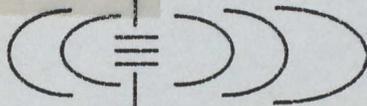


QUEEN  
QC  
925.1  
.C2  
L35  
1993

**UNIVERSITÉ LAVAL**  
**FACULTÉ DES SCIENCES ET DE GÉNIE**  
**DÉPARTEMENT DE GÉNIE ÉLECTRIQUE**



---

**Laboratoire de Radiocommunications  
et de Traitement du Signal**

Rapport final - Phase II

**Évaluation des conditions de propagation des liaisons  
terrestres dans la bande de 23,6 à 34,5 GHz en fonction  
de la propagation micro-onde terrestre**

par

Pierre Lahaie et Michel Lecours

pour

Le Gouvernement du Canada -  
Communications Canada

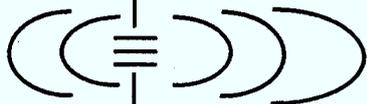
Sous

Contrat 36300-2-0164  
Approvisionnement et Service Canada

Avril 1992 à mars 1993

QC  
925.1  
.C2  
L35  
1993  
~~500K~~

**UNIVERSITÉ LAVAL**  
**FACULTÉ DES SCIENCES ET DE GÉNIE**  
**DÉPARTEMENT DE GÉNIE ÉLECTRIQUE**



---

**Laboratoire de Radiocommunications  
et de Traitement du Signal**

Rapport final - Phase II

**2. Évaluation des conditions de propagation des liaisons  
terrestres dans la bande de 23,6 à 34,5 GHz en fonction  
de la propagation micro-onde terrestre ...)**

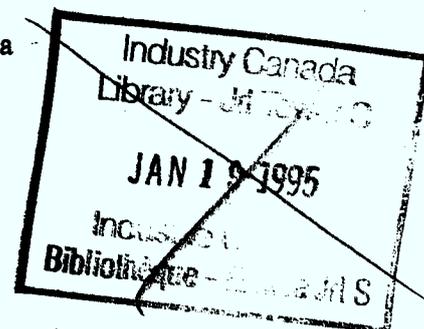
par

/s/ Pierre Lahaie et Michel Lecours



pour

Le Gouvernement du Canada  
Communications Canada



Sous

Contrat 36300-2-0164  
Approvisionnement et Service Canada

Avril 1992 à mars 1993

QC  
925.1  
C2  
L35  
1993

DD 12872568

**Table des matières**

<b>Titre</b>	<b>Page</b>
<b>Table des matières</b>	<b>i</b>
<b>Liste des figures</b>	<b>ii</b>
<b>1.0 Introduction</b>	<b>1</b>
<b>2.0 Avancement des travaux</b>	<b>2</b>
<b>2.1 Système de mesure de la taille des gouttes de pluie</b>	<b>3</b>
<b>2.2 La mesure de l'affaiblissement en fonction du taux de pluie</b>	<b>5</b>
<b>2.3 Description des images radars et des méthodes envisagées pour leur utilisation</b>	<b>6</b>
<b>3.0 Travail nécessaire pour l'élaboration d'un modèle de prédiction des affaiblissements par la pluie et de la fiabilité de ces méthodes.</b>	<b>11</b>
<b>4.0 Conclusion</b>	<b>13</b>
<b>Annexe 1</b>	<b>14</b>
<b>Annexe 2</b>	<b>30</b>
<b>Références bibliographiques</b>	<b>41</b>

**Liste des figures**

<b>Figure</b>	<b>Page</b>
Figure 1) Schéma du système de mesure de la taille des gouttes de pluie	2
Figure 2) Images d'une même suite de gouttes avec augmentation du contraste	3
Figure 3) Schéma du système de mesure des affaiblissements	4
Figure 4) Comparaison entre affaiblissement et mesure du taux de pluie	5
Figure 5) Image radar d'un évènement pluvieux	6
Figure 6) Régions à éliminer	7
Figure 7) Fonctionnement du radar météorologique	8
Figure 8) Séquence de quatre images subséquentes espacées de cinq minutes débutant à 20h le 4 août 1989	9

## 1.0) Introduction

Les travaux sur l'évaluation des conditions de propagation des liaisons micro-onde terrestres dans la bande 23.6 à 34.5 GHz dont il est question ici, ont débuté en avril 1991. Au moment de la conception du système de gestion du spectre de Communications Canada, il existait peu de modèles pour les bandes de fréquence supérieures à 13 GHz et, en particulier pour la bande micro-onde supérieure de 23.6 à 34.5 GHz. Cependant, en raison de la congestion du spectre dans les bandes de fréquence jusqu'à 20 GHz, on s'intéresse de plus en plus à la partie du spectre au dessus de 20 GHz. C'est ainsi que l'Union Internationale des Télécommunications (UIT) a, depuis 1979, encouragé le développement des bandes au dessus de 20 GHz grâce à une révision de la table des attributions de fréquence. Les ondes électromagnétiques au dessus de 20 GHz sont fortement atténuées par la pluie, ce qui représente une contrainte majeure pour leur utilisation. De plus, il y a une grande incertitude quant aux modèles de prédiction, et les modèles d'atténuation par la pluie du Comité Consultatif International sur la Radio (CCIR) sont relativement déficients dans cette bande de fréquence. Au Canada, le Centre de Recherches en Communications (CRC), qui agit comme chef de file pour les recherches en propagation micro-onde, a lancé un projet de "Modèle d'orage synthétique", basé sur vingt années de recherche, d'observations radar et de mesures de propagation, mais ce projet n'est que partiellement complété.

Le but du présent projet est de modéliser la propagation dans la bande de 23.6 à 34.5 GHz à l'aide des informations scientifiques les plus récentes en vue de l'intégration de ces informations dans le système de gestion du spectre électromagnétique. Au début des travaux, en avril 1991, on a considéré que le projet pouvait être divisé en trois étapes: - L'étape I consistait en une analyse exhaustive de la littérature scientifique sur la propagation dans la bande 23.6 à 34.5 GHz et à l'identification des travaux connexes en vue des étapes ultérieures. - Une deuxième étape serait consacrée à l'établissement ou à l'adaptation de modèles adéquats. - Une troisième étape serait consacrée à la synthèse des résultats en vue de leur intégration au système de gestion du spectre. Un système de gestion du spectre électromagnétique applicable à la bande de fréquence allant de 23.6 à 34.5 GHz doit pouvoir tenir compte de l'effet de l'atténuation causée par la pluie à ces fréquences. À titre d'exemple, à 30 GHz, l'atténuation causée par la pluie pourra être de l'ordre de 1.7 dB/km à 16.8 dB/km pour des pluies ayant des intensités allant de 10 mm/hre à 100 mm/hre.

Suite aux travaux réalisés pour la phase I (avril 1991 - mars 1992), et tel qu'expliqué dans le rapport de ces travaux, on peut considérer que la prédiction de l'atténuation causée par la pluie - qui est nécessaire à l'élaboration d'un système de gestion du spectre applicable à la bande de fréquence considérée - est limitée par deux facteurs principaux. i) D'une part, on a une connaissance imprécise de paramètres physiques (forme, grosseur, vitesse, intensité) des gouttes de pluie, de leur distribution pour des pluies survenant dans différentes conditions climatiques, et de l'effet de ces paramètres et de leur distribution sur l'atténuation micro-onde dans cette bande de fréquence. ii) D'autre part, on a une connaissance incomplète de la structure macroscopique des cellules de pluie, c'est-à-dire de la distribution spatiale des cellules, et de la configuration des cellules de pluie individuelles pour les différents climats. Même si des données météorologiques et hydrographiques sont disponibles, et même si des modèles d'orages synthétiques ont été développés, il reste un travail considérable à accomplir pour obtenir une description globale de la structure des pluies.

Pour la phase II (avril 1992 - mars 1993), nous avons procédé à la fois à la réalisation d'expériences en vue d'améliorer nos connaissances des paramètres physiques de la pluie au Canada et de leur corrélation avec l'atténuation micro-onde, tel que discuté en i) ci-dessus, et au développement de techniques de traitement d'images radar météo en vue de l'établissement de recommandations précises tenant compte des connaissances statistiques actuelles sur la structure des cellules de pluie. En ce qui concerne l'atténuation micro-onde, nous avons établi un lien micro-onde CW dans la bande considérée opérant sur une courte distance de l'ordre de 100 mètres en vue d'être assuré d'une pluie relativement uniforme sur la longueur du parcours. Ce lien fonctionne en conjonction avec un système de caméra vidéo avec obturateur rapide pour mesurer les paramètres physiques de la pluie (grosesse, forme, vitesse des gouttes) et avec un pluviomètre. Des contraintes de coût (et de temps) nous ont forcé à utiliser en partie des équipements de transmission-réception et d'acquisition de données qui étaient déjà disponibles à notre laboratoire mais qui devraient être partagés avec d'autres projets. D'autre part, la mise au point de ces équipements n'a été complétée que vers le milieu de l'été et le nombre d'événements de pluie qui ont été mesurés était donc limité. Par ailleurs, nous avons obtenu, grâce à la collaboration du Professeur A. Bellon de l'Université McGill, des images radar météo numérisées d'activités orageuses portant sur une région de 240 x 240 km autour de Montréal; nous avons étudié l'application de telles données radar au développement d'un système de gestion du spectre adapté aux phénomènes de propagation dans la bande de 23.6 à 34.5 GHz.

## 2.0) Avancement des travaux

Les travaux présentés dans ce rapport ont été effectués durant la période de avril 1992 à mars 1993. Ces travaux ont consisté dans un premier temps en la mise au point de systèmes de mesure de la distribution en taille des gouttes de pluie ainsi que dans la mise au point d'un système de mesure de l'affaiblissement linéique des micro-ondes par la pluie sur de courtes distances. Nous avons aussi travaillé à la conception d'outils informatiques permettant de traiter les données provenant de ces mesures. Dans un deuxième temps, nous avons travaillé à la conception d'outils permettant de traiter les images de radars météorologiques provenant de l'observatoire radiométéorologique de l'Université McGill grâce à la collaboration de M. Aldo Bellon professeur à l'Université McGill. Ces outils permettent, entre autres, de visualiser les images radars pour les valider ainsi que de calculer un modèle pour la prédiction des affaiblissements par la pluie.

