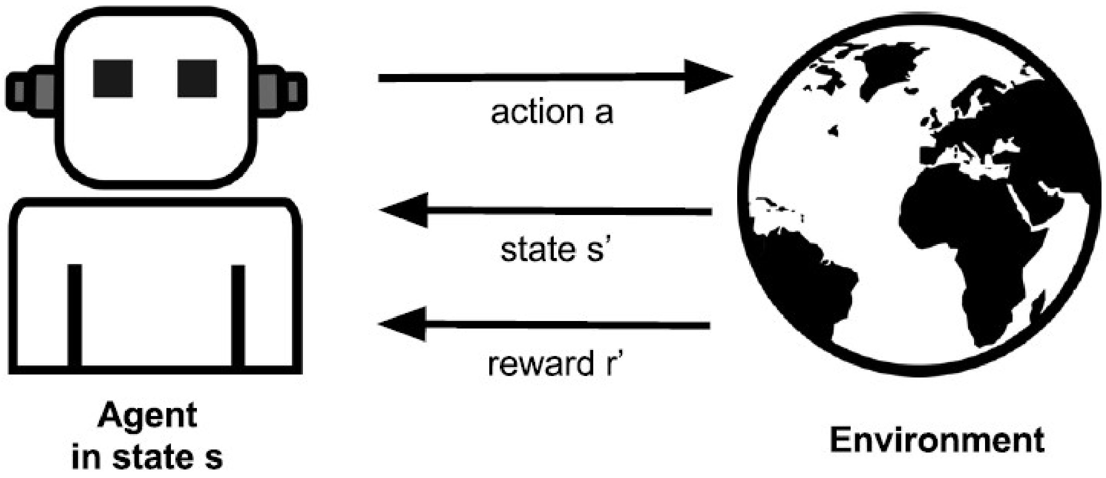
Reinforcement Learning and Robot Vision Recap

COMP3411/9814: Artificial Intelligence

Reinforcement Learning

COMP3411/9814: Artificial Intelligence

Reinforcement Learning



Elements of reinforcement learning

There are four essential elements:

- **Policy**

  - Informs how to act in a particular situation.
  - Set of stimulus-response rules or associations.
  - Can be stochastic.

Elements of reinforcement learning

There are four essential elements:

- **Reward function**

    - Defines the aim of an RL problem.
    - Maps each perceived state (or state-action pair) into a number, the reward.
    - The goal is to maximize the long-term reward.
    - In biological systems may correspond to pain and pleasure feelings.
    - Can be stochastic.

Elements of reinforcement learning

There are four essential elements:

- **Value function**

  - Shows what is good in the long run (the reward in an immediate sense).
  - In biological systems corresponds to more refined judgments or foresight about the future from one state.
  - Actions are decided based on the value.
  - It’s much harder to determine values than rewards.

Elements of reinforcement learning

There are four essential elements:

- **Optionally, a model of the environment**

    - Imitates the environment behaviour.
    - Can predict states and reward obtained.
    - The use of models of the environment is still relatively new.

Action-value estimation methods

Greedy method

- The simplest way to choose an action: the action with the highest estimated value.
- A_t* where $Q_t(A_t^*) = \max_a Q_t(a)$.

ε-greedy method

- A simple alternative: to choose the best action most of the time, and sometimes (with a small probability ε) a random one.
- $Q_t(a)$ converges to $q_*(a)$ with probability 1- ε.

Action-value estimation methods

Softmax method

- ε-greedy effectively trades off exploration and exploitation, but the selection is equitable (or fair) among actions.
- Sometimes, the worst action is very bad.
- Softmax uses action probabilities as a Boltzmann distribution.

$$\frac{e^{Q_t(a)/\tau}}{\sum_{i=1}^{n} e^{Q_t(i)/\tau}}$$

Action-value estimation methods

Softmax method

- High temperatures give almost equal probability for all actions.
- Low temperatures make a bigger difference in the probability.
- It is not clear if softmax or ε-greedy performs better.
- However, softmax shows dissimilar performance making difficult the selection of the temperature.

$$\frac{e^{Q_t(a)/\tau}}{\sum_{i=1}^{n} e^{Q_t(i)/\tau}}$$

Action-value estimation methods

- In sumary, two possible action selection methods (among many others):
- ϵ-greedy method:

$$P(s_t, a) = \begin{cases} 1 - \epsilon & \text{if } a = \underset{a_i \in A(s_t)}{argmax}\ Q(s_t, a_i) \\ \epsilon & \text{otherwise} \end{cases}$$

- Softmax method:

$$P(s_t, a) = \frac{e^{Q(s_t,a)/T}}{\sum_{a_i \in A} e^{Q(s_t,a_i)/T}}$$

- Given a set of Q values, could you calculate the selection for all Q values using either ε-greedy or softmax methods?

