

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
KATEDRA KYBERNETIKY A UMELEJ INTELIGENCIE

**Mapovanie priestoru pomocou zhlukovania
obrazov Kohonenovou sieťou**

BAKALÁRSKA PRÁCA

KN Map

ZDROJOVÝ KÓD PROGRAMU

Michal Puheim

Vedúci bakalárskej práce:

Ing. Rudolf Jakša, PhD.

Konzultant bakalárskej práce:

Ing. Rudolf Jakša, PhD.

Košice 2011

```

/*****
* * * *
* * * * Mapovanie priestoru pomocou zhukovania obrazov Kohonenovou sietou * * * */
* * * *
* * * * Bakalarska Praca * * * */
* * * *
* * * * KN_Map.c * * * */
* * * * verzia 1.0.0 * * * */
* * * *
* * * * Michal Puheim * * * */
* * * * maj 2011 * * * */
* * * *
*****/

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<math.h>

/* Konstanty */
#define H_T 1 /* Adaptacna vyska */
#define DL_RET 50 /* Standardna dlzka retazcov */
#define MAX_POC_OBRAZKOV 999 /* Maximalny pocet obrazkov */
#define MAX_ROZMER_OBRAZKA 100000 /* Max. velkost vektoru vstupneho obrazka */
#define MAX_ROZMER_VYSTUPU 20 /* Max. velkost hrany vystupnej matice */
#define MAX_DELENI_KRUZNICE 36 /* Max. pocet deleni kruznice na vystupe */

/* Globalne premenne */
char nazov_experimentu[DL_RET];
int rozmer_vystupu_x, rozmer_vystupu_y, poc_deleni_kruznice, poc_obrazkov,
rozmer_obrazka_x, rozmer_obrazka_y, farebny_rozsah, inicial, spolu_cyklov_ucenia;
double gama, ri_t;
double vystupna_vrstva_x_y_d[MAX_ROZMER_VYSTUPU][MAX_ROZMER_VYSTUPU][MAX_DELENI_KRUZNICE];
int obrazok[MAX_ROZMER_OBRAZKA], shuffle[MAX_POC_OBRAZKOV];
double *vaha_x_y_d_v[MAX_ROZMER_VYSTUPU][MAX_ROZMER_VYSTUPU][MAX_DELENI_KRUZNICE];

/* Prototypy funkcii */
void vytvorit_experiment(void);
void nacitat_experiment(void);
void clr_input(void);
char nacitaj_znak(void);
int minimum(int a, int b);
int spocitaj_obrazky(void);
void zisti_rozmer_obrazka(void);
void alokuj_pamat(void);
void nacitaj_obrazok(int cislo);
int nacitaj_inicializacny_obrazok(int x, int y, int d);
void inicializuj_siet(void);
void nahodne_poradie(int min_poc, int max_poc);
void vizualizuj_vahy(void);
void ucenie(void);
void testuj_siet(void);
void hlavne_menu(void);

/* Hlavna funkcia */
main()
{
    /* Inicializacia generatora nahodnych cisel */
    srand ( (unsigned)time ( NULL ) );
    /* Uvodne menu */
    char vstup = EOF;
    while (vstup != '0')
    {
        printf("\nUvodne MENU\n -1- Vytvorit novy experiment\n -2- Nacitat existujuci
experiment\n -0- Koniec\n \nZadajte vasu volbu: ");
        vstup = nacitaj_znak();
        switch(vstup)
        {
            case '1':
                vytvorit_experiment();
                break;
            case '2':
                nacitat_experiment();
                break;
            case '0':
                return;
            default:
                printf("\nNespravne zadana hodnota. Opakujte volbu.\n");
                break;
        }
        vstup = EOF;
    }
}

```

```

/* Vycisti nepotrebné znaky na standardnom vstupe */
void clr_input()
{
    while(getchar()!='\n'); /* Precistujeme celý vstupný zásobník */
}

/* Nacítanie znaku zo standardného vstupu */
/* Funkcia vráti jeden (prvý) znak bezohľadu na to, koľko bolo zadávaných zna- */
/* kov alebo či bol stlačený iba enter. Zbytočné znaky zo zásobníka vymaze. */
char nacitaj_znak()
{
    char zn;
    zn=getchar(); /* Nacítame znak */
    if (zn!='\n')
        while(getchar()!='\n'); /* Precistujeme celý vstupný zásobník */
    return zn;
}

/* Funkcia vráca minimim z dvoch čísel */
int minimum(int a, int b)
{
    if (a>b)
        return b;
    else
        return a;
}

/* Funkcia zabezpečuje načítanie údajov potrebných na vytvorenie novej neuronovej siete */
void vytvorit_experiment()
{
    printf("\nVYTvorenie NOVEHO EXPERIMENTU\n\n");
    printf("Zadajte názov experimentu: ");
    while (scanf("%s",nazov_experimentu)==0)
    {
        clr_input();
        printf("\nChyba pri načítaní údajov!");
        printf("\nZadajte názov experimentu: ");
    }
    clr_input();
    printf("Zadajte rozmer výstupnej matice: \n X: ");
    while (scanf("%d",&rozmer_vystupu_x)==0||rozmer_vystupu_x<1||
rozmer_vystupu_x>MAX_ROZMER_VYSTUPU)
    {
        clr_input();
        printf("\nChyba pri načítaní údajov! Zadajte celé číslo v rozmedzí od 1 do
%d.",MAX_ROZMER_VYSTUPU);
        printf("\n X: ");
    }
    clr_input();
    printf(" Y: ");
    while (scanf("%d",&rozmer_vystupu_y)==0||rozmer_vystupu_y<1||
rozmer_vystupu_y>MAX_ROZMER_VYSTUPU)
    {
        clr_input();
        printf("\nChyba pri načítaní údajov! Zadajte celé číslo v rozmedzí od 1 do
%d.",MAX_ROZMER_VYSTUPU);
        printf("\n Y: ");
    }
    clr_input();
    printf("Zadajte počet delení kružníc na výstupnej matici: ");
    while (scanf("%d",&poc_deleni_kruznice)==0||poc_deleni_kruznice<1||
poc_deleni_kruznice>MAX_DELENI_KRUZNICE)
    {
        clr_input();
        printf("\nChyba pri načítaní údajov! Zadajte celé číslo v rozmedzí od 1 do
%d.",MAX_DELENI_KRUZNICE);
        printf("\n Zadajte počet delení kružníc na výstupnej matici: ");
    }
    clr_input();
    printf("Zadajte typ inicializácie neuronovej siete: \n -1- Nahodna\n -2- Nahodne
obrazky\n -3- Manualne urcena\nZadajte vašu voľbu: ");
    while (scanf("%d",&inicial)==0||inicial<1||inicial>3)
    {
        clr_input();
        printf("\nChyba pri načítaní údajov!");
        printf("\nZadajte typ inicializácie neuronovej siete: \n -1- Nahodna\n -2- Nahodne
obrazky\n -3- Manualne urcena\nZadajte vašu voľbu: ");
    }
    clr_input();
    printf("\n-----");
    printf("\nNázov experimentu: %s",nazov_experimentu);
    printf("\nRozmer výstupnej matice: %d x %d",rozmer_vystupu_x,rozmer_vystupu_y);
    printf("\nTyp inicializácie neuronovej siete: ");
}

