

université de BORDEAUX	<p style="text-align: center;">ANNEE UNIVERSITAIRE 2015 / 2016 SESSION 1 DE PRINTEMPS</p> <p>PARCOURS : L3 Informatique Code UE : J11N6014 (Epreuve Image)</p> <p>Date : 25 avril 2016 Heure : 14h Durée : 1h30</p> <p>Documents : autorisés. Epreuve de Mme : C. Blanc. Consignes : Temps recommandé 1h30 sur 3h. Vous devez répondre directement sur le sujet qui comporte 6 pages. Insérez ensuite votre réponse dans une copie d'examen comportant tous les renseignements administratifs</p>	<p style="text-align: center;">Collège Sciences et Technologies</p>
-----------------------------------	--	---

Numéro d'anonymat :

Questions de cours (5 pts) :

1. On souhaite réaliser sur une image en niveaux de gris une quantification en 6 niveaux de gris et une inversion vidéo. De quel type de transformation s'agit-il ? Dessinez la courbe permettant de réaliser cette transformation.

2. Parmi les différents domaines vus en cours et en TD, citez deux exemples de méthodes basées sur les caractéristiques de l'œil humain.

3. En analyse d'images pour détecter les contours le masque de Prewitt donne de meilleurs résultats que le masque de Sobel. Expliquez précisément pourquoi ?

4. Dans le domaine de l'image numérique, quel est le nombre de pixels voisins d'un pixel ? Justifiez

5. En synthèse d'images quel est l'intérêt d'utiliser les coordonnées homogènes pour réaliser les transformations géométriques utiles lors de l'étape de visualisation ?

Exercice 1 (5 pts) :

On considère une image en « vraies couleurs RGB » de taille $x*y$ pixels.

1. Indiquez en le justifiant quel est l'espace mémoire nécessaire au stockage de cette image ?

2. Sur cette image pour chaque pixel P de couleur (R_P, G_P, B_P) on calcule la moyenne $M_P = (R_P + G_P + B_P)/3$ et on remplace la couleur de P par le triplet (M_P, M_P, M_P) .
- a. Quel sera le résultat obtenu ?

Numéro d'anonymat :

- b. Quelle est l'espace mémoire nécessaire au stockage de l'image après cette transformation ?

- c. Expliquer les défauts de cette transformation et proposez une amélioration.

3. On souhaite maintenant obtenir une représentation en noir et blanc de l'image. Parmi les méthodes vues en cours :

- a. Indiquez celle(s) qui permettra d'optimiser l'espace mémoire nécessaire au stockage de l'image finale

- b. En le justifiant, classez les différentes méthodes de binarisation de la plus rapide à la plus lente.

Exercice 2 (5 pts) :

1. Rappelez le principe de l'algorithme de binarisation de Floyd & Steinberg.

2. Précisez quelles sont les variantes possibles lors de l'application de cet algorithme ainsi que leurs conséquences.

3. Expliquez pourquoi cet algorithme est un algorithme de binarisation global.

4. Lorsqu'on applique cet algorithme sur l'image 1 ci-dessous, des pixels blancs apparaissent dans les zones sombres comme on peut le voir sur l'image 2, expliquez pourquoi.

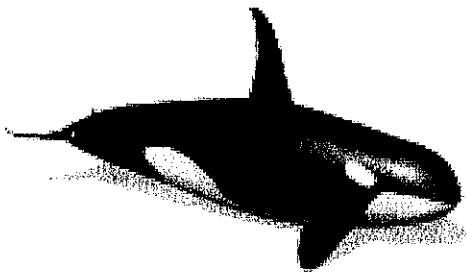


Image 1

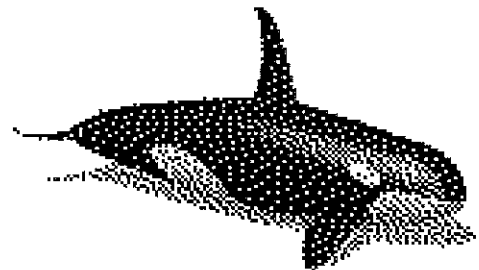


Image 2

Numéro d'anonymat :

