

최적화된 쿼드트리 이용한 2차원 연기 데이터의 효율적인 슈퍼 해상도 기법

최유연⁰, 김동희*, 김종현*
⁰강남대학교 소프트웨어응용학부,
^{*}강남대학교 소프트웨어응용학부
 e-mail: jonghyunkim@kangnam.ac.kr

Efficient Super-Resolution of 2D Smoke Data with Optimized Quadtree

YooYeon Choe⁰, Donghui Kim*, Jong-Hyun Kim*
⁰Dept. of Software Application, Kangnam University,
^{*}Dept. of Software Application, Kangnam University

● 요약 ●

본 논문에서는 SR(Super-Resolution)을 계산하는데 필요한 데이터를 효율적으로 분류하고 분할하여 빠르게 SR 연산을 가능하게 하는 쿼드트리 기반 최적화 기법을 제안한다. 제안하는 방법은 입력 데이터로 사용하는 연기 데이터를 다운스케일링(Downscaling)하여 쿼드트리 연산 소요 시간을 감소시키며, 이때 연기의 밀도를 이진화함으로써, 다운스케일링 과정에서 밀도가 손실되는 문제를 피한다. 학습에 사용된 데이터는 COCO 2017 Dataset이며, 인공지능망은 VGG19 기반 네트워크를 사용한다. 컨볼루션 계층을 거칠 때 데이터의 손실을 막기 위해 잔차(Residual)방식과 유사하게 이전 계층의 출력 값을 더해주며 학습한다. 결과적으로 제안하는 방법은 이전 결과 기법에 비해 약15~18배 정도의 속도향상을 얻었다.

키워드: 쿼드트리(Quadtree), 이진화(Binarization), 다운스케일링(Downscaling), 합성곱 신경망(Convolutional neural network), 슈퍼 해상도(Super-resolution)

I. Introduction

최근에 물리 기반 유체 시뮬레이션의 연산 비용 절약을 위해 쿼드트리 기반의 Super-Resolution 기법이 제안되었다.[1] 이 기법은 쿼드트리를 활용하여 효율적으로 데이터를 분할하고 유의미한 데이터만을 연산에 사용하기 때문에 연산 속도의 향상은 물론이고, 고해상도의 데이터가 입력 데이터로 사용되어도 메모리 부족 문제를 피할수 있다는 장점이 있다 (Fig. 1 참조). 본 논문에서는 이러한 장점을 제공하는 쿼드트리를 계산하는 과정에서 이진화와 다운스케일링 기법을 추가함으로써 쿼드트리 연산 소요 시간을 월등히 감소시킬 수 있는 최적화 기법을 제안한다.

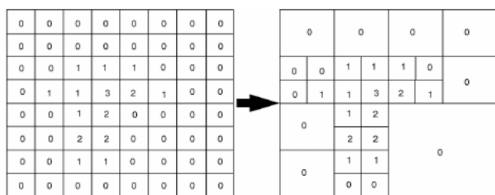


Fig. 1. Comparison with regular grid and quadtree structures.

II. The Proposed Scheme

본 연구에서는 우선 입력 데이터가 주어지면 이진화와 다운스케일링을 통해 연기 데이터를 압축한다. 여기서 이진화를 통해 연기의 밀도 데이터가 손실되는 문제를 최소화 하면서, 시뮬레이션 공간의 크기를 줄이기 때문에 쿼드트리 연산 속도 최적화가 가능하다. Fig. 2에서 보듯이 이진화 과정은 연한 밀도 데이터의 손실 없이 검출된 결과를 보여준다 (Fig. 2a 참조). 그 다음 데이터를 8×8, 16×16과 같은 작은 사이즈로 분할 한 뒤, 각 분할 데이터의 밀도가 임계값을 초과하는지에 따라, 초과한다면 FD(Full Density), 초과하지 않으면 ED(Empty Density)로 분류한다. 본 논문에서는 상향식 병합을 기반으로 쿼드트리를 계산하며, 여기서 일련의 규칙에 따라 노드를 병합한다. 쿼드트리가 구성되면, 연산이 끝난 노드들을 업스케일링하여 원본 데이터의 크기로 복원하고, 원본에 맞게 수정된 위치 정보에 해당하는 밀도 데이터들을 이용한다. 이 과정을 마친 데이터는 네트워크를 통해 고해상도 데이터로 업스케일링한다[2].

