(54) **Title**: IMAGE PROCESSING SYSTEM FOR DOWNSCALING IMAGES USING PERCEPTUAL DOWNSCALING METH-
OD
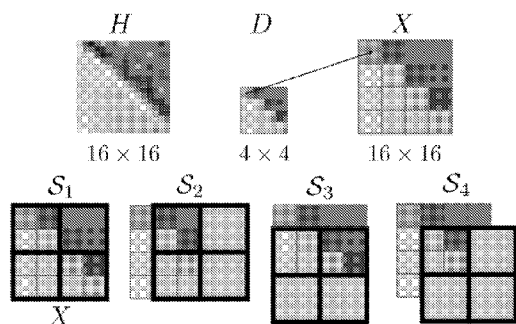
FIG. 3
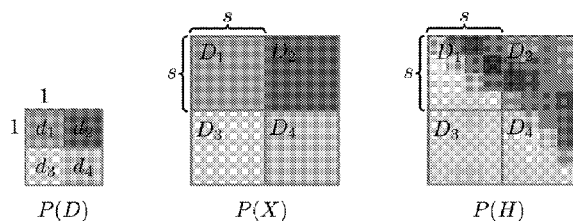
FIG. 4

(57) **Abstract**: An image processor inputs a first image and
outputs a downscaled second image by upscaling the second
image to a third image, wherein the third image is substan-
tially the same size as the first image size with a third resolu-
tion, associating pixels in the second image with a corres-
ponding group of pixels from the third set of pixels, sampling
a first image area at a first location of the first set of pixels to
generate a first image sample, sampling a second image area
of the third set of pixels to generate a second image sample,
measuring similarity between the image areas, generating a
perceptual image value, recursively adjusting values of third
set of pixels until the image perception value matches a per-
ceptual standard value, and adjusting pixel values in the
second image to a representative pixel value of each of the
corresponding group of pixels.

IMAGE PROCESSING SYSTEM FOR DOWNSCALING IMAGES USING PERCEPTUAL DOWNSCALING METHOD

CROSS-REFERENCES TO PRIORITY AND RELATED APPLICATIONS

[0001] This application claims priority from and is a non-provisional of U.S. Provisional Patent Application No. 62/196,640 filed July 24, 2015 entitled "Perceptually Based Downscaling of Images". The entire disclosure of application recited above is hereby incorporated by reference, as if set forth in full in this document, for all purposes.

FIELD OF THE INVENTION

[0002] The present disclosure generally relates to image processing. The disclosure relates more particularly to apparatus and techniques for performing downscaling of images wherein an input image file is processed to generate a downscaled output image file.

BACKGROUND

[0003] Image downscaling is a fundamental operation performed constantly in digital imaging. The abundance of high resolution capture devices and the variety of displays with different resolutions make it an essential component of virtually any application involving images or video. However, this problem has so far received substantially less attention than other sampling alterations.

[0004] Classical downscaling algorithms aim at minimizing aliasing artifacts by linearly filtering the image via convolution with a kernel before subsampling and subsequent reconstruction, following the sampling theorem [Shannon 1998]. However, along with aliasing, these strategies also smooth out some of the perceptually important details and features since the kernels used are agnostic to the image content.

[0005] A solution to this problem is adapting the kernel shapes to local image patches [Kopf et al. 2013] in the spirit of bilateral filtering [Tomasi and Manduchi 1998], so that they are better aligned with the local image features to be preserved. This strategy can significantly increase the crispness of the features while avoiding ringing artifacts typical for post-sharpening filters. However, it still cannot capture all perceptually relevant details, and as a result, might distort some of the perceptually important features and the overall look of the input image or lead to artifacts such as jagged edges [Kopf et al. 2013].

[0006] Loss of some of the perceptually important features and details stems from the common shortcoming of these methods that they operate with simple error metrics that are known to correlate poorly with human perception [Wang and Bovik 2009]. Significant improvements have been obtained for many problems in image processing by replacing these classical metrics with perceptually based image quality metrics [Zhang et al. 2012; He et al. 2014].