```

```

        if (inicial == 1)
            printf("Nahodna.");
        else if (inicial == 2)
            printf("Nahodne poprehadzovane obrazky.");
        else if (inicial == 3)
            printf("Manualne urcena.");
        printf("\n-----");
        printf("\n\nPre pokracovanie stlacte enter... ");
        nacitaj_znak();
        poc_obrazkov = spocitaj_obrazky();
        zisti_rozmer_obrazka();
        alokuj_pamat();
        inicializuj_siet();
        printf("\n\nPre pokracovanie stlacte enter... ");
        nacitaj_znak();
        spolu_cyklov_ucenia = 0;
        hlavne_menu();
    }

/* Funkcia zabezpecuje nactanie ulozeného experimentu - zatiaľ neimplementované */
void nactat_experiment()
{
}

/* Funkcia vracia počet obrazkov v podpriecinku './obrazky/' s názvom v tvare 'image000' */
int spocitaj_obrazky()
{
    int i;
    FILE *f;
    char nazov_suboru[DL_RET];
    printf("\nZisťujem počet obrazkov v adresari './obrazky/'...");
    for (i=0; i<MAX_POC_OBRAZKOV; i++)
    {
        if (i<9)
            strcpy(nazov_suboru, "obrazky/image00");
        else if (i<99)
            strcpy(nazov_suboru, "obrazky/image0");
        else if (i<999)
            strcpy(nazov_suboru, "obrazky/image");
        sprintf(nazov_suboru, "%s%d%s", nazov_suboru, i+1, ".ppm");
        /*printf("\nHľadám subor './%s'", nazov_suboru);*/
        /* Otvoríme subor */
        if ((f=fopen(nazov_suboru, "r"))==NULL)
        {
            /*printf(" - Subor sa nepodarilo otvoriť!");*/
            printf(" %d obrazkov.", i);
            return i;
        }
        if ((fclose(f))==EOF)
            printf("\nChyba pri zatváraní suboru!\n");
    }
    printf(" %d obrazkov.", i);
    return i;
}

/* funkcia stanoví rozmer obrazkov na základe údajov v prvom načítanom obraze */
void zisti_rozmer_obrazka()
{
    char c;
    FILE *f;
    printf("\nZisťujem rozmer obrazkov...");
    /* Otvoríme subor */
    if ((f=fopen("obrazky/image001.ppm", "r"))==NULL)
    {
        printf("\nCHYBA! - Subor sa nepodarilo otvoriť!");
        printf("\nRozmer obrazku nebolo možné zistiť!");
        printf("\nKritická chyba. Stlačte enter pre ukončenie programu...");
        nacitaj_znak();
        exit(EXIT_FAILURE);
    }
    /* Citáme zo suboru */
    while (((c=fgetc(f))!='\n') && (c!=EOF)); /* Preskocenie riadka */
    while (((c=fgetc(f))!='\n') && (c!=EOF)); /* Preskocenie riadka */
    if (fscanf(f, "%d %d %d", &rozmer_obrazka_x, &rozmer_obrazka_y, &farebny_rozsah)!=3)
    {
        printf("\nChyba pri citaní suboru!");
        printf("\nRozmer obrazku nebolo možné zistiť!");
        printf("\nKritická chyba. Stlačte enter pre ukončenie programu...");
        nacitaj_znak();
        exit(EXIT_FAILURE);
    }
    if ((fclose(f))==EOF)
        printf("\nChyba pri zatváraní suboru!\n");
}

