

## 영상 분할을 통한 Richardson-Lucy 디컨벌루션 개선 알고리즘

\*김정환 \*\*박대준 \*\*\*정제창

\*한양대학교 공과대학 융합전자공학부

\*konstantine0927@gmail.com \*\*daejoon12@gmail.com \*\*\*jjeong@hanyang.ac.kr

## An Image Segmentation Method for Richardson-Lucy Deconvolution Algorithm Improvement

\*Jeonghwan Kim \*\*Daejun Park \*\*\*Jechang Jeong

Department of Electronic Engineering, College of Engineering, Hanyang University

## 요약

본 논문에서는 Non-blind 디컨벌루션 알고리즘 중 하나인 Richardson-Lucy(RL) 디컨벌루션을 영상 분할을 통해 성능을 향상시킨 알고리즘을 제안한다. RL 디컨벌루션은 영상의 크기가 커질수록 연산 양이 크게 증가한다. 따라서 크기가 큰 영상의 RL 디컨벌루션은 계산에 많은 시간을 필요로 한다. 이를 개선하기 위하여 영상을 적절한 크기로 분할하여 각각 RL 디컨벌루션을 계산한다. 또한 분할 시 생기는 왜곡을 줄이기 위해 리플 제거를 위한 알고리즘을 추가한다. 이를 통해 기존의 알고리즘보다 연산 양을 줄여 빠른 RL 디컨벌루션이 가능하도록 개선한다.

## 1. 서론

영상 블러란 영상을 촬영할 시 초점을 잘못 맞추거나 손의 떨림, 피사체의 움직임 등으로 인하여 영상이 흐려지는 현상을 말한다. 그리고 영상 블러를 제거하는 과정을 디블러링이라고 한다. 영상 블러는 다음 수식과 같이 모델링할 수 있다.<sup>[1]</sup>

$$g = f * k + n \quad (1)$$

$g$ 는 열화된 영상,  $f$ 는 원본 영상,  $k$ 는 블러 커널,  $n$ 은 노이즈를 나타내며  $*$ 는 컨벌루션 연산을 나타낸다. 열화된 영상에서 원본 영상을 추정하기 위해서는 컨벌루션의 역과정이 필요하다. 이 과정을 디컨벌루션이라 한다..

디컨벌루션을 하기 위해서는 열화된 영상과 함께 블러 커널과 노이즈를 알아야한다. 블러 커널을 미리 알고 영상을 복원하는 알고리즘을 non-blind 디컨벌루션이라 한다. 그 반대로 블러 커널을 추정해나가며 영상을 복원하는 알고리즘은 blind 디컨벌루션이라고 한다. 본 논문에서 다루는 Richardson-Lucy(RL) 디컨벌루션은 non-blind 디컨벌루션의 대표적인 알고리즘으로 현재의 blind 디컨벌루션 알고리즘의 기본으로 응용된다.<sup>[3]</sup>

RL 디컨벌루션은 연산 양이 많아 연산 시 많은 시간이 소요된다. 최근 촬영되는 영상은 100만 화소 이상의 높은 화질의 사진이 많고 이러한 사진들을 RL 디컨벌루션을 이용하여 디블러링 하는데 수 분이 소요된다. 따라서 RL 디컨벌루션을 이용하면 즉각적으로 결과를 도출할 수 없다.

그렇기에 본 논문에서 영상 분할을 적용하여 속도 개선을 하는 알고리즘을 제안한다.

또한 RL 디컨벌루션은 푸리에 변환 과정을 포함하기 때문에 영상 경계와 에지로부터 리플이 발생한다.<sup>[2]</sup> 본 논문에서는 이러한 리플을 효과적으로 제거하기 위하여 Chan이 제안한 뉴먼(Neumann) 경계 조건에서의 아포다이제이션(Apodization)을 이용하였다.<sup>[4],[5]</sup>

본 논문의 구성은 다음과 같다. 2장에서는 RL 디컨벌루션 알고리즘에 대한 설명 및 한계점에 대해 설명한다. 3장에서는 이를 개선하는 알고리즘을 자세히 설명한다. 4장에서는 알고리즘에 대한 실험 결과를 제시한다. 마지막으로 5장에서 결론을 맺는다.

