

## PROJECT 1 OPENMP



ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΟΜΑΔΑΣ: ΝΤΩΝΕΣ ΣΑΒΒΑΣ, ΠΕΤΡΟΥΔΗΣ ΝΙΚΟΛΑΟΣ  
ΥΠΕΥΘΥΝΟΣ ΚΑΘΗΓΗΤΗΣ: ΗΡΑΚΛΗΣ ΣΠΗΛΙΩΤΗΣ  
ΜΑΘΗΜΑ: ΤΕΧΝΟΛΟΓΙΑ ΠΑΡΑΛΛΗΛΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ  
ΘΕΜΑ: ΔΙΑΔΙΚΟΠΟΙΗΣΗ ΕΙΚΟΝΑΣ ΜΕ ΤΗ ΜΕΘΟΔΟ OTSU  
ΕΡΓΑΣΤΗΡΙΟ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΥΨΗΛΩΝ  
ΕΠΙΔΟΣΕΩΝ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΙΣΤΩΝ, Δ.Π.Θ  
ΑΚΑΔΗΜΑΙΚΟ ΕΤΟΣ: 2022 – 2023



ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΡΑΚΗΣ | ΤΜΗΜΑ ΗΜ&ΜΥ

## ΕΙΣΑΓΩΓΗ

Μία δυαδική εικόνα δημιουργείται από την διάκριση μίας gray-scale εικόνας σε δύο επίπεδα, επανομαζόμενα συνήθως 0 και 1 [2]. Το thresholding είναι σημαντική τεχνική της όρασης υπολογιστών και της επεξεργασίας εικόνων, με την οποία διαχωρίζεται ένα αντικείμενο-στόχος από την εικόνα φόντου [1]. Με τη μέθοδο Otsu βρίσκεται αυτόματα ένα κατώφλι ώστε να επιτευχθεί η διαδικοποίηση της εικόνας [2]. Το βέλτιστο κατώφλι επιλέγεται από το κριτήριο ταξινόμησης, έτσι ώστε να γίνει ο διαχωρισμός των δύο κατηγοριών foreground και background (φόντο). Ειδικότερα, το κατώφλι καθορίζεται ελαχιστοποιώντας τη διασπορά «εντός κατηγορίας» ή μεγιστοποιώντας τη διασπορά «μεταξύ κατηγοριών», που ορίζεται ως μία weighted διαφορά των μέσων τιμών των εντάσεων των δύο κατηγοριών (foreground και φόντου) [1].

Αποτέλεσμα της μεθόδου είναι η εικόνα να χωρίζεται σε δύο τμήματα, το λευκό  $T_0$  και το σκοτεινό  $T_1$ . Η περιοχή  $T_0$  περιέχει επίπεδα έντασης από 0 έως  $t$ , ενώ η περιοχή  $T_1$  περιέχει επίπεδα έντασης από  $t$  έως  $l$ , όπου  $t$  το κατώφλι και  $l$  το μέγιστο επίπεδο έντασης (έστω 256). Το  $T_0$  και  $T_1$  μπορεί να αντιστοιχούν σε αντικείμενο και φόντο ή αντίστροφα (η φωτεινή περιοχή δεν αντιστοιχεί πάντα στο αντικείμενο) [4].



ΕΙΚΟΝΑ 1 – Mountain 1024x1024



ΕΙΚΟΝΑ 2 – Mountain 1024x1024 IMAGE WITH OTSU

## ΨΕΥΔΟΚΩΔΙΚΑΣ

ΒΗΜΑ 1: Υπολογισμός ιστογράμματος της gray-scale εικόνας.

ΒΗΜΑ 2: Υπολογισμός της διασποράς του foreground και background για ένα μοναδικό κατώφλι .

- i) Υπολογισμός των βαρών των foreground και background pixels.
- ii) Υπολογισμός της μέσης τιμής των foreground και background pixels .
- iii) Υπολογισμός της διασποράς των foreground και background pixels.

ΒΗΜΑ 3: Υπολογισμός της διασποράς «μεταξύ των κατηγοριών» και εύρεση του threshold όταν βρίσκεται η μέγιστη τιμή διασποράς.