Précisons dès maintenant les buts recherchés ici. Il nous faut connaître la valeur de l'affaiblissement linéique en fonction du taux de pluie pour détecter les erreurs systématiques qui se glisseraient dans le modèle de prédiction des affaiblissements. Pour ce faire il nous faut connaître la distribution en taille des gouttes de pluie en fonction du taux de pluie, car c'est cela qui détermine l'affaiblissement. Idéalement pour calculer l'affaiblissement des micro-ondes par la pluie, il nous faudrait disposer de données précises qui nous donneraient les détails des précipitations sur une surface étendue. Malheureusement, les données des pluviomètres installés par les organismes de surveillance de l'environnement ne sont pas satisfaisantes à cette fin. Il existe par contre les radars météorologiques qui permettent d'obtenir des données sur la pluie à

tous les kilomètres carrés. Ces données sont moins fiables prises individuellement que les données provenant des pluviomètres, mais l'erreur sur la mesure semble distribuée uniformément autour de celle-ci. Aussi nous avons choisi d'utiliser ces données radar dans la suite de nos travaux.

Nous montrons donc dans ce rapport la faisabilité des mesures que nous préconisons, ceci à l'aide de figures et nous terminons avec une description des travaux futurs nécessaires dans une prochaine étape pour élaborer des modèles fiables de prédictions des affaiblissements par la pluie.

### 2.1) Système de mesure de la taille des gouttes de pluie.

La figure 1 montre le système de mesure de la taille des gouttes de pluie. Ce système est composé d'une caméra, d'un abri pour caméra, d'un système d'éclairage pour illuminer les gouttes de pluie, d'un écran noir utilisé pour le fond et d'un système d'enregistrement des images vidéo générées par la caméra.

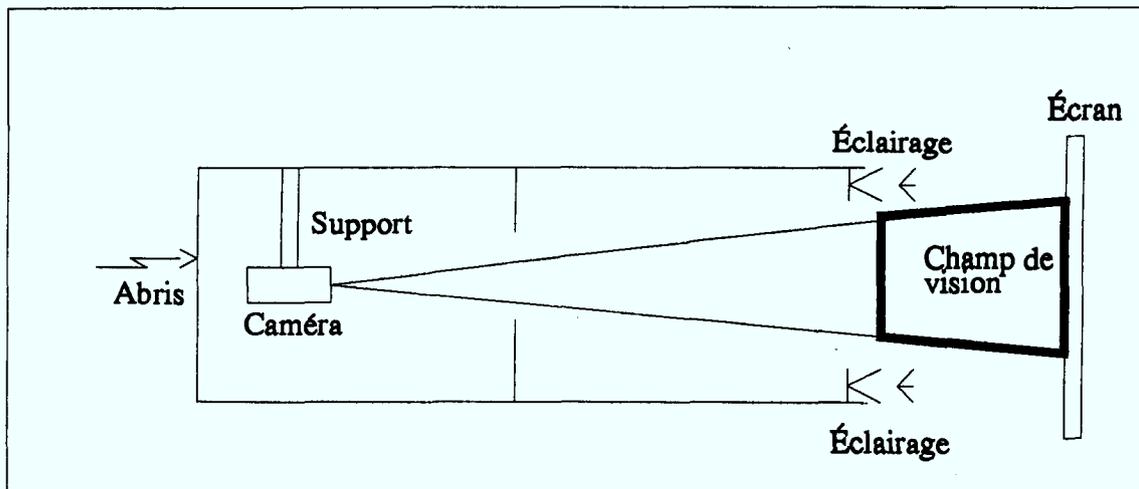


Figure 1: Schéma du système de mesure de la taille des gouttes de pluie.

La dimension du champ de vision dépend de la profondeur de champ qui est disponible grâce à la caméra. La profondeur de champ d'une lentille dépend de l'ouverture du diaphragme de cette lentille. Plus le diaphragme est fermé plus le champ peut être grand, mais en revanche il y a moins de lumière qui atteint la cellule sensible au fond de la caméra. La dimension du champ de vision dépend donc d'un compromis entre la quantité de lumière qui atteint la cellule et la netteté que l'on désire dans l'image. Plus il y a de lumière, plus les gouttes seront nettes à une grande distance de l'objectif de la caméra. Un autre paramètre important est la durée d'exposition pour chaque image. En effet, si cette durée est trop grande, les gouttes laisseront une trainée sur l'image, et cette trainée nous fera perdre de la netteté dans l'image.

Les cellules sensibles à la lumière fonctionnent selon un procédé d'intégration de l'énergie lumineuse incidente. Donc si nous diminuons le temps d'exposition, nous perdons du contraste

entre les points qui forment le fond noir et les points qui forment la goutte. Nous devons donc éclairer les gouttes avec un très fort éclairage pour à la fois obtenir une profondeur de champ suffisante pour saisir des gouttes de pluie pratiquement à chaque image dans le cas d'une pluie normale et saisir les gouttes sans qu'il y ait impression de mouvement de leur part sur l'image qui l'aura saisie.

La figure 2 montre une image de goutte de pluie saisie à l'aide de la caméra vidéo et une série de traitements sur cette image montre que le contraste entre les gouttes et le fond peut être amélioré pour permettre de saisir les gouttes plus aisément. La figure 2.a est l'image sans traitement et la figure 2.b est l'image avec un traitement visant à augmenter le contraste entre les gouttes et le fond noir.

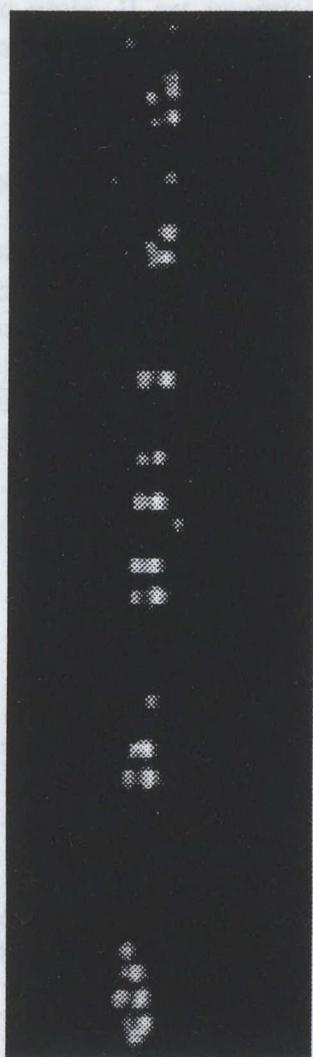


figure 2.a

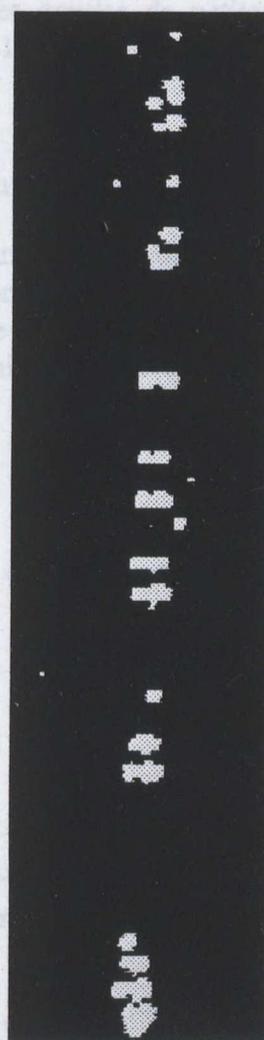


figure 2.b

Figure 2: Images d'une même suite de gouttes avec augmentation du contraste

Ces images sont obtenues à l'aide d'une caméra qui génère un signal selon le standard NTSC puis elles sont saisies à l'aide d'une carte qui la transforme en image selon le format de sauvegarde SUN RASTERFILE, qui est une spécification de la quantité de lignes et de colonnes puis une suite qui constitue l'intensité des pixels.

Pour identifier les gouttes à l'intérieur des images nous appliquons en premier lieu un seuil pour séparer les pixels d'intensité élevée des pixels d'intensités faibles ensuite nous appliquons un traitement qui vise à détecter les gouttes et leurs surfaces. En calculant la surface de la goutte en pixels nous utilisons la formule permettant de relier le diamètre d'un cercle avec sa surface, ce qui nous permettra d'obtenir, une fois le système calibré, la dimension des gouttes selon un histogramme de la taille des gouttes de pluie et donc une courbe de la distribution en taille des gouttes de pluie. Il est important de spécifier que les gouttes de petites tailles seront fort probablement sous estimées si leur taille est inférieure à la dimension d'un pixel.

## 2.2) La mesure de l'affaiblissement en fonction du taux de pluie.

Pour mesurer l'affaiblissement en fonction du taux de pluie nous utilisons en parallèle un pluviomètre et un système de transmission réception échantillonné (voir figure 3). Nous superposons alors les données provenant du pluviomètre et les données du système d'acquisition pour comparer l'affaiblissement par la pluie et l'affaiblissement qui est prévu selon la loi d'affaiblissement du CCIR (équation 1).

$$A = \alpha R^\beta \quad (1)$$

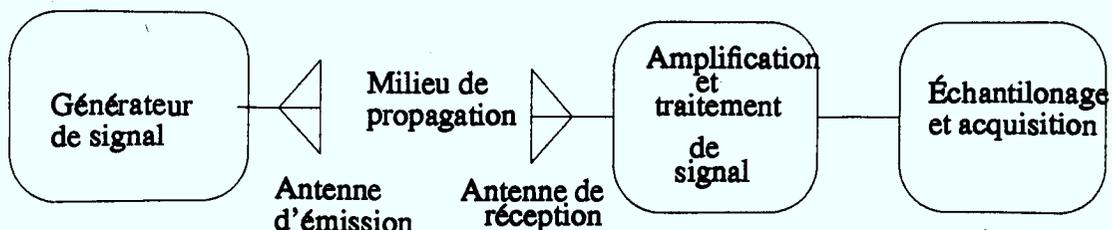


Figure 3: Schéma du système de mesure des affaiblissements.