Incremental implementation

- A simple implementation: record all the rewards.

$$Q_t(a) = \frac{R_1 + R_2 + \cdots + R_{K_a}}{K_a}$$

- Problem: growing use of memory and computational cost over time.

Incremental implementation

- Denote Q_k the estimated reward at time-step k, i.e., the average of the k-1 first rewards, then:

$$
\begin{aligned}
Q_{k+1} &= \frac{1}{k}\sum_{i=1}^{k} R_i \\
&= \frac{1}{k}\left(R_k + \sum_{i=1}^{k-1} R_i\right) \\
&= \frac{1}{k}\Big(R_k + (k-1)Q_k\Big) \\
&= \frac{1}{k}\Big(R_k + kQ_k - Q_k\Big) \\
&= Q_k + \frac{1}{k}\Big[R_k - Q_k\Big],
\end{aligned}
$$

Returns

- If the reward sequence received is R_{t+1}, R_{t+2}, R_{t+3}, …. We want to maximise the expected return G_t.

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T$$

- In tasks with final state and that can be divided into subsequences (episodes)
- Each episode finishes in the final state and the task starts over from an initial state.
- These tasks are known as episodic tasks.

- Can you calculate a Q value based on average reward?

Returns

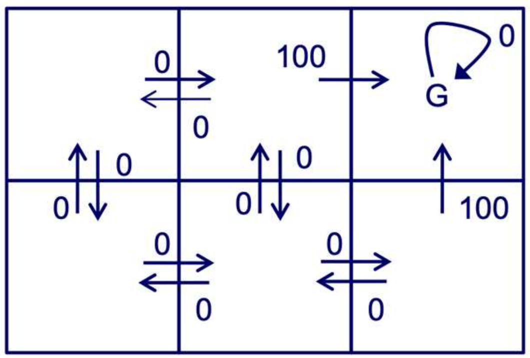
- Tasks intended to be performed continuously without limit are referred to as continuous tasks (or non-episodic).
- The return could be infinite, given that $T = \infty$.
- In this case, the agent maximises the discounted rewards, choosing actions to maximise the discounted return:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \quad = \quad \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

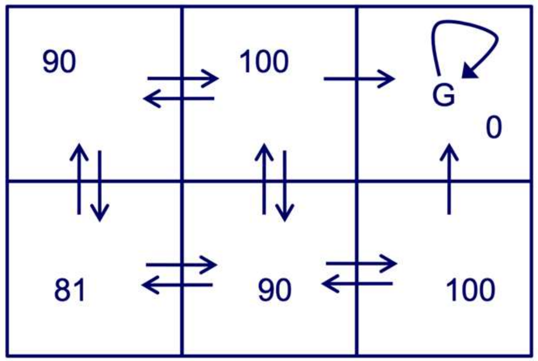
- Discount rate $0 \leq \gamma < 1$. Determine the present value of future rewards. If $\gamma = 0$, the agent is myopic. If $\gamma \rightarrow 1$ the agent is foresighted.

- Can you calculate G_i for a give i?