```

```

        printf("    Rozmer    je    %d    x    %d.    Farebny    rozsah    je
%d.",rozmer_obrazka_x,rozmer_obrazka_y,farebny_rozsah);
    if (3 * rozmer_obrazka_x * rozmer_obrazka_y > MAX_ROZMER_OBRAZKA)
    {
        printf("\nChyba! Rozmer obrazka 3 * %d * %d = %d je vacsi ako maximalna povolená
hodnota %d.",rozmer_obrazka_x,rozmer_obrazka_y,3 * rozmer_obrazka_x * rozmer_obrazka_y,
MAX_ROZMER_OBRAZKA);
        printf("\nJe potrebne zmensit rozmer vstupnych obrazkov.");
        printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
        nacitaj_znak();
        exit(EXIT_FAILURE);
    }
}

/* Alokacia pamate pre neuronovu siet */
void alokuj_pamat()
{
    int x,y,d;
    /* Alokacia pamate pre vahy */
    printf("\nAlokujem pamat...");
    for (d = 0; d < poc_deleni_kruznice; d++)
    {
        for (x = 0; x < rozmer_vystupu_x; x++)
        {
            for (y = 0; y < rozmer_vystupu_y; y++)
            {
                if((vaha_x_y_d_v[x][y][d] = (double *) malloc(3 * rozmer_obrazka_x *
rozmer_obrazka_y * sizeof(double)))==NULL)
                {
                    printf("\nChyba pri alokacii pamate! Zrejme nie je dostatok
pamate."); /* Kontrola alokacie */
                    /*printf("\nx: %d, y: %d",x,y);*/
                    printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
                    nacitaj_znak();
                    exit(EXIT_FAILURE);
                }
            }
        }
    }
    printf(" Alokacia pamate prebehla uspesne.");
}

/* Nacitanie udajov obrazku zo suboru so pamate programu */
void nacitaj_obrazok(int cislo)
{
    char c;
    FILE *f;
    char nazov_suboru[DL_RET];
    int rozm_x,rozm_y,pom,i;
    if (cislo<10)
        strcpy(nazov_suboru, "obrazky/image00");
    else if (cislo<100)
        strcpy(nazov_suboru, "obrazky/image0");
    else if (cislo<1000)
        strcpy(nazov_suboru, "obrazky/image");
    sprintf(nazov_suboru, "%s%d%s", nazov_suboru, cislo, ".ppm");
    /*DEBUG printf("\nNacitam obrazok zo suboru './%s'",nazov_suboru);*/
    /* Otvorime subor */
    if ((f=fopen(nazov_suboru,"r"))==NULL)
    {
        printf("\nChyba! Subor './%s' sa nepodarilo otvorit!",nazov_suboru);
        printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
        nacitaj_znak();
        exit(EXIT_FAILURE);
    }
    /* Citame zo suboru */
    while (((c=fgetc(f))!='\n')&&(c!=EOF)); /* Preskocenie riadka */
    while (((c=fgetc(f))!='\n')&&(c!=EOF)); /* Preskocenie riadka */
    if (fscanf(f,"%d %d",&rozm_x,&rozm_y)!=2)
    {
        printf("\nChyba pri nacitani obrazku zo suboru './%s'! Rozmer obrazku nebolo mozne
nacitat.",nazov_suboru);
        printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
        nacitaj_znak();
        exit(EXIT_FAILURE);
    }
    if (fscanf(f,"%d",&pom)!=1)
    {
        printf("\nChyba pri nacitani obrazku zo suboru './%s'! Farebny rozsah nebolo mozne
nacitat.",nazov_suboru);
        printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
        nacitaj_znak();
        exit(EXIT_FAILURE);
    }
}

```

```

    if ((rozm_x != rozmer_obrazka_x) || (rozm_y != rozmer_obrazka_y))
    {
        printf("\nChyba pri nactani obrazku zo suboru './%s!',nazov_suboru);
        printf("\nRozmer obrazku %dx%d nesedi s pozadovanim rozmerom %dx
%d.",rozm_x,rozm_y,rozmer_obrazka_x,rozmer_obrazka_y);
        printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
        nactaj_znak();
        exit(EXIT_FAILURE);
    }
    if (pom != farebny_rozsah)
    {
        printf("\nChyba pri nactani obrazku zo suboru './%s!',nazov_suboru);
        printf("\nFarebny rozsah obrazku %d nesedi s pozadovanim rozsahom
%d.",pom,farebny_rozsah);
        printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
        nactaj_znak();
        exit(EXIT_FAILURE);
    }
    for (i = 0; i < 3 * rozmer_obrazka_x * rozmer_obrazka_y; i++)
    {
        if (fscanf(f,"%d",&pom)!=1)
        {
            printf("\nChyba pri nactani obrazku zo suboru './%s!',nazov_suboru);
            printf("\nNactanie %d. bodu farebneho rastra zlyhalo.",i);
            printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
            nactaj_znak();
            exit(EXIT_FAILURE);
        }
        if (pom>farebny_rozsah)
        {
            printf("\nChyba pri nactani obrazku zo suboru './%s!',nazov_suboru);
            printf("\nHodnota (%d) %d. bodu farebneho rastra je vyssia ako maximum
%d.",pom,i,farebny_rozsah);
            printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
            nactaj_znak();
            exit(EXIT_FAILURE);
        }
        obrazok[i]=pom;
    }
    if((fclose(f))==EOF)
        printf("\nChyba pri zatvarani suboru!\n");
}

/* Specialne nactanie pre ucely manualnej inicializacie - subor musi mat masku v tvare
"x_y_d" */
int nactaj_inicializacny_obrazok (int x, int y, int d)
{
    char c;
    FILE *f;
    char nazov_suboru[DL_RET];
    int rozm_x,rozm_y,pom,i;
    /* Urcenie nazvu suboru */
    sprintf(nazov_suboru, "inicializacia/%d_%d_%d.ppm",x,y,d);
    /*DEBUG printf("\nOtvaram subor './%s' pre nactanie obrazku",nazov_suboru);*/
    /* Otvorime subor */
    if((f=fopen(nazov_suboru,"r"))==NULL)
    {
        /* Ak sa subor nepodarilo otvorit, zrusime nactanie. */
        /* printf("\nSubor './%s' sa nepodarilo otvorit!",nazov_suboru); */
        return 0;
    }
    /* Citame zo suboru */
    /* printf("\nNactam obrazok zo suboru './%s'",nazov_suboru);*/
    while (((c=fgetc(f))!='\n')&&(c!=EOF)); /* Preskocenie riadka */
    while (((c=fgetc(f))!='\n')&&(c!=EOF)); /* Preskocenie riadka */
    if (fscanf(f,"%d %d",&rozm_x,&rozm_y)!=2)
    {
        printf("\nChyba pri nactani obrazku zo suboru './%s'! Rozmer obrazku nebolo mozne
nactat.",nazov_suboru);
        printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
        nactaj_znak();
        exit(EXIT_FAILURE);
    }
    if (fscanf(f,"%d",&pom)!=1)
    {
        printf("\nChyba pri nactani obrazku zo suboru './%s'! Farebny rozsah nebolo mozne
nactat.",nazov_suboru);
        printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
        nactaj_znak();
        exit(EXIT_FAILURE);
    }
    if ((rozm_x != rozmer_obrazka_x) || (rozm_y != rozmer_obrazka_y))
    {
        printf("\nChyba pri nactani obrazku zo suboru './%s!',nazov_suboru);