Exercice 3 (5 pts) :

POV-Ray est un logiciel de synthèse d'images permettant de générer une image à partir de fichiers sources contenant la description d'une scène 3D. Le rendu consiste en l'image de la scène perçue par une caméra. Ainsi, pour définir une scène 3D, on doit placer dans un repère 3D: une caméra, des sources de lumière, les objets définissant la scène ainsi que les éventuelles constructions géométriques

On doit aussi définir les propriétés optiques de chacun des objets de la scène. Ce qui inclus, entre autres: l'intensité des sources lumineuses, la quantité de lumière que les objets vont réfléchir, la couleur, le degré de transparence, la brillance, la rugosité d'un objet, etc.

L'extrait de fichier POV-Ray ci-dessous décrit la géométrie et la modélisation optique de plusieurs objets utilisés pour représenter les images 1, 2 et 3.

```
//Obj 1
sphere { <-2.0, 0.7, -1>, 0.7
  finish {
    ambient 0.2
    diffuse 0.8
    specular 0.2
    phong 50 }
  pigment { color red 1 green 0 blue 0 }
}

//Obj 2
sphere { <1.3, 0.9, -1.75>, 0.7
  finish {
    ambient 0.2
    diffuse 0.8
    specular 1
    phong 1 }
  pigment { color red 0 green 1 blue 0 }
}

//Obj 3
sphere { <-0.4, 1.8, -1.5>, 0.7
  finish {
    ambient 0.2
    diffuse 0.8
    specular 0.2
    phong 1 }
  pigment { color red 0 green 0 blue 1 }
}

//Obj 4
sphere { <0.0, 0.0, 0.0>, 2
  finish {
    ambient 0.2
    diffuse 0.8
    specular 0.2
    phong 1 }
  pigment { color red 0.5 green 0 blue 0.5 }
}

// Obj 5
box { <-2.0, -0.2, -2.0>, <2.0, 0.2, 2.0>
  finish {
    ambient 0.2
    diffuse 0.8
  }
  pigment { color red 1 green 1 blue 0 }
}
```

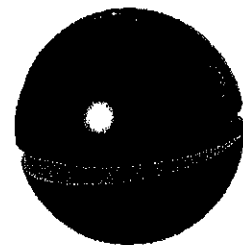


Image 1

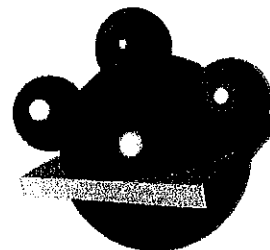


Image 2

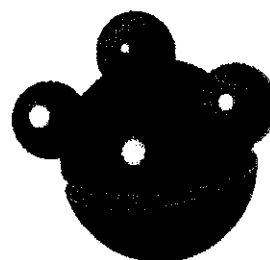
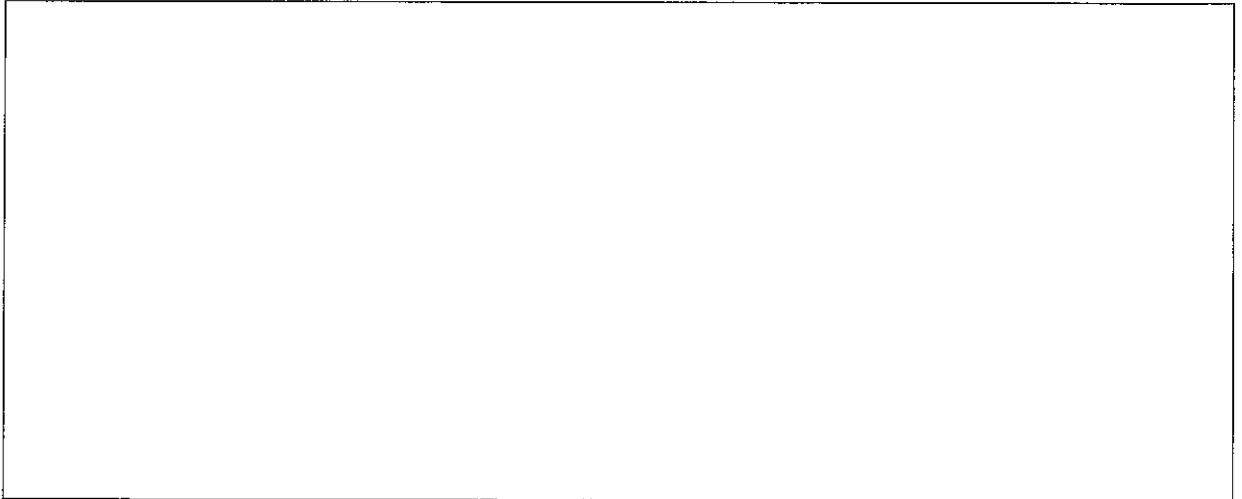


