

# All-In-Focus-Image

Hannah Bergenroth<sup>1</sup>, Evelyn Bankell<sup>2</sup>

## Abstract

This paper presents an efficient algorithm for multi-focus image fusion called the Laplacian pyramid transform. The algorithm uses Sum-Modified-Laplacian as focus measure, which is used to obtain a respective weight for each pixel that are later saved in a weight map. Thereafter, a Laplacian pyramid is found for each image in an image stack with different depth of fields and a Gaussian pyramid is obtained by the corresponding weight map for each image. Lastly, the images are combined by using image fusion and the fused image is then obtained by inverse Laplacian pyramid transform. Two methods are implemented in order to examine the result, the transform based method and a more simple pixel-based method. By studying and analyzing the results, it showed that the transform based method forms less noise and gives a better result both visually as quantitatively.

**Source code:** <https://github.com/hannahbergenroth/All-in-focus-image>

## Authors

<sup>1</sup>Media Technology Student at Linköping University, [hanbe563@student.liu.se](mailto:hanbe563@student.liu.se)

<sup>2</sup>Media Technology Student at Linköping University, [eveba996@student.liu.se](mailto:eveba996@student.liu.se)

## Keywords

Focus measure — Sum-modified-Laplacian — Laplacian-Gaussian pyramid — Image fusion

## Contents

<b>Introduktion</b>	<b>1</b>
0.1 Föreslagen metod	1
<b>1 Bakgrund</b>	<b>2</b>
1.1 Fokusmått	2
SML som fokusmått	
1.2 Laplacian Pyramid Transform	2
Gaussian Pyramid • Laplacian Pyramid	
1.3 Focus stacking	3
Pixelbaserad focus stacking • Transformbaserad focus stacking • Image fusion med Laplacian-Gaussian pyramid	
<b>2 Implementation</b>	<b>4</b>
2.1 Sum-modified-Laplacian	4
2.2 Gaussian Pyramid	4
2.3 Laplacian Pyramid	4
2.4 Image fusion	4
Pixelbaserad fusion • Laplacian-Gaussian fusion	
<b>3 Resultat</b>	<b>5</b>
<b>4 Diskussion och Slutsats</b>	<b>5</b>
<b>References</b>	<b>6</b>

## Introduktion

Under senare år har digitala bildbehandlingstekniker i stor utsträckning använts inom många områden. På grund av begränsat skärpedjup för optiska linser i CCD-enheter, är det ofta svårt att få en bild där alla relevanta objekt är i fokus. Därmed kommer den erhållna bilden inte vara i fokus överallt, det vill säga om ett objekt i scenen är i fokus, kommer ett annat vara ur fokus.

För att skapa en bild där alla objekt är i fokus, krävs en *fusion* för att se till att alla fokuserade objekt väljs. Det problem som *image fusion* i allmänhet försöker lösa, är att kombinera kompletterande information från flertalet bilder tagna från samma scen för att på så sätt få en ny, sammansatt bild som innehåller den bästa informationen från originalbilderna. Detta medför att den sammansatta bilden har bättre kvalitet än någon av originalbilderna.

### 0.1 Föreslagen metod

*Focus stacking* (även känd som *focal plane merging*, *z-stacking* och *focus blending*) är en digital bildbehandlingsteknik som kombinerar flertalet bilder tagna på

olika fokusavstånd för att ge en bild som resulterar i ett längre skärpedjup (DoF) än någon av de enskilda bilderna.

I denna rapport introduceras en metod för att skapa en helt fokuserad bild. Utgångspunkten är baserad på algoritmen Laplacian-Gaussian pyramider med *blending* som föreslås i [1], med modifieringen att vikterna nu ges direkt av fokusmättet. Först mäts den relativa graden av fokus i varje bildpunkt för varje bild med SML som därefter används för att beräkna respektive vikt för varje bildpunkt och som sparas ned i en så kallad *weight map*. Vi skapar en Laplacian pyramid av varje bild i bildserien och en Gaussian pyramid för dess motsvarande *weight map*. Bilderna sätts därefter samman med användning av *Laplacian-Gaussian multi-resolution fusion* för att slutligen få den helt fokuserade bilden.