```

```

        printf("\nRozmer obrazku %dx%d nesedi s pozadovanim rozmerom %dx
%d.", rozm_x, rozm_y, rozmer_obrazka_x, rozmer_obrazka_y);
        printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
        nacitaj_znak();
        exit(EXIT_FAILURE);
    }
    if (pom != farebny_rozsah)
    {
        printf("\nChyba pri nacistani obrazku zo suboru './%s!', nazov_suboru);
        printf("\nFarebny rozsah obrazku %d nesedi s pozadovanim rozsahom
%d.", pom, farebny_rozsah);
        printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
        nacitaj_znak();
        exit(EXIT_FAILURE);
    }
    for (i = 0; i < 3 * rozmer_obrazka_x * rozmer_obrazka_y; i++)
    {
        if (fscanf(f, "%d", &pom) != 1)
        {
            printf("\nChyba pri nacistani obrazku zo suboru './%s!', nazov_suboru);
            printf("\nNacistanie %d. bodu farebneho rastra zlyhalo.", i);
            printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
            nacitaj_znak();
            exit(EXIT_FAILURE);
        }
        if (pom > farebny_rozsah)
        {
            printf("\nChyba pri nacistani obrazku zo suboru './%s!', nazov_suboru);
            printf("\nHodnota (%d) %d. bodu farebneho rastra je vyssia ako maximum
%d.", pom, i, farebny_rozsah);
            printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
            nacitaj_znak();
            exit(EXIT_FAILURE);
        }
        obrazok[i] = pom;
    }
    if ((fclose(f)) == EOF)
        printf("\nChyba pri zatvarani suboru!\n");
    return 1;
}

/* Inicializacia hodnot vah pred ucenim siete */
void inicializuj_siet()
{
    int x, y, d, v, r;
    double r_double;
    /* Inicializacia vah */
    /* -nahodna */
    printf("\nInicializujem vahy.");
    for (v = 0; v < 3 * rozmer_obrazka_x * rozmer_obrazka_y; v++)
    {
        for (d = 0; d < poc_deleni_kruznice; d++)
        {
            for (x = 0; x < rozmer_vystupu_x; x++)
            {
                for (y = 0; y < rozmer_vystupu_y; y++)
                {
                    r = rand() / ( RAND_MAX / 10000 + 1 ); /*Nahodne cele cislo od 0 do
10000*/
                    r_double = (double)r; /*Pretypovanie celeho cisla na realne*/
                    vaha_x_y_d_v[x][y][d][v] = r_double/10000; /*Nahodne realne cislo od 0
do 1*/
                }
            }
        }
    }
    /*DEBUG if (v % 1000 == 1) printf("\n%d - %f", v, vaha_x_y_d_v[x-1][y-1][d-1][v]);*/
    if (v % 100000 == 0)
        printf(".");
}
/* -nahodne poprehadzovane obrazky */
if (inicial == 2)
{
    for (d = 0; d < poc_deleni_kruznice; d++)
    {
        nahodne_poradie(1, poc_obrazkov);
        for (x = 0; x < rozmer_vystupu_x; x+=2)
        {
            for (y = 0; y < rozmer_vystupu_y; y+=2)
            {
                /*DEBUG printf("\n%d | shuffle> %d", 1 + (y/2) + (x/2) *
(rozmer_vystupu_y/2), shuffle[1 + (y/2) + (x/2) * (rozmer_vystupu_y/2)]);*/
                /*DEBUG printf(" x> %d y> %d", x, y);*/
                /*DEBUG nacitaj_znak();*/
                if (1 + (y/2) + (x/2) * (rozmer_vystupu_y/2) <= poc_obrazkov)