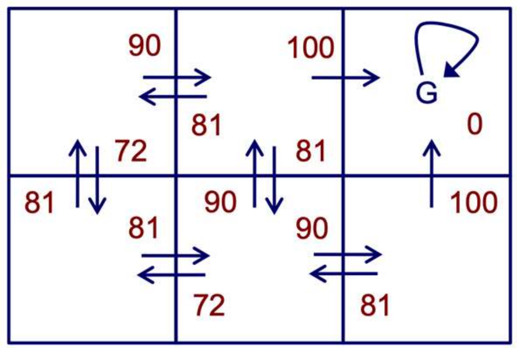
Q-values



$r(s, a)$ (immediate reward) values



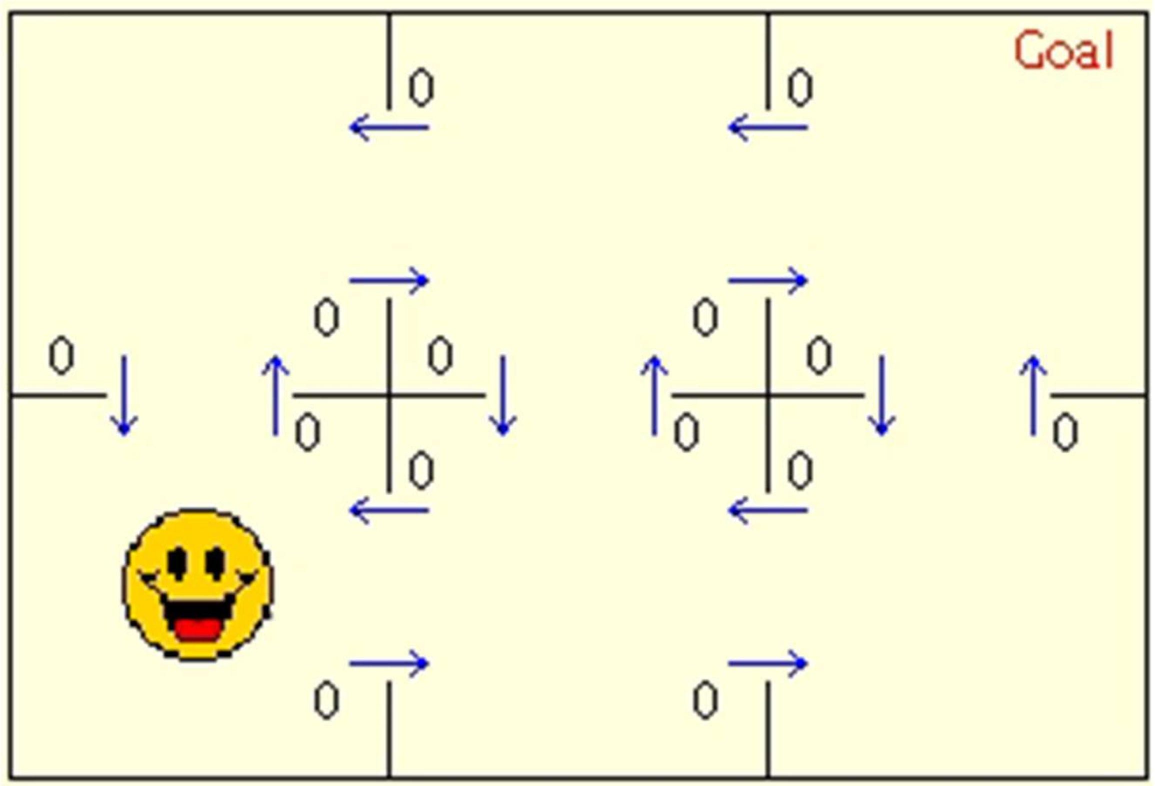
$V^*(s)$ values



$Q(s, a)$ values

$$\gamma = 0.9$$

Grid world example



- Can you calculate the sequence a Q table of values for each action?

Sarsa: On-Policy TD Control

- Updates after each transition from a non-terminal S_t.
- If S_{t+1} is terminal, $Q(S_{t+1}, A_{t+1})$ is defined as zero.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\Big]$$

- Each element of the 5-tuple $(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1})$ is used, this gives the name to the algorithm.
- On-policy methods continuously estimate q_π for policy π, and at the same time change π greedily towards q_π.

Q-Learning: Off-Policy TD Control

- A simple but important breakthrough is an off-policy TD algorithm.
- The simplest way is *one-step Q-learning*:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \Big]$$

- The learned action-value function Q directly approximates q*, the optimal action-value function, regardless the followed policy π.
- The policy still has an effect in which state-action pairs are visited and updated.

- Can you calculate the sequence a Q table of values for each action?
- How about for SARSA?

