

Multiscale Ultrawide Foveated Video Extrapolation

Amit Aides
Electrical Engineering Dept.
Technion - Israel Inst. Tech.
Haifa 32000, Israel
amitibo@tx.technion.ac.il

Tamar Avraham
Computer Science Dept.
Technion - Israel Inst. Tech.
Haifa 32000, Israel
tammya@cs.technion.ac.il

Yoav Y. Schechner
Electrical Engineering Dept.
Technion - Israel Inst. Tech.
Haifa 32000, Israel
yoav@ee.technion.ac.il

Abstract

Video extrapolation is the task of extending a video beyond its original field of view. Extrapolating video in a manner that is consistent with the original video and visually pleasing is difficult. In this work we aim at very wide video extrapolation which increases the complexity of the task. Some video extrapolation methods simplify the task by using a rough color extrapolation. A recent approach focuses on artifact avoidance and run time reduction using foveated video extrapolation, but fails to preserve the structure of the scene. This paper introduces a multi-scale method which combines a coarse to fine approach with foveated video extrapolation. Foveated video extrapolation reduces the effective number of pixels that need to be extrapolated, making the extrapolation less time consuming and less prone to artifacts. The coarse to fine approach better preserves the structure of the scene while preserving finer details near the domain of the input video. The combined method gains improvement both visually and in processing time.

1. Introduction

The ever increasing popularity of video media brought great advances to the video display industry. Nowadays, televisions enable the viewer to enjoy video with higher definition and in three dimensions. Video extrapolation offers a new way to enhance the viewer experience by extending the video beyond its original field of view (FOV), embedding the viewer in the video. A first rough step in this direction is the Ambilight television system [16]. There, the wall around the television screen is illuminated in real-time with light that matches the colors that exist at the margins of each frame. Related setups are discussed in [7] and [18].

Recently, [2] proposed extrapolation which is an actual video. Video extrapolation relates to *video inpainting*, a.k.a. video completion, (e.g., [9, 10, 15, 23]) and *video retargeting* (e.g., [20, 22]). However, in video inpainting the missing parts are inside the FOV, while extrapolation aims to ex-

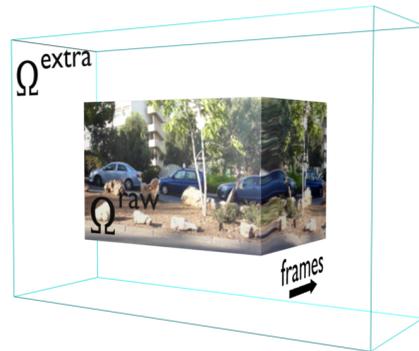


Figure 1: Spatio-temporal (ST) representation of the domain of the raw video Ω^{raw} , and the output domain Ω^{extra} .

tend the video beyond the FOV. Video inpainting methods have focused on filling in areas that occupy a rather small percentage (usually holes) of the raw FOV. When applied to large areas (see Fig. 1), traditional inpainting methods are prone to disturbing and distracting artifacts and usually require a long run time.

Ref. [2] approaches these issues by introducing a foveated method for (very wide) video extrapolation. The foveated method imitates the behavior of the human fovea. That is, the resolution of the extrapolated video diminishes toward the boundaries of the extrapolated region. Ref. [2] starts at the finest resolution level, i.e. the *innermost* domain, and proceeds outwards from *fine* to *coarse*. We refer to it as an *Inside-Out* method. The video extrapolated over each inner domain is used as the input video for the extrapolation of the next, coarser and outer, domain. While it successfully showed reduction in artifacts and processing time, the *Inside-Out* method has several shortcomings. In particular, it often does not preserve the structure of the scene and tends to result in a smooth extrapolation that does not preserve textures.

In this paper, we propose an opposite approach. The extrapolation starts with the coarsest scale, processing the

whole extrapolation domain. Then the other resolution scales are used consecutively from coarse to fine, where a finer scale applies to a smaller (inner) domain. We refer to this algorithm as the *Outside-In* method. The *Outside-In* method uses the information extrapolated in coarse scales to initialize extrapolation in finer scales. Therefore, the inner domains of the output video are processed several times, in different resolution scales, gradually refining the result.

There are several benefits to using the *Outside-In* method, compared to the *Inside-Out* method:

1. Extrapolating the coarse scale first has better chance of preserving the general structure of the scene.
2. The *Outside-In* method tends to preserve visual details and textures, in the areas close to the raw video, better than the *Inside-Out* method.
3. Faster run time. There is reduction in time-demanding efforts for avoiding the creation of distractions.

2. Background

2.1. Foveated Video Extrapolation

The human fovea occupies $\approx 1\%$ of the retina. It has high density of photo-receptors in a narrow region. Outside this narrow region, the density of photoreceptors falls off, and the perceived signals there are spatially much coarser [19]. This characteristic is exploited for foveated video extrapolation. The resolution of the extrapolated video diminishes toward the boundaries of the extrapolated domain. In [2], this approach significantly decreases the extrapolation time, when compared to methods that extrapolate all the output domain in full resolution. It also reduces significantly the amount of distracting artifacts in the extrapolated video.

2.2. Completion Algorithm

The extrapolation method in this paper makes use of a completion algorithm as a basic operation in the extrapolation of the video. As in Ref. [2], we use the patch-based algorithm suggested by Criminisi et al. [4], but other completion algorithms can be used as well (e.g., [3, 9, 10, 11, 12, 13, 15, 17]). The completion algorithm used in this work is described here briefly.

