Original Article

Received: July 22, 2020
Revised: July 22, 2020
Accepted: September 12, 2020

Correspondence to:

Huijun Chen, Ph.D.
Center for Biomedical Imaging
Research, Department of
Biomedical Engineering, Room
109, School of Medicine, Tsinghua
University, Haidian District,
Beijing 100084, China.
Tel. +86-18311161088
Fax. +86-01062798332
E-mail: chenhj_cbir@mail.tsinghua.edu.cn

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © 2020 Korean Society of Magnetic Resonance in Medicine (KSMRM)

A Self-Supervised Learning Framework for Under-Sampling Pattern Design Using Graph Convolution Network

Yuze Li, Huijun Chen

Center for Biomedical Imaging Research, Department of Biomedical Engineering, School of Medicine, Tsinghua University, Haidian District, Beijing 100084, China

Purpose: To generate the under-sampling pattern using a self-supervised learning framework based on a graph convolutional network.

Materials and Methods: We first decoded the k -space data into the graph and put it into the network. After the processing of graph convolution layers and graph pooling layers, the network generated the under-sampling pattern for MR reconstruction. We trained the network on the simulated brain dataset enabled by the self-supervised learning strategy. We did simulation along with the *in vivo* brain and liver experiments under different noise levels and accelerating factors to compare the performance between the proposed method and traditional methods using the PSNR and SSIM index.

Results: The simulation experiments showed that the proposed method can achieve the best performance with low accelerating factors (2 and 3) at all noise levels and in high accelerating factors (4 and 5) at high noise levels (50 and 70 dB). In *in vivo* experiments, the proposed method attained the highest PSNR and SSIM in the brain dataset as well as in the liver dataset after fine tuning on a small liver dataset.

Conclusion: The self-supervised learning framework based on a graph convolutional network was able to design the under-sampling mask for MR reconstruction. The superior performance in the simulation and *in vivo* experiments demonstrated the feasibility and flexibility of the proposed method and its potential in clinical use.

Keywords: Undersampling Pattern Design; Graph Convolutional Network; Self-supervised Learning

INTRODUCTION

Reduction of scan time is always a popular topic in MRI development. Many fast imaging methods, such as parallel imaging and compressed sensing (CS) (1-3), adopt partial sampling k -space data to reconstruct images without sacrificing their quality. As important as the reconstruction algorithm is, the design of the under-sampling pattern also has a huge effect on the final reconstruction results. Thus, how to design an optimized under-sampling pattern is an important question. Within the CS framework, the pattern of a randomized sampling Poisson disk (RANP) (3) is widely used because it is incoherent, which is suitable for CS theory and can do good reconstructions. The design of RANP uses the probability density function (pdf), in which the sampling

position is randomly drawn on the k -space gridding according to the pdf, which in practice it samples more densely in the inner than in the outer region of the k -space. However, using pdf requires tuning one or more parameters whose different settings may affect the quality of reconstructed images and is also a time-consuming process.

Some groups have proposed learning-based pattern-design methods. In (4–6), the training was used data to generate a *a posteriori* distribution to replace the pdf for sampling the data. In (7), the mutual information of the training data was calculated and used to sample the rows in the k -space. However, these methods extract only the intensity or energy information of the training data. The geometric information contained in the k -space gridding is not well used when the information of different anatomies may be encoded in the distribution of the sampled k -space data.

Deep learning has recently been attracting more and more attention in medical imaging analysis (8). A graphic convolution network (GCN) is also a useful tool, especially in dealing with problems involving structural data (9); so it may be able to better optimize the k -space sampling pattern.

In this study, we propose a GCN-based method with self-supervised learning to help design the sampling pattern for MRI reconstruction. We based this method on training samples; so the setting of parameters is through learning but not by human tuning, which brings less randomness and is inefficient. In the proposed GCN, we used the self-supervised learning strategy. During training, we used Monte Carlo dropout to generate the temporary ground truth under-sampling pattern after every iteration; so there was no need to produce the unavailable gold-standard under-sampling patterns off-line. We did the simulation and *in vivo* experiments under different noise levels and accelerating factors. We compared the proposed method with traditional methods using qualitative and quantitative analysis.