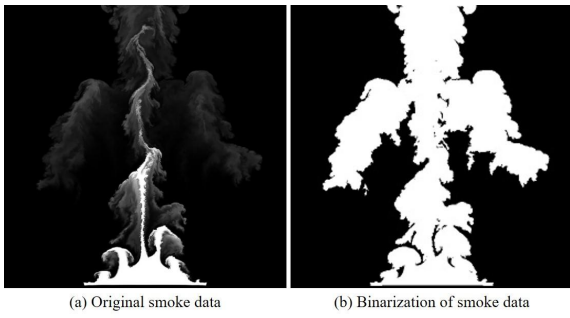


Fig. 2. Comparison with original and binarization of smoke data.

1. Binarization and downscaling of smoke simulation space

본 논문에서는 연기 시뮬레이션의 가로×세로 해상도를 같은 구조로 사용했기 때문에, 연산 소요 시간을 줄이기 위해 데이터를 압축하는 과정에서 한 스텝을 다운스케일링할 시, 이전 크기의 $\frac{1}{4}$ 배로 데이터 사이즈를 줄여나갔다. 본 논문에서는 다운스케일링 할 때 인접한 4개 노드의 밀도 평균을 계산하여 밀도 데이터를 압축했으며, 총 3번의 다운스케일링을 사용했다. 자세한 과정은 아래와 같다.

첫번째 스텝에서는 0.0, 0.01, 0.1, 0.05 등의 다양한 임계값을 이용한 테스트를 통해 가장 좋은 데이터 양상을 보여주는 0.05를 임계값으로 선정하였고 두번째와 세 번째 스텝에서는 0과 1로 이진화 된 인접한 4개 노드의 밀도 평균을 계산해 다운스케일링을 하였기 때문에, 0.0, 0.25, 0.5, 0.75, 1.0의 값으로만 이루어져 있으므로 임계값을 0.5로 지정하였다. 따라서, 이진화를 통해 데이터의 손실을 최소화 하고, 다운스케일링을 통해 데이터의 크기를 줄여 연산을 대폭 감소시켰기 때문에 쿼드트리 연산 속도의 최적화가 가능하였다. Fig. 3은 이진화와 다운스케일링을 했을 때의 결과이다. 다운스케일링한 결과는 시뮬레이션 공간이 줄어들어도 불고하고 밀도 손실 없이 원형의 형태를 그대로 유지했다 (Fig. 3b의 작은 삽입 이미지 참조). Fig. 3b는 다운스케일된 데이터를 원본 연기 데이터의 크기로 키운 결과이며, 약간 흐릿하기는 하지만 밀도의 형태를 그대로 유지하였다.

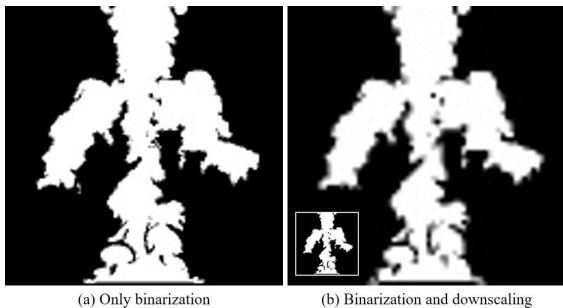


Fig. 3. Visualization of data changes due to iterative binarization and downscaling.

2. Classification of FD and ED in smoke data

본 논문에서는 이진화와 다운스케일링을 통해 압축 된 데이터를 쿼드트리 최하단 노드의 크기에 맞게 분할한다. 최하단 노드의 크기를 16×16 사이즈로 설정하였기 때문에 (입력 데이터가 512×512 기준), 3차례의 압축을 거친 데이터를 기준으로 했을 때 16개의 노드가 생성된다. 생성된 노드들을 이용하여 상향식으로 병합해야 되기 때문에, 우선 노드들의 밀도 존재 유무를 파악하여, 임계값과 비교해 FD와 ED로 분류한다.

3. Quadtree construction with bottom-up style for merging

앞에서 설명한 방법으로 분류 된 노드들을 상향식으로 병합한다. 쿼드트리이기 때문에 4개의 노드를 하나의 노드로 병합하는데 있어서 우리는 다음과 규칙을 사용하였다.