```

```

        {
            nacitaj_obrazok(shuffle[1 + (y/2) + (x/2) * (rozmer_vystupu_y/2)]);
            for (v = 0; v < 3 * rozmer_obrazka_x * rozmer_obrazka_y; v++)
                vaha_x_y_d_v[x][y][d][v] = ((double)obrazok[v])/farebny_rozsah;
        }
    }
    printf(".");
}
}
/* -manualne urcena podla nazvov suborov inicializacnych obrazkov */
else if (inicial == 3)
{
    for (d = 0; d < poc_deleni_kruznice; d++)
    {
        for (x = 0; x < rozmer_vystupu_x; x++)
        {
            for (y = 0; y < rozmer_vystupu_y; y++)
            {
                if (nacitaj_inicializacny_obrazok(x,y,d) == 1)
                {
                    for (v = 0; v < 3 * rozmer_obrazka_x * rozmer_obrazka_y; v++)
                        vaha_x_y_d_v[x][y][d][v] = ((double)obrazok[v])/farebny_rozsah;
                }
            }
        }
        printf(".");
    }
    spolu_cyklov_ucenia = 0;
    printf(" Inicializacia ukoncena.");
}

/* Pseudo-nahodne poprehadzovanie poradia v poli */
void nahodne_poradie(int min_poc, int max_poc)
{
    int i,j,pom,opakuje_sa;
    for (i = min_poc; i <= max_poc; i++)
        shuffle[i] = -1;
    for (i = min_poc; i <= max_poc; i++)
    {
        /* Priradime nahodne cislo */
        while ((pom = min_poc + rand() / ( RAND_MAX / (1+max_poc-min_poc))) > max_poc);
        /*Nahodne cele cislo od min_poc do max_poc*/
        opakuje_sa = 0;
        /* Overime, ci sa neopakuje */
        for (j = min_poc; j <= max_poc; j++)
        {
            if ((pom == shuffle[j]) && (j != i))
                opakuje_sa = 1;
        }
        /* Ak sa opakuje, priradime prve volne cislo. (Postupujeme od min_poc vyssie.)*/
        if (opakuje_sa == 1)
        {
            for (pom = min_poc; pom <= max_poc; pom++)
            {
                opakuje_sa = 0;
                for (j = min_poc; j <= max_poc; j++)
                {
                    if (pom == shuffle[j])
                    {
                        opakuje_sa = 1;
                        break;
                    }
                }
                if (opakuje_sa == 0)
                    break;
            }
        }
        /* Pre dany prvok v poli priradime jeho nahodne premiestnenie */
        shuffle[i] = pom;
        /*DEBUG printf("\n%d %d",i,shuffle[i]);*/
    }
    /*DEBUG nacitaj_znak();*/
}

/* Vizualizacia hodnot vah do suboru */
void vizualizuj_vahy()
{
    FILE *f;
    char nazov_suboru[DL_RET];
    int x,y,d,v,xv,yv,xi,yi,zmensenie;
    int pixel_R, pixel_G, pixel_B;

```



```

/* Adaptivne nastavenie zmensenia obrazkov na vystupnej matici podla velkosti vstupneho
obrazka */
if (rozmer_obrazka_x*rozmer_obrazka_y>=640*480)
    zmensenie = 10;
else if (rozmer_obrazka_x*rozmer_obrazka_y>=320*240)
    zmensenie = 5;
else if (rozmer_obrazka_x*rozmer_obrazka_y>=160*120)
    zmensenie = 2;
else
    zmensenie = 1;
/* Pre vsetky diely kruznice "d", tj. vsetky uhly otocenia, vytvorime subor s maticou
modelov obrazkov v danom uhle */
for (d = 0; d < poc_deleni_kruznice; d++)
{
    /* Urcenie nazvu suboru*/
    if ((spolu_cyklov_ucenia < 10)&&(d < 10))
        sprintf(nazov_suboru, "vystup/%s_%dx%d_cyklus-0%d_smer-0%d.ppm",
nazov_experimentu, rozmer_vystupu_x, rozmer_vystupu_y, spolu_cyklov_ucenia, d);
    else if ((spolu_cyklov_ucenia >= 10)&&(d < 10))
        sprintf(nazov_suboru, "vystup/%s_%dx%d_cyklus-%d_smer-0%d.ppm",
nazov_experimentu, rozmer_vystupu_x, rozmer_vystupu_y, spolu_cyklov_ucenia, d);
    else if ((spolu_cyklov_ucenia < 10)&&(d >= 10))
        sprintf(nazov_suboru, "vystup/%s_%dx%d_cyklus-0%d_smer-%d.ppm",
nazov_experimentu, rozmer_vystupu_x, rozmer_vystupu_y, spolu_cyklov_ucenia, d);
    else if ((spolu_cyklov_ucenia >= 10)&&(d >= 10))
        sprintf(nazov_suboru, "vystup/%s_%dx%d_cyklus-%d_smer-%d.ppm",
nazov_experimentu, rozmer_vystupu_x, rozmer_vystupu_y, spolu_cyklov_ucenia, d);
    /* Otvorenie suboru pre zapis*/
    printf("\nVizualizujem vahy do suboru './%s'...",nazov_suboru);
    if((f=fopen(nazov_suboru,"w"))==NULL)
    {
        printf("\nChyba pri otvarani suboru './%s'.",nazov_suboru);
        return;
    }
    /* Hlavicka suboru */
    fprintf(f,"P3\n");
    fprintf(f,"# Vizualizacia vah kohonenovej neuronovej siete. Nazov experimentu
'%s'.\n",nazov_experimentu);
    fprintf(f,"%d %d\n",rozmer_vystupu_x * rozmer_obrazka_x /
zmensenie,rozmer_vystupu_y * rozmer_obrazka_y / zmensenie);
    fprintf(f,"%d\n",farebny_rozsah);
    /* Hodnoty pixelov */
    for (y = 0; y < rozmer_vystupu_y; y++)
    {
        for (yv = 0; yv < rozmer_obrazka_y; yv+=zmensenie)
        {
            for (x = 0; x < rozmer_vystupu_x; x++)
            {
                /*
for (v = 0; v < 3 * rozmer_obrazka_x * rozmer_obrazka_y; v++)
{
    fprintf(f,"%d", (int) (farebny_rozsah*vaha_x_y_d_v[x][y][d][v]));
    if ((v+1)%3 != 0)
        fprintf(f," ");
    else
        fprintf(f,"\n");
}
*/
for (xv = 0; xv < rozmer_obrazka_x; xv+=zmensenie)
{
    pixel_R = (int) (farebny_rozsah*vaha_x_y_d_v[x][y][d][3 * (yv *
rozmer_obrazka_x + xv)    ]);
    pixel_G = (int) (farebny_rozsah*vaha_x_y_d_v[x][y][d][3 * (yv *
rozmer_obrazka_x + xv) + 1]);
    pixel_B = (int) (farebny_rozsah*vaha_x_y_d_v[x][y][d][3 * (yv *
rozmer_obrazka_x + xv) + 2]);
    fprintf(f,"%d %d %d\n",pixel_R,pixel_G,pixel_B);
}
}
}
}
if((fclose(f))==EOF)
    printf("\nChyba pri zatvarani suboru './%s'.\n",nazov_suboru);
}
printf(" Hotovo.");
}

/* Ucenie neuronovej siete */
void ucenie ()
{
    printf("\nUCENIE NEURONOVEJ SIETE");
    char pom;
    int poc_cyklov_ucenia,x,y,d,x_vit,y_vit,d_vit,v,iteracia,akt_obr,vizualizacia;
    double dj; /*vzdialenost medzi neuronmi*/