Pour les mesures qui ont été prises durant l'été 1992 le générateur de signal fonctionnait à la fréquence de 37 GHz, ce qui est la fréquence à laquelle un système de transmission-réception était disponible au laboratoire<sup>1</sup>. La figure 4 montre un exemple de mesure et la comparaison entre la mesure et l'affaiblissement calculé à partir des données du CCIR pour la fréquence de 37,2 GHz. Les deux courbes contenues dans la figure 4 correspondent en: La courbe supérieure est la puissance mesurée au récepteur et la courbe inférieure est la puissance qui devrait être lues au récepteur compte tenu du taux de pluie à ce moment, le niveau de puissance pour la courbe 2 a été déplacé vers le bas. Pour obtenir l'affaiblissement par la pluie il nous faut comparer les niveaux de puissance avant qu'il y ait de la pluie.

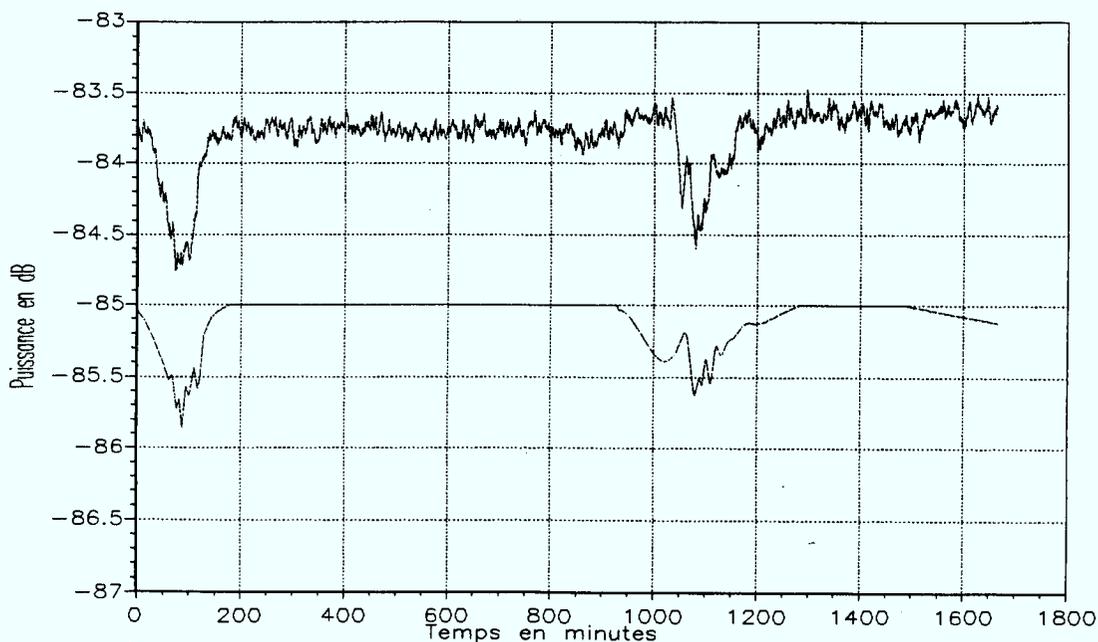


Figure 4: Comparaison entre affaiblissement et mesure du taux de pluie.

### 2.3) Description des images radars et des méthodes envisagées pour leur utilisation.

Nous envisageons de nous servir des images provenant du radar météorologique de l'Université McGill pour l'élaboration d'un modèle de prédiction des affaiblissements par la pluie. Nous disposons pour le présent projet d'un enregistrement correspondant à plusieurs heures d'un

1. Note: Étant donné que ces installations étaient requises pour plusieurs projets de recherche. Elles devaient être partagées entre différents utilisateurs, ce qui a constitué une limitation majeure au cours de la période du projet.

évènement pluvieux survenue à Montréal le 4 août 1989. Les images sont constituées de pixels représentant une surface de 1 km carré, et l'intensité du pixel représente l'intensité de pluie en cet endroit. La résolution disponible est en terme de dixième de mm/heure de pluie, mais il est possible qu'une erreur aléatoire de 30% sur les valeurs de l'intensité de pluie existe. Un avantage indéniable du radar est que celui-ci nous fournit l'équivalent d'un pluviomètre à tous les km carrés de surface. Nous devons donc évaluer la fiabilité d'un modèle construit à partir de ces données. La figure 5 présente en tons de gris un évènement pluvieux qui s'est produit à Montréal le 4 août 1989 à 20h 30.

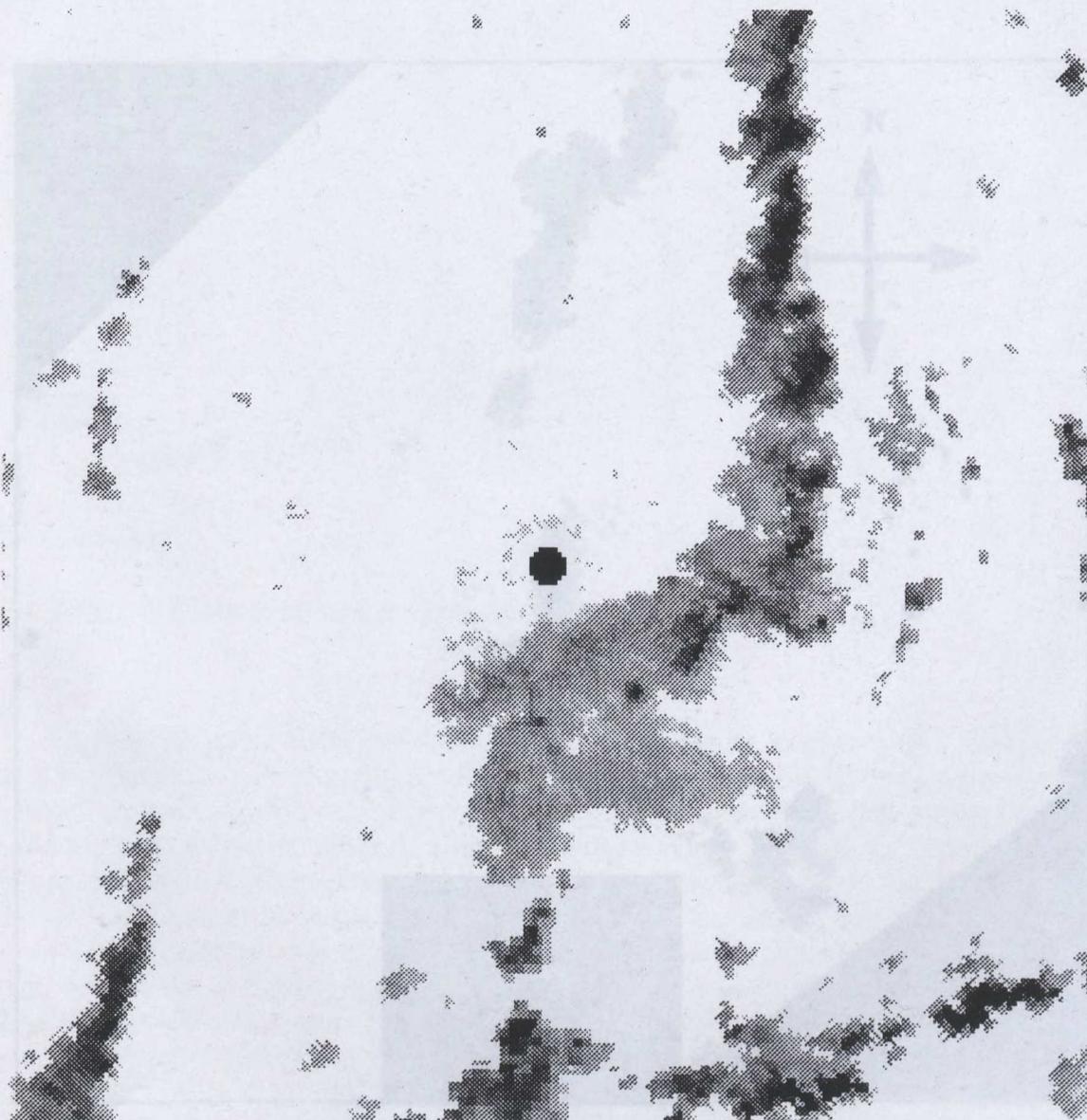


Figure 5: Image radar d'un évènement pluvieux.

Dans la figure 5, nous devons tout d'abord remarquer certains points importants. Tout d'abord, le point octogonal entièrement noir au centre de la figure est la position du radar; cet endroit ne peut être utilisé car les échos de sol sont trop forts, ce qui surestime la pluie. Les données qui proviennent de cet endroit sont donc annulées et un marqueur est placé dans le fichier pour les points constituant cette région.