Consider Fig. 1. Let Ω^{raw} be the spatio-temporal (ST) domain of an input video I^{raw} . We wish to synthesize a new video I^{extra} , by extending I^{raw} to domain $\Omega^{\text{extra}} \supset \Omega^{\text{raw}}$. The algorithm is iterative. At any iteration, let Ω^{assigned} be the set of pixels to which a value was already assigned. Let $\Omega^{\text{missing}} = \Omega^{\text{extra}} / \Omega^{\text{assigned}}$ be the set of pixels to which an assigned value is still missing, as illustrated in Fig. 2. In any iteration, a target ST block $\hat{\mathcal{M}}$ is selected for completion, and another ST block $\hat{\mathcal{S}}$ is selected as the data source

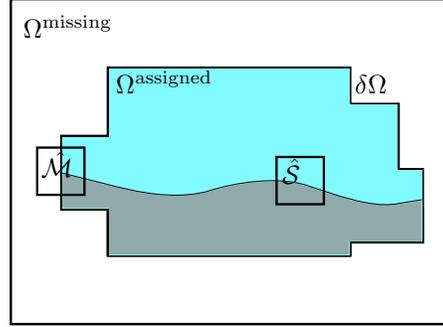


Figure 2: The completion algorithm. The domain Ω^{assigned} is already completed; Ω^{missing} is yet to be completed; $\delta\Omega$ is the boundary between them. The block $\hat{\mathcal{M}}$ is selected as the next target block, due to a strong edge crossing $\delta\Omega$ at its center point. The block $\hat{\mathcal{S}}$ is the selected source block.

for this completion. The ST blocks are of constant size: $w_x \times w_y \times w_t$. The target block is always on the boundary $\delta\Omega$ between Ω^{assigned} and Ω^{missing} . For better propagation of the scene structure, blocks that include strong ST edges crossing $\delta\Omega$, are selected first. Since the target block $\hat{\mathcal{M}}$ is always on the boundary $\delta\Omega$, only a subregion of it is known: $\hat{\mathcal{M}}_{\text{assigned}} = \hat{\mathcal{M}} \cap \Omega^{\text{assigned}}$. After $\hat{\mathcal{M}}$ is selected, a source $\hat{\mathcal{S}}$ is searched, throughout Ω^{raw} . The search is performed by comparing $\hat{\mathcal{M}}_{\text{assigned}}$ to the set of corresponding pixels in \mathcal{S} , which is termed $\mathcal{S}_{\text{assigned}}$. A cost function, $r(\mathcal{S})$ is defined,

$$r(\mathcal{S}) \triangleq \text{SSD_LAB}(\mathcal{S}_{\text{assigned}}, \hat{\mathcal{M}}_{\text{assigned}}), \quad (1)$$

where SSD_LAB is the sum-of-square-differences (SSD) in the CIE-Lab color space. The selected source block $\hat{\mathcal{S}}$ is the one for which

$$\hat{\mathcal{S}} \triangleq \underset{\mathcal{S} \subset \Omega^{\text{raw}}}{\text{argmin}} r(\mathcal{S}). \quad (2)$$

Pixels from the selected source $\hat{\mathcal{S}}$ are used for filling the missing pixels in $\hat{\mathcal{M}}$. Eq. (2) yields a source block having a subregion that is visually similar to the known part of the target block. Using $\hat{\mathcal{S}}$ usually results in a completed target block which is similar to a real block from Ω^{raw} .

2.3. Peripheral Avoidance of Distractions

It is very important that the viewer’s attention will not be distracted by the extrapolated image periphery. The *Peripheral Avoidance of Distractions* (PAD) criterion, introduced in [2] is used for reducing attention-drawing artifacts. PAD penalizes candidate source blocks, \mathcal{S} , that introduce strong spatial or temporal gradients. It also rejects outliers by seeking a source block that has high similarity to several other candidate source blocks.

This PAD criterion is implemented by introducing two new elements to Eq. (1). The first element, $g(\mathcal{S})$, penalizes

blocks that have prominent ST edges (calculated as the ℓ_1 norm of the ST gradient of \mathcal{S}). The second element, $h(\mathcal{S})$, penalizes outlier blocks by calculating the distance of each block from the center of its cluster:

$$\text{PAD}(\mathcal{S}) \triangleq \frac{r(\mathcal{S})}{\min_{\tilde{\mathcal{S}} \in \mathcal{S}} r(\tilde{\mathcal{S}})} + \frac{g(\mathcal{S})}{\min_{\tilde{\mathcal{S}} \in \mathcal{S}} g(\tilde{\mathcal{S}})} + \frac{h(\mathcal{S})}{\min_{\tilde{\mathcal{S}} \in \mathcal{S}} h(\tilde{\mathcal{S}})}. \quad (3)$$

All elements of Eq. (3), i.e. $r(\mathcal{S})$, $g(\mathcal{S})$ and $h(\mathcal{S})$, are normalized to have equal influence. The reader is referred to Ref. [2] for more on the calculation of $g(\mathcal{S})$ and $h(\mathcal{S})$.

To avoid repeated selection of the same source block, the algorithm introduces randomness (Efros and Leung [5]). This is achieved by selecting a source block randomly out of the N_{top} candidates graded by Eq. (3).