```

```

printf("\nZadajte parameter ucenia gama: ");
while (scanf("%lf",&gama)==0)
{
    clr_input();
    printf("\nChyba pri nacistani udajov!");
    printf("\nZadajte parameter ucenia gama: ");
}
clr_input();
printf("Zadajte polomer susednosti ri_t: ");
while (scanf("%lf",&ri_t)==0)
{
    clr_input();
    printf("\nChyba pri nacistani udajov!");
    printf("\nZadajte polomer susednosti ri_t: ");
}
clr_input();
printf("Zadajte pocet cyklov ucenia: ");
while (scanf("%d",&poc_cyklov_ucenia)==0)
{
    clr_input();
    printf("\nChyba pri nacistani udajov!");
    printf("\nZadajte pocet cyklov ucenia: ");
}
clr_input();
printf("Prajete vizualizovat vahy do suboru po kazdom cykle ucenia?(A/N): ");
while (((vizualizacia=nacitaj_znak())!='A')&&(vizualizacia!='N'))
{
    printf("\nChyba pri nacistani udajov!");
    printf("\nPrajete vizualizovat vahy do suboru po kazdom cykle ucenia?(A/N): ");
}
    printf("\nZadane parametre: gama = %.3f, ri_t = %.3f, pocet iteracii = %d.",gama,ri_t,poc_cyklov_ucenia);
    printf("\nPrajete si zacat ucenie?(A/N): ");
    while (((pom=nacitaj_znak())!='A')&&(pom!='N'))
    {
        printf("\nChyba pri nacistani udajov!");
        printf("\nPrajete si zacat ucenie?(A/N): ");
    }
    if (pom == 'N')
    {
        printf("\nUcenie siete bolo zrusene. Pre pokracovanie stlacte enter... ");
        nacitaj_znak();
        return;
    }
    /* Ucenie siete */
    printf("\nZacinam ucenie neuronovej siete...");
    if (vizualizacia == 'A')
        vizualizuj_vahy();
    for (iteracia = 0; iteracia < poc_cyklov_ucenia; iteracia++) /* opakuj pre vsetky cykly ucenia */
    {
        /* Nahodne poprehadzovanie poradia vstupnych obrazkov*/
        nahodne_poradie(1,poc_obrazkov);
        /* Vypocet vystupu siete */
        for (akt_obr = 1; akt_obr <= poc_obrazkov; akt_obr++) /* Opakuj pre vsetky obrazky */
        {
            printf("\n%d.   cyklus   ucenia,   %d.   obrazok (%d)...",iteracia+1,akt_obr,shuffle[akt_obr]);
            nacitaj_obrazok(shuffle[akt_obr]);
            /* Nastavenie vystupov na 0 */
            for (x = 0; x < rozmer_vystupu_x; x++)
            {
                for (y = 0; y < rozmer_vystupu_y; y++)
                {
                    for (d = 0; d < poc_deleni_kruznice; d++)
                        vystupna_vrstva_x_y_d[x][y][d] = 0;
                }
            }
            /* Hladanie vitazneho neuronu */
            x_vit = 0;
            y_vit = 0;
            d_vit = 0;
            for (d = 0; d < poc_deleni_kruznice; d++)
            {
                for (x = 0; x < rozmer_vystupu_x; x++)
                {
                    for (y = 0; y < rozmer_vystupu_y; y++)
                    {
                        for (v = 0; v < 3 * rozmer_obrazka_x * rozmer_obrazka_y; v++)
                        {
                            vystupna_vrstva_x_y_d[x][y][d] += pow((vaha_x_y_d_v[x][y][d][v] - ((double)obrazok[v])/farebny_rozsah), 2);
                        }
                    }
                }
            }
        }
    }