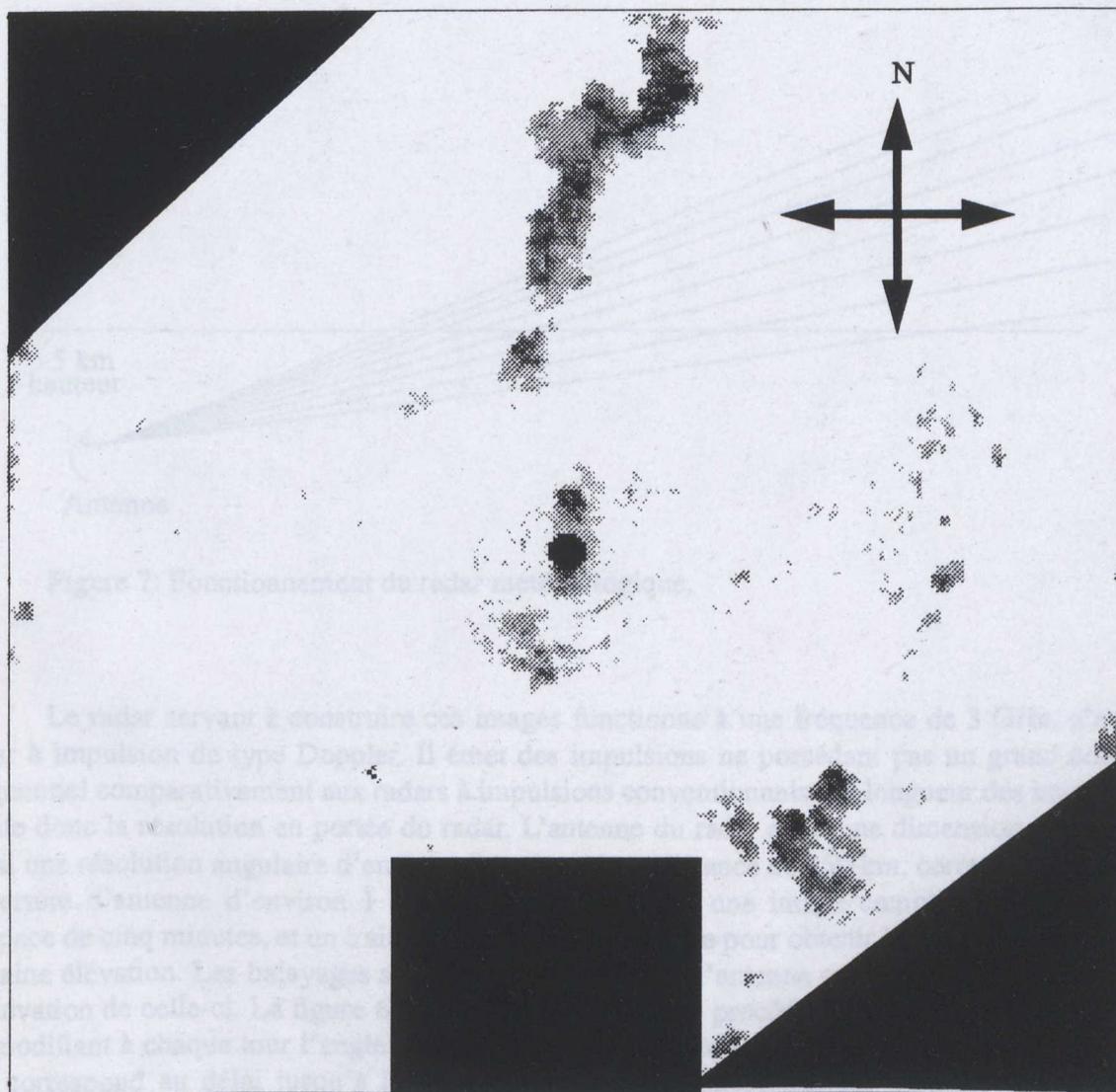


Figure 6: Régions à éliminer.

Les données provenant d'autres régions doivent aussi être annulées, les échos provenant de celles-ci étant trop fort. Ces régions sont montagneuses, elles sont situées dans les parties sud-est, sud et nord-ouest de l'image. Le radar est situé près de Ste-Anne de Bellevue sur l'île de Montréal et, dans le cas de la partie nord-ouest, deux montagnes situées à environ une dizaine de kilomètres du radar causent de l'ombre pour l'illumination de la partie nord-ouest éloignée du radar. Pour la partie sud-est, ce sont les "White Mountains", situées dans le Vermont qui causent de forts échos ceux-ci étant dus aux lobes secondaires de l'antenne. Ce phénomène est observable aussi dans la partie sud de l'image à cause de montagnes de faibles altitude situées en Ontario. Ces régions sont identifiées dans la figure 6 à partir d'une image contenant de faibles précipitation au nord de l'image. Nous pouvons donc voir que ces images contiennent encore beaucoup d'endroits qui serviront pour établir un modèle.

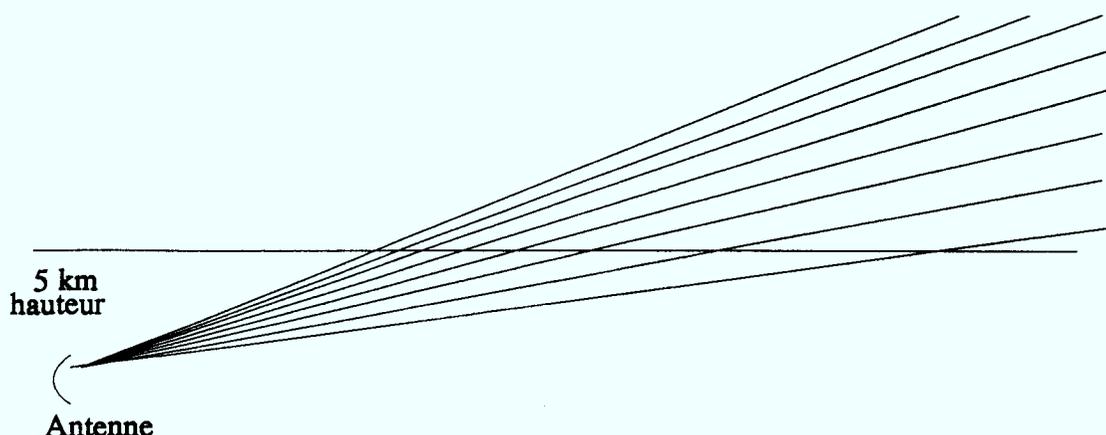


Figure 7: Fonctionnement du radar météorologique.

Le radar servant à construire ces images fonctionne à une fréquence de 3 GHz, c'est un radar à impulsion de type Doppler. Il émet des impulsions ne possédant pas un grand contenu fréquentiel comparativement aux radars à impulsions conventionnels. La longueur des impulsions limite donc la résolution en portée du radar. L'antenne du radar est d'une dimension qui permet aussi une résolution angulaire d'environ deux km à une distance de 150 km, correspondant à une ouverture d'antenne d'environ 1 degré. Ce radar obtient une image complète en volume en l'espace de cinq minutes, et un traitement est alors nécessaire pour obtenir les précipitations à une certaine élévation. Les balayages successifs effectués par l'antenne se font en modifiant l'angle d'élévation de celle-ci. La figure 6 donne un exemple de ce procédé. L'antenne balaie le volume en modifiant à chaque tour l'angle d'élévation, on choisit alors l'intensité de rétrodiffusion radar qui correspond au délai jusqu'à l'intersection avec la ligne de hauteur cinq km pour obtenir l'image radar des précipitations à la hauteur de cinq km. Nous faisons ensuite l'hypothèse que les précipitations à cinq km sont les mêmes qu'au niveau du sol.

### 3.0) Travail nécessaire pour l'élaboration d'un modèle de prédiction des affaiblissements

Un des aspects intéressant du radar météorologique est qu'il permet dans une certaine mesure de voir la dynamique des précipitations (voir la figure 8). L'évolution des cellules de pluie et leur déplacements sont ces deux aspects. Les radars pourraient donc nous permettre d'évaluer la dynamique des affaiblissements, bien sur en admettant que des intervalles de cinq minutes sont utiles pour ce genre d'études.

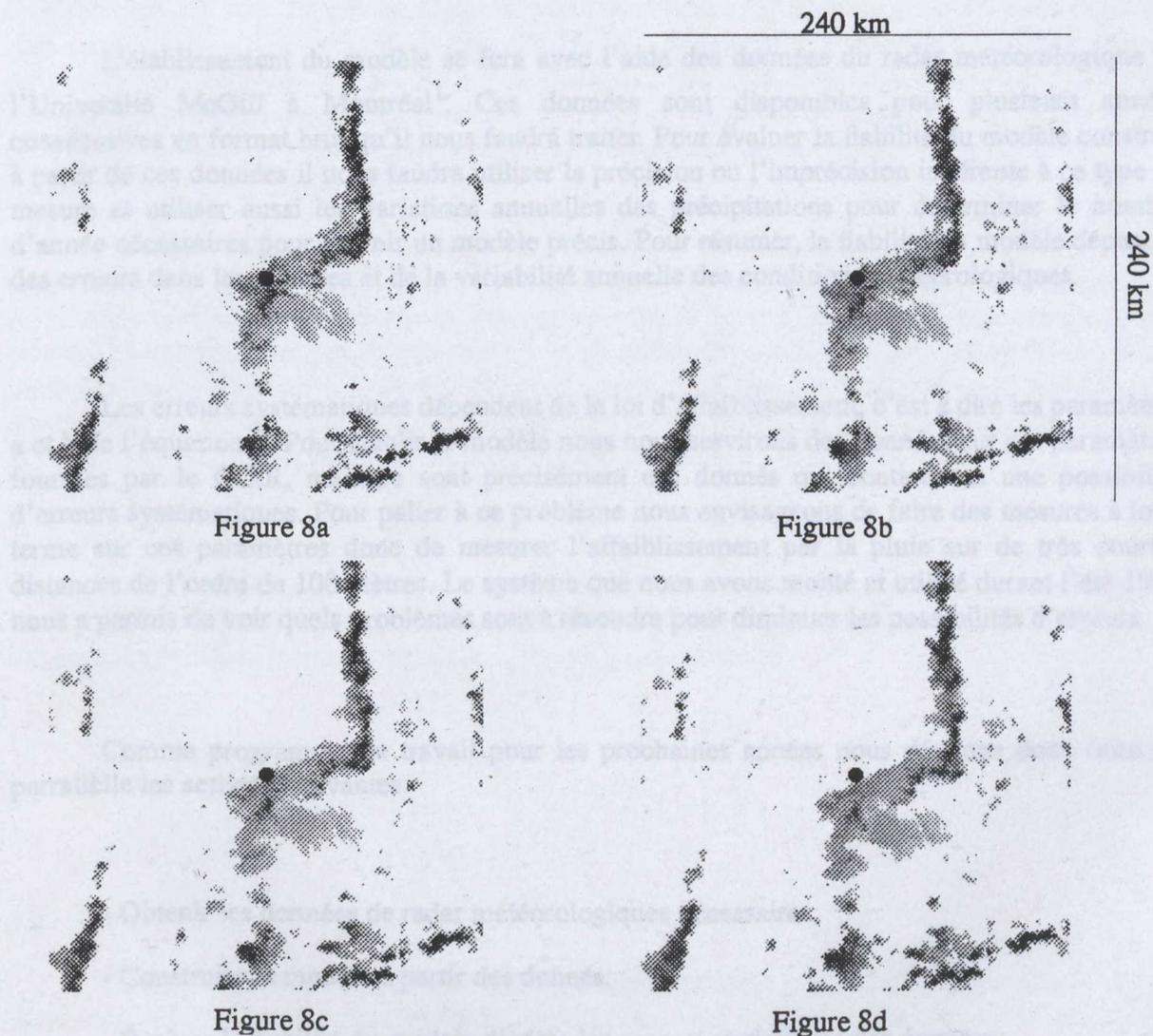


Figure 8: Séquence de quatre images subséquentes espacées de cinq minutes débutant à 20 h le 4 août 1989.

### **3.0) Travail nécessaire pour l'élaboration d'un modèle de prédiction des affaiblissements par la pluie et de la fiabilité de ces méthodes.**

Beaucoup de travaux sont nécessaires pour permettre de développer un modèle de prévision des affaiblissements par la pluie. Ces travaux concernent le modèle proprement dit. Il nous faut aussi évaluer la fiabilité de ce modèle et détecter les erreurs systématiques qui pourraient s'y glisser.