3. Outside-In Video Extrapolation

3.1. Foveated Coarse to Fine Extrapolation

The Inside-Out method [2] uses the above completion algorithm at different scales. When extrapolating regions close to the raw video, it uses fine resolution, and when extrapolating regions close to the boundary it uses coarse resolution. Even though the *Inside-Out* method uses multiple resolution scales, each location is processed using only a single scale. If the object to be completed is much larger than the patch size, the completion algorithm may create an extrapolation that is locally correct but doesn't match the structure of the object. Consider for example the synthetic example of chairs in Fig. 3. The 13×13 px patches used by the *Inside-Out* method are much smaller than the size of a chair. Therefore, the *Inside-Out* method fails to reconstruct the cropped chairs. On the other hand, increasing the size of the patch reduces the ability to reconstruct small objects.

The *Outside-In* method addresses this problem by using a multi-scale approach. This approach allows the extrapolation of both fine and coarse details. Each domain is processed several times, each time using a finer resolution scale. We now detail the *Outside-In* method for foveated video extrapolation.

Following [6], we define a scale map that represents the resolution decay over the support of I^{extra} . This scale map is quantized to N levels $\{f^\sigma\}_{\sigma=0}^{N-1}$. Here, $\sigma \in [0, N_{\text{scales}} - 1]$ is a scale index, and $0 < f < 1$ is a constant which determines the resolution quantization steps. The number of scale levels N is determined by f and the extent of the support of I^{extra} . Accordingly, we define N spatial domains $\{\Psi_\sigma\}_{\sigma=0}^{N-1}$ as illustrated in Fig. 4. The innermost domain Ψ_0 has the same scale as the input video, which is in domain Ω^{raw} . The widest domain, $\Psi_{N-1} = \Omega^{\text{extra}}$, relates to the coarsest resolution level: f^{N-1} . According to this definition, $\Omega^{\text{raw}} \subset \Psi_0 \subset \dots \subset \Psi_{N-1}$.

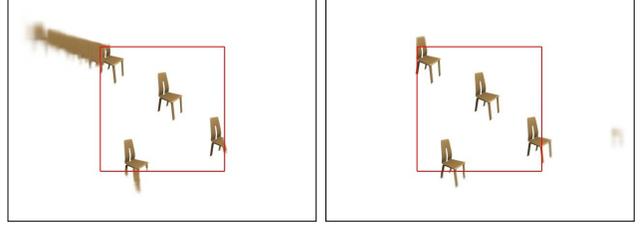


Figure 3: [Left] Extrapolation *Inside-Out*. [Right] Extrapolation *Outside-In*. The red square marks the raw domain. Synthetic image produced using a *Blender* chair model [14].

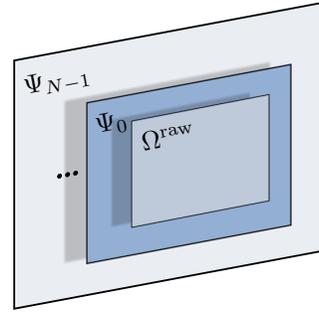


Figure 4: Extrapolation domains used by the *Outside-In* method: $\Omega^{\text{raw}} \subset \Psi_0 \subset \dots \subset \Psi_{N-1}$.

Our *Outside-In* method extrapolates domain Ψ_{N-1} first, using the coarsest resolution: f^{N-1} . Then, the algorithm extrapolates domains $\Psi_{N-2}, \dots, \Psi_0$, in this order, using their corresponding scales: f^{N-2}, \dots, f^0 . This results in a coarse to fine extrapolation. The inner domains which have finer details are contained in the coarse domains, i.e., $\Psi_\sigma \subset \Psi_{\sigma+1}$. The *Outside-In* method uses the video extrapolated in the coarse scales to initialize the extrapolation in finer inner domains.

The *Outside-In* method has three major stages as illustrated in Fig. 5:

Stage 1

The raw video is scaled down repeatedly, forming a Gaussian pyramid of source videos:

$$I_\sigma^{\text{src}} \triangleq \begin{cases} I^{\text{raw}} & \sigma = 0 \\ \text{REDUCE}(I_{\sigma-1}^{\text{src}}) & \text{otherwise} \end{cases}, \quad (4)$$

where REDUCE refers to a downscale¹ operation by a factor f (in our experiments we used $f = 0.5$).

¹The downscale and upscale operations can be performed either only in the spatial domain or both in the spatial and temporal domains (see discussion in Sec. 5.1).