[0007] The standard approach to image downscaling involves limiting the spectral bandwidth of the input high resolution image by applying a low-pass filter, subsampling, and reconstructing the result. As is well-known in signal processing, this avoids aliasing in the frequency domain and can be considered optimal if only smooth image features are desired. Approximations of the theoretically optimum sinc filter, such as the Lanczos filter, or filters that avoid ringing artifacts such as the bicubic filter are typically used in practice [Mitchell and Netravali 1988]. However, these filters often result in oversmoothed images as the filtering kernels do not adapt to the image content. The same is true for more recent image interpolation techniques [Thévenaz et al. 2000; Nehab and Hoppe 2011].

[0008] Recently, Kopf et al. [2013] showed that significantly better downscaling results with crisper details can be obtained by adapting the shapes of these kernels to the local input image content. Since the kernels better align with the features in the input image, they capture small scale details when present. However, the method does not take perceptual importance of the features into account, resulting in loss of apparent details and hence leading to a rather abstract view of the input image. Indeed, the method is shown to provide excellent results for generating pixel-art images [Kopf et al. 2013].

[0009] Improvements in this image processing are desirable, in that they can reduce the amount of computing effort needed to obtain pleasing downscaled images.

References

[0010] BANTERLE, F., ARTUSI, A., AYDIN, T., DIDYK, P., EISEMANN, E., GUTIERREZ, D., MANTIUK, R., and MYSZKOWSKI, K. 2011. Multidimensional image retargeting. In ACM SIGGRAPH Asia 2011 Courses, ACM, ACM SIGGRAPH Asia.

[0011] BONNIER, N., SCHMITT, F., BRETTEL, H., and BERCHE, S. 2006. Evaluation of spatial gamut mapping algorithms. In Proc. 14th Color Imag. Conf., 56–61.

[0012] BRUNET, D., VRSCAY, E., and WANG, Z. 2010. Structural similarity-based approximation of signals and images using orthogonal bases. In Image Analysis and Recognition, A. Campilho and M. Kamel, Eds., vol. 6111 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, 11–22.

[0013] BRUNET, D., VRSCAY, E., and WANG, Z. 2012. On the mathematical properties of the structural similarity index. Image Processing, IEEE Trans. on 21, 4 (April), 1488–1499.

[0014] BRUNET, D. 2012. A Study of the Structural Similarity Image Quality Measure with Applications to Image Processing. PhD thesis, University of Waterloo.

[0015] CHAI, L., SHENG, Y., and ZHANG, J. 2014. SSIM performance limitation of linear equalizers. In Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, 1220–1224.

[0016] CHANNAPPAYYA, S., BOVIK, A., and HEATH, R. 2006. A linear estimator optimized for the structural similarity index and its application to image denoising. In *Image Processing, 2006 IEEE International Conference on*, 2637–2640.

[0017] CHANNAPPAYYA, S., BOVIK, A., CARAMANIS, C., and HEATH, R. 2008. SSIM-optimal linear image restoration. In *Acoustics, Speech and Signal Processing (ICASSP), 2008. IEEE International Conference on*, 765–768.

[0018] CHANNAPPAYYA, S., BOVIK, A., and HEATH, R. 2008. Rate bounds on SSIM index of quantized images. *Image Processing, IEEE Trans. on 17*, 9 (Sept), 1624–1639.

[0019] CHANNAPPAYYA, S. S., BOVIK, A. C., CARAMANIS, C., and JR., R. W. H. 2008. Design of linear equalizers optimized for the structural similarity index. *Image Processing, IEEE Trans. on 17*, 6, 857–872.

[0020] CHEN, G.-H., YANG, C.-L., and XIE, S.-L. 2006. Gradient-based structural similarity for image quality assessment. In *Image Processing, IEEE International Conference on*, 2929–2932.