## 1. Bakgrund

### 1.1 Fokusmått

För att kunna skapa en helt fokuserad bild är det nödvändigt att kunna mäta fokuskvaliteten i ett mindre bildområde och därmed ha ett mått för fokus. Ett fokusmått returnerar ett värde för varje bild som en indikator till vilken grad en bild, eller ett område av den, är i fokus. Eftersom en bild i fokus representeras av höga frekvenser, bör ett fokusmått ge högre värden för mer fokuserade områden och ett maximum för områden helt i fokus.

Det finns två typer av fokusmått, fokusmått i den spatiala domänen och transform-baserade fokusmått. I den spatiala domänen kan beräkningarna göras direkt på en inmatningsbild, medan ett transform-baserat fokusmått kräver en transformering som utförs med exempelvis en fouriertransform eller liknande. Gemensamt för ett bra fokusmått är att det ska ge ett extremvärde för det bäst fokuserade området och olika värden för olika nivåer av fokusering, måttet måste därmed vara mindre känsligt för brus för att kunna uppfylla detta. Dessutom, ska måttet vara oberoende av scenen som fångas, snabb och exakt samt ha god reproducerbarhet [2].

#### 1.1.1 SML som fokusmått

Det går att högpasfiltrera en bild genom att beräkna dess andraderivata. För tvådimensionella bilder används ofta *Laplacian*:

$$\Delta^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (1)$$

där  $I(x, y)$  är bildintensiteten i punkt  $(x, y)$ . Noterbart är i fallet då andraderivatorna i  $x$ - och  $y$ -led har motsatta tecken och tenderar att ta ut varandra. Detta kan ofta förekomma för bland annat texturerade bilder vilket gör att Laplacian beter sig instabilt. Detta problem löses genom att definiera den *modifierade Laplacian*:

$$\Delta_M^2 I = \left| \frac{\partial^2 I}{\partial x^2} \right| + \left| \frac{\partial^2 I}{\partial y^2} \right| \quad (2)$$

Den modifierade Laplacian är alltid större eller lika stor som Laplacian. Den diskreta approximationen av den modifierade Laplacian fås av:

$$ML(x, y) = |2I(x, y) - I(x - step, y) - I(x + step, y)| \\ + |2I(x, y) - I(x, y - step) - I(x, y + step)| \quad (3)$$

För att tillgodose möjliga variationer i storleken på strukturelement, beräknade författarna av [3] de partiella derivatorna genom att använda ett variabelt avstånd (steg) mellan pixlarna för att beräkna den modifierade Laplacian. Fokusmått för en punkt  $(i, j)$  beräknas slutligen som summan av den modifierade laplacian, i ett "litet" område kring punkten  $(i, j)$ :

$$F(i, j) = \sum_{x=i-N}^{i+N} \sum_{y=j-N}^{j+N} ML(x, y) \quad (4)$$

där parametern  $N$  bestämmer fönsterstorleken som används för att beräkna fokusmättet. Vanligtvis används ett litet fönster av storlek 3x3 eller 5x5, det vill säga  $N = 1$  eller  $N = 2$ . Fokusmättet refereras hädanefter som *sum-modified-Laplacian* (SML).

I [4] jämfördes flera fokusmått efter fokusmättets förmåga att urskilja bildområden i fokus från suddiga bildområden. Det experimentella resultat som presenteras i [4] visar att SML ger bättre resultat än andra fokusmått (Energy of image Laplacian (EIL), Tenenbaum gradient (TG), Energy of image gradient (EIG)) när exekveringstiden inte togs hänsyn till i utvärderingen.

### 1.2 Laplacian Pyramid Transform

En effektiv metod för att beskriva en bild med olika fokusområden är *the image pyramid* presenterad av Burt och Adelson, [1]. Avsikten med metoden är att bryta ned originalbilden till flera del-bilder med olika upplösning.

För att skapa en *Laplacian pyramid* behövs först en *Gaussian pyramid* framtas, där bilden lågpassfiltreras och halveras i dess bildstorlek. De olika nivåerna i pyramiden skapas genom upprepande filtrering och halvering utav bilden.