MATERIALS AND METHODS

In this study, we selected the GCN with a first-order graph Laplacian (10) as the backbone network of the model. It worked well in the task of graph-node classification and had a lower computation burden to train. The following equation defines the basic forward operation of one step:

$$X_{l+1} = \sigma(\hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2} X_l W_l) \quad [1]$$

where $\hat{A}=A+I$ and A is the adjacency matrix of the input graph data, X_l is the feature matrix of the l_{th} layer, $\hat{D}$ is the diagonal node degree matrix, W_l is the trainable parameters, and σ is the activation function.

To apply the GCN to under-sampling pattern design, first, we had to convert the k -space data into data with a graph structure. We used the full sampling k -space data kf with complex values and a Cartesian grid G_c to generate the initial graph G as the network input. We first converted the k -space data $kf \in \mathbb{R}^{n \times n \times 2}$ to the 2-channel real-value data, since n was the square image size. Then we processed kf by a two-layer CNN (3×3 conv layer + ReLU layer) then generated a feature map with c channels. Here c is set to 16. The Cartesian grid G_c was flattened into a vector and converted to the adjacency matrix $A \in \mathbb{R}^{N \times N}$ where $N=n \times n$, and we considered the 4-neighbor as the adjacency relationship of A . Specifically, in adjacency matrix A , the item $a(p,q)$ described the adjacency relationship of position p and q in the Cartesian grid G_c . If p and q in G_c were a 4-neighbor, then $a(p,q)=1$, otherwise 0.

The graph G can be composed using the Cartesian grid G_c and the feature map obtained from kf . Each node of graph G was positioned on grid G_c , and the feature of the node was the corresponding item in the feature map. The weight of edges between every two nodes p can be set according to the connecting strength between two nodes, which is inversely proportional to its distance and intensity (or feature map value) difference. It can be described as follows:

$$\rho_i = \sqrt{\frac{1}{(d_i/n)} \times \frac{1}{(v_i/v_{max})}}$$

where d_i is the distance between the nodes, v_i is the intensity difference between the nodes, and v_{max} is the maximum feature value of all nodes.

Considering that the GCN needs to conduct the data sampling operation, the graph has to be shrunk by reducing the nodes and links in the original graph data. So, we adopted the graph pooling layer (11) in our method. Not like the pooling operation applied on the grid data in the CNN, we used here the projection vector p_i , which determines how much information of X_i can be transmitted through pooling and the k -maxpooling operation. The following equation defines the forward behavior of the graph pooling layer:

$$\begin{aligned}
y &= X_p / \|p\| \\
idx &= \text{rank}(y, k) \\
\tilde{y} &= \text{sigmoid}(y(idx)) \\
\tilde{X} &= X_i(idx, :) \\
A_{i+1} &= A_i(idx, idx) \\
X_{i+1} &= \tilde{X} \odot (\tilde{y} 1_C^T)
\end{aligned} \quad [2]$$

where y is the projected vector, which describes how much information remains, and k is the number of nodes left after k -maxpooling, which can be used as an adjustment of the acceleration factor in this work; $\text{rank}(\cdot, \cdot)$ is the ranking operation; $1_C \in \mathbb{R}^C$ is the vector containing C elements being 1; and $\odot$ is the element-wise matrix multiplication.

Figure 1 shows the framework of the proposed method.

When we constructed the initial graph, we applied the graph convolution operation and the graph pooling operation and repeated them three times. This process actually was to imitate the under-sampling. After three times of the “under-sampling” operations, many nodes and edges were removed, and we put the remaining nodes back into the initial graph to generate the final under-sampling mask. Considering that the ground truth for the pattern is unknown, we adopted a self-supervised learning method.

We used Monte-Carlo (MC) dropout (12) in this study to conduct the self-supervised learning. MC dropout was added only in the last layer of the GCN, and after each iteration, the network output several under-sampling patterns by randomly activating or repressing the neuron

Fig. 1. The proposed pattern-design framework. Initial graph G is generated from k -space data kf and a Cartesian grid Gc . Then, GCN and graph pooling operations are done to output the under-sampling pattern. After Monte-Carlo (MC) dropout, the optimized pattern is obtained to calculate the image and k -space loss with the learned pattern. The loss is then back-propagated through the whole network.

connection. Then, we obtained several reconstructed images using these masks with the reconstruction algorithm. Here in this study, we adopted total variation (TV) and l1 constraint CS reconstruction employing the split Bregman iteration method (13) because of its fast convergence. Next, we selected the best quality reconstructed image along with its mask as the temporary ground truth. Actually, we used the MC dropout to find the best generated mask based on the current network. After every iteration, the network worked better than in the last iteration.