[0021] DEMIRTAS, A., REIBMAN, A., and JAFARKHANI, H. 2014. Full-reference quality estimation for images with different spatial resolutions. *Image Processing, IEEE Trans. on 23*, 5 (May), 2069– 2080.

[0022] DIDYK, P., RITSCHEL, T., EISEMANN, E., and MYSZKOWSKI, K. 2012. *Perceptual Digital Imaging: Methods and Applications*. CRC Press, ch. Exceeding Physical Limitations: Apparent Display Qualities.

[0023] DONG, J., and YE, Y. 2012. Adaptive downsampling for high-definition video coding. In *ICIP 2012*, 2925–2928.

[0024] GERSTNER, T., DECARLO, D., ALEXA, M., FINKELSTEIN, A., GINGOLD, Y., and NEALEN, A. 2012. Pixelated image abstraction. In *NPAR 2012, Proc. of the 10th International Symposium on Non-photorealistic Animation and Rendering*.

[0025] HE, L., GAO, F., HOU, W., and HAO, L. 2014. Objective image quality assessment: A survey. *Int. J. Comput. Math. 91*, 11 (Nov.), 2374–2388.

[0026] KOPF, J., SHAMIR, A., and PEERS, P. 2013. Content-adaptive image downscaling. *ACM Trans. Graph. 32*, 6 (Nov.), 173:1– 173:8.

[0027] KRAWCZYK, G., MYSZKOWSKI, K., and SEIDEL, H.-P. 2007. Contrast restoration by adaptive countershading. In *Proc. of Eurographics 2007*, Blackwell, vol. 26 of *Computer Graphics Forum*.

[0028] LISSNER, I., PREISS, J., URBAN, P., LICHTENAUER, M. S., and ZOLLIKER, P. 2013. Image-difference prediction: From grayscale to color. *Image Processing, IEEE Trans. on 22*, 2, 435–446.

[0029] LIU, T., YUAN, Z., SUN, J., WANG, J., ZHENG, N., TANG, X., and SHUM, H.-Y. 2011. Learning to detect a salient object. *Pattern Analysis and Machine Intelligence, IEEE Trans. on 33*, 2 (Feb), 353–367.

[0030] MITCHELL, D. P., and NETRAVALI, A. N. 1988. Reconstruction filters in computer-graphics. In *Proc. of SIGGRAPH '88*, ACM, New York, NY, USA, 221–228.

[0031] NEHAB, D., and HOPPE, H. 2011. Generalized sampling in computer graphics. Tech. Rep. MSR-TR-2011-16, February.

[0032] OGAWA, T., and HASEYAMA, M. 2013. Image inpainting based on sparse representations with a perceptual metric. *EURASIP Journal on Advances in Signal Processing 2013*, 1.

[0033] PANG, W.-M., QU, Y., WONG, T.-T., COHEN-OR, D., and HENG, P.-A. 2008. Structure-aware halftoning. *ACM Trans. Graph. 27*, 3 (Aug.), 89:1–89:8.

[0034] POLESEL, A., RAMPONI, G., and MATHEWS, V. J. 1997. Adaptive unsharp masking for contrast enhancement. In *ICIP '973-Volume Set-Volume 1 - Volume 1*, IEEE Computer Society, Washington, DC, USA, 267–.

[0035] REHMAN, A., WANG, Z., BRUNET, D., and VRSCAY, E. 2011. SSIM-inspired image denoising using sparse representations. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, 1121–1124.

[0036] RITSCHEL, T., SMITH, K., IHRKE, M., GROSCH, T., MYSZKOWSKI, K., and SEIDEL, H.-P. 2008. 3D Unsharp Masking for Scene Coherent Enhancement. *ACM Trans. Graph. (Proc. of SIGGRAPH 2008) 27*, 3.

[0037] SHANNON, C. 1998. Communication in the presence of noise. *Proc. of the IEEE 86*, 2 (Feb), 447–457.

[0038] SHAO, Y., SUN, F., LI, H., and LIU, Y. 2014. Structural similarity-optimal total variation algorithm for image denoising. In *Foundations and Practical Applications of*

Cognitive Systems and Information Processing, vol. 215. Springer Berlin Heidelberg, 833–843.