## 2. 기존 알고리즘

## 2.1 Richardson-Lucy 디컨벌루션

Richardson과 Lucy는 베이저언 기본 구조(Bayesian framework)를 기반으로 반복 기댓값 최대화 디컨벌루션 알고리즘을 개선시킨 RL 디컨벌루션을 제시하였다.<sup>[6],[7]</sup> 이 알고리즘은 이전의 Weiner 디컨벌루션보다 더 높은 성능을 발휘하였다.<sup>[8]</sup> RL 디컨벌루션은 다음과 같은 수식으로 나타낼 수 있다.<sup>[9]</sup>

$$\hat{f}_{p+1}(x) = \hat{f}_p(x) \left[ k(-x) * \frac{g(x)}{k * \hat{f}_p}(x) \right] \quad (2)$$

$\hat{f}$ 는 RL 디컨벌루션을 통해 구한 추정 영상이고,  $p$ 는 반복 횟수를 의미한다. 초기  $f$ 에는 열화된 영상을 사용하며, 반복 횟수가 증가할수록  $\hat{f}_{p+1}$ 은  $\hat{f}_p$ 로 수렴하게 되며 원본에 가까운 영상을 얻을 수 있다.

### 2.2 RL 디컨벌루션 알고리즘의 한계

RL deconvolution은 컨벌루션 연산을 위해 FFT를 사용하기 때문에 영상의 경계에 의한 두루마리 효과(Wrap-around effect)가 나타난다.<sup>[2]</sup> 이는 2차 푸리에 변환 시에 영상의 경계면은 불연속한 값으로 인해 나이퀴스트(Nyquist)주파수 또는 이보다 큰 신호가 되풀이되어 공간적 왜곡(리플)이 발생하는 현상이다. 따라서 RL 디컨벌루션에서는 리플을 제거하는 것이 필수적이다.

리플을 제거하는 방법은 간단하고 잘 알려져 있다. 길이가  $N$ 인 영상에서 영상 경계 외부에 0을 최소  $N$ 개 패딩을 하면 FFT 계산 뒤 역 FFT를 수행하였을 때 생기는 리플이 발생하지 않는다. 그러나 이는 영상을 크게 만들어 연산 양을 크게 늘리는 부작용이 뒤따르기 때문에 리플이 보이지 않는 적절한 개수의 0을 패딩해야 한다. 이 과정을 통해 얻은 결과가 그림 1 (b)와 같다. 그러나 그림 1 (c)와 같이 복원 영상의 테두리 부분을 확대하여 확인하면 약간의 리플이 남아있음을 확인할 수 있다.



그림 1. (a) 리플 문제가 발생한 결과 영상  
(b) 가중치를 통해 리플을 해결한 결과 영상  
(c) (b)의 경계면을 확대한 영상

RL 디컨벌루션은 연산 양이 많다는 단점을 가지고 있다. 이 알고리즘은 한 번 반복에 4개의 FFT연산이 필요하다. 즉,  $N \times N$  영상에서는 한 번 시행에  $4N^2 \log_2 N$ 의 연산이 필요하다. 여기에 위에서 언급한 패딩을 적용하게 되면 연산의 수는 더 증가하게 된다. 본 논문은 이 문제점에 주목하여 영상 분할 알고리즘을 적용한 개선된 알고리즘을 제안한다.

## 3. 제안하는 알고리즘

### 3.1 영상 분할 알고리즘

영상의 크기에 따라 RL 디컨벌루션의 연산 양이 증가하는 것을 개선하기 위해 기존 알고리즘에 영상 분할 알고리즘을 추가하였다. 영상을 가로 세로  $D$ 등분하여 RL 디컨벌루션을  $p$ 번 반복 시행하면 이론적으로 연산의 양은  $4pN^2 \log_2(N/D)$ 개로 줄어들게 된다. 따라서 기존 알고리즘보다  $4pN^2 \log_2 D$ 개가 줄어든다. 그러나 리플을 제거하기 위해 패딩이 추가되기 때문에 연산 양은 이론 값보다 증가한다. 과도한 분할은 패딩의 증가로 연결되어 연산 양을 증가시킬 수 있으므로 적절한 분할 수가 필요하다. 본 논문에서 실험을 통해 분할 수를 바꿔가며 연산 시간을 비교하였고,  $N/2^7$ 이 연산 양을 줄이는 최적의  $D$ 값임을 도출하였다.