### 1.2.1 Gaussian Pyramid

För att skapa en Gaussian pyramid nyttjas originalbilden,  $G_0$ , vilken motsvarar den lägsta nivån i pyramiden. Nästa nivå i pyramiden skapas genom att lågpassfiltrera bilden med faltning och därefter halvera bildstorleken. Metodiken upprepas sedan återigen med hjälp av den föregående nivån i pyramiden och därigenom kan ett ämnat antal nivåer i pyramiden skapas. Genom att använda Gaussian Pyramider reduceras oönskade artefakter vilka vanligen uppstår vid nedsampling [1].

### 1.2.2 Laplacian Pyramid

Den översta nivån i Gaussian pyramiden återanvänds sedan som den översta nivån i Laplacian pyramiden. För en nivå lägre tas däremot differensen mellan den motsvarande lågpassfiltrerade bilden och bilden i den övre nivån, vilken expanderas till en fördubblad bildstorlek. Detta upprepas för varje nivå i Gaussian pyramiden till och med den lägsta nivån, originalbilden. Detta visas i Ekvation 5 nedan, där  $G$  motsvarar Gaussian,  $LP$  Laplacian,  $l$  anger nivån medan  $N$  anger antalet nivåer i pyramiden och  $*$  indikerar att bilden är expanderad. De tre stegen illustreras även genom flödesschemat i figur 1. Laplacian pyramider används framförallt för att återfå de höga frekvenserna som reducerades vid filtrering [1].

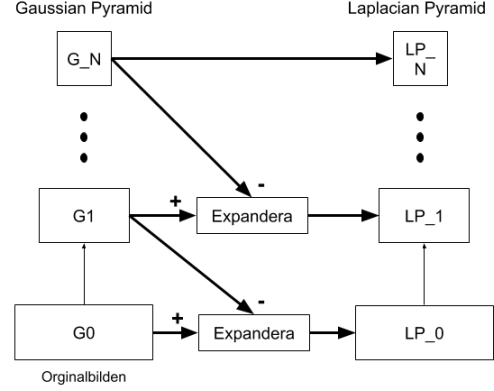
$$\begin{cases} LP_l = G_l - * G_{l+1}, & l \leq 0 < N \\ LP_N = G_N, & l = N \end{cases} \quad (5)$$

## 1.3 Focus stacking

Enligt fusionsmetoder kan multifokusbildsfusionen kategoriseras i två typer: transformdomän baserade och spatialdomän baserade metoder.

### 1.3.1 Pixelbaserad focus stacking

Den mest grundläggande metoden för *focus stacking* är den pixelbaserade. Varje pixelvärde på motsvarande position i alla bilder jämförs för att bestämma det bästa fokusvärdet för varje pixel i den erhållna slutliga bilden. För en bildserie bestäms det bästa representativa pixelvärdet med användning av en objektiv funktion och urvalskriterier. Fokusbått används ofta som objektiv



**Figure 1.** Konstruktion av en Laplacian pyramid visat i ett flödesschema

funktion. Fokusbåttet beräknas för alla bilder i bildserien där de maximala urvalskriterierna används för att bestämma den bästa representativa pixeln för varje position.

### 1.3.2 Transformbaserad focus stacking

I den transformbaserade *focus stacking* metoden transformeras varje inmatningsbild från den spatial domänen till en annan domän. Bildkvaliteten och detaljerna jämförs sedan i den domänen med hjälp av objektiva funktioner och jämförelsemetoder. Efter att ha kombinerats utifrån givna riktlinjer transformeras bilden åter till den spatiala domänen med en invers transformation för att erhålla den förväntade fusionsbilden. Det finns många olika transformbaserade algoritmer som används, så som *Laplacian pyramid* (LP) och *Discrete Wavelet Transform* (DWT).

### 1.3.3 Image fusion med Laplacian-Gaussian pyramid

För fusion används Laplacian-Gaussian fusion föreslagen av Burt och Adelson [1]. Varje inmatningsbild tilldelas en vikt givet viktfunktionen  $W_i$ :

$$W_i(x,y) = F(x,y)^{W_c} \quad (6)$$

där  $W_c$  är vald viktkoefficient och  $F(x,y)$  det resulterande fokusbåttet i varje bildpunkt. Utifrån viktfunktionen i Ekvation 6, tilldelas en vikt för varje bild  $W_{N,i}$  baserat på summan av alla bilders vikter enligt:

$$W_{N,i}(x,y) = \frac{W_i(x,y)}{\sum_{k=1}^N W_k(x,y)} \quad (7)$$

där  $N$  representerar antalet bilder i bildserien.