```

```

        vystupna_vrstva_x_y_d[x][y][d] = sqrt(vystupna_vrstva_x_y_d[x][y]
[d]);

        if (vystupna_vrstva_x_y_d[x][y][d] < vystupna_vrstva_x_y_d[x_vit]
[y_vit][d_vit])
        {
            x_vit = x;
            y_vit = y;
            d_vit = d;
        }
    }
}

/* Zmena vah */
for (d = 0; d < poc_deleni_kruznice; d++)
{
    for (x = 0; x < rozmer_vystupu_x; x++)
    {
        for (y = 0; y < rozmer_vystupu_y; y++)
        {
            dj = pow((double)(x - x_vit), 2) + pow((double)(y - y_vit), 2) +
pow((double)(minimum(abs(d - d_vit), poc_deleni_kruznice - abs(d - d_vit))), 2);
            for (v = 0; v < 3 * rozmer_obrazka_x * rozmer_obrazka_y; v++)
            {
                vaha_x_y_d_v[x][y][d][v] += gama * H_T * exp(-dj/ri_t) *
(((double)obrazok[v])/farebny_rozsah - vaha_x_y_d_v[x][y][d][v]);
            }
        }
    }
}

spolu_cykvov_ucenia++;
if (vizualizacia == 'A')
    vizualizuj_vahy();
}
printf("\nUcenie bolo ukoncene.");
printf("\n\nPre pokracovanie stlacte enter... ");
nacitaj_znak();
}

/* Testovanie siete na zaklade obrazka zadaneho uzivatelom */
void testuj_siet()
{
    char c;
    FILE *f;
    char nazov_suboru[DL_RET];
    char nazov_obrazku[DL_RET];
    int rozm_x, rozm_y, pom, i, sur_x, sur_y, sur_d, x, y, d, x_vit, y_vit, d_vit, v;
    double chyba;
    /* Nacitanie nazvu testovacieho obrazku */
    printf("\nZadajte nazov testovacieho obrazku: ");
    while (scanf("%s", nazov_obrazku) == 0)
    {
        clr_input();
        printf("\nChyba pri nacistani udajov!");
        printf("\nZadajte nazov experimentu: ");
    }
    clr_input();
    /* Doplnenie nazvu suboru obrazku o priecinok a priponu */
    sprintf(nazov_suboru, "test/%s.ppm", nazov_obrazku);

    /* Nacitanie suboru */
    /*DEBUG printf("\nNacitam obrazok zo suboru './%s'", nazov_suboru);*/
    /* Otvorime subor */
    if ((f = fopen(nazov_suboru, "r")) == NULL)
    {
        printf("\nChyba! Subor './%s' sa nepodarilo otvorit!", nazov_suboru);
        return;
    }
    /* Nacitanie suradnic testovacieho suboru od uzivatela */
    printf("Zadajte suradnice suboru: \n X: ");
    while (scanf("%d", &sur_x) == 0 || sur_x < 0 || sur_x >= rozmer_vystupu_x)
    {
        clr_input();
        printf("\nChyba pri nacistani udajov! Zadajte cele cislo v rozmedzi od 0 do
%d.", rozmer_vystupu_x - 1);
        printf("\n X: ");
    }
    clr_input();
    printf(" Y: ");
    while (scanf("%d", &sur_y) == 0 || sur_y < 0 || sur_y >= rozmer_vystupu_y)
    {
        clr_input();

```

```

        printf("\nChyba pri nacistani udajov! Zadajte cele cislo v rozmedzi od 0 do
%d.",rozmer_vystupu_y-1);
        printf("\n Y: ");
    }
    clr_input();
    printf(" Smer: ");
    while (scanf("%d",&sur_d)==0||sur_d<0||sur_d>=poc_deleni_kruznice)
    {
        clr_input();
        printf("\nChyba pri nacistani udajov! Zadajte cele cislo v rozmedzi od 0 do
%d.",poc_deleni_kruznice-1);
        printf("\n Smer: ");
    }
    clr_input();
    /* Citanie zo suboru */
    while (((c=fgetc(f))!='\n')&&(c!=EOF)); /* Preskocenie riadka */
    while (((c=fgetc(f))!='\n')&&(c!=EOF)); /* Preskocenie riadka */
    if (fscanf(f,"%d %d",&rozm_x,&rozm_y)!=2)
    {
        printf("\nChyba pri nacistani obrazku zo suboru './%s'! Rozmer obrazku nebolo mozne
nacistat.",nazov_suboru);
        printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
        nacitaj_znak();
        exit(EXIT_FAILURE);
    }
    if (fscanf(f,"%d",&pom)!=1)
    {
        printf("\nChyba pri nacistani obrazku zo suboru './%s'! Farebny rozsah nebolo mozne
nacistat.",nazov_suboru);
        printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
        nacitaj_znak();
        exit(EXIT_FAILURE);
    }
    if ((rozm_x != rozmer_obrazka_x)|| (rozm_y != rozmer_obrazka_y))
    {
        printf("\nChyba pri nacistani obrazku zo suboru './%s'!",nazov_suboru);
        printf("\nRozmer obrazku %dx%d nesedi s pozadovanim rozmerom %dx
%d.",rozm_x,rozm_y,rozmer_obrazka_x,rozmer_obrazka_y);
        printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
        nacitaj_znak();
        exit(EXIT_FAILURE);
    }
    if (pom != farebny_rozsah)
    {
        printf("\nChyba pri nacistani obrazku zo suboru './%s'!",nazov_suboru);
        printf("\nFarebny rozsah obrazku %d nesedi s pozadovanim rozsahom
%d.",pom,farebny_rozsah);
        printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
        nacitaj_znak();
        exit(EXIT_FAILURE);
    }
    for (i = 0; i < 3 * rozmer_obrazka_x * rozmer_obrazka_y; i++)
    {
        if (fscanf(f,"%d",&pom)!=1)
        {
            printf("\nChyba pri nacistani obrazku zo suboru './%s'!",nazov_suboru);
            printf("\nNacistanie %d. bodu farebneho rastra zlyhalo.",i);
            printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
            nacitaj_znak();
            exit(EXIT_FAILURE);
        }
        if (pom>farebny_rozsah)
        {
            printf("\nChyba pri nacistani obrazku zo suboru './%s'!",nazov_suboru);
            printf("\nHodnota (%d) %d. bodu farebneho rastra je vyssia ako maximum
%d.",pom,i,farebny_rozsah);
            printf("\nKriticka chyba. Stlacte enter pre ukoncenie programu...");
            nacitaj_znak();
            exit(EXIT_FAILURE);
        }
        obrazok[i]=pom;
    }
    if((fclose(f))==EOF)
        printf("\nChyba pri zatvarani suboru!\n");

    /* Zhlukovanie obrazku sietou */
    /* Nastavenie vystupov na 0 */
    for (x = 0; x < rozmer_vystupu_x; x++)
    {
        for (y = 0; y < rozmer_vystupu_y; y++)
        {
            for (d = 0; d < poc_deleni_kruznice; d++)
                vystupna_vrstva_x_y_d[x][y][d] = 0;
        }
    }