### 3.2 리플 제거 알고리즘

영상 분할 알고리즘에서 발생하는 치명적인 문제점은 2.2장에서 언급한 리플로 인하여 분할 흔적이 남는 점이다. 기존 알고리즘에서는 영상에서 리플을 완전히 제거하지 않았으나 이대로 영상 분할 알고리즘을 적용할 시 분할 선 부근의 리플이 생겨 복원 영상에 왜곡이 발생한다. 따라서 영상 분할 알고리즘을 적용하기 위해 기존보다 더 강력한 리플 제거 알고리즘을 적용하여 분할선이 보이지 않도록 개선하였다.

본 논문에서 제안하는 알고리즘은 Chan이 제안한 뉴먼(Neumann) 경계 조건에서의 아포다이제이션 알고리즘<sup>[4]</sup>을 응용하여 패딩하는 값을 0이 아닌 영상에서의 반사되는 값으로 패딩하였다. 이를 통해 기존의 알고리즘보다 더 효과적으로 리플을 제거하였다. 패딩 수가 증가할수록 리플이 효과적으로 제거되는 장점이 존재하지만 연산 양이 늘어난다는 단점이 있기 때문에 적절한 트레이드오프가 필요하다.

### 3.2 제안한 알고리즘의 도식

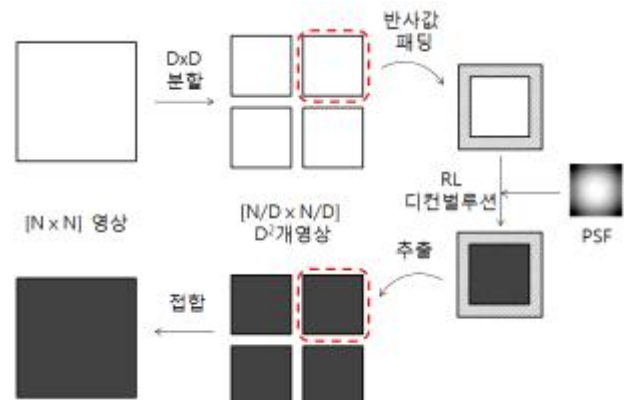


그림 2. 제안하는 알고리즘의 도식

그림 2는 본 논문에서 제안하는 알고리즘을 도식화한 것이다.  $N \times N$  크기의 영상을 입력 받으면  $D^2$ 개의  $2^7 \times 2^7$  크기의 영상으로 분할한다. 분할한 영상은 리플 제거를 위해 반사 값을 이용한 패딩을 거쳐 RL 디컨벌루션을 계산한 뒤 결과 영상들을 다시 접합시켜 복원 영상을 얻는다.

#### 4. 실험 결과

실험은 그림 3 (a)의 여러 크기의 마을 사진에 가우시안 블러를 추가하여 RL 디컨벌루션을 100회 반복하는 방법으로 진행하였다. 그림 3은 마을 사진을 RL 디컨벌루션과 제안하는 알고리즘을 적용한 결과를 나타낸다. 그림 3 (a)는 원본 영상, (b)는  $7 \times 7$  크기의 가우시안 블러 커널을 적용한 영상, (c)는 기존 RL 디컨벌루션을 적용한 결과, (d)는 본 논문이 제안하는 알고리즘을 적용한 결과 영상이다.