L'établissement du modèle se fera avec l'aide des données du radar météorologique de l'Université McGill à Montréal<sup>1</sup>. Ces données sont disponibles pour plusieurs années consécutives en format brut qu'il nous faudra traiter. Pour évaluer la fiabilité du modèle construit à partir de ces données il nous faudra utiliser la précision ou l'imprécision inhérente à ce type de mesure et utiliser aussi les variations annuelles des précipitations pour déterminer le nombre d'année nécessaires pour fournir un modèle précis. Pour résumer, la fiabilité du modèle dépendra des erreurs dans les mesures et de la variabilité annuelle des conditions météorologiques.

Les erreurs systématiques dépendent de la loi d'affaiblissement, c'est à dire les paramètres a et b de l'équation 1. Pour établir le modèle nous nous servirons des données sur ces paramètres fournies par le CCIR, mais ce sont précisément ces données qui contiennent une possibilité d'erreurs systématiques. Pour palier à ce problème nous envisageons de faire des mesures à long terme sur ces paramètres donc de mesurer l'affaiblissement par la pluie sur de très courtes distances de l'ordre de 100 mètres. Le système que nous avons monté et utilisé durant l'été 1992 nous a permis de voir quels problèmes sont à résoudre pour diminuer les possibilités d'erreurs.

Comme programme de travail pour les prochaines années nous désirons donc faire en parallèle les activités suivantes:

- Obtenir les données de radar météorologiques nécessaires;
- Construire le modèle à partir des données;
- Évaluer la fiabilité du modèle d'après les aspects statistiques des données;
- Détecter les possibilités d'erreurs systématiques;

---

1. Note: Nous tenons à souligner la collaboration du professeur Aldo Bellon de l'observatoire radiométéorologique de l'Université McGill.

Pour obtenir les données de radar météorologiques nous devons nous déplacer à Montréal, à l'observatoire météorologique de l'Université McGill pour une certaine période de temps, nécessaire pour mettre les données brutes disponibles sur ruban magnétique en données utilisable par nos ordinateurs. Les logiciels qui devront être utilisés sont disponibles à l'observatoire.

Le modèle qui sera construit à l'aide des données doit faire l'objet d'un certain travail, nous travaillons présentement à la conception d'un logiciel qui permettra de calculer à partir des données radars les affaiblissements par la pluie. Pour ce faire nous implantons des liaisons micro-onde fictives dans le logiciel et nous calculons à partir des images radars quel aurait été l'affaiblissement subi. Nous compilons ensuite ces résultats à l'intérieur d'un fichier. Le modèle sera construit avec approximativement 100 liaisons micro-ondes fictives à trois fréquences, 5 longueurs différentes de 5 km à 20 km de distance entre l'émetteur et le récepteur, et avec six orientations différentes pour les systèmes. Ceci nous permettra de compiler les variations des affaiblissements par la pluie dépendamment de la fréquence, de la longueur et de l'orientation des systèmes de communications micro-ondes.

La partie théorique du travail, soit l'évaluation de la fiabilité du modèle, elle, se fera durant toute les parties expérimentales et calculatoires. Nous devons pour ce faire connaître les paramètres techniques du radar pour évaluer les erreurs de mesures et le comportement de ces erreurs. La variation des précipitations d'année en année devra elle aussi faire l'objet d'une étude pour prédire la fiabilité. Toutes ces données sont disponibles dans la littérature technique la contribution originale viendra de l'application de ces connaissances au radar météorologique et à la prédiction des affaiblissements des micro-ondes par la pluie.

Pour détecter les erreurs systématiques, il nous faut valider pour notre climat les données sur l'affaiblissement linéique fournies par le CCIR, ces données n'ayant pas été validées de façon exhaustive par des mesures durant de longues périodes de temps.

#### **4.0) Conclusion**

Ce rapport présente les travaux qui ont été faits durant la période s'étendant d'avril 1992 à décembre 1992. Il nous a fallu travailler à l'affichage des images radars que nous possédions, construire les montages pour évaluer la faisabilité des mesures comme celles de l'affaiblissement linéique des micro-ondes par la pluie; nous avons aussi travaillé pour monter le système de détermination des tailles de gouttes de pluie; et enfin nous avons élaboré certaines méthodes pour calculer à partir des images radars les affaiblissements par la pluie.

Pour afficher les images radars nous avons conçu un programme fonctionnant sur ordinateurs SUN avec interface graphique X-Windows, ce programme nécessite encore quelques améliorations pour être entièrement fonctionnel, nous le donnons à l'annexe 1. Pour le traitement des données provenant des images de gouttes de pluie nous avons développé quelques algorithmes de traitement des images et quelques programmes qui vont être utilisés dans les prochaines études sur la distribution en taille des gouttes de pluie. Ces programmes sont donnés à l'annexe 2 de ce document. Pour les programmes qui vont servir au calcul des affaiblissements par la pluie, nous fournissons un prototype de programme qui utilise la méthode décrite à la section précédente.

## Annexe 1

### Programme d'affichage des images radars

```

/* Ce programme permet d'afficher des images radar. */
/* Il permet dans un premier temps de les décompresser */
/* Ensuite il les affiche dans une fenêtre X-Windows */

/* La ligne de compilation est la suivante */

/*          cc Image_radar.c -o Image_radar -I/gel/mistassini/logiciels/openwin/ */
/*          share/include -lX -lxview -lolgx -lm */

/* Les fichiers de librairies suivants doivent être inclus dans le programme */

#include <stdio.h>
#include <X11/Xlib.h>
#include <xview/xview.h>
#include <sys/param.h>
#include <sys/types.h>
#include <xview/panel.h>
#include <xview/canvas.h>
#include <xview/icon.h>
#include <xview/icon_load.h>
#include <xview/tty.h>
#include <xview/svrmage.h>
#include <xview/termsw.h>
#include <xview/text.h>
#include <xview/textsw.h>
#include <xview/seln.h>
#include <xview/notice.h>
#include <xview/xv_xrect.h>
#include <xview/rectlist.h>
#include <xview/cms.h>
#include <sys/stat.h>
#include <sys/timeb.h>
#include <sys/file.h>
#include <math.h>
#include <malloc.h>
#include <string.h>
#include <stdlib.h>

```

```

/* Le Pixel d'affichage est défini comme étant composé de deux pixels écran */
/* Ce paramètre peut devoir être modifié dépendamment de la machine */
#define taille_pixel 2

/* Les couleurs sont définies ici dans ce programme il s'agit de couleurs subjectives. */
#define BLANC 0
#define JAUNE 1
#define SERIN 2
#define VERT 3
#define FORET 4
#define OLIVE 5
#define CIEL 6
#define ROYAL 7
#define MARINE 8
#define VIOLET 9
#define ORANGE 10
#define ORANGE_FONCE 11
#define BRUN 12
#define ECARLATE 13
#define SANG 14
#define NOIR 15

#define NOMBRE_COULEUR 16

/* Les fonctions et constantes sont définies comme existantes ici */

unsigned short icon_image[]={
#include "verif.icon"
};

void montre_carte();
void fabrique_panneau();
void decompresse_image_radar();
void affiche_image_radar();
void montre_position();

Icon icon;
short image_radar[240][240];
Panel panneau;
Panel_item it_rep_image,it_rep_sequence,it_nom_image,it_nom_sequence;

```

```

Panel_item    it_x_depart,it_y_depart,it_x_arivee,it_y_arivee;
Panel_item    it_bouton_change,it_bouton_prochaine,it_bouton_sauvegarde;
Panel_item    it_bouton_quitter;
Canvas        canevas;
Frame         cadre;
Cms           cms;
unsigned long *table_pixel;
Pixmap        point_mesure;
GC            gc;
FILE          *sequence;
FILE          *image;
Server_image  closed_image;

```

```

static Xv_singlecolor couleurs[]={
    {255,255,255},
    {238,238,238},
    {221,221,221},
    {204,204,204},
    {187,187,187},
    {170,170,170},
    {153,153,153},
    {136,136,136},
    {119,119,119},
    {102,102,102},
    {85 ,85 ,85 },
    { 68, 68, 68},
    { 51, 51, 51},
    { 34, 34, 34},
    { 17, 17, 17},
    {0 ,0 ,0 }};

```

```

static unsigned char pixel_pluie[] = {0xff,0xff};

```

```

/*          La fonction "couleur" retourne la valeur de couleur associée          */
/*          à un pixel. Cette couleur est reliée au taux de pluie                */
*/
*/

```

```

int couleur(valeur)
short valeur;

{
    if(valeur==0)
        return(BLANC);
    if(valeur>0 && valeur<=5)
        return(JAUNE);
    if(valeur>5 && valeur<=10)

```

```

    return(SERIN);
    if(valeur>10 && valeur<=30)
        return(VERT);
    if(valeur>30 && valeur<=50)
        return(FORET);
    if(valeur>50 && valeur<=80)
        return(OLIVE);
    if(valeur>80 && valeur<=130)
        return(CIEL);
    if(valeur>130 && valeur<=200)
        return(ROYAL);
    if(valeur>200 && valeur<=300)
        return(MARINE);
    if(valeur>300 && valeur<=500)
        return(VIOLET);
    if(valeur>500 && valeur<=750)
        return(ORANGE);
    if(valeur>750 && valeur<=1000)
        return(ORANGE_FONCE);
    if(valeur>1000 && valeur<=1500)
        return(BRUN);
    if(valeur>1500 && valeur<=2000)
        return(ECARLATE);
    if(valeur>2000 && valeur<=5000)
        return(SANG);
    if(valeur>5000 )
        return(NOIR);

}

/*          Le corps du programme se trouve ici          */

main(argc,argv)
int argc;
char *argv[];
{

XGCValues gc_val,gcvalues;

xv_init(XV_INIT_ARGC_PTR_ARGV, &argc,argv,NULL);

/*          Définition et ouverture des objets gérés par le programme          */

cms = (Cms) xv_create(NULL,CMS,
    CMS_SIZE,16,
    CMS_COLORS, couleurs,