ΒΗΜΑ 4: Καθορισμός της τιμής έντασης 0 για τα pixel με ένταση μικρότερη του threshold και 255 για τα pixel με ένταση μεγαλύτερη ή ίση του threshold.

```
void copy_in_2_out_img (length, width, inimg, outimg)
{
    unsigned long length, width;
    unsigned char inimg[length][width], outimg[length][width];

    int total=0;
    int top=256;
    int sumB=0;
    int wB=0;
    int maximum=0;
    int sum1=0;

    int hist[256];
    int i,j,temp;
    double start,end;
    start = omp_get_wtime();

    for (i=0;i<=255;i++)
        hist[i] = 0;

    for(i=0;i<length;i++)
    {
        for(j=0;j<width;j++)
        {
            temp = inimg[i][j];
            hist[temp] += 1;
        }
    }

    for(i=0;i<top;i++)
    {
        sum1=sum1+i*hist[i];
        total=total+hist[i];
    }
```

ΕΙΚΟΝΑ 3 - ΣΕΙΡΙΑΚΟΣ ΚΩΔΙΚΑΣ 1

```
int wF,mF;
int level,val;
for(i=1;i<=top;i++)
{
    wF=total-wB;
    if(wB>0 && wF>0)
    {
        mF=(sum1-sumB)/wF;
        val=wB*wF*((sumB/wB)-mF)*((sumB/wB)-mF);
        if(val>=maximum)
        {
            level=i;
            maximum=val;
        }
    }

    wB=wB+hist[i];
    sumB=sumB+(i-1)*hist[i];
}

for(i=0;i<length;i++)
{
    for(j=0;j<width;j++)
    {
        if(inimg[i][j]<level)
        {
            outimg[i][j]=0;
        }
        else
        {
            outimg[i][j]=255;
        }
    }
}
```

ΕΙΚΟΝΑ 4 - ΣΕΙΡΙΑΚΟΣ ΚΩΔΙΚΑΣ 2

Ο αλγόριθμος έχει χρονική πολυπλοκότητα  $O(N*M + 2*K)$  όπου N,M οι διαστάσεις της εικόνας και K η μέγιστη τιμή έντασης της gray εικόνας.



## ΠΑΡΑΛΛΗΛΟΠΟΙΗΣΗ ΤΟΥ ΚΩΔΙΚΑ

```
for(P = 2; P < 65; P *= 2)
{
    double start2,end2;
    int total=0;
    int top=256;
    int maximum = INT_MIN;

    int hist[256];
    int i,j,temp;
    start2 = omp_get_wtime();

    for (i=0;i<=255;i++)
        hist[i] = 0;

    #pragma omp parallel for num_threads(P) private(i,j,temp) collapse(2) \
    reduction(+:hist) schedule(static,length*width/P)
    for(i=0;i<length;i++)
    for(j=0;j<width;j++)
    {
        temp = inimg[i][j];
        hist[temp] += 1;
    }
}
```

ΕΙΚΟΝΑ 5 - ΠΑΡΑΛΛΗΛΟΣ ΚΩΔΙΚΑΣ 1

```
#pragma omp parallel num_threads(P) private(i,hist) reduction(+:wB,sumB,sum1,total)
{
    #pragma omp for schedule(static,top/P)
    for (int i = 0; i < top; i++) {
        sum1 += i * hist[i];
        total += hist[i];
    }

    #pragma omp for schedule(static,top/P)
    for (int i = 1; i <= top; i++) {
        wB += hist[i];
        sumB += (i-1)*hist[i];
        int wF = total - wB;
        if (wB > 0 && wF > 0) {
            int mF=(sum1-sumB)/wF;
            int val=wB*wF*((sumB/wB)-mF)*((sumB/wB)-mF);
            if (val > maximum) {
                maximum = val;
                level = i;
            }
        }
    }

    #pragma omp for collapse(2) schedule(static,length*width/P)
    for(i=0;i<length;i++)
    for(j=0;j<width;j++)
    {
        if(inimg[i][j]<level)
        {
            outimg[i][j]=0;
        }
        else
        {
            outimg[i][j]=255;
        }
    }
}
```

ΕΙΚΟΝΑ 6 - ΠΑΡΑΛΛΗΛΟΣ ΚΩΔΙΚΑΣ 2

## METPIKES

Θα παρουσιαστούν οι μετρικές για διάφορες εικόνες που έγιναν στο rcheon2 που περιέχει έως 64 πυρήνες και 128GB RAM.