[0039] SILVESTRE-BLANES, J. 2011. Structural similarity image quality reliability: Determining parameters and window size. *Signal Processing 91*, 4, 1012 – 1020.

[0040] SMITH, K., LANDES, P.-E., THOLLOT, J., and MYSZKOWSKI, K. 2008. Apparent greyscale: A simple and fast conversion to perceptually accurate images and video. *Computer Graphics Forum (Proc. of Eurographics 2008) 27*, 2 (APR).

[0041] TH'EVENAZ, P., BLU, T., and UNSER, M. 2000. Interpolation revisited. *Medical Imaging, IEEE Trans. on 19*, 7, 739–758.

[0042] TOMASI, C., and MANDUCHI, R. 1998. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, 839–846.

[0043] TRENTACOSTE, M., MANTIUK, R., and HEIDRICH, W. 2011. Blur-Aware Image Downsizing. In *Proc. of Eurographics*.

[0044] WANG, Z., and BOVIK, A. 2009. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *Signal Processing Magazine, IEEE 26*, 1 (Jan), 98–117.

[0045] WANG, Z., and LI, Q. 2007. Video quality assessment using a statistical model of human visual speed perception. *J. Opt. Soc. Am. A 24*, 12, B61B69.

[0046] WANG, Z., BOVIK, A., SHEIKH, H., and SIMONCELLI, E. 2004. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Trans. on 13*, 4 (April), 600– 612.

[0047] WANG, S., REHMAN, A., WANG, Z., MA, S., and GAO, W. 2011. Rate-AAIM optimization for video coding. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, 833–836.

[0048] WU, X., ZHANG, X., and WANG, X. 2009. Low bit-rate image compression via adaptive down-sampling and constrained least squares upconversion. *Trans. Img. Proc. 18*, 3 (Mar.), 552–561.

[0049] YEGANEH, H. 2014. *Cross Dynamic Range and Cross Resolution Objective Image Quality Assessment with Applications*. PhD thesis, University of Waterloo.

[0050] ZHANG, Y., ZHAO, D., ZHANG, J., XIONG, R., and GAO, W. 2011. Interpolation-dependent image downsampling. *Image Processing, IEEE Trans. on 20*, 11 (Nov), 3291–3296.

[0051] ZHANG, L., ZHANG, L., MOU, X., and ZHANG, D. 2012. A comprehensive evaluation of full reference image quality assessment algorithms. In *ICIP 2012*, 1477–1480.

[0052] ZHOU, F., and LIAO, Q. 2015. Single-frame image super-resolution inspired by perceptual criteria. *Image Processing. IET 9*, 1, 1–11.

SUMMARY

[0053]   An image processor inputs a first image and outputs a downscaled second image by upscaling the second image to a third image, wherein the third image is defined by a third set of pixels derived from the second set of pixels establishing a third image size substantially the same size as the first image size with a third resolution, associating individual pixels in the second image with a corresponding group of pixels from the third set of pixels, sampling a first image area at a first location of the first set of pixels to generate a first image sample, sampling a second image area of the third set of pixels corresponding to the first image area location and size to generate a second image sample, measuring the similarity between the first image area and the second image area by processing the first image sample and the second image sample to generate a perceptual image value, recursively adjusting the values of third set of pixels until the image perception value matches a perceptual standard value within a pre-defined threshold, and adjusting individual pixel values in the second image to a representative pixel value of each of the corresponding group of pixels.

[0054] The following detailed description together with the accompanying drawings will provide a better understanding of the nature and advantages of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0055] Various embodiments in accordance with the present disclosure will be described with reference to the drawings, in which:

[0056] FIG. 1 illustrates artifacts of downscaling.

[0057] FIG. 2 illustrates various approaches to downscaling.

[0058] FIG. 3 illustrates a process an image downscaling engine might use to perform a piecewise constant interpolation.