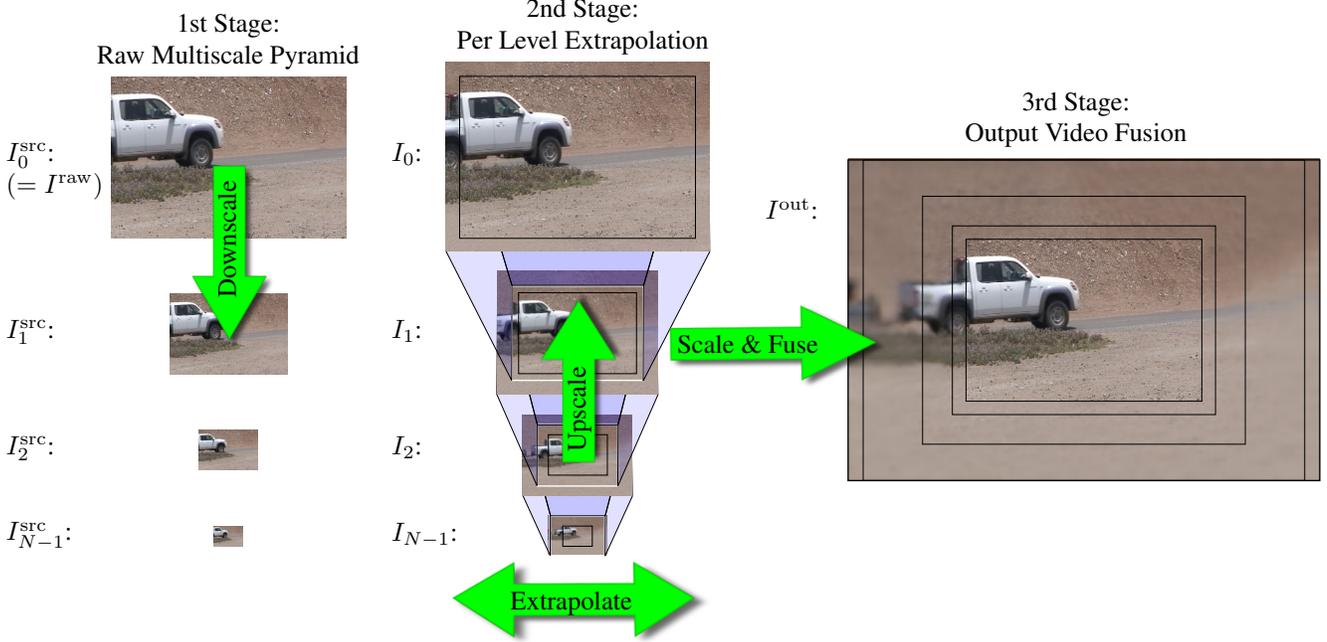


Figure 5: A schematic description of *Outside-In* foveated video extrapolation. First, a Gaussian video pyramid $\{I_\sigma^{\text{src}}\}_{\sigma=0}^{N-1}$ is created. Then each level of the pyramid, starting bottom up, is extrapolated over its corresponding domain. The information from the coarser scale initializes the extrapolation in the finer level. Finally, I^{raw} and the extrapolated videos $\{I_\sigma\}_{\sigma=0}^{N-1}$ are merged to create the output video I^{out} .

Stage 2

In this stage, extrapolation is performed. A completion algorithm is used for extrapolating each domain around the source videos. At scale σ this domain is:

$$\Psi_\sigma^{\text{extra}} \triangleq \Psi_\sigma / \Omega^{\text{raw}}. \quad (5)$$

The extrapolation starts with the coarsest (and widest) domain, $\Psi_{N-1}^{\text{extra}}$ and proceeds in order from coarse to fine. First, we obtain I_{N-1} in domain $\Psi_{N-1}^{\text{extra}}$, then we obtain I_{N-2} in domain $\Psi_{N-2}^{\text{extra}}$ (see Fig. 5). Overall this results in a set of target videos $\{I_\sigma\}_{\sigma=0}^{N-1}$:

$$I_\sigma = \begin{cases} \text{COMPLETE}(I_{N-1}^{\text{src}}, \emptyset) & \sigma = N - 1 \\ \text{COMPLETE}(I_\sigma^{\text{src}}, I_\sigma^{\text{expand}}) & \text{otherwise} \end{cases}. \quad (6)$$

Here COMPLETE refers to a completion algorithm which we describe later in Sec. 3.2. The COMPLETE operator in Eq. (6) has two inputs. The first is the source video (see Eq. (4)). The source patches used in the extrapolation of I_σ are taken only from the corresponding I_σ^{src} . This way, each target video I_σ is extrapolated only in the corresponding resolution, f^σ . The second input to the COMPLETE operator is an *initialization* of I_σ . It is based on an extrapolated video $I_{\sigma+1}$, which was obtained previously in the coarser (and wider) domain $\Psi_{\sigma+1}^{\text{extra}}$. Since $I_{\sigma+1}$ is extrapolated at

coarser resolution than I_σ , an upscale operation is necessary. Therefore, in each spatial and scale, the completion algorithm is initialized by an expanded video:

$$I_\sigma^{\text{expand}}(\Psi_\sigma^{\text{extra}}) \triangleq \text{EXPAND}[I_{\sigma+1}(\Psi_{\sigma+1}^{\text{extra}})]. \quad (7)$$

Here, EXPAND is an upscale operation by a factor of f^{-1} .

Stage 3

Finally, when all the ST domains $\{\Psi_\sigma\}_{\sigma=0}^{N-1}$ have been extrapolated, the resulting videos in these domains $\{I_\sigma\}_{\sigma=0}^{N-1}$, are scaled up to a resolution consistent with I^{raw} . They are then merged to create I^{extra} . The process of scaling and merging of $\{I_\sigma\}_{\sigma=0}^{N-1}$ is identical to the corresponding stage described in [2]. The output video I^{out} is defined as:

$$I^{\text{out}} \triangleq I^{\text{extra}} \cup I^{\text{raw}}. \quad (8)$$

3.2. Coarse Initialization

The *Outside-In* approach extends the basic completion algorithm, described in Sec. 2.2, in two major ways:

3.2.1 Coarse to Fine Completion

As explained in Sec. 3.1, in the *Outside-In* method, the finer resolution levels base their extrapolation on the results of

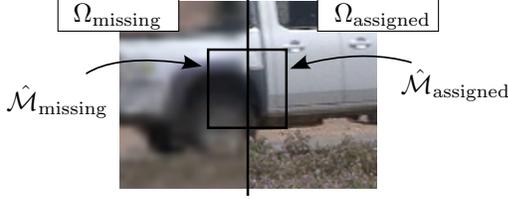


Figure 6: The subregion $\hat{\mathcal{M}}_{\text{missing}}$ is already completed with coarse resolution information.