We calculated the combination of the binary cross entropy (BCE) loss and mean absolute error (MAE) loss between the learned under-sampling pattern and the temporary ground truth pattern, and between the image from the learned under-sampling pattern and the temporary ground truth image.

The total loss can be defined as

$$Loss = Loss_I + Loss_K$$

where $Loss_I$ means the image-based MAE loss and $Loss_K$ means the k -space-based BCE loss. We heuristically found the weight of the loss.

We did the training on simulated brain phantoms (14). A total of 5000 T1w images were generated from 20 brain-anatomy models with data augmentation, including random flipping, rotation, and mirroring. Then we transformed these images into k -space data using FFT as the network input. The training, validation, and testing numbers were 3500, 1000, and 500, respectively. For training and validation samples, we added Gaussian noise with random levels of 0–70 dB. We did the training on a server with a I7-6850K CPU, 32 GB ram, and 2 Titan XP GPUs using the PyTorch 1.3.0 platform. The total training time was about 20 hours with 15000 epochs.

Experiment Settings

First, we compared the performance on simulation data between the proposed GCN-based method and traditional methods, including randomized sampling Gaussian (RANG) (3), an RANP (3), and a template-based Power Poisson disk (POWP) (4). It should be noted that, to make the comparison fair, we manually added the center area to the RANG pattern to be sure that the inner part of the k -space was densely sampled while the outer region was sparsely sampled.

The POWP method used the magnitude of the Fourier

transform of template images (or training images) as the cumulative probability density function to sample the data points. It actually was the process of sampling more points on the high-magnitude region or the high-energy region. So, we called it the power-based method. The training samples used in POWP were the same as in the GCN method.

Additionally, we tested multiple settings, including different accelerating factors (2, 3, 4 and 5) and different Gaussian noise levels (0, 30, 50, and 70 dB), added in the k -space in the simulation experiments during testing.

Then, we used existing *in vivo* brain datasets of 10 healthy volunteer and liver datasets of 8 healthy volunteer to further validate these methods. We obtained written informed consent from these volunteers. We scanned all the images on the 3T MR system. Brain images were acquired using the T1 FLASH sequence with the approval of the local institutional review board. The protocols used in this study were: in-plane resolution $1.5 \times 1.5 \text{ mm}^2$, matrix size 256×256 , slice thickness 2 mm, and TR/TE 300/2.5 ms. Liver images were acquired using the T1 GRE sequence, and the protocols were: in-plane resolution $1.5 \times 1.5 \text{ mm}^2$, matrix size 320×320 , slice thickness 2 mm, and TR/TE 191/4.5 ms. We evaluated and compared reconstruction performances between three settings: (1) training on brain + testing on brain; (2) training on brain + testing on liver; (3) training on brain + fine tuning on liver + testing on liver. In fine tuning, we used only 50 liver images as the training samples. It should be noted that fine tuning of POWP was retraining using all simulation brain images and *in vivo* liver images.

For simulation and *in vivo* experiments, we used the generated under-sampling masks to under-sample the fully acquired k -space, and we reconstructed the MR images using the TV and l1 constraint CS reconstruction algorithm (13). We assessed the image quality using the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) index (15) compared with the images reconstructed with the full k -space.

RESULTS

Simulation Experiments

Figure 2 shows the different sampling patterns for all methods at sampling rates of 2, 3, 4 and 5 at noiseless setting. All patterns were more likely to sample points near the center of the k -space. However, the proposed GCN method distributes points more evenly in the whole k -space.