```

```

NULL);

cadre= (Frame) xv_create(NULL,FRAME,
    FRAME_LABEL,"Image de Pluie",
    XV_WIDTH, 1000,
    XV_HEIGHT, 500,
    XV_SHOW,TRUE,
    NULL);

fabrique_panneau();

canevas =(Canvas) xv_create(cadre,CANVAS,
    XV_X,2,
    XV_Y,2,
    XV_WIDTH,484,
    XV_HEIGHT,484,
    XV_SHOW,TRUE,
    NULL);

xv_set(canvas_paint_window(canevas),
    WIN_EVENT_PROC, montre_position,
    WIN_MOUSE_BUTTONS, WIN_CONSUME_EVENTS,NULL,
    NULL);

table_pixel = (unsigned long *) xv_get(cms,CMS_INDEX_TABLE);

point_mesure = XCreateBitmapFromData(
    xv_get(cadre,XV_DISPLAY),xv_get(cadre,XV_XID),
    pixel_pluie,taille_pixel,taille_pixel);

gcvalues.graphics_exposures = False;
gcvalues.background = table_pixel[BLANC];
gc = XCreateGC(xv_get(cadre,XV_DISPLAY),xv_get(cadre,XV_XID),
    GCBackground | GCGraphicsExposures, &gcvalues);

closed_image = (Server_image)xv_create(NULL, SERVER_IMAGE,
    XV_WIDTH, 64,
    XV_HEIGHT, 64,
    SERVER_IMAGE_DEPTH,1,
    SERVER_IMAGE_BITS, icon_image,
    NULL);

icon = (Icon)xv_create(cadre, ICON,
    ICON_IMAGE,closed_image,

```

```

        NULL);

        xv_set(cadre,FRAME_ICON,icon,NULL)
xv_main_loop(cadre);
}
/*      Dans cette fonction, nous chargeons l'image radar compressé      */
/*      Cette image est décompressée puis ensuite affichée                */

void chargement_d_image(item,event)

Frame item;
Event *event;
{

int longueur;
char repertoire_image[100];
char nom_image[100];
char *nom_fichier;

strcpy (repertoire_image, (char *)xv_get(it_rep_image,PANEL_VALUE));
strcpy (nom_image, (char *)xv_get(it_nom_image,PANEL_VALUE));
fprintf(stderr,"nom du repertoire : %s \n",repertoire_image);
fprintf(stderr,"nom de l'image   : %s \n",nom_image);

longueur = strlen(repertoire_image) + strlen(nom_image) +1 ;
nom_fichier =(char *)calloc(longueur,sizeof(char));

strcpy(nom_fichier,repertoire_image);
strcat(nom_fichier,nom_image);
fprintf(stderr,"nom du fichier   : %s \n",nom_fichier);
decompresse_image_radar(nom_fichier);
fprintf (stderr,"je desire afficher une image \n");
free(nom_fichier);
if (image == NULL)
    return;
affiche_image_radar();
}

/*      Cette fonction permet d'incrémenter la prochaine image a chargé      */
*/

void prochaine_image(item,event)

```

```

Frame item;

```

```

Event *event;
{
    char *nom_image;
    char minutes[3],heures[3],jours[3];
    int jour,minute,heure;
    char *nom2_image;
    char tampon[2];

    nom_image = (char *)calloc(13,sizeof(char));
    nom2_image = (char *)calloc(13,sizeof(char));
    strncpy (nom_image, (char *)xv_get(it_nom_image,PANEL_VALUE),13);
    jours[0] = nom_image[6];
    jours[1] = nom_image[7];
    jours[2] = '\0';
    heures[0] = nom_image[8];
    heures[1] = nom_image[9];
    heures[2] = '\0';
    minutes[0] = nom_image[10];
    minutes[1] = nom_image[11];
    minutes[2] = '\0';
    minute = atoi(minutes);
    heure = atoi(heures);
    jour = atoi(jours);
    if (minute == 55)
    {
        minute = 0;
        heure = heure + 1;
        if(heure == 24)
        {
            jour = jour + 1;
            heure = 0;
            minute = 0;
        }
    }
    else
    minute = minute + 5;
    sprintf(jours,"%-02d",jour);
    sprintf(minutes,"%-02d",minute);
    sprintf(heures,"%-02d",heure);
    strncpy(nom2_image,nom_image,6);
    strcat(nom2_image,jours);
    strcat(nom2_image,heures);
    strcat(nom2_image,minutes);
    xv_set(it_nom_image,PANEL_VALUE,nom2_image,NULL);
    free(nom_image);
}

```

```

    free(nom2_image);

}

/*          Cette fonction permet de décompresser les images radar          */
/*          en utilisant l'algorithme de décompression                       */

void decompresse_image_radar(nom_fichier)
char *nom_fichier;
{
    int i,j,k;
    short bidon;

    if ((image = fopen(nom_fichier,"rb"))==NULL)
        {
            fprintf(stderr,"Le fichier %s ne peut etre ouvert \n",nom_fichier);
            return;
        }

    fseek(image,512,0);
    for(i=0;i<57600;i++)
        {
            j=(int) (i/240);
            k=i- 240*j;
            image_radar[j][k] =0;
        }
    i=0;
    do
        {
            fread(&bidon,sizeof(short),1,image);
            if (bidon>0)
                {
                    k=(int) (i/240);
                    j= i - 240*k;
                    image_radar[j][k] = bidon;
                    ++i;
                }
            else
                i= i- (int)bidon;
        } while (i<57600);
}

```

```

/*          Cette fonction permet d'afficher l'image radar dans le          */
/*          canevas de l'environnement de travail                          */

```

```
void affiche_image_radar()
```

```

{
    xv_destroy(canevas);

    canevas =(Canvas) xv_create(cadre,CANVAS,
        XV_X,2,
        XV_Y,2,
        XV_WIDTH,484,
        XV_HEIGHT,484,
        CANVAS_X_PAINT_WINDOW,TRUE,
        CANVAS_REPAINT_PROC,monstre_carte,
        WIN_CMS,cms,
        XV_SHOW,TRUE,
        NULL);

    xv_set(canvas_paint_window(canevas),
        WIN_EVENT_PROC, monstre_position,
        WIN_MOUSE_BUTTONS, WIN_CONSUME_EVENTS,NULL,
        NULL);
}

```

```

/*          Cette fonction permet de sauvegarder une image radar          */
/*          ainsi qu'une région particulière qui a été sélectionner        */

```

```
void sauvegarde(item,event)
```

```

Frame item;
Event *event;
{
    char *nom_fichier;
    char buffer_sequence[100];
    char buffer_repert[100];
    char nom_image[100];
    int pos_x_dep,pos_x_arr,pos_y_dep,pos_y_arr;
    int longueur;

    pos_x_dep = (int) atoi((char *)xv_get(it_x_depart,PANEL_VALUE));
    pos_x_arr = (int) atoi((char *)xv_get(it_x_arivee,PANEL_VALUE));
}

```

```

pos_y_dep = (int) atoi((char *)xv_get(it_y_depart,PANEL_VALUE));
pos_y_arr = (int) atoi((char *)xv_get(it_y_arivee,PANEL_VALUE));
strcpy(buffer_sequence,(char *)xv_get(it_nom_sequence,PANEL_VALUE));
strcpy(buffer_repert,(char *)xv_get(it_rep_sequence,PANEL_VALUE));
strcpy(nom_image,(char *)xv_get(it_nom_image,PANEL_VALUE));
longueur = strlen(buffer_sequence) + strlen(buffer_repert) +1;
nom_fichier = (char *)calloc(longueur,sizeof(char));
strcpy(nom_fichier,buffer_repert);
strcat(nom_fichier,buffer_sequence);
if ((sequence = fopen(nom_fichier,"ab"))==NULL)
{
    if((sequence = fopen(nom_fichier,"wb"))==NULL)
    {
        fprintf("Il m'est impossible d'ouvrir le fichier %s pour écriture \n",
            nom_fichier);
        return;
    }
}
fprintf(sequence,"%s",nom_image);
fwrite(&pos_x_dep,sizeof(int),1,sequence);
fwrite(&pos_x_arr,sizeof(int),1,sequence);
fwrite(&pos_y_dep,sizeof(int),1,sequence);
fwrite(&pos_y_arr,sizeof(int),1,sequence);

fclose(sequence);
}

```

```

/*          Cette fonction permet de quitter le programme selon le protocole          */
/*          convenable à l'environnement de travail                                  */

```

```
void quitter(item,event)
```

```
Frame item;
Event *event;
```

```
{
    fprintf(stderr,"Je quitte \n");
    xv_destroy_safe(cadre);

```

}

```
/*          Cette fonction affiche pixel par pixel l'image          */
```

```
void montre_carte(canvas,pw,display,win,xrects)
Canvas canvas;
Xv_window pw;
Display *display;
Window win;
Xv_xrectlist *xrects;

{
  int i,j;
  for(j=0;j<240; j++)
  {
    for(i=0; i<240 ; i++)
    {
      XSetForeground(display,gc,table_pixel[couleur(image_radar[j][i])]);
      XCopyPlane(display, point_mesure,win,gc,0,0,taille_pixel,taille_pixel,
                 i*2,j*2,1);
    }
  }
}
```

```
/*          Les objets composant le panneau de contrôle sont construits ici          */
```

```
void fabrique_panneau()
{
/*          Création du panneau de contrôle          */
panneau = (Panel) xv_create(cadre,PANEL,
                           XV_X,485,
                           XV_Y,2,
                           XV_HEIGHT,480,
                           XV_WIDTH,520,
                           NULL);
```

```
/*          Création du bouton de commande de chargement d'image          */
```

```
xv_create(panneau,PANEL_BUTTON,
          PANEL_NOTIFY_PROC,chargement_d_image,
          XV_X,30,
          XV_Y,30,
```

```
PANEL_LABEL_STRING,"Chargement d'image",
NULL);
```

```
/*          Création du bouton de commande d'incrémentat
```

```
ion d'image          */
xv_create(panneau,PANEL_BUTTON,
          PANEL_NOTIFY_PROC,prochaine_image,
          XV_X,30,
          XV_Y,60,
          PANEL_LABEL_STRING," Prochaine image ",
          NULL);
```

```
/*          Création du bouton de commande de sauvegarde d'ima
```

```
ge          */
xv_create(panneau,PANEL_BUTTON,
          PANEL_NOTIFY_PROC,sauvegarde,
          XV_X,30,
          XV_Y,90,
          PANEL_LABEL_STRING," Sauvegarde ",
          NULL);
```

```
/*          Création du bouton de commande de terminaison de programme
```

```
          */
xv_create(panneau,PANEL_BUTTON,
          PANEL_NOTIFY_PROC,quitter,
          XV_X,30,
          XV_Y,120,
          PANEL_LABEL_STRING," Quitter  ",
          NULL);
```

```
/*          Création de la note de répertoire des images          */
```

```
it_rep_image = (Panel_item) xv_create(panneau,PANEL_TEXT,
          PANEL_LABEL_STRING,"Répertoire image  :",
          XV_X,30,
          XV_Y,150,
          PANEL_VALUE_DISPLAY_LENGTH,40,
          PANEL_VALUE_STORED_LENGTH,100,
          XV_SHOW,TRUE,
          NULL);
```

```
*/          Création de la note de répertoire de sauvegarde          */
```

```
it_rep_sequence = (Panel_item) xv_create(panneau,PANEL_TEXT,
          PANEL_LABEL_STRING,"Répertoire séquence :",
```

```

XV_X,30,
XV_Y,180,
PANEL_VALUE_DISPLAY_LENGTH,40,
PANEL_VALUE_STORED_LENGTH,100,
XV_SHOW,TRUE,
NULL);