coarser resolution levels (see Eq. (6) and (7)). This is implemented by the following modification to Eq. (1):

$$r(\mathcal{S}) \triangleq \begin{cases} \text{SSD_LAB}(\mathcal{S}_{\text{assigned}}, \hat{\mathcal{M}}_{\text{assigned}}) & \sigma = N - 1 \\ \text{SSD_LAB}(\mathcal{S}, \hat{\mathcal{M}}) & \text{otherwise} \end{cases} \quad (9)$$

When extrapolating the coarsest level, $\sigma = N - 1$, the algorithm uses the same cost function, Eq. (1). In any of the other scale levels, the domain to be extrapolated $\Psi_{\sigma}^{\text{extra}}$ is pre-filled with $I_{\sigma+1}^{\text{expand}}$. Therefore, the subregion $\hat{\mathcal{M}}_{\text{missing}}$,

$$\hat{\mathcal{M}}_{\text{missing}} \triangleq \hat{\mathcal{M}} \cap \Omega^{\text{missing}}, \quad (10)$$

is already completed with coarse resolution information (see Fig. 6). Therefore, it is possible to use *all* pixels of the source and destination patches in the calculation of the distance function. The coarse resolution information in $\hat{\mathcal{M}}_{\text{missing}}$ is a coarse initialization to the missing higher resolution information.

3.2.2 Selective Use of the PAD Criterion

The PAD criterion significantly increases the processing time of the extrapolation. The *Outside-In* method is able to dispose of the PAD criterion immediately after the coarse level. This is possible because the finer resolution levels base their extrapolation on the results of the coarse resolution level.

The coarsest resolution level has the least pixels to process of all scales. Hence, using PAD only in the coarsest scale leads to a considerable gain in processing speed.

4. Cascaded Outside-In

The number of scale levels N is fixed by the foveation parameters and video dimensions (see Sec. 3.1). Using less resolution levels results in a lateral resolution decay which is insufficient. On the other hand, due to the discrete nature of images, there is a lower limit to the patch size that can be used in the completion algorithm. Consider Fig. 7. The source patches are derived only from the source video, I_{σ}^{src} . As the scale coarsens, the size of the input video I_{σ}^{src} approaches the size of the patch. At some stage, it becomes

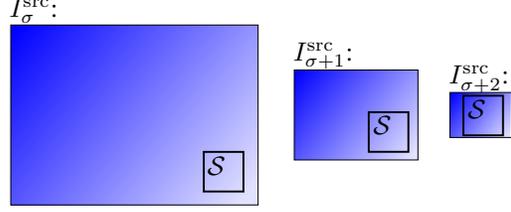


Figure 7: As the scale coarsens, the size of the input video I_{σ}^{src} approaches the size of the patch. At some stage, it becomes impossible for the completion algorithm to select source patches \mathcal{S} with small gradients.

impossible for the completion algorithm to select source patches with small gradients. In these cases, the resulting extrapolated videos $\{I_{\sigma}\}_{\sigma=0}^{N-1}$ may contain many artifacts. Hence, there is a limit to the practical number of scale levels that can be used.

To overcome this issue, we use a cascaded variant of the *Outside-In* method. The domain set $\{\Psi_{\sigma}\}_{\sigma=0}^{N-1}$ is split into several subsets² (see Fig. 8):

$$T_l \triangleq \{\Psi_{\sigma}\}_{\sigma=Kl}^{K(l+1)-1}, \quad 0 \leq l \leq \lfloor N/K \rfloor. \quad (11)$$

A corresponding set of raw videos $\{I_l^{\text{input}}\}_{l=0}^{\lfloor N/K \rfloor}$ is defined:

$$I_l^{\text{input}} \triangleq \begin{cases} I^{\text{raw}} & l = 0 \\ \text{REDUCE}(I_{Kl-1}) & \text{otherwise} \end{cases} \quad (12)$$

Let us start at T_0 . The extrapolation of the domains in T_0 (see Fig. 8), is performed using the *Outside-In* method using I_0^{input} as the raw video. Then, the next domains in T_1 are extrapolated using I_1^{input} as the raw video. This process is repeated for all subsets.

What is gained by this? The gain is that I_l^{input} is larger than I^{raw} for all $l > 0$. This is because I_l^{input} was already created by extrapolation. The *Cascaded Outside-In* method enables the use of a large number of scale levels N , without worrying that the support of I_{σ}^{src} might approach the size of a patch. The video I_{Kl-1} in Eq. (12) is extrapolated using the PAD criterion. Therefore, the completion algorithm is able to select patches with small gradients. Fig. 9 demonstrates the improvement of the *Cascaded Outside-In* over the basic *Outside-In*.

5. Experiments

5.1. Visual Comparison

Figures 10 and 11 show results of extrapolating still images and video sequences using the *Inside-Out* and *Cascaded Outside-In* methods. The latter method better preserves the structure of the scene.

²The number of domains in each subset is denoted K . Its value depends on the size of I^{raw} . In our work we used $K = 3$.