- 1) 모든 노드가 FD일 경우, 깊이 레벨이 같은 FD 노드들은 병합 (Merge) 후 부모 노드를 제외한 자식 노드들은 삭제한다.
- 2) 모든 노드가 ED일 경우, 깊이 레벨이 같은 ED 노드들은 별도의 병합 없이 삭제한다.

4. Upscaling quadtree and collecting meaningful data

공간 분할 기법을 사용하여 데이터를 분할 및 분류한 것은 연산이 필요한 데이터가 존재하는 위치 정보를 파악하기 위함 이었으므로, 쿼드트리 연산이 종료된 데이터는 압축 된 상태가 아닌, 원본 사이즈로 업스케일링 한다. 그리고 업스케일된 데이터의 위치 정보에 해당하는 원본 데이터를 가져와 조합하여 SR연산에 필요한 준비를 마친다.

5. Synthesizing stage

본 논문에서 사용 된 SR모델은 다음과 같다 (Fig 4 참조). 첫번째 컨볼루션 연산이 끝난 특징 맵(Feature map)을 이후 2번의 컨볼루션층을 거쳐서 나온 결과 값과 다시 더하는 방법을 사용한다. 컨볼루션 연산을 통해 이미지가 약간의 정보가 손실될 수가 있는데, 이를 최소화 하기 위해 이전 값과 더해준다.

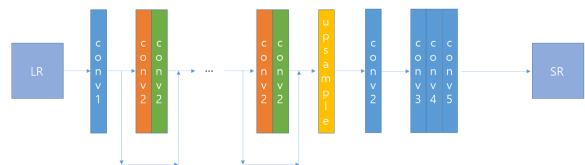


Fig. 4. VGG neural network.

이러한 사이클을 10번 반복하는데, 한 번의 주기 당 2개의 컨볼루션 층을 지나므로 20번의 컨볼루션 연산이 진행된다. 처음에는 첫 컨볼루션이 끝난 결과 값이 더해지지만 이후에는 이전 결과 값이 더해진다. 그 다음 업스케일링을 통해 사이즈를 2배 키우고, 이후 4번의 컨볼루션 연산을 끝으로 종료된다. 학습 데이터로는 COCO 2017 dataset을

사용하였고, SR Scale은 2, Batch Size는 32, Learning rate는 0.0001, 총 10만 스텝을 반복하였다. 학습을 진행하며 사용한 손실 함수는 MSE(Mean squared error)를 사용하였다 (수식 1 참조).

$$L = \frac{1}{W \times H} \sum_i^W \sum_j^H (SR_{ij} - HR_{ij})^2 \quad (1)$$

III. Conclusions

본 연구에서는 연기 데이터에서 유의미한 밀도가 있는 부분만을 연산에 사용하여 계산 속도를 향상시키고, 고해상도의 시뮬레이션 데이터가 입력 데이터로 사용되어도 메모리를 효율적으로 사용할 수 있는 쿼드트리 기반 프레임워크를 제안했다. 특히, 쿼드트리기법의 전처리 단계에서 이진화와 다운스케일링 기법을 추가로 적용하여 쿼드트리 연산 시간을 월등히 감소시켰다.

본 논문에서는 와류가 있는 512×512 시뮬레이션을 이진화와 다운스케일링 하였고, 이와 같은 결과를 품질과 계산속도 측면에서 비교하였다. Fig. 5와 6의 경우 전통적인 쿼드트리만을 사용했을 때보다, 우리의 방법이 약 15배의 성능 향상을 보여주었으며, Fig. 7과 8에서는 18배의 성능 향상을 보여주었다.

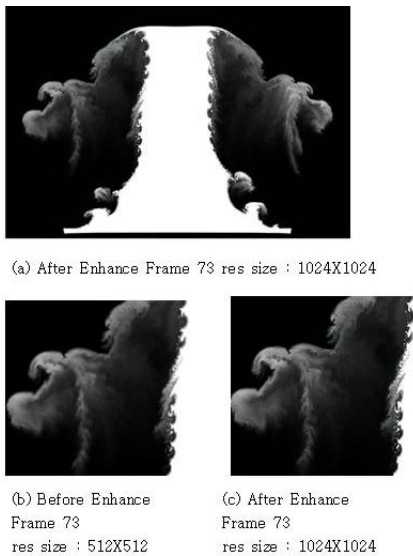


Fig. 5. Result 1.