```

```

/*          Création de la note de nom de fichier de l'image          */

```

```

it_nom_image = (Panel_item) xv_create(panneau,PANEL_TEXT,
PANEL_LABEL_STRING,"Nom de l'image      :",
XV_X,30,
XV_Y,210,
PANEL_VALUE_DISPLAY_LENGTH,40,
PANEL_VALUE_STORED_LENGTH,100,
XV_SHOW,TRUE,
NULL);

```

```

/*          Création de la note de nom de fichier sauvegarde        */

```

```

it_nom_sequence= (Panel_item) xv_create(panneau,PANEL_TEXT,
PANEL_LABEL_STRING,"Nom de la séquence :",
XV_X,30,
XV_Y,240,
PANEL_VALUE_DISPLAY_LENGTH,40,
PANEL_VALUE_STORED_LENGTH,100,
XV_SHOW,TRUE,
NULL);

```

```

/*          Création des notes d'affichage de la position de la souris */

```

```

xv_create(panneau,PANEL_TEXT,
PANEL_LABEL_STRING,"Position de départ  ",
XV_X,30,
XV_Y,270,
PANEL_VALUE_DISPLAY_LENGTH, 0,
PANEL_VALUE_STORED_LENGTH, 0,
XV_SHOW,TRUE,
NULL);

```

```

xv_create(panneau,PANEL_TEXT,
PANEL_VALUE_DISPLAY_LENGTH, 0,
PANEL_VALUE_STORED_LENGTH, 0,
PANEL_LABEL_STRING,"Position d'arivee  ",
XV_X,250,
XV_Y,270,

```

```
XV_SHOW,TRUE,  
NULL);
```

```
it_x_depart = (Panel_item) xv_create(panneau,PANEL_TEXT,  
    PANEL_LABEL_STRING,"X :",  
    XV_X,30,  
    XV_Y,290,  
    XV_SHOW,TRUE,  
    NULL);
```

```
it_y_depart = (Panel_item) xv_create(panneau,PANEL_TEXT,  
    PANEL_LABEL_STRING,"Y :",  
    XV_X,110,  
    XV_Y,290,  
    PANEL_VALUE_DISPLAY_LENGTH,4,  
    PANEL_VALUE_STORED_LENGTH,4,  
    XV_SHOW,TRUE,  
    NULL);
```

```
it_x_arivee = (Panel_item) xv_create(panneau,PANEL_TEXT,  
    PANEL_LABEL_STRING,"X :",  
    XV_X,250,  
    XV_Y,290,  
    PANEL_VALUE_DISPLAY_LENGTH,4,  
    PANEL_VALUE_STORED_LENGTH,4,  
    XV_SHOW,TRUE,  
    NULL);
```

```
it_y_arivee = (Panel_item) xv_create(panneau,PANEL_TEXT,  
    PANEL_LABEL_STRING,"Y :",  
    XV_X,330,  
    XV_Y,290,  
    PANEL_VALUE_DISPLAY_LENGTH,4,  
    PANEL_VALUE_STORED_LENGTH,4,  
    XV_SHOW,TRUE,  
    NULL);
```

```
}
```

```
/*          Fonction permettant de montrer la position sélectionner avec la souris      */  
  
void montre_position(window,event)  
Xv_window window;  
Event *event;  
  
{  
    int position_x_depart,position_y_depart,position_x_arivee,position_y_arivee;  
  
    char no1[10],no2[10];  
    if (event_action(event) == ACTION_SELECT)  
    {  
        if (event_is_down(event))  
        {  
            position_x_depart = (int) (event_x(event) / 2);  
            position_y_depart = (int) (event_y(event) / 2);  
            sprintf(no1,"%d",position_x_depart);  
            sprintf(no2,"%d",position_y_depart);  
            xv_set(it_x_depart,PANEL_VALUE,no1,NULL);  
            xv_set(it_y_depart,PANEL_VALUE,no2,NULL);  
        }  
        if(event_is_up(event))  
        {  
            position_x_arivee = (int) (event_x(event) / 2);  
            position_y_arivee = (int) (event_y(event) / 2);  
            sprintf(no1,"%d",position_x_arivee);  
            sprintf(no2,"%d",position_y_arivee);  
            xv_set(it_x_arivee,PANEL_VALUE,no1,NULL);  
            xv_set(it_y_arivee,PANEL_VALUE,no2,NULL);  
        }  
    }  
}
```

Image de Plate



Chargement d'image

Prochain image

Sauvegarde

Quitter

Répertoire image : /opt/miracini/scr/plate/imagenes\_radar/image/

Répertoire séquence :

Nom de l'image : 180400-42026

Nom de la séquence :

Position de départ

X : 79 Y : 51

Position d'arrivée

X : 141 Y : 143

## Annexe 2

### Programme de calcul des affaiblissements

```

/*      Programme permettant de calculer les affaiblissements          */
/*      totaux sur des liens micro-ondes et de les compiler à        */
/*      l'intérieur d'un fichier pour analyse                          */
/*
/*      Ce programme devra permettre de calculer a partir            */
/*      d'une liste de fichier image de pluie de calculer l'affaiblissement */
/*      que subiront plusieurs liens micro-ondes. Le procédé         */
/*      de calcul consiste a simuler une intégration sur le parcours  */
/*      suivit par le lien. Chacune des valeurs ainsi obtenues sera  */
/*      ensuite traitée pour obtenir un histogramme.                  */
/*
/*      L'histogramme sera une fonction à quatre dimensions.         */
/*
/*      dimension:  taille                                           */
/*
/*      Fréquence      7      de 10 a 40 GHz (intervalle de 5)       */
/*      Longueur      10     de 2 a 20 km (intervalle de 2)          */
/*      Angle          18     de 0 a 340 degrés (intervalle de 20 degrés) */
/*      Affaiblissement 200   de 0 a 200 dB (intervalle de 1 dB)     */
/*
/*      Processus de fonctionnement:                                  */
/*
/*      Pour chaque valeur de (fréquence,longueur,angle) calculer la  */
/*      valeur de l'affaiblissement que subira chacun des liens et intégrer */
/*      la valeur de cet affaiblissement en additionnant 1 dans la case  */
/*      correspondante au triple et inférieure ou égale a la valeur    */
/*      de l'affaiblissement.                                          */
/*
/*      Ensuite il faut normaliser les valeurs contenues dans l'histogramme */
/*      en divisant chacune de ces valeurs par le nombre de lien utilisé  */
/*      pour faire les calculs. Il suffira alors de multiplier par 5 les nombres */
/*      obtenues pour obtenir le temps total moyen pendant lequel un lien  */
/*      aura subit le niveau d'affaiblissement donné.                 */

```

```

/*          Les fonctions nécessaires pour le fonctionnement du programme sont          */
/*
/*          -- Obtenir la liste des images radar                                     */
/*          -- Fonction d'intégration dans les images radar                         */
/*          -- Fonction de répartition des valeurs dans l'histogramme              */
/*          -- Fonction de sauvegarde des résultats (histogramme)                  */
/*          -- Fonction de chargement des images radar                             */
/*          -- Fonction de normalisation de l'histogramme                          */
/*          -- Fonction de chargement des positions d'origine                       */
/*          des liens micro-ondes                                                  */
/*          -- Fonction pour obtenir la valeur des paramètres                       */
/*          de la loi d'affaiblissement                                            */
*/

```

```

*/
/*
/*          Le processus sera composer de plusieurs boucles imbriquées.           */
/*          Ces boucles commenceront par la longueur (boucle intérieure),         */
/*          pour ensuite faire les angles et enfin les fréquences.                */
*/

```

```

#include <stdio.h>
#include <malloc.h>
#include <math.h>
#define Pi 3.141592

```

```

/*          Implantation des boucles                                             */
*/
for(frequence = 10;frequence <=40 ; frequence = frequence + 5)
{
for(angle = 0; angle <=340; angle = angle +20)
{
for(longueur=2;longueur<=20; longueur=longueur+2)
{
for(lien=0;lien<1000;lien++)
{
affaiblissement=integre(origine[lien],longueur,angle,frequence);
classifie(affaiblissement,longueur,angle,frequence);
}
}
}
}
}
*/

```

```
typedef struct {
    float x;
    float y;
} Point;
```

```
typedef struct{
    char **phrse;
    int nomb_image;
} definition;
```

```
typedef struct{
    int nomb_sys;
    Point *premier;
} Micro;
```

```
/*          Définition des variables globales du programme/          */
/*          CCIR contient les coefficients et exposants d'atténuation          */
/*          image est l'image radar en traitement          */
/*          histogramme contient les résultats          */
```

```
float CCIR[2][7];
float image[240][240];
int histogramme[7][10][18][200];
```

```
/*          La fonction intègre, calcule l'affaiblissement total sur un lien,          */
/*          il faut cependant que l'image soit chargée en mémoire, de même          */
/*          que les valeurs des coefficients et exposant de la loi d'affaiblissement          */
```

```
float integre(orig,longu,ang,freq)
Point orig;
int longu;
int ang;
int freq;
{
    float pas = 0.1,affaib=0.0;
    float a,b;
    float rad_ang;
    int freq_red;
    Point position;
    int i,j,k,N;

    position = orig;
```

```

N = (int)((float)longu/pas) + 1;
pas = (float)longu/(float)N;
freq_red = freq/5 -2;
a = CCIR[0][freq_red];
b = CCIR[1][freq_red];
printf(" a= %f b= %f\n",a,b);