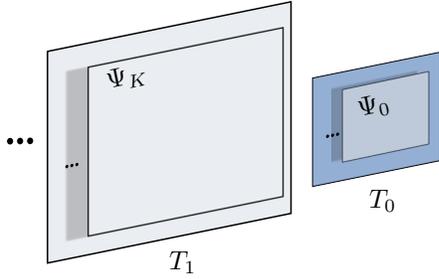


Figure 8: The domain set $\{\Psi_\sigma\}_{\sigma=0}^{N-1}$ is split into several subsets T_l .

The *Cascaded Outside-In* seems to also better preserve textures (e.g. the grass region of the top image in Fig. 10). The reason is that the *Outside-In* method does not use PAD when extrapolating the inner domains. Source patches are not penalized for having prominent ST edges. This results in more details in the extrapolated video.

The *Cascaded Outside-In* method has faults. In the coarsest scale the proper source patches have much less details to distinguish themselves from other patches. In this scale level, the completion algorithm uses the PAD criterion. Therefore, it might prefer improper patches just because they are smoother than the proper patches. This results in visual errors that propagate to the final result.

As described in Sec. 3.1 the REDUCE and EXPAND operations are in the heart of the foveated framework. These operations can be applied either solely in the spatial domain, or in the spatial and temporal domain simultaneously. We found out that temporal coarsening significantly improves results during extrapolation of scenes with global motion. When the scene includes more complex motions, e.g., objects that enter and/or exit the FOV, better results are obtained when the scaling is done only in the spatial domain. Extrapolated images and video clips, including results of a user study, are available in [1].

5.2. Run Time Comparison

Table 1 compares some run time results of the *Inside-Out* and *Outside-In* methods. The *Outside-In* method improves the run time of the extrapolation algorithm. The improvement ranges from 10% to 46% depending on the length of the video sequence. The *Outside-In* method enjoys the same run time superiority as the *Inside-Out* method when compared to methods that do not use the foveated approach (see [2]).

6. Discussion

The foveated approach significantly reduces the number of distracting artifacts and significantly improves run time over other extrapolation methods. By incorporating a coarse

Video Size	Inside-Out	Outside-In	Improvement
$342 \times 234 \times 50$	29007	17513	39%
$310 \times 230 \times 76$	15154	13514	10%
$320 \times 292 \times 32$	55134	29689	46%

Table 1: Run time comparison of the *Inside-Out* and *Outside-In* methods. Video sizes are given in Width \times Height \times Frames format. Measured time is in CPU seconds.

to fine approach into the foveated approach, *Outside-In* extrapolation can better preserve the scene structure, while maintaining visual details in the area close to the input video. This results in a more visually pleasing extrapolated video. In addition, *Outside-In* extrapolation also shows run time improvement over the *Inside-Out* method.

The *Outside-In* extrapolation is not without drawbacks. The coarse-to-fine approach bases extrapolation of the finer scales on the result of a coarse extrapolation. Sometimes this results in artifacts, as explained in Sec. 5. A better approach may be to process the video in several scales *simultaneously* (as in [8]) or use global optimization (as in [23]).

Acknowledgments

We thank Ad de Vann, Hans van Mourik and Guido Volleberg for useful discussions. This research was supported by Philips Consumer Lifestyle, Eindhoven, The Netherlands. Yoav Schechner is a Landau Fellow - supported by the Taub Foundation. This work was also supported by the Israel Science Foundation (Grant 1031/08) and was conducted in the Ollendorff Minerva Center. Minerva is funded through the BMBF.

References

- [1] <http://www.cs.technion.ac.il/~tammya/extrapolation.html>. 6
- [2] T. Avraham and Y. Y. Schechner. Ultrawide foveated video extrapolation. *IEEE Journal of Selected Topics in Signal Processing*, Special Issue on Recent Advances in Processing for Consumer Displays, to appear 2011. 1, 2, 3, 4, 6
- [3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28, Aug. 2009. 2
- [4] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based inpainting. *IEEE Trans. Image Processing*, 13:1200–1212, 2004. 2
- [5] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Proc. IEEE ICCV*, pages 1033–1038, 1999. 3
- [6] W. S. Geisler and J. S. Perry. A real-time foveated multi-resolution system for low-bandwidth video communication.

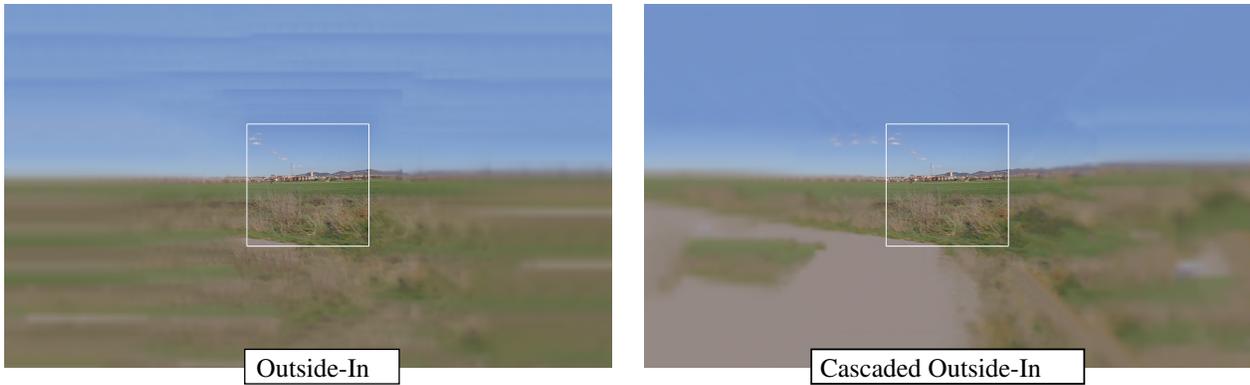


Figure 9: [Left] Extrapolation *Outside-In*. The extrapolation exhibits many artifacts, particularly in the sky. [Right] Extrapolation *Cascaded Outside-In*. This method exhibits much less artifacts. The white square in the middle marks the raw domain. The image is from the LabelMe database [21].