### METPIKES EIKONAS 256X256

256x256	Serial	P=2	P=4	P = 8	P = 16	P = 32	P = 64
Time(sec)	0,00034	0,00032	0,00018	0,00022	0,00038	0,00071	0,03567
SpeedUp	-	1,0535	1,8794	1,5223	0,8915	0,4796	0,0096
Efficiency	-	0,5267	0,47	0,19	0,0557	0,015	0,0001

### METPIKES EIKONAS 512X512

512x512	Serial	P=2	P=4	P = 8	P = 16	P = 32	P = 64
Time(sec)	0,00153	0,00083	0,00054	0,00037	0,00049	0,00077	0,03587
SpeedUp	-	1,8378	2,8097	4,0788	3,1096	1,9852	0,0426
Efficiency	-	0,9189	0,7024	0,51	0,1943	0,062	0,0007

### METPIKES EIKONAS 1024X1024

1024x1024	Serial	P=2	P=4	P = 8	P = 16	P = 32	P = 64
Time(sec)	0,00515	0,00269	0,00207	0,00124	0,00159	0,00108	0,03883
SpeedUp	-	1,9142	2,4913	4,1518	3,2367	4,7882	0,1327
Efficiency	-	0,957	0,6228	0,519	0,2023	0,1496	0,0021

### METPIKES EIKONAS 2048X2048

2048x2048	Serial	P=2	P=4	P = 8	P = 16	P = 32	P = 64
Time(sec)	0,01889	0,01043	0,0063	0,0034	0,00205	0,00151	0,03591
SpeedUp	-	1,811	2,9997	5,5185	9,206	12,514	0,5263
Efficiency	-	0,9057	0,75	0,69	0,575	0,39	0,0082

### METPIKES EIKONAS 4096X4096

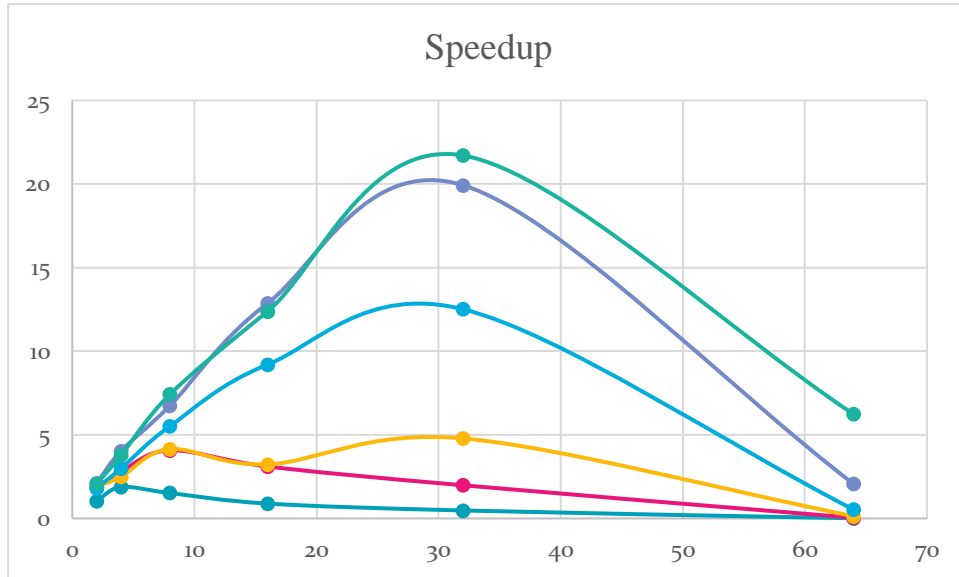
4096x4096	Serial	P=2	P=4	P = 8	P = 16	P = 32	P = 64
Time(sec)	0,08055	0,0378	0,0199	0,0119	0,00626	0,00404	0,272
SpeedUp	-	2,13	4,043	6,7485	12,872	19,93	2,058
Efficiency	-	1,0647	1,011	0,8436	0,8045	0,6228	0,0322