Figure 3 shows the example reconstruction images of a

Fig. 2. The under-sampling patterns for all methods at different accelerating factors with a noiseless setting. To make the comparison fair, we manually added the center area to the RANG pattern. Other methods are more likely to gather sampling points in the center k -space. GCN = graphic convolution network; POWP = power Poisson disk; RANG = randomized sampling Gaussian; RANP = randomized sampling Poisson disk.

simulation brain phantom for all methods at sampling rates of 2, 3, 4 and 5 using a noise level of 50 dB. The images reconstructed by RANP and RANG were noisier and could not restore fine details. Both the proposed GCN method and the POWP method can get clearer images, but the GCN method showed better contrast at the brain sulcus and gyrus region.

The quantitative results for other settings are shown in Table 1. It can be noticed that with increasing noise levels and the accelerating factors, the reconstruction quality became worse, as PSNR and SSIM both decreased for all methods. When the accelerating factor was relatively low (2 or 3), the proposed GCN method achieved the best reconstruction results among all methods in all noise levels. However, at high accelerating factors (4 and 5) with lower noise levels (0 and 30 dB), the POWP method behaved the best, while the proposed GCN method performed much like the POWP and was the second best. Moreover, when noise level went up to 50 dB and 70 dB, the proposed GCN

method had the best performance among all the methods.

In vivo Experiments

The results of the *in vivo* experiment were similar, as shown in Figure 4. The first row shows testing images of *in vivo* livers with training on the simulated brain phantom. Good reconstruction quality with more details and better contrast can be seen from the GCN method. The clearer and more consecutive brain sulci can be seen in the GCN method. The second row shows testing images of *in vivo* livers with training on simulated brain phantom. Images from the POWP and GCN method show the comparable results. Both methods showed small liver vessels, and the bifurcation can be differentiated, whereas the other two methods failed to reconstruct the small vessels. The third row shows testing images of *in vivo* livers with training on the simulated brain phantom and fine tuning on *in vivo* liver images. After fine tuning, both learning-based methods, POWP and GCN, behaved better than that trained on only

Fig. 3. Reconstruction results for all methods at sampling rates 2, 3, 4 and 5 using at noise level of 50 dB. A yellow square indicates the zooming region, and the red arrow indicates the brain sulcus and gyrus region. GCN = graphic convolution network; POWP = power Poisson disk; RANG = randomized sampling Gaussian; RANP = randomized sampling Poisson disk.

Table 1. PSNR and SSIM for Reconstructed Images on Different Settings for Multiple Accelerating Factor and Noise Level

Acc. factor	Method	Noise (dB)	0	30	50	70	0	30	50	70
Evaluation matrix			PSNR				SSIM			
2	RANG		37.6	37.0	35.9	34.9	89.2	86.7	84.2	81.9
	RANP		39.3	38.5	36.7	35.2	90.1	87.2	85.0	83.1
	POWP		39.6	38.6	36.9	35.6	93.6	90.2	89.1	87.9
	GCN		39.9	39.0	37.2	36.1	94.5	90.9	90.1	88.3
3	RANG		34.5	34.0	32.8	31.9	86.5	85.3	84.6	83.1
	RANP		36.7	35.2	34.3	32.7	86.9	85.6	84.7	83.2
	POWP		36.9	35.3	34.3	32.9	89.9	87.7	86.3	85.2
	GCN		37.1	36.0	35.0	33.7	90.2	88.1	86.9	86.0
4	RANG		32.6	31.5	30.9	30.1	83.0	80.3	78.1	77.9
	RANP		34.0	32.3	31.2	30.6	83.2	80.5	78.2	77.9
	POWP		34.5	33.1	32.9	31.7	86.1	85.3	83.9	81.3
	GCN		34.1	32.9	33.2	32.0	85.7	85.0	84.2	81.7
5	RANG		30.1	29.5	28.7	28.1	80.1	78.6	77.5	73.9
	RANP		31.2	30.0	29.2	28.3	80.3	78.9	77.6	74.1
	POWP		32.1	31.7	29.9	28.6	84.0	82.9	80.6	79.1
	GCN		31.6	31.3	30.5	29.1	83.8	82.5	80.9	79.5

The best performance among 4 methods in different setting is shown in bold and underlined.