Robot Vision

Image processing: Averaging

- It can use a threshold.
- Larger rectangles achieve more smoothing.
- Broad lines are thickened and thin lines eliminated

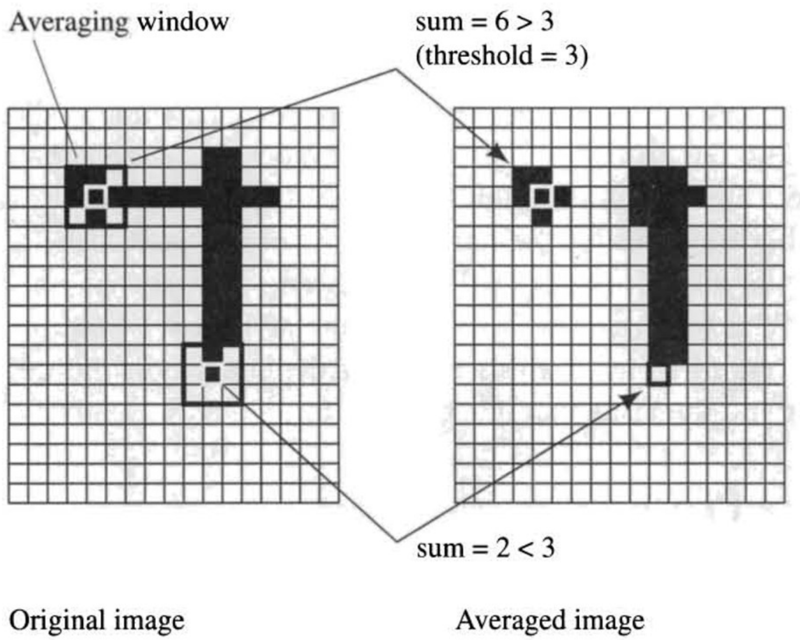


Image processing: Averaging

- Given a simple 4 x 4 picture matrix:

9	9	9	3
9	9	3	3
9	3	3	3
3	3	3	3

- Smooth this matrix using an averaging technique and a 3 x 3 pixel window.

Image processing: Averaging

- There are four 3 x 3 pixel windows in the matrix.

- Replace middle value in each window by average of all the values in the window.

$$\begin{matrix} 9 & 9 & 9 & 3 \\ 9 & 9 & 3 & 3 \\ 9 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \end{matrix} \longrightarrow \begin{matrix} 9 & 9 & 9 & 3 \\ 9 & 7 & 5 & 3 \\ 9 & 5 & 4 & 3 \\ 3 & 3 & 3 & 3 \end{matrix}$$

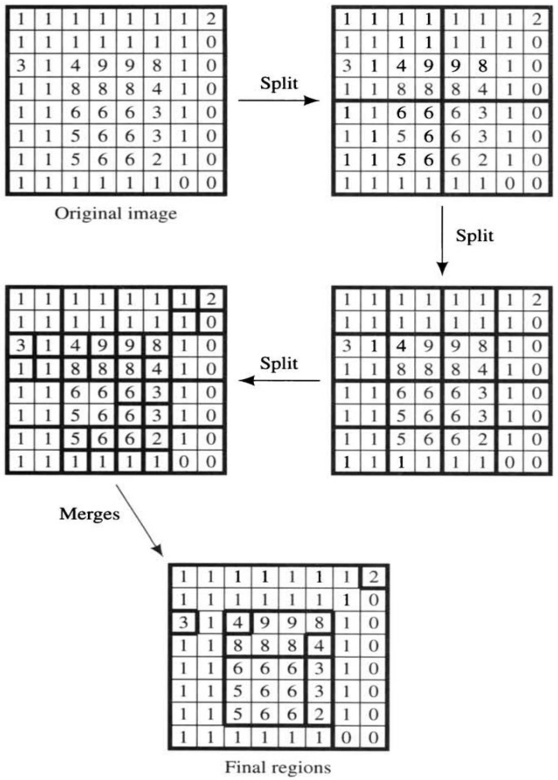
- Given a really small image can you demonstrate how smoothing works?