표 1은 사진 크기별로 분할 수를 조정해가며 연산 시간을 측정하고 기존 알고리즘에 비해 증가 혹은 감소한 속도의 비율을 나타낸다. 굵게 표시된 값은 본 논문이 제안하는 가장 최적의 조건일 때의 결과다. 표 2는 객관적인 화질 비교를 위해 기존 알고리즘에 의한 결과와 본 논문에서 제안한 최적의 알고리즘을 이용한 결과의 원본대비 PSNR(Peak signal-to-noise ratio)을 계산한 값이다. 이 결과를 통하여 제안한 알고리즘을 적용하여도 결과에 왜곡이 발생하지 않음을 확인할 수 있다. 표 3은 다양한 크기의 서로 다른 영상들에 대한 기존 알고리즘과 제안한 알고리즘의 값을 비교한 결과를 나타낸다. 그림 4는 표3의 실험 중 "City" 영상에 대한 주관적 화질 비교를 위한 결과 영상이다.

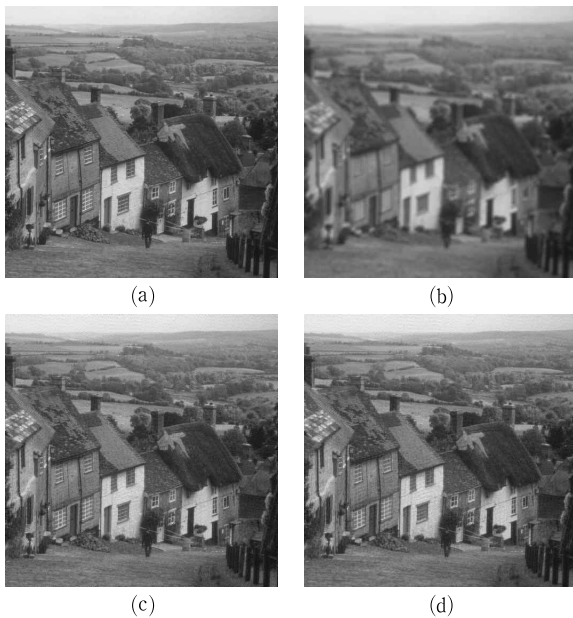


그림 3. (a) 원본 영상 (b) 블러 영상 (c) 기존 알고리즘으로 디블러링한 결과 영상 (d) 제안된 알고리즘으로 디블러링한 결과 영상

D	128×128	256×256	512×512	1024×1024	2048×2048
기준	<b>0.3936</b>	1.9412	9.4021	37.4633	151.2904
2	0.9585 (-58.94%)	<b>1.7282</b> (12.32%)	12.3407 (-23.81%)	50.2131 (-25.39%)	148.8282 (1.65%)
4	0.9075 (-56.63%)	3.6584 (-46.94%)	<b>6.8975</b> (36.31%)	51.7155 (-27.56%)	212.4913 (-28.80%)
8	2.2106 (-82.19%)	3.5805 (-45.78%)	15.3377 (-38.70%)	<b>28.8020</b> (30.07%)	212.3555 (-28.76%)
16	6.3066 (-93.76%)	8.5239 (-77.23%)	14.4768 (-35.05%)	58.7626 (-36.25%)	<b>117.1908</b> (29.10%)

[표 1] 영상의 크기와 분할 수에 따른 연산 시간 (단위:초)

영상 크기	기준	제안	차이
256×256	32.2345	32.5426	0.3081
512×512	32.8419	34.7890	1.9471
1024×1024	34.3554	34.3472	-0.0082
2048×2048	33.7477	34.0385	0.2908

[표 2] 표 1 실험의 PSNR

영상	구분	연산 시간(초)	PSNR
House 256×256	RL	2.0719	33.7977
	제안	1.7382	34.1703
Cameraman 256×256	RL	1.9702	31.3591
	제안	1.7176	31.3098
Lena 512×512	RL	9.3432	33.4263
	제안	6.9431	33.4781
City 512×512	RL	9.4682	32.6578
	제안	6.8844	32.5577
Car 1024×512	RL	19.0025	33.8008
	제안	14.3061	33.6386
Pebbles 2048×2048	RL	153.8032	35.7710
	제안	117.2223	35.5791