Acc. = accelerating; GCN = graphic convolution network; POWP = power Poisson disk; PSNR = peak signal-to-noise ratio; RANG = randomized sampling Gaussian; RANP = randomized sampling Poisson disk; SSIM = structural similarity

Fig. 4. Reconstructed images for *in vivo* brain and liver at accelerating of 3. First row shows testing images of *in vivo* brains with training on a simulated brain phantom; the best image quality can be seen in the GCN method. The second row shows the testing images of *in vivo* livers with training on a simulated brain phantom; images from the POWP and GCN methods show the comparable quality. The third row shows testing images of *in vivo* livers with training on a simulated brain phantom and fine-tuning on *in vivo* liver images; the results from GCN show stronger contrast than do those from the POWP method. The yellow square indicates the zooming region, and the red arrow indicates the brain sulcus and the liver vessel. GCN = graphic convolution network; POWP = power Poisson disk; RANG = randomized sampling Gaussian; RANP = randomized sampling Poisson disk.

Table 2. PSNR and SSIM for All Methods at Different Accelerating Factors in Different Training Settings

PSNR												
Setting	Brain (Train) + Brain (Test)				Brain (Train) + Liver (Test)				Brain (Train) + Liver (F.T. + Test)			
Acc. F.	2	3	4	5	2	3	4	5	2	3	4	5
RANG	36.5	33.5	31.0	29.6	37.1	34.9	32.2	30.2	(37.1)	(34.9)	(32.2)	(30.2)
RANP	38.2	34.7	31.0	30.0	38.9	35.0	32.5	30.5	(38.9)	(35.0)	(32.5)	(30.5)
POWP	38.3	34.9	33.2	31.5	39.5	36.3	34.7	32.6	39.8	36.6	35.0	32.9
GCN	38.9	35.5	33.5	31.7	39.2	35.8	34.1	32.1	40.2	36.9	35.5	33.3
SSIM												
Setting	Brain (Train) + Brain (Test)				Brain (Train) + Liver (Test)				Brain (Train) + Liver (F.T. + Test)			
Acc. F.	2	3	4	5	2	3	4	5	2	3	4	5
RANG	86.2	85.0	80.0	78.2	87.0	86.1	81.7	79.3	(87.0)	(86.1)	(81.7)	(79.3)
RANP	86.5	85.2	80.1	78.5	87.1	86.5	82.0	79.6	(87.1)	(86.5)	(82.0)	(79.6)
POWP	89.5	87.2	84.2	82.4	90.2	89.6	85.6	83.5	90.6	90.0	85.9	83.9
GCN	89.8	87.9	84.9	82.9	89.7	89.3	85.1	83.1	91.0	90.5	86.2	84.3

The number in brackets remains the same after fine tuning because RANG and RANP methods don't need training when generating under-sampling mask; The best performance among 4 methods in different setting is shown in bold and underlined.

Acc. F. = accelerating factor; F.T. = fine tuning; GCN = graphic convolution network; POWP = power Poisson disk; PSNR = peak signal-to-noise ratio; RANG = randomized sampling Gaussian; RANP = randomized sampling Poisson disk; SSIM = structural similarity

the brain dataset, whereas images from the GCN method have the stronger contrast, especially in the region of small vessels.

The quantitative comparison (Table 2) between these methods also showed that the POWP and GCN had similar results on all settings and worked far better than did the RANG and RANP methods. In the condition of training on brain + testing on brain, the GCN method behaved the best among all methods. However, in the condition of training on brain + testing on liver, the POWP method behaved the best, and the GCN method was the second best and performed like POWP. After POWP and GCN were fine-tuned on the liver dataset, GCN attained the best performance.

DISCUSSION AND CONCLUSION

In this study, we have proposed a novel GCN and self-supervised learning-based method to generate the under-sampling pattern for MR imaging. We did simulation and *in vivo* experiments under different noise levels and accelerating factors. We also tested generalization of the model in the *in vivo* liver dataset. The proposed method achieved the best results in most experiments, and its feasibility and flexibility were demonstrated.