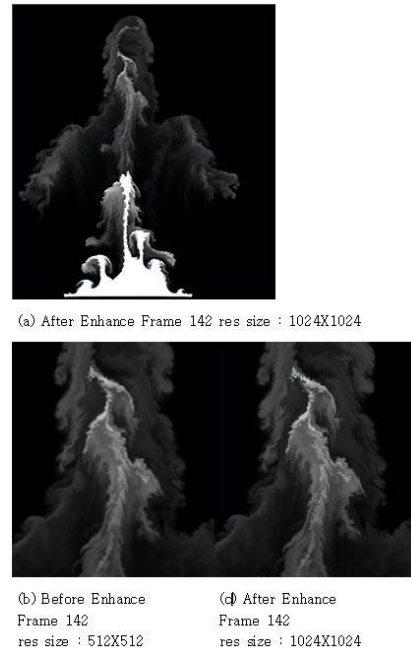


Fig. 6. Result 2.

Enhance 결과로는 2배 업스케일링 한 1024×1024 해상도의 데이터를 얻었다. 다음은 각 장면 별 Enhance 전후를 비교하기 위한 장면의 일부 프레임이다. 가장 자리의 디테일이 눈에 띄게 또렷하고 세밀해졌으며 경계가 뚜렷해진 결과를 확인할 수 있다.

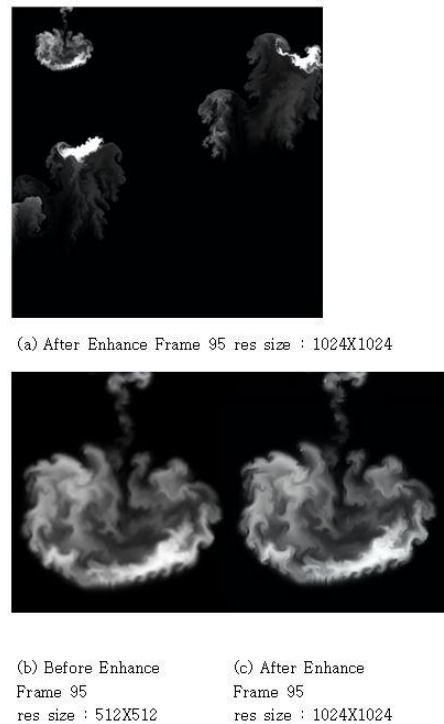


Fig. 7. Result 3.

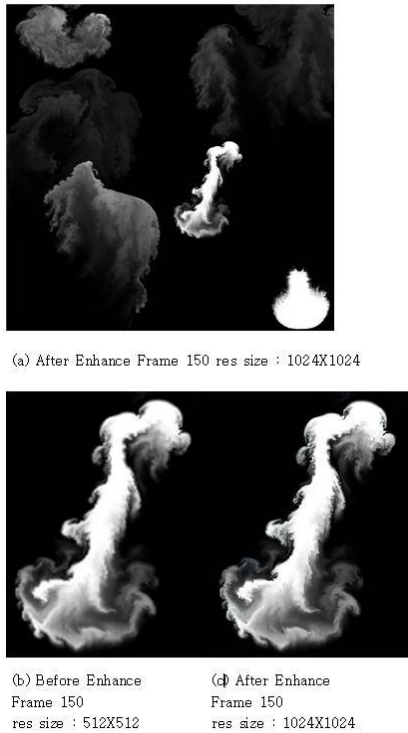


Fig. 8. Result 4.

제안하는 방법은 쿼드트리 기법을 통해 입력 데이터로 고해상도 데이터가 주어졌을 때도 메모리 부족 문제를 효율적으로 완화할 수 있는 방안을 제시했으며, 전처리 단계에 최적화 기법을 적용하여 쿼드트리 연산 시간을 감소시켰다. 향후, 외류의 움직임의 디테일한 움직임을 좀 더 보존하고 표현할 수 있는 방법에 대해 연구할 예정이다.

REFERENCES

- [1] Hong Byeongsun, Park Jihyeok, Choi Myungjin, Kim Changhun. Quad Tree Based 2D Smoke Super-resolution with CNN. J Korea Comput Graph Soc. vol. 25, pp. 105-113, 2019.
- [2] Dong, Chao and Loy, Chen Change and He, Kaiming and Tang, Xiaoou, Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence, pp. 295-307, 2015.