```

```

    }
    /* Hladanie vitazneho neuronu */
    printf("\nZhlukujem...\n");
    x_vit = 0;
    y_vit = 0;
    d_vit = 0;
    for (d = 0; d < poc_deleni_kruznice; d++)
    {
        for (x = 0; x < rozmer_vystupu_x; x++)
        {
            for (y = 0; y < rozmer_vystupu_y; y++)
            {
                for (v = 0; v < 3 * rozmer_obrazka_x * rozmer_obrazka_y; v++)
                {
                    vystupna_vrstva_x_y_d[x][y][d] += pow((vaha_x_y_d_v[x][y][d][v] -
                    ((double)obrazok[v])/farebny_rozsah), 2);
                }
                vystupna_vrstva_x_y_d[x][y][d] = sqrt(vystupna_vrstva_x_y_d[x][y][d]);

                if (vystupna_vrstva_x_y_d[x][y][d] < vystupna_vrstva_x_y_d[x_vit][y_vit]
[d_vit])
                {
                    x_vit = x;
                    y_vit = y;
                    d_vit = d;
                }
            }
        }
    }

    /* Informacny vypis */
    printf("\nSkutocne suradnice zadaneho obrazka:      X:%2d,   Y:%2d,   Smer:
%2d",sur_x,sur_y,sur_d);
    printf("\nVypocitane suradnice zadaneho obrazka:   X:%2d,   Y:%2d,   Smer:
%2d",x_vit,y_vit,d_vit);
    /* Vypocet chyby (vzdialenosti)*/
    chyba = sqrt(pow((double)(sur_x - x_vit), 2) + pow((double)(sur_y - y_vit), 2) +
pow((double)(minimum(abs(sur_d - d_vit),poc_deleni_kruznice - abs(sur_d - d_vit)))/4, 2));
    printf("\nChyba:%f",chyba);
}

/* Hlavne menu programu */
void hlavne_menu()
{
    /* Hlavne menu */
    char vstup = EOF;
    while (vstup != '0')
    {
        printf("\n\nHLAVNE MENU\n -1- Znovu inicializovat siet\n -2- Ucit siet\n -3-
Vizualizovat vahy\n -4- Testovat siet");
        printf("\n -0- Ukoncit experiment\n \nZadajte vasu volbu: ");
        vstup = nacitaj_znak();
        switch(vstup)
        {
            case '1':
                printf("\nINICIALIZACIA SIETE");
                printf("\nZadajte typ inicializacie neuronovej siete: \n -1- Nahodna\n -2-
Nahodne obrazky\n -3- Manualne urcena\nZadajte vasu volbu: ");
                while (scanf("%d",&inicial)==0||inicial<1||inicial>3)
                {
                    clr_input();
                    printf("\nChyba pri nacistani udajov!");
                    printf("\nZadajte typ inicializacie neuronovej siete: \n -1- Nahodna\n
-2- Nahodne obrazky\n -3- Manualne urcena\nZadajte vasu volbu: ");
                }
                clr_input();
                inicializuj_siet();
                printf("\n\nPre pokračovanie stlacte enter... ");
                nacitaj_znak();
                break;
            case '2':
                ucenie();
                break;
            case '3':
                printf("\nVIZUALIZACIA VAH");
                vizualizuj_vahy();
                printf("\n\nPre pokračovanie stlacte enter... ");
                nacitaj_znak();
                break;
            case '4':
                printf("\nTESTOVANIE SIETE");
                testuj_siet();
                printf("\n\nPre pokračovanie stlacte enter... ");
                nacitaj_znak();
                break;
        }
    }
}

```

```
        case '0':
            return;
        default:
            printf("\nNespravne zadana hodnota. Opakujte volbu.\n");
            break;
    }
    vstup = EOF;
}
```