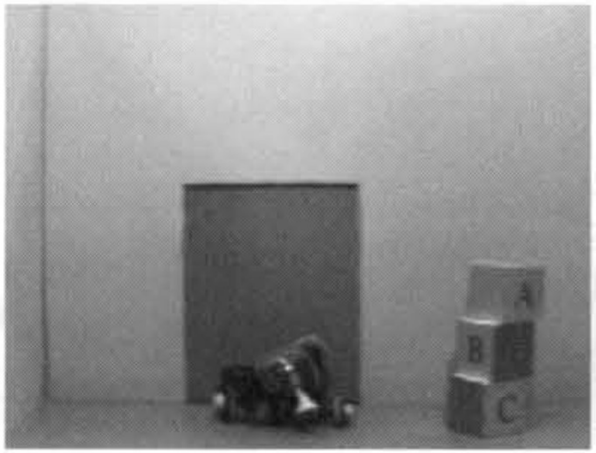
Image processing: Region finding

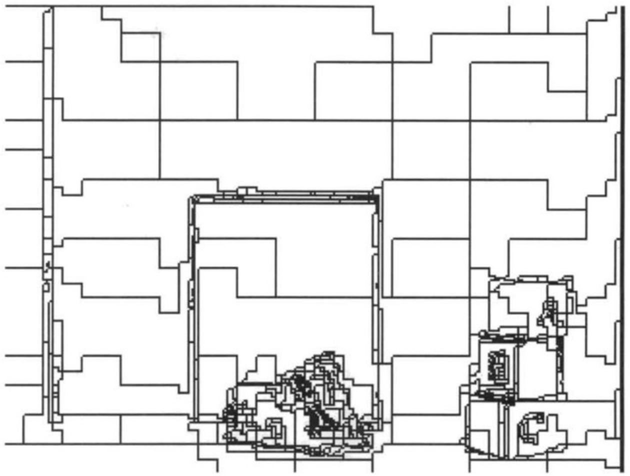
- To find regions in which a property does not change abruptly.
- A region is homogeneous. Intensity difference no more than some ε
- Split-and-merge method. $2^n \times 2^n$ array of pixels.
  - Each no homogeneous region is split in four.
  - Splits continues until no more splits need to be made.
  - Adjacent regions are merged if homogeneous.

Image processing: Region finding

- Splitting and merging candidate regions.
- In this example, intensities may not vary more than 1 unit. Therefore, ε <= 1.







- Given a really small image know how region splitting and merging works?

Stereo vision

- Under perspective projection large, distant objects might produce the same image as similar but smaller, closer ones.
- Distance estimation from single images is problematic, but sometimes possible.
- e.g., If we know an object is on the floor and the camera height.

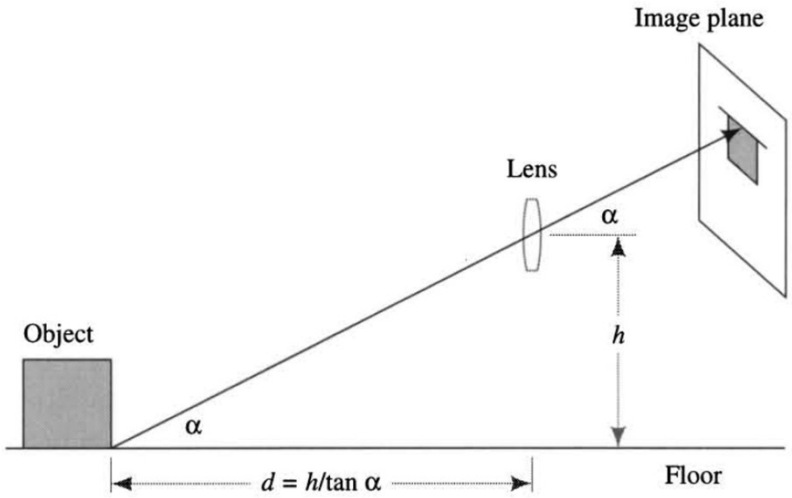


Figure 6.20

Depth Calculation from a Single Image

Stereo vision

- Depth information from stereo vision.
- Two-dimensional setup.
- Two lenses with distance b.
- Correspondence problem for pairs of points.

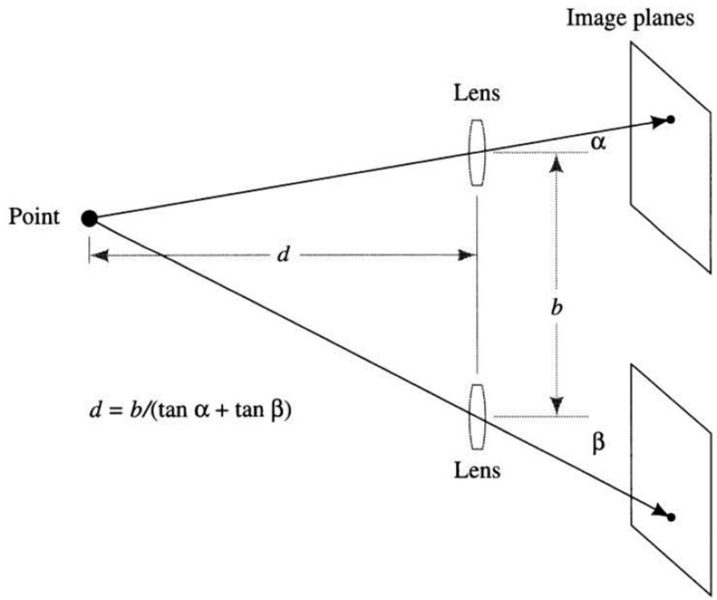


Figure 6.21

Triangulation in Stereo Vision

- Can you calculate the distance to an object from the disparity data (angles and distance between cameras)?