```

```

rad_ang = (float)ang * Pi/180;

```

```

for (i=0;i<N;i++)

```

```

{

```

```

    affaib = affaib + a * pow(image[(int)(position.y + pas * cos(rad_ang)/2.0)][(int)(position.x + pas
* sin(rad_ang) /2)],b) * pas;

```

```

    position.x = position.x + pas *cos(rad_ang);

```

```

    position.y = position.y + pas*sin(rad_ang);

```

```

}

```

```

return(affaib);

```

```

}

```

```

/*          La fonction classifie permet de placer dans l'histogramme
/*          les résultats de calculs

```

```

*/

```

```

*/

```

```

void classifie(aff,longu,ang,freq)

```

```

float aff;

```

```

int longu;

```

```

int ang;

```

```

int freq;

```

```

{

```

```

    int affaib;

```

```

if(aff==0.0)

```

```

{

```

```

    histogramme[freq][longu][ang][0] ++;

```

```

}

```

```

else

```

```

{

```

```

    affaib = (int)aff + 1;

```

```

    histogramme[freq][longu][ang][affaib] ++;

```

```

}

```

```

}

```

```

/*          Cette fonction permet de charger les données provenant          */
/*          de la norme CCIR concernant la propagation dans la pluie        */
/*          le coefficient "a" est situé dans la colonne 0 et l'exposant     */
/*          "b" est situé dans la colonne 1                                  */

```

```

void chargement_CCIR()

```

```

{
  int i;
  float bidon;
  char *nom_fichier;
  FILE *fichier;

  nom_fichier=(char *)calloc(200,sizeof(char));
  nom_fichier = "/gel/mistassini/usr/plahaie/rep_donnee/loi_CCIR.data";
  if((fichier=fopen(nom_fichier,"rb"))==NULL)
  {
    printf("Le fichier donnee CCIR ne peut etre ouvert \n");
    exit(0);
  }

  for(i=0;i<7;i++)
  {
    fread(&bidon,sizeof(float),1,fichier);
    CCIR[0][i] = bidon;
    printf("%f \n",bidon);
    fread(&bidon,sizeof(float),1,fichier);
    CCIR[1][i] = bidon;
    printf("%f \n",bidon);
  }
  fclose(fichier);
}

```

```

/*          Cette fonction permettra de lire séquentiellement la          */
/*          liste des images radar les noms d'images radar sont contenues  */
/*          dans un fichier de type "ascii" et sont séquentiellement placées */
/*          dans ce fichier. Le fichier contient une donné de type "integer" */
/*          donnant le nombre d'images contenues dans le fichier les       */
/*          noms d'images contiennent un nombre toujours identiques        */
/*          de caractères (199208041552) soit 12. La fonction retourne     */
/*          des noms d'images ainsi que le nombre de ces images dans une   */
/*          structure de type définition                                    */
*/

```

```

definition nom_image_radar(nom_de_fichier_image)
char *nom_de_fichier_image;
{
    definition resultat;
    char **phrase,*chbid;
    FILE *fichier;
    int i,j,bidon;

    if ((fichier=fopen(nom_de_fichier_image,"rb"))==NULL)
    {
        printf("le fichier %s ne peut etre ouvert \n",nom_de_fichier_image);
        exit(0);
    }

    fread(&bidon,sizeof(int),1,fichier);
    phrase = (char **)calloc(bidon,sizeof(char *));

    for(i=0;i<bidon;i++)
    {

        chbid = (char *)calloc(13,sizeof(char));
        phrase[i] = (char *)calloc(13,sizeof(char));
        fscanf(fichier,"%s",chbid);
        strcpy(phrase[i],chbid);
        free(chbid);
    }

    resultat.phrse = phrase;
    resultat.nomb_image = bidon;
    return(resultat);
}

```

```
/*          Fonction permettant de lire une image radar de 240*240 compressée */
```

```
void chargement_image_radar(fichier_radar)
FILE *fichier_radar;
{
short vecteur[57600];
int i,j;
short bidon;

fseek(fichier_radar,512,0);
for(i=0;i<57600;i++)
    vecteur[i]=0;

for(i=0;i<57600;i++)
    {
    fread(&bidon,sizeof(short),1,fichier_radar);
    if(bidon<0)
        i=i-(int)bidon;
    else
        vecteur[i] = bidon;
    }
for(i=0;i<240;i++)
    {
    for(j=0;j<240;j++)
        image[i][j] = (float)(vecteur[i*240 +j]/10);
    }
}
```

```

/*          Fonction permettant de charger les positions et le nombre          */
/*          de systèmes micro-ondes qui seront utilisés pour calculer          */
/*          les statistiques d'affaiblissement                                  */

```

```

/*          Le premier nombre contenu dans le fichier est le nombre          */
/*          de systèmes il s'agit d'un entier. Les autres sont du type Point  */
/*          soit des "float" la position x et la position y étant placés à la suite */
/*          l'un de l'autre.                                                  */

```

```

/*          La fonction retourne une structure contenant le nombre de          */
/*          systèmes ainsi que le pointeur à la première structure.          */

```

```

Micro position_systemes_micro_onde()
{
  char *nom_de_fichier = "/gel/usr/plahaie/rep_donnee/origine_systeme_micro_onde.dta";
  FILE *fichier;
  int nombre_systeme;
  Point *bidon_fl;
  int i;
  Micro resultat;

  if((fichier=fopen(nom_de_fichier,"rb"))==NULL)
  {
    printf("Le fichier contenant les systemes micro_ondes ne peut etre ouvert \n");
    exit(0);
  }
  fread(&nombre_systeme,sizeof(int),1,fichier);
  resultat.nomb_sys = nombre_systeme;
  bidon_fl = (Point *)calloc(nombre_systeme,sizeof(Point));
  for(i=0;i<nombre_systeme;i++)
  {
    fread(&bidon_fl[i],sizeof(Point),1,fichier);
  }
  fclose(fichier);
  resultat.premier = bidon_fl;
  return(resultat);
}

```

```
/*          Fonction de sauvegarde des résultats des calculs d'affaiblissement          */
```

```
void sauve_normal_histo(nombre_systeme,nom_de_fichier)
int nombre_systeme;
char *nom_de_fichier;
{
int i,j,k,l;
float bidon;
FILE *fichier;

if((fichier==fopen(nom_de_fichier,"wb"))==NULL)
{
printf("Ca fait trois jours que je calcule pour rien \n");
exit(0);
}

for (i=0;i<7;i++)
{
for (j=0;j<10;j++)
{
for(k=0;k<18;k++)
{
for(l=0;l<200;l++)
{
bidon= (float)histogramme[i][j][k][l] / nombre_systeme;
fwrite(&bidon,sizeof(float),1,fichier);
}
}
}
}
}
fclose(fichier);
}
```

```

/*          Ceci est la fonction de contrôle général et de calcul du programme          */

main()
{
  definition liste_image;
  char *chemin_image = "/gel/mistassini/usr/plahaie/images_radar/image/";
  char *nom_fichier_image;
  char *nom_fichier_liste_image= "/gel/mistassini/usr/plahaie/rep_donnee/fichier_image.dta";
  FILE *fichier_image_radar;
  int i,j, fre, longu, angl, voyant, longueur_nom;
  Micro systeme_micro_onde;
  float affaiblissement;
  /*chargement des noms des images radar  */
  /*

printf("Entrez le nom du fichier contenant les images radar \n");
scanf("%s",nom_fichier_liste_image);
*/
chargement_CCIR();
liste_image = nom_image_radar(nom_fichier_liste_image);
systeme_micro_onde = position_systemes_micro_onde();
for(i=0;i<liste_image.nomb_image;i++)
{
  longueur_nom=strlen(chemin_image)+strlen(liste_image.phrse[i]) +1;
  nom_fichier_image=(char *)calloc(longueur_nom,sizeof(char));
  strcpy(nom_fichier_image,chemin_image);
  strcat(nom_fichier_image,liste_image.phrse[i]);
  voyant=0;
  if((fichier_image_radar=fopen(nom_fichier_image,"rb"))==NULL)
  {
    printf("Le fichier %s ne peut etre ouvert \n",liste_image.phrse[i]);
    voyant=1;
  }
  free(nom_fichier_image);
  if(voyant==0)
  {
    chargement_image_radar(fichier_image_radar);
    fclose(fichier_image_radar);
    for(j=0;j<systeme_micro_onde.nomb_sys;j++)
    {
      for(fre=0;fre<7;fre++)
      {
        for(longu=0;longu<10;longu++)
        {
          for(angl=0;angl<18;angl++)
          {

```

```
    affaiblissement=integre(systeme_micro_onde.premier[i],
        (longu+1)*2,angl*10,(fre+2)*5);

    classifie(affaiblissement,longu,angl,fre);
}
}
}
}
}
}

sauve_normal_histo(systeme_micro_onde.nomb_sys,"/gel/usr/plahaie/rep_donnee/
histogramme_affaiblissement.dta");
}
```

## Références bibliographiques

- [1] J.O. Laws and D.A. Parsons, "The relation of raindrop size to intensity", Transaction of the American Geophysical Union, vol. 24, pp. 452-460, 1943
- [2] J.S. Marshall and W. McK. Palmer, "The relation of raindrop with size", Journal of Meteorology, vol. 5, pp. 165-166, Aug. 1948
- [3] R.L. Olsen, D.V. Rogers and D.B. Hodge, "The  $aR^b$  relation in the calculation of rain attenuation", IEEE Transactions on Antennas and Propagation, vol. 26, no2, pp. 318-329, March 1978
- [4] R.K. Crane, "The prediction of attenuation by rain", IEEE Transactions on Communications, vol. 28, no9, septembre 1980
- [5] A. Kilambi and J.S. Marshall, "A radar study on terrestrial microwave links", Radio Science, vol. 12, no 1, pp. 53-61, Jan 1977.
- [6] P. Walteuffel, "Attenuation des ondes hyperfréquences par la pluie: une mise au point", Annales des Télécommunications, vol 28, no 8, pp. 254-272, 1973