The GCN used in this study shows advantages over other methods using only intensity or energy information. To form the initial data graph, we used both the intensity information and the geometric information in the *k*-space data. The graph convolution and graph pooling operations

learned not only the energy information for each data point on the *k*-space but also its local and global connection information. The POWP method considered the energy distribution, but it simply sampled more points on the high-energy region and sampled sparsely on the low-energy region. The GCN method covered the whole *k*-space region and sampled points that may have more influence on the quality of the reconstructed images. The reason that GCN can behave this way may be that it considered both energy and geometric information of the *k*-space. The results in Figure 2 show that the GCN method sampled points more uniformly, which can be a proof that GCN was not considering only the simple energy distribution of the *k*-space data.

In the *in vivo* liver dataset, the POWP showed the best results when trained only on the brain dataset; however, after fine tuning on a few liver images, the GCN worked the best, indicating that the proposed method seems to learn more structural information. The proposed method can be easily transferred to other anatomies and work well as long as it is fine-tuned on a few images of the given anatomy.

Traditional pattern-design methods rely mainly on pdf or other human-designed functions to sample the data. The tuning of parameters needs much time and may attain only sub-optimal performance. A learning-based method can overcome this issue, in that through learning, the parameters can be automatically settled, and once the model finishes the training, the pattern can be generated within a few seconds.

The proposed method uses the self-supervised learning

strategy, which makes the proposed method more useful. The strategy we used in this study is different from traditional semi-supervised and supervised methods, in that no single-labeled samples are required. All ground truths of under-sampling patterns are generated through MC dropout after each iteration; so it is free of searching for gold standard under-sampling patterns. This strategy is also very flexible, in that it can be easily transferred to other clinical conditions and be applied to different anatomies, different image contrasts, or different scanners.

There remain some limitations in the proposed method. First, we used simple and shallow network structures in the study. Though our model worked well on most experiments, more advanced GCN structures should be considered in future work. Second, we used only the split Bregman TV CS reconstruction algorithm in the under-sampling methods tests in this study. More experiments should be conducted to further validate our framework on different reconstruction algorithms. Last, multi-coil data with a sensitivity map were not considered. The GCN may learn more information from the correlation encoded in different coils to improve the performance.

In conclusion, the initial results of the GCN-based self-supervised-learning pattern-design framework demonstrated its feasibility and robustness, showing that GCN can be adopted in MRI under-sampling pattern design. Our results highlight that GCN can learn the energy distribution of the k -space and properly model it. In the future, more experiments should be conducted to support its potential in clinical use.

REFERENCES

1. Pruessmann KP, Weiger M, Scheidegger MB, Boesiger P. SENSE: sensitivity encoding for fast MRI. *Magn Reson Med* 1999;42:952-962
2. Griswold MA, Jakob PM, Heidemann RM, et al. Generalized autocalibrating partially parallel acquisitions (GRAPPA). *Magn Reson Med* 2002;47:1202-1210
3. Lustig M, Donoho D, Pauly JM. Sparse MRI: the application of compressed sensing for rapid MR imaging. *Magn Reson Med* 2007;58:1182-1195
4. Knoll F, Clason C, Diwoky C, Stollberger R. Adapted random sampling patterns for accelerated MRI. *MAGMA* 2011;24:43-50
5. Zhang Y, Peterson BS, Ji G, Dong Z. Energy preserved sampling for compressed sensing MRI. *Comput Math Methods Med* 2014;2014:546814
6. Vellagoundar J, Machireddy RR. A robust adaptive sampling method for faster acquisition of MR images. *Magn Reson Imaging* 2015;33:635-643
7. Seeger M, Nickisch H, Pohmann R, Scholkopf B. Optimization of k -space trajectories for compressed sensing by Bayesian experimental design. *Magn Reson Med* 2010;63:116-126
8. Litjens G, Kooi T, Bejnordi BE, et al. A survey on deep learning in medical image analysis. *Med Image Anal* 2017;42:60-88
9. Zhang Z, Cui P, Zhu W. Deep learning on graphs: a survey. *arXiv*, 2018;1812.04202
10. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. *ICLR* 2017
11. Gao H, Ji S. Graph U-nets. *ICML* 2019
12. Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. *ICML* 2016
13. Goldstein T, Osher S. The split Bregman method for L1-regularized problems. *Siam J Imaging Science* 2009;2:323-343
14. <https://brainweb.bic.mni.mcgill.ca/brainweb/>. Accessed November 22, 2019
15. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process* 2004;13:600-612