Logic and Automated Reasoning

- Can you translate English sentences to and from expressions in propositional and first order logic?
- Can you work out the truth table of an expression in propositional logic?
- Do you know what soundness, completeness and decidability mean?
- Can you show something like:

$$P \wedge (Q \rightarrow R) \Leftrightarrow \neg(P \rightarrow Q) \vee (P \wedge R)$$

Do you remember these identities?

Commutativity:	$p \wedge q \leftrightarrow q \wedge p$	$p \vee q \leftrightarrow q \vee p$
Associativity:	$p \wedge (q \wedge r) \leftrightarrow (p \wedge q) \wedge r$	$p \vee (q \vee r) \leftrightarrow (p \vee q) \vee r$
Distributivity:	$p \wedge (q \vee r) \leftrightarrow (p \wedge q) \vee (p \wedge r)$	$p \vee (q \wedge r) \leftrightarrow (p \vee q) \wedge (p \vee r)$
Implication:	$(p \rightarrow q) \leftrightarrow (\neg p \vee q)$	
Idempotent:	$p \wedge p \leftrightarrow p$	$p \vee p \leftrightarrow p$
Double negation:	$\neg \neg p \leftrightarrow p$	
Contradiction:	$p \wedge \neg p \leftrightarrow \text{FALSE}$	
Excluded middle:		$p \vee \neg p \leftrightarrow \text{TRUE}$
De Morgan:	$\neg (p \wedge q) \leftrightarrow (\neg p \vee \neg q)$	$\neg (p \vee q) \leftrightarrow (\neg p \wedge \neg q)$

Can you work out what the models of a formula are?

$S \vDash P$

Each row is an interpretation of *S*.
Only the first row is a model of *S*.

$\boldsymbol{S} = \{\mathbf{p} \rightarrow \mathbf{q}, \mathbf{q} \rightarrow \mathbf{p}, \mathbf{p} \lor \mathbf{q}\}$
$\boldsymbol{P} = \mathbf{p} \land \mathbf{q}$

p	q	p → q	q → p	p ∨ q	*S*	p ∧ q
T	T	T	T	T	T	T
T	F	F	T	T	F	~~F~~
F	T	T	F	T	F	~~F~~
F	F	T	T	F	F	~~F~~

Proofs (propositional and first-order)

- Can you take any propositional or first order expression and turn it into conjunctive normal form?

- Given a knowledge base and a query, can you show all the steps in a resolution proof of the query?

Prolog

- Can you write a simple Prolog program?

- E.g. merge two sorted lists:

```
?- merge([1, 3, 7, 15], [5, 9, 13], X).
X = [1, 3, 5, 7, 9, 13, 15]
```

Planning

- Can you write the pre-conditions and postconditions of an action in STRIPS (PDDL) representations (i.e. preconditions and effects).
- Do you know the difference between a forward chaining and backward chaining planner?
- Given a simple planning problem, can you draw a graph showing how a partial order planner works?
- Can you explain how constraint solving can improve the efficiency of a planning algorithm?

Inductive Logic Programming

- Given two definite clauses, can find the least general generalisation?
- Can you explain why intra_construction, absorption and truncation are generalisation operations?

Decision Trees

- Can you calculate the entropy resulting from split so that you can choose the most informative attribute to split on?
- Can you calculate the Laplace error and determine when to stop growing a branch in a decision tree?
- What is k-fold cross-validation and how is it used to estimate the classification accuracy of a machine learning algorithm?

NLP

- Given a simple grammar, can you determine if an input sentence is recognised by the grammar?
- Given a simple grammar, can you generate a sentence that is recognised by the grammar?
- Can you demonstrate a left-most or right-most derivation?
- Given a definite clause grammar can you show what the output would be after recognising an input sentence?