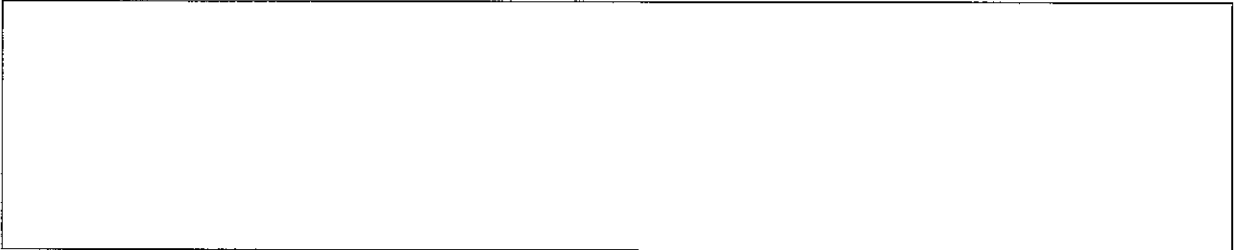
Image 3

Pas de panique je sais qu'à priori vous ne connaissez pas POV-Ray. Pour répondre aux questions suivantes seuls votre cours et un bon sens de l'observation sont utiles.

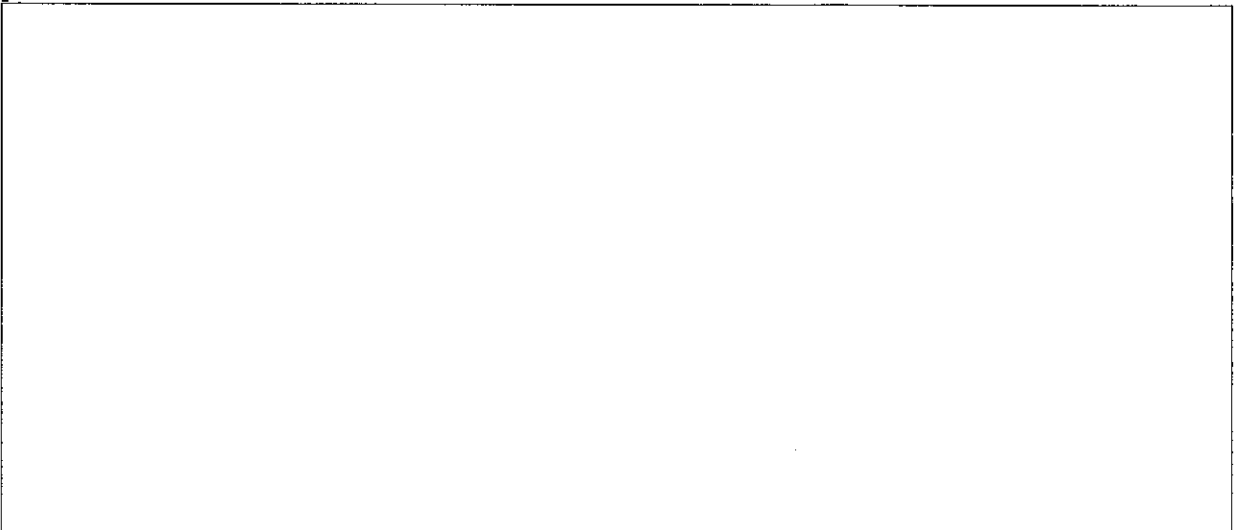
1. Sur l'image 2 indiquez où se trouve chacun des 5 objets.
2. En comparant les effets visuels sur les différentes sphères expliquez le rôle de la composante « phong » dans la description optique des objets.