[0059] FIG. 4 illustrates a process the engine performs to compute various values.

[0060] FIG. 5 illustrates post-sharpening after filtering.

[0061] FIG. 6 illustrates the effects of the patch size on downscaled images.

[0062] FIG. 7 illustrates results of deviations.

[0063] FIG. 8 illustrates an example image optimized over different patches.

[0064] FIG. 9 illustrates a downscaled edge of a picture frame.

[0065] FIG. 10 illustrates other aspects of image processing.

[0066] FIG. 11 illustrates other aspects of image processing.

[0067] FIG. 12 illustrates user study results.

[0068] FIG. 13 illustrates test results.

[0069] FIG. 14 illustrates image smoothing.

[0070] FIG. 15 illustrates results of downscaling.

[0071] FIG. 16 illustrates examples of images used for the user study.

[0072] FIG. 17 examples of downscaling with adaptively adjusting local details.

[0073] FIG. 18 is a block diagram that illustrates a computer system upon which an embodiment of the invention may be implemented.

[0074] FIG. 19 is a simplified functional block diagram of a storage device having an application that can be accessed and executed by a processor in a computer system with which an embodiment of the invention may be implemented.

[0075] FIG. 20 illustrates an example of a downscaling engine that takes in an input image file and outputs an output image file using processes described herein.

DETAILED DESCRIPTION

[0076] In embodiments described herein, an image processing engine inputs an image, processes it and outputs an output image where the output image is of smaller resolution than the input image and does so in a perceptually pleasing way, i.e., minimizing artifacts.

[0077] FIG. 1 illustrates some such artifacts. In each of FIG. 1A and 1B, there is an input image on the left and four results of downscaling on the right. The subsampling output images, bicubic output images and content-adaptive output images use conventional approaches, whereas the perceptual output images use novel and improved techniques described herein in more detail.

[0078] The subsampling output image might be produced using classical downscaling algorithms that linearly filter the image via convolution with a kernel before subsampling and subsequent reconstruction, but can result in smoothing out some of the perceptually important details and features since the kernels used are agnostic to the image content.

[0079] Kernel shapes can be non-agnostic to local image patches and increase the crispness of the features while avoiding ringing artifacts typical for post-sharpening filters, but still might not capture all perceptually relevant details, and as a result, might distort some of the perceptually important features and the overall look of the input image, as in the content-adaptive output images. The bicubic output images also end up with undesirable artifacts.

[0080] As explained in more detail below, the perceptual image preserves perceptually important features and the overall look of the original images. A perceptual image quality measure can be used in the processing, instead of standard metrics.

[0081] Loss of some of the perceptually important features and details stems from the common shortcoming of these methods that they operate with simple error metrics that are known to correlate poorly with human perception [Wang and Bovik 2009]. Significant improvements have been obtained for many problems in image processing by replacing these classical metrics with perceptually based image quality metrics [Zhang et al. 2012; He et al. 2014].

[0082] A standard approach to image downscaling involves limiting the spectral bandwidth of the input high resolution image by applying a low-pass filter, subsampling, and reconstructing the result. As is well-known in signal processing, this avoids aliasing in the frequency domain and can be considered optimal if only smooth image features are desired. Approximations of the theoretically optimum sinc filter, such as the Lanczos filter, or filters that avoid ringing artifacts, such as the bicubic filter, are typically used in practice. However, these filters often result in oversmoothed images as the filtering kernels do not adapt to the image content.

[0083] For natural images, the methods described herein can perform significantly better and provide crisper depictions of a high resolution input image by incorporating a perceptual metric. These methods also can provide better spatio-temporal consistency with less apparent aliasing artifacts, and run orders of magnitude faster with a simple and robust implementation, thus saving on computing resources.