[표 3] 다양한 영상에서의 결과 비교

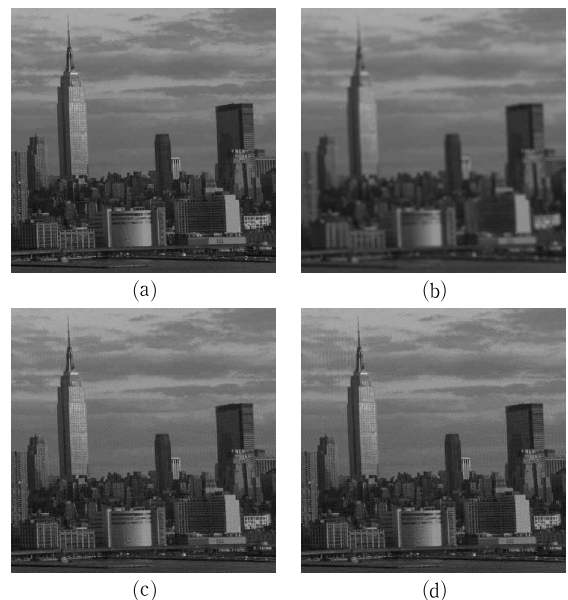


그림 4. (a) 원본 영상 (b) 블러 영상 (c) 기존 알고리즘 결과 영상 (d) 제안된 알고리즘 결과 영상

## 5. 결론

본 논문에서는 RL 디컨벌루션의 연산 양을 줄이기 위하여 영상 분할 알고리즘과 리플 제거 알고리즘을 추가하였다. 그 결과 영상의 손상 없이 연산 속도를 개선할 수 있었다. 특히 최근에는 해상도가 높은 영상이 많고, 영상의 크기가 클수록 제안한 알고리즘을 적용 시 연산 양의 감소가 크기 때문에, 연산 시간 단축에 효과적이다.

RL 디컨벌루션은 다른 여러 디컨벌루션 알고리즘에 포함되는 방법이기 때문에 본 논문에서 제안된 알고리즘이 해당 알고리즘에 적용될 가능성이 있다. 또한 제안된 알고리즘에 더 효율적인 리플 제거 방법을 적용한다면 더 좋은 결과를 얻을 수 있을 것이라 전망된다.

## 감사의 글

“이 논문은 2015년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임“(NRF-2015R1A2A2A01006004)

## 참 고 문 헌

- [1] R. Gonzalez, R. Woods “Digital Image Processing, 2<sup>nd</sup> ed.”, Prentice Hall, 2002.
- [2] P. A. Jansson, “Deconvolution of Images and Spectra, 2<sup>nd</sup> ed.”, Academic Press, 1997.
- [3] D. A. Fish, A. M. Brinicombe, E. R. Pike, J. G. Walker, “Blind deconvolution by means of the Richardson - Lucy algorithm”, *Journal of the Optical Society of America A*, vol. 12, no. 1, pp. 58-65, 1995
- [4] M. Ng, R. Chan, W. C. Tang, “A fast algorithms for deblurring models with Neumann boundary conditions”, *SIAM Journal on Scientific Computing*, v. 21, no. 3, pp.851-866, 1999
- [5] M. Bertero, P. Boccacci, “A simple method for the reduction of boundary effects in the Richardson-Lucy approach to image deconvolution”, *Astronomy and Astrophysics*, vol. 437, no. 1, pp.369-374, 2005.
- [6] W. H. Richardson, “Bayesian-Based Iterative Method of Image Restoration”, *Journal of the Optical Society of America*, vol. 62, no. 1, 1972.
- [7] L. B. Lucy, “An iterative technique for the rectification of observed distributions”, *The Astronomical Journal*, vol. 79, no. 6, 1974.
- [8] M. Thakur, S. Datar, “Image restoration based on deconvolution by Richardson Lucy Algorithm”,

*International Journal of Engineering Trends and Technology(IJETT)*, vol. 14, no. 4, pp. 161-165, 2014.

[9] M. Temerinac-Ott, “Tile-based Lucy-Richardson deconvolution Modelling a Spatially-Varying PSF for Fast Multiview Fusion of Microscopical Images”,(Technical report 260, University of Freiburg, 2010). <http://lmb.informatik.uni-freiburg.de/Publications/2010/Tem10a>