3. Quelle est le nom de la méthode de modélisation géométrique employée pour construire les objets représentés sur les images ?



4. Pour chaque image expliquez comment l'objet rendu a été construit avec la méthode précédente.



Année Universitaire	2015-2016	Période	Session de Printemps
Code Etape	MIAGE L3	Code UE	J1IN6026
Nom de l'Epreuve	Utilisation des Réseaux		
Date / Heure	03/05/2016 à 14h00	Durée	1h30
Documents	Non autorisés	Calculatrice	Non
Nombre de pages	3	Enseignant	A. Esnard

Nota Bene : Le barème est donné à titre indicatif.

Ex 1 (Cours, 2 pt)

1. Expliquez brièvement les différentes couches du modèle standard TCP/IP, en citant des exemples de protocole à chaque niveau.
2. Expliquez brièvement le rôle du protocole ICMP. A quelle couche appartient ce protocole ?

Ex 2 (Code de Hamming, 4 pt)

On considère le code de Hamming (11,7) étudié en cours. On détaillera tous les calculs.

1. Quel est le code de Hamming correspondant au mot 1110001 ?
2. Quel est le message correspondant au code 10001110101 ? Y a-t-il une erreur ? Si oui, corrigez-la (si possible).

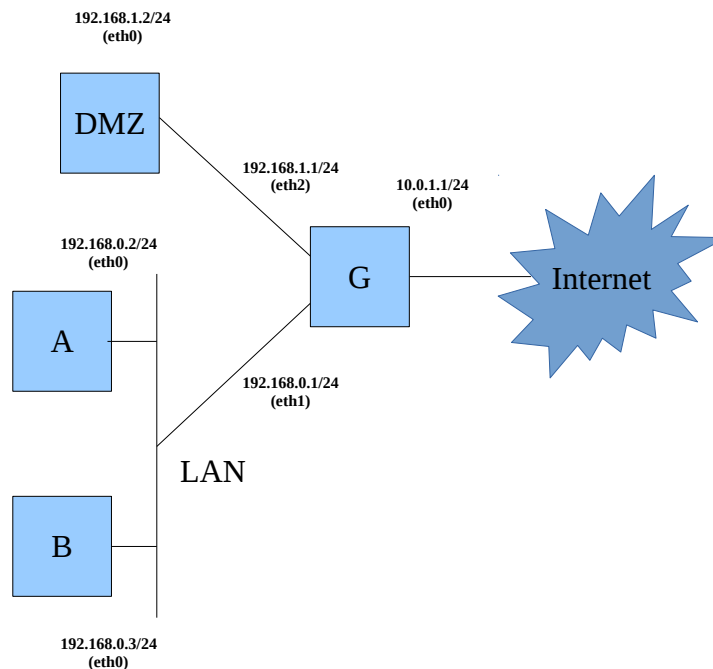
Ex 3 (Gigabit Ethernet, 4 pt)

Deux PCs sont reliés à un réseau par une liaison Gigabit Ethernet (débit à 1 Gb/s). Le support de transmission est un câble Ethernet classique sur lequel le signal électrique circule à 200 000 km/s. Ce réseau supporte des paquets IP de taille 576 octets.