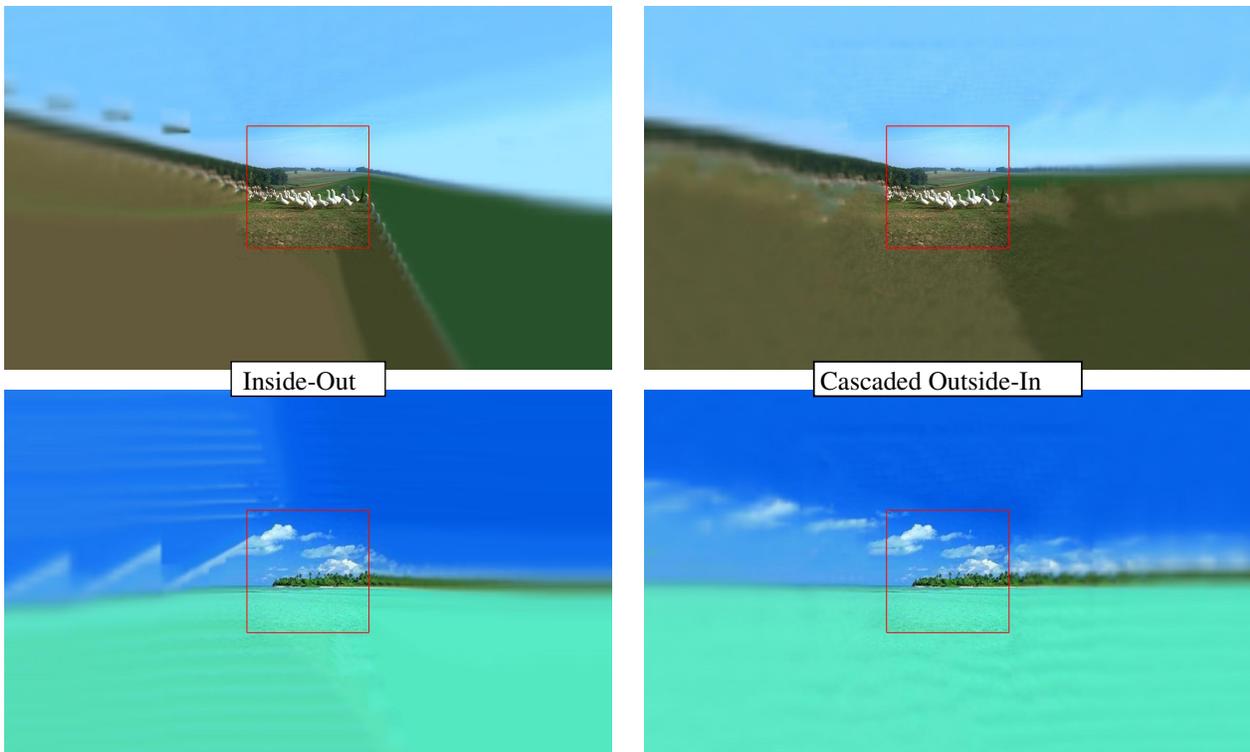


Figure 10: Still images extrapolated *Inside-Out* and *Outside-In*. The red square in the middle marks the raw domain. The *Outside-In* extrapolation better preserves the structure of the scene. The images are from the LabelMe database [21].

- Proc. SPIE Human Vision and Electronic Imaging*, pages 294–305, 1998. 3
- [7] A. Ghosh, M. Trentacoste, H. Seetzen, and W. Heidrich. Real illumination from virtual environments. In *Eurographics Workshop on Rendering*, pages 243–252, 2005. 1
- [8] M. Holtzman-Gazit and I. Yavneh. A scale-consistent approach to image completion. *Int. J. Multiscale Comput. Eng.*, 6:617–628, 2008. 6
- [9] J. Jia, T. P. Wu, Y. W. Tai, and C. Tang. Video repairing: Inference of foreground and background under severe occlusion. In *Proc. IEEE CVPR*, pages 364–371, 2004. 1, 2
- [10] Y. Jia, S. Hu, and R. R. Martin. Video completion using tracking and fragment merging. *The Visual Computer*, 21:601–610, 2005. 1, 2
- [11] A. C. Kokaram, B. Collis, and S. Robinson. Automated rig removal with bayesian motion interpolation. *IEEE VISIP*, 152:407–414, 2005. 2
- [12] W. F. L. Zhu, Y. Chen and A. Torralba. Nonparametric