[0084] Downscaling operators are also designed for other related problems. Several algorithms carefully tune the downscaling operators and filters to the interpolation method used for subsequent upscaling. Those methods do not really address perceptual quality of the downscaled image itself. Thumbnail generation tries to preserve the imperfections, in particular blurriness in the original images for accurate quality assessment from the downscaled images. In contrast, the downscaling problem can be regarded as selectively adjusting the blur to preserve the important details and overall look of an input image. Another related set of algorithms deals with retargeting images by changing the aspect ratios of input images [Banterle et al. 2011], while preserving important parts such as foreground objects in the image by carefully modifying the image content.

[0085] The embodiments described herein are able to keep the image content close to that of the original image and target resolution reductions far more than the retargeting algorithms

are normally designed for. Image abstraction methods can be used to generate artistic depictions of an input image such as via pixel art [Gerstner et al. 2012] by reducing the resolution as well as the color palette. The embodiments described herein provide better results by targeting realistic depictions of the input image.

5   [0086] As explained herein, the image processing engine can treat image downscaling as an optimization problem with SSIM as the error metric. This can provide a significant advantage for preserving perceptually important features. Also, a closed-form solution can be derived for the downscaling problem. This provides a perceptually based method for downscaling images that provides a better apparent depiction of the input image. Image downscaling can

10   be treated as an optimization problem where the difference between the input and output images is measured using a perceptual image quality metric. The downscaled images retain perceptually important features and details, resulting in an accurate and spatio-temporally consistent representation of the high resolution input. Our downscaling method preserves perceptually important fine details and features that cannot be captured with other metrics,

15   resulting in crisper images that provide a better depiction of the original image.

[0087] The image processing engine can derive the solution of the optimization problem in closed-form, which leads to a simple, efficient and parallelizable implementation with sums and convolutions. The process has computer run times similar to linear filtering and is orders of magnitude faster than the state-of-the-art for image downscaling. Herein, validation of the

20   effectiveness of the technique is provided with test results from extensive tests on many images, video, and by results of a user study, which indicates a clear preference for the results of the processes described herein.

[0088] The downscaling problem is treated as an optimization that solves for the downscaled output image given the input image. The error between the two images is measured using the

25   structural similarity (SSIM) index [Wang et al. 2004]. The use of SSIM in optimization problems has been hindered by the resulting non-linear non-convex error functions [Brunet et al. 2012]. However, as explained herein, for the downscaling problem, it is possible to derive a closed-form solution to this optimization. The solution leads to a non-linear filter, which involves computing local luminance and contrast measures on the original and a smoothed

30   version of the input image. Although the filter is seemingly different than SSIM without any covariance term, it maximizes the mean SSIM between the original and downscaled images.

[0089] The downscaled images do not exhibit disturbing aliasing artifacts for natural images and are spatio-temporally more coherent than methods based on kernel optimizations [Kopf et al. 2013]. This allows the engine to apply the technique to video downscaling as well. The

resulting process has a very simple, efficient, and parallelizable implementation with sums and convolutions. It thus has a computational complexity similar to the classical filtering methods, and runs orders of magnitude faster than the state-of-the-art [Kopf et al. 2013].

[0090] Standard error metrics such as the mean squared error is well-known to correlate poorly with human perception when measuring image differences [Wang and Bovik 2009]. Instead, for the assessment of the quality of images and video, a variety of perceptually based image quality metrics has been proposed. Full reference quality metrics refer to the assumption that an input image can be compared to an available reference image for quality assessment. For the downscaling problem, the input image is the reference, and the downscaled output is the image to be assessed.

[0091] The engine uses the structural similarity (SSIM) index [Wang et al. 2004], which is one of the most widely used and successful full reference image quality metrics [Brunet et al. 2012], but other metrics might be used as well or instead. SSIM represents a matching score between two images by local luminance, contrast, and structure comparisons. Given a high resolution input image H, the engine seeks to find the down-scaled output image D that is as close as possible to H as measured by the SSIM index. The dissimilarity measure between images H and D is denoted $d(H, D)$. A goal is the image D^* that minimizes this measure $d(H, D)$. This measure can be obtained using images that are single-channel images, such that each pixel of H and D contains a single number in the dynamic range [0, 1], and further assume for simplicity that the width and height of H are downscaled by an integer factor, s, to produce D. If the actual downscaling factor is not an integer, the engine can preprocess and upscale the input image by bicubic filtering such that the factor becomes an integer. Similar approaches can be taken for multi-channel images.