1. Quel temps faut-il pour émettre un paquet IP ? On rappelle que la couche Ethernet rajoute 18 octets à la taille du paquet IP.
2. Les deux PCs sont à une distance raisonnable de 10m. Quelle est la durée de propagation du signal ?
3. Pourquoi le protocole CSMA/CD impose-t-il une taille minimale des trames Ethernet ?
4. Les câbles Gigabit Ethernet classiques (norme 1000Base-T) supportent une longueur maximale de 100m. En déduire la taille minimale des trames Giga Ethernet.

Ex 4 (Administration, 10 pt)

On souhaite mettre en place un réseau domestique connecté à Internet par la passerelle G et organisé selon le schéma ci-dessous. La machine DMZ dispose d'un serveur SSH (port 22) et les machines du LAN utilisent régulièrement le web.



Pour chaque machine (sauf B), on demande de lister les commandes Linux nécessaires pour réaliser les tâches suivantes :

1. Configurer les IP.
2. Configurer le routage pour accéder à Internet.
3. Configurer le firewall sur la passerelle G conformément aux besoins énoncés.
4. Configurer le NAT dynamique sur la passerelle G. En quoi, est-ce utile ?
5. Configurer le port forwarding de G:2222 vers DMZ:22. En quoi, est-ce utile ?

Nota Bene : Vous disposez d'une annexe pour vous aider.

Annexes

Memento Routage

- Activer le routage sur une machine (ip forward) : `echo 1 > /proc/sys/net/ipv4/ip_forward`
- Afficher la table de routage : `route -n`
- Définir une route par défaut : `route add default gw <@gateway>`
- Ajouter une route vers un réseau : `route add -net <@network> netmask <mask> gw <@gateway>`
- Ajouter une route vers une machine particulière : `route add -host <@host> gw <@gateway>`
- Pour supprimer une règle, il faut taper la commande `route del <...>` avec exactement les mêmes arguments que pour la commande `add`.

Memento Firewall

Voici quelques notes concernant l'utilisation d'iptables pour configurer un firewall. La configuration du firewall se base sur la table "filter" et est subdivisée en 3 chaînes (notée <CHAIN>) : INPUT : tout ce qui rentre dans la machine ; OUTPUT : tout ce qui sort dans la machine ; FORWARD : tout ce qui traverse la machine (i.e. lors du routage).

- Pour afficher les règles de la table filter : `iptables -t filter -L`
- Pour effacer toutes les règles ajoutées : `iptables -t filter -F`
- Pour chaque règle que l'on ajoute, trois actions sont possibles (notée <ACTION>) :
 - ACCEPT : on accepte ;
 - REJECT : on rejette poliment (réponse d'erreur envoyé à l'émetteur) ;
 - DROP : on jette à la poubelle (pas de réponse d'erreur).
- Pour modifier la politique par défaut du firewall : `iptables -t filter -P <CHAIN> <ACTION>`
- Pour ajouter une nouvelle règle à une chaîne du firewall (attention à l'ordre des règles) : `iptables -t filter -A <CHAIN> <SRC> <DST> <...> -j <ACTION>`
 - avec <SRC> des indications sur la provenance des paquets IP, comme par exemple `"-i eth0"` ou `"-s 192.168.0.0/24"` ou encore `"-s 0/0"` ;
 - avec <DST> des indications sur la destination des paquets IP, comme par exemple : `"-o eth1"` ou `"-d 147.210.0.0/24"` ;
 - avec <...> des infos complémentaires sur par exemple la nature du protocole `"-p icmp"` ou `"-p tcp"`, avec éventuellement des précisions spécifiques à ces protocoles (`"-dport 80"` pour TCP) ou encore sur l'état `"-m state -state NEW"`, ...