### METPIKES EIKONAS 8192X8192

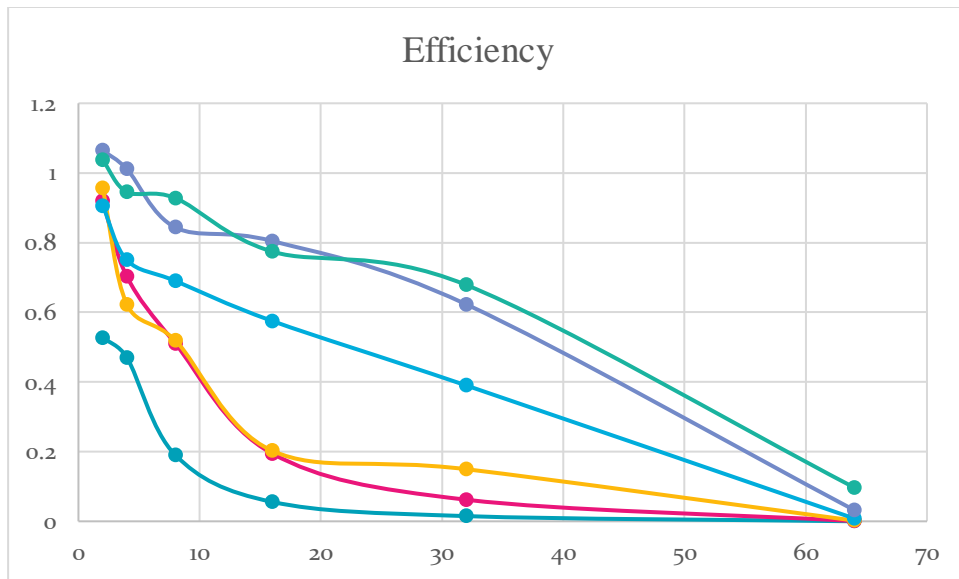
8192x8192	Serial	P=2	P=4	P = 8	P = 16	P = 32	P = 64
Time(sec)	0,2992	0,1443	0,07912	0,0403	0,02415	0,01378	0,048
SpeedUp	-	2,0738	3,7816	7,4235	12,388	21,7088	6,2326
Efficiency	-	1,0369	0,9454	0,9279	0,7742	0,6784	0,097

## ΓΡΑΦΗΜΑΤΑ ΔΙΑΣΠΟΡΑΣ ΜΕ ΟΜΑΛΕΣ ΓΡΑΜΜΕΣ

Παρακάτω θα παρουσιαστούν τα γραφήματα speedup ως προς το σύνολο των πυρήνων καθώς και το efficiency ως προς το σύνολο των πυρήνων, για κάθε εικόνα.



ΓΡΑΦΗΜΑ 1 - SPEEDUP - CORES



ΓΡΑΦΗΜΑ 2 - EFFICIENCY - CORES

## ΣΥΜΠΕΡΑΣΜΑΤΑ

Ο κώδικας που δημιουργήθηκε για να επιτελέσει τη διαδικοποίηση εικόνας με τη μέθοδο Otsu παρατηρείται ότι για εικόνες με ανάλυση 2048x2048 (δηλαδή με μία πολυπλοκότητα) και πάνω είναι παραλληλοποιήσιμος μέχρι τους 32 πυρήνες.

Όσον αφορά την επεκτασιμότητα, ο παράλληλος κώδικας που δημιουργήθηκε είναι επεκτάσιμος για εικόνες με ανάλυση μεγαλύτερη των 1024x1024 αφού το speedup αυξάνεται με την προσθήκη περισσότερων πυρήνων στην παραλληλοποίηση [3].

## ΒΙΒΛΙΟΓΡΑΦΕΙΑ

1) Nobuyuki Otsu (1979). "A threshold selection method from gray-level histograms". *IEEE Trans. Sys. Man. Cyber.* **9** (1): 62–66.

<https://ieeexplore.ieee.org/document/4310076>

2) Sunil L. Bangare, Amruta Dubal, Pallavi S. Bangare, Dr. S.T. Patil, Reviewing Otsu's Method For Image Thresholding, International Journal of Applied Engineering Research, <https://dx.doi.org/10.37622/IJAER/10.9.2015.21777-21783>

3) Εισαγωγή στον παράλληλο προγραμματισμό, Peter S. Pacheco

4) Jamileh Yousefi, Image Binarization using Otsu Thresholding Algorithm, <http://dx.doi.org/10.13140/RG.2.1.4758.9284>