Då  $L\{I\}^l$  och  $G\{W_N\}^l$  betecknar nivå  $l$  i Laplacian-respektive Gaussian pyramiden, ges den slutliga, *fused*, Laplacian pyramid av:

$$L\{I_F(x,y)\}^l = \sum_{i=1}^N G\{W_{N,i}(x,y)\}^l L\{I_i(x,y)\}^l \quad (8)$$

## 2. Implementation

Den föreslagna algoritmen har implementerats i MATLAB och testats på en bildserie bestående av 26 bilder.

### 2.1 Sum-modified-Laplacian

Fokusrått implementerades genom att skapa två separata funktioner, en användes för att beräkna den modifierade Laplacian och den andra för att beräkna summan av den modifierade Laplacian i ett område kring punkten. Dessa implementerades i ML respektive SML. För att beräkna den modifierade Laplacian behövdes varje inmatningsbild först utökas med en rad och kolumn på vardera sida för att undkomma fallet då beräkningar utfördes på de allra yttersta pixelpunkterna, detta utfördes med MATLAB-funktionen *padarray* med värdet *replicate*. Därefter beräknades den modifierade Laplacian enligt Ekvation 3.

Utifrån den modifierade Laplacian beräknades SML för varje bildpunkt i ett 3x3 stort fönster, parametern  $N$  i Ekvation 4 valdes därmed till 1. SML beräknades i enlighet med ekvationen och den tidigare utökade bilden beskars till utgångsstorlek innan fokusrått slutligen returnerades.

### 2.2 Gaussian Pyramid

Implementationen av Gaussian pyramiderna gjordes i *gaussian\_pyramid*. Under framställningen av pyramiderna användes nio nivåer, detta eftersom originalbilden bestod av 512x512 pixlar och att den översta nivån därmed utgjorde 2x2 pixlar. Bilderna filtrerades med användning av MATLAB-funktionen *imfilter*, där filtret  $\omega$  användes:

$$\omega = \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \quad (9)$$

Därefter halverades bildstorleken genom den inbyggda MATLAB-funktionen *imresize*. Detta upprepades för varje nivå, där den föregående nivån används för att skapa den nästkommande nivån.

### 2.3 Laplacian Pyramid

Samma antal nivåer som för Gaussian pyramiden, användes för att skapa Laplacian pyramiden där principen från Ekvation 5 användes. Implementationen av pyramiderna skedde i *laplacian\_pyramid*.

Bilderna uppsamlades i *UpSize* med inparametrar  $g\_pyr(n)$  och *odd*. Argumenten  $g\_pyr(n)$  motsvarar bilden på nivå  $n$  i Gaussian pyramiden och *odd* beskriver en 2-vektor som indikerar om bilden bör ha en udda storlek för respektive dimension och beräknas enligt följande:

$$2 * size(g\_pyr()) - size(g\_pyr(n - 1))$$

Bilden utökades först med MATLAB-funktionen *padarray* med 1x1 genom värdet *replicate* och sedan skapades en noll-matris med det dubbla antalet rader och kolumner från inmatningsbilden. Därefter fylldes originalbilden i de udda positionerna av den uppsamlade bilden. Bilderna filtrerades sedan med samma filter från Ekvation 9 och slutligen beskars bilden till utgångsstorlek genom att ta bort den tillägda kanten.

### 2.4 Image fusion

För den pixel- och transformbaserade *image fusion* skapades en så kallad *weight map*. Denna implementerades i *weight\_map*. Viktkoefficienten från Ekvation 6 valdes till 2 i båda metoder efter experimenterande med tillfredsställande resultat. För att undvika division med 0, adderades ett litet tal om  $1e - 12$  till varje vikt innan vikterna normaliserades för varje pixel så att vikterna tillsammans adderades upp till 1 genom Ekvation 7.