Memento NAT & Port Forwarding

- NAT Dynamique : `G$ iptables -t nat -A POSTROUTING -o <iface> -j MASQUERADE` (avec <iface> l'interface réseau permettant d'accéder à Internet)
- Port forwarding : `G$ iptables -t nat -A PREROUTING -i <iface> -p tcp --dport <P> -j DNAT --to <S>:<Q>` (redirige le trafic de la connexion TCP/IP sur G:P vers S:Q avec G et S des adresses IP et P et Q des numéros de port)

Examen Informatique Théorique 2

Documents non autorisés

1 Minimisation d'automates

Soit l'automate $M = (Q, \Sigma, \delta, \{q_0\}, F)$ où $Q = \{1, 2, 3, 4\}$, $\Sigma = \{a, b\}$, $q_0 = 1$, $F = \{1, 2, 4\}$ et $\delta = \{(1, a, 3), (1, b, 2), (2, a, 4), (2, b, 2), (3, a, 3), (3, b, 1), (4, a, 3), (4, b, 2)\}$.

- Représenter l'automate sous la forme d'un graphe orienté étiqueté. Cet automate est-il complet ?
- Quel est le langage reconnu par cet automate ?
- Cet automate est-il déterministe ? Si nécessaire, déterminer-le.
- Minimiser l'AFD. Quel est le langage reconnu par l'automate minimal obtenu ?

2 Des expressions aux automates - l'algorithme de Glushkov

En utilisant l'algorithme de Glushkov, représenter un automate (sous la forme d'un graphe orienté étiqueté) acceptant les mots décrits par chacune des expressions régulières suivantes :

- $a(ab + bb)^*b$;
- $(a(bb)^*b)^*(aa + bba)^*$.

3 Langages et grammaires algébriques

Soit $L = \{b^p c^q a^{p+q}, p, q \geq 0\}$

- En utilisant les propriétés de clôture des langages rationnels prouver que L n'est pas rationnel.
- Donner une grammaire algébrique pour L .

4 Pièces

On se place sur l'alphabet $A = \{a, b, c\}$. Étant donné deux mots $u, v \in A^*$ on dit que u est une **pièce** de v si il existe des mots $v_0, \dots, v_n \in A^*$ et des lettres $a_0, \dots, a_n \in A$ t. q. $u = a_0 a_1 a_2 \dots a_n$ et $v = v_0 a_0 v_1 a_1 v_2 \dots a_n v_n$. Par exemple $ababc$ est une pièce de $aaabaccabca$.

Étant donné un mot $u \in A^*$, on note L_u le langage de tous les mots qui ont u pour pièce:

$$L_u = \{v \mid u \text{ est une pièce de } v\}$$

1. On considère le mot $u = abbc$. Construire un automate qui accepte le langage L_u .

2. Plus généralement si on donne $u \in A^*$ un mot quelconque, comment construire un automate qui accepte L_u .

3. On définit les *langages testables par pièces* de la façon suivante:

- pour tout $u \in A^*$, L_u est testable par pièces
- si L_1 et L_2 sont des langages testables par pièces alors $L_1 \cap L_2$ est également testables par pièces.
- Si L est testable par pièces alors \bar{L} (le complémentaire de L) est testable par pièces.

Dites si l'affirmation suivante est vraie ou fausse: "tout langage testable par pièces est régulier (c'est-à-dire reconnaissable par un automate fini)". La réponse devra être justifiée.

4. Étant donné un langage L , on note $\uparrow L$ le langage,

$$\uparrow L = \{w \in A^* \mid \text{il existe un mot } u \in L \text{ tel que } u \text{ est une pièce de } w\}$$

Montrer que si L est régulier alors $\uparrow L$ est également régulier (*Indice:* expliquer comment construire un automate qui accepte $\uparrow L$ partir d'un automate qui accepte L).

5. Étant donné un langage L , on note $\downarrow L$ le langage,

$$\downarrow L = \{w \in A^* \mid \text{il existe un mot } u \in L \text{ tel que } w \text{ est une pièce de } u\}$$

Montrer que si L est régulier alors $\downarrow L$ est également régulier (*Indice:* expliquer comment construire un automate qui accepte $\downarrow L$ partir d'un automate qui accepte L).