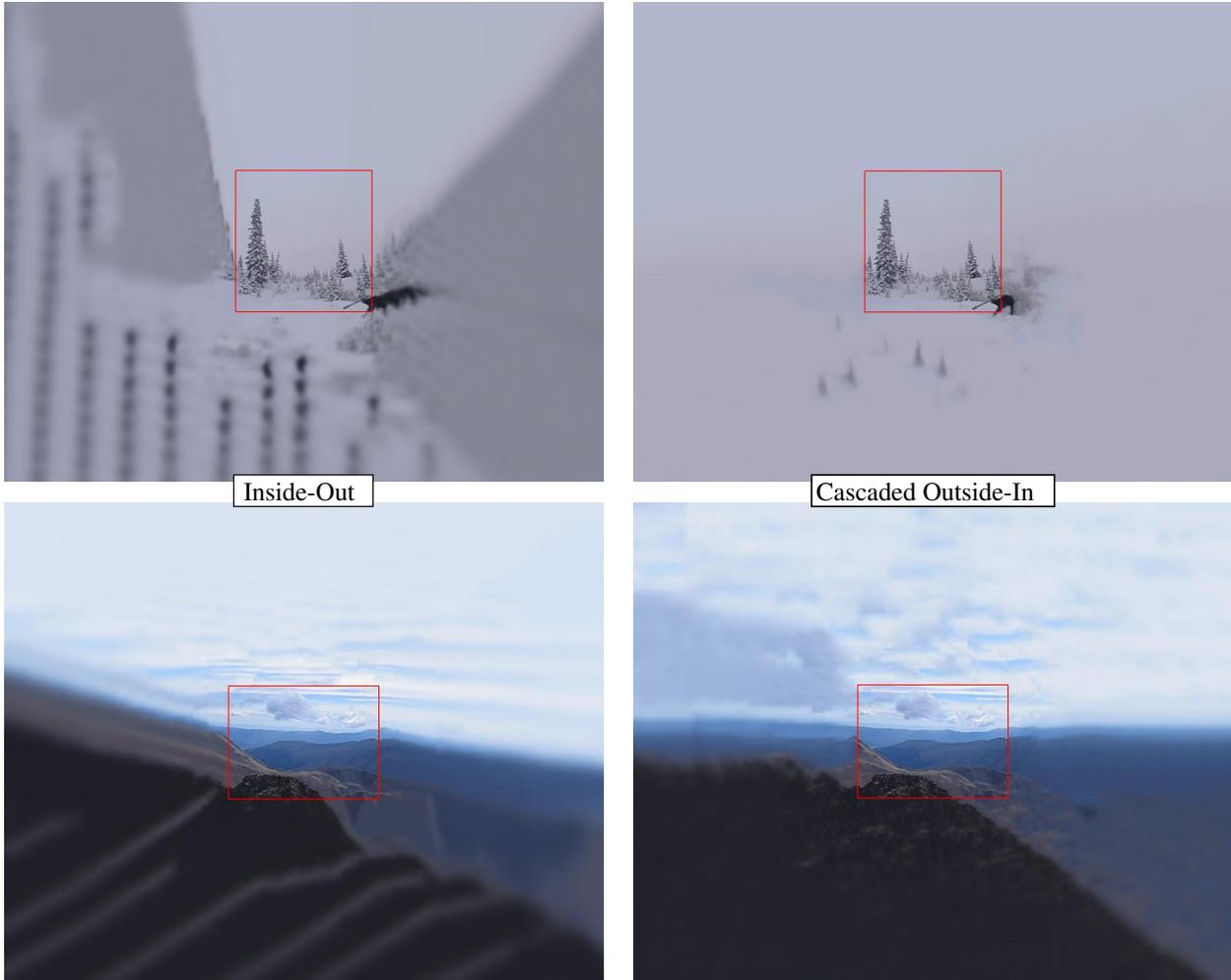


Figure 11: Sample frames from videos extrapolated *Inside-Out* and *Outside-In*. The red square marks the raw video. The latter method better preserves the structure of the scene. The videos are taken with permission from <http://forrest.stonedahl.com/index.html> and <http://hamilton.uchicago.edu/~sethi/>.

- bayesian texture learning and synthesis. *NIPS*, 2009. 2
- [13] Y. Matsushita, E. Ofek, W. G. X. Tang, and H. Shum. Full-frame video stabilization with motion inpainting. *IEEE TPAMI*, 28:1150–63, 2006. 2
- [14] M. Myftiu. Multi-d thinking. <http://www.freewebs.com/multid/>. 3
- [15] K. A. Patwardhan, G. Sapiro, and M. Bertalmio. Video inpainting of occluding and occluded objects. In *Proc. IEEE ICIP*, volume 2, pages 69–72, 2005. 1, 2
- [16] Philips-Ambilight. Philips has unveiled 'ambilight technology' for TVs. <http://www.televisionpoint.com/news/newsfullstory.php?id=1124298851>. 1
- [17] Y. Pritch, E. Kav-Venaki, and S. Peleg. Shift-map image editing. In *Proc. IEEE ICCV*, pages 151–158, 2009. 2
- [18] A. G. Rempel, W. Heidrich, H. Li, and R. Mantiuk. Video viewing preferences for hdr displays under varying ambient illumination. In *ACM APGV*, pages 45–52, 2009. 1
- [19] J. G. Robson and N. Graham. Probability summation and regional variation in contrast sensitivity across the visual field. *Vision Research*, 21:409–418, 1981. 2
- [20] M. Rubinstein, A. Shamir, and S. Avidan. Improved seam carving for video retargeting. *ACM TOG*, 27(3):1–9, 2008. 1
- [21] B. Russell and A. Torralba. Labelme: a database and web-based tool for image annotation. *IJCV*, 77:157–173, 2008. 7
- [22] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. Summarizing visual data using bidirectional similarity. In *Proc. IEEE CVPR*, 2008. 1
- [23] Y. Wexler, E. Shechtman, and M. Irani. Space-time completion of video. *IEEE TPAMI*, 29:463–476, 2007. 1, 6