#### 2.4.1 Pixelbaserad fusion

Utifrån den nyligen skapade *weight map*, användes MATLAB-funktionen *max(weightmap, [], 3)* för att ta fram indexet och fokusvärdet på den bild i bildserien som ger den högsta vikten (vilken är baserad på fokusvärdet) i varje bildpunkt. Den resulterande, helt fokuserade, bilden utgår därmed uteslutande från denna funktion där den bäst fokuserade pixeln i bildserien och dess fokusvärde används i varje position i den resulterande bilden.

#### 2.4.2 Laplacian-Gaussian fusion

Den transformbaserade *image fusion* med Laplacian-Gaussian pyramider implementerades i *fusion* och utfördes i enlighet med Ekvation 8 efter att en Gaussian pyramid skapats för varje *weighting map* och en Laplacian pyramid tagits fram för varje bild i bildserien. Den resulterande, *fused*, Laplacian pyramiden ingick

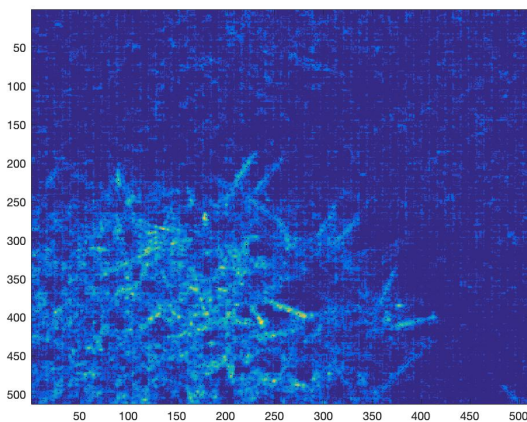


därefter en invers transformation genom funktionen som implementerades i `reconstruct_laplacian_pyramid`. I funktionen togs den slutliga, helt fokuserade, bilden fram genom att addera ihop de olika nivåerna i pyramiden efter att de lägre nivåerna först uppsamlats i `UpSize` där de lågpasfilterades med samma filter från Ekvation 9 som också användes vid nedsamlingen och skapandet av pyramiderna.

### 3. Resultat

I *figur 2 och 3* visas två olika visualiseringsmetoder för att belysa vart i en bild som den relativa fokusnivån är hög. I *figur 2* används MATLAB-funktionen `imagesc()`, där varje bildpunkt visar en färg som representerar den relativa fokusnivån. I de bildpunkter där färgen går mot ljusare och grönare är den relativa fokusnivån hög, däremot representeras en låg fokusnivå i en mörkare blå färg. I *figur 3* illustreras fokusnivån genom funktionen `mesh()`, en tredimensionell graf, där x- och y-led motsvarar den två dimensionella bilden och z-led den relativa fokusnivån.

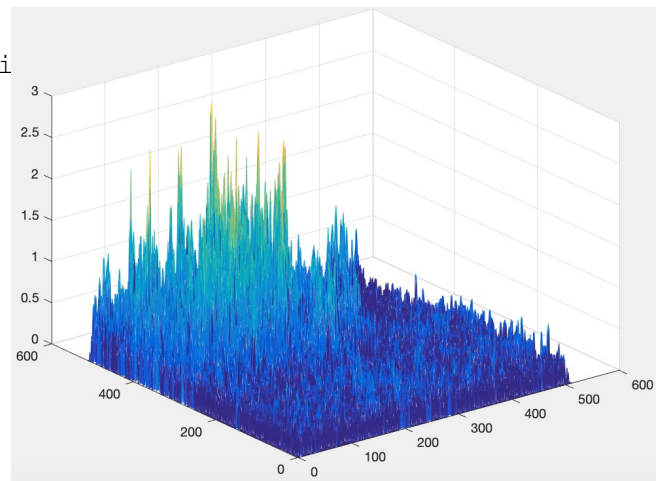
Resultatet av de två fusionsmetoderna visas i *figur 4 och 5*. *Figur 4* visar resultatet av den pixelbaserade metoden medan *figur 5* visar resultatet från den transformbaserade metoden där Laplacian och Gaussian pyramider använts.



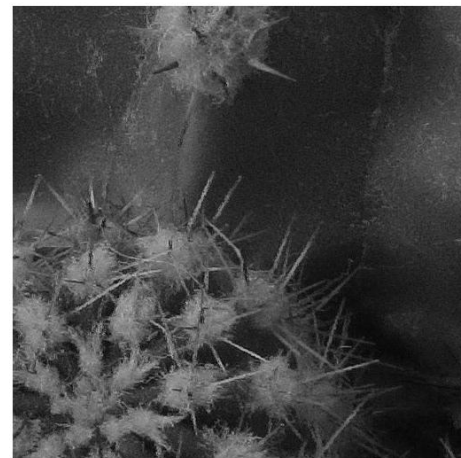
**Figure 2.** Visualisering av den relativa fokusnivån i en bild med `imagesc()`

### 4. Diskussion och Slutsats

Den relativa fokusnivån för en bild går tydligt att urskilja genom de två olika visualiseringsmetoderna från *figur 2 och 3*. Vart i bilden de bildpunkter med högst fokus



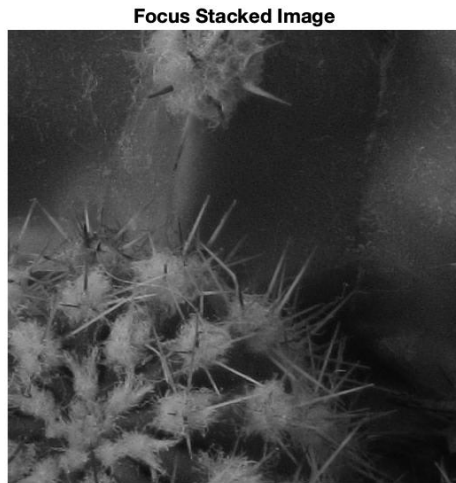
**Figure 3.** Visualisering av den relativa fokusnivån i en bild med `mesh()`



**Figure 4.** Resultatet av den pixelbaserade metoden

förekommer går tydligast att lokalisera genom metoden från *figur 2*. Att däremot enkelt lokalisera ett område med en högre fokusnivå används metoden från *figur 3* och dessutom uppskattas mätningen av den relativa fokusnivån lättare. Visualiseringen av den relativa fokusnivån som gjorts med dessa metoder har även underlättat vid implementationen av de två olika metoderna för *fusion*. Man har därmed kunnat försäkra sig om att det använda fokusmättet verkligen lokaliserat de områden i respektive bild som är i fokus. Detta medför att man kan garantera att de kommande beräkningarna för *image fusion* bygger på korrekt data.

Genom att jämföra *figur 4 och 5*, betraktas den pixelbaserade metoden innehålla betydligt mer brus, medan



**Figure 5.** Resultatet av den transformbaserade metoden med Laplacian pyramider

den transformbaserade metoden återskapar ett mer homogent resultat. Att den pixelbaserade metoden är känsligare för brus beror på att metoden jämför respektive pixel mellan bilderna i bildserien och därmed inte tar hänsyn för dess närliggande pixlar. Det beror även på att metoden inte använder sig utav låpassfiltrering vilket förekommer i den transformbaserade metoden. Det går därmed att fastställa att den transformbaserade metoden med Laplacian pyramider ger ett bättre resultat. Emellertid är det viktigt att ha i åtanke att exekveringstid är betydligt längre för den transformbaserade metoden och att det därmed är en avvägning vilken metod som är lämpligast att använda beroende på applikationen och användningstillfälle.

Sammanfattningsvis ger metoden, vilken presenteras i denna rapport, ett bra resultat både visuellt och genom mätbarhet. Detta bevisades även genom att jämföra med en enklare metod, den pixelbaserade.

## References

- [1] P. Burt and T. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communication*, COM-31:532–540, 1983.
- [2] Y. Tian. Monte carlo evaluations of ten focus measures. *Digital Photography III*, Proc. SPIE 6502, 2007.
- [3] S. K. Nayar and Y. Nakagawa. Shape from focus. *IEEE Trans. Pattern Anal. Mach. Intell.*, Lett. 16(8):824–831, 1994.

- [4] W. Huang and Z. Jing. Evaluation of focus measures in multi-focus image fusion. *Pattern Recogn.*, Lett. 28(4):493–500, 2007.