[0092] FIG. 2 illustrates various approaches to downscaling. FIG. 2A is the input image and FIG. 2B illustrates eight examples of output images, wherein the one in the lower right is produced by optimizing the perceptual metric. Commonly used filters for downscaling, such as the box or bicubic filter, result in oversmoothing. Trying to avoid oversmoothing by post-sharpening the downscaled images (Sharpened image) or using the Lanczos filter can lead to ringing artifacts and the small-scale features can still not be recovered. Generalized sampling [Nehab and Hoppe 2011] and content-adaptive downscaling [Kopf et al. 2013] can produce crisper images, but cannot preserve perceptually important details. In contrast to the others, the by using the perceptual metric, a perceptually optimum image as measured by this metric is generated.

[0093] Most image quality assessment measures are not designed to compare images of different spatial resolutions [Yeganeh 2014]. For images of different resolutions, there are two common simple approaches: downscaling the higher resolution image, or upscaling the lower resolution one [Demirtas et al. 2014]. To not lose the information present in H, the engine upscales D to form an upscaled image X that has the same dimensions as H.

[0094] FIG. 3 illustrates a process the engine might use to perform a piecewise constant interpolation, with each pixel of D replicated in s^2 pixels of X. In the upper left is the input high resolution image H (16 by 16 pixels), the downscaled image D (4 by 4 pixels) in the center, and its scaled version X (16 by 16 pixels) on the right. Each pixel of D is replicated in 16 pixels of X. Also illustrated there are "patch sets." A patch set, S_k ($k=1, 2, ..., n_p$) contains patches that do not overlap (tiles). In this example, there are 2 by 2 patches on D, so $n_p=4$. The patch sets are shifted by 4 pixels in X and H, which corresponds to a shift of 1 pixel in D.

[0095] The SSIM index is a local measure of similarity computed between local patches of images. These similarity scores are then summed for all patches to compute the mean SSIM. Denoting the ith patch of image X by $P_i(X)$, the downscaling problem can thus be written as finding the optimum X^* that satisfies Equation 1 for some set, S, of patches with the constraint that each group of pixels of X that corresponds to a single pixel of D has the same pixel value.

$$X^* = \mathrm{argmin}_X \sum\nolimits_{P_i \in S} d(P_i(H), P_i(X)) \qquad\text{(Equ. 1)}$$

[0096] The pixel values of X do not have to be constrained to be in [0,1] and the optimized D might contain a small number of pixels negligibly outside of the dynamic range. The shapes and the set of the patches can be defined in various ways, depending on the application considered [Silvestre-Blanes 2011]. For a given patch size n_p, the engine uses the set S of all possible square patches of width (and height) $s\sqrt{n_p}$ (excluding the patches not completely within the image limits), but in patch sets such that each patch set S_k contains only non-overlapping patches, and where S is the union of all of the patch sets S_k. The final X^* is computed by averaging the solutions X_k^* of the problem in Equation 1 for different patch sets. Since each group of s^2 pixels in X actually corresponds to a single pixel in D, integer patch shifts in D lead to shifts by s in H and X. The patch sets S_k for a small example image with $n_p=4$ are shown in the bottom row of FIG. 3. The solution does not deviate much for other choices of patch sets, with n_p chosen as described herein.

[0097] FIG. 4 illustrates the process the engine performs to compute the various values. Each pixel, d_i, in the output downscaled image patch $P(D)$ is mapped to a set, D_i, of s^2 pixels in the

patches $P(X)$ and $P(H)$. All s^2 pixels in D_i of $P(X)$ have value d_i. Since the patches in S do not

overlap, the pixels of each patch can be optimized independently of the other patches. Hence,

for a patch, P, in S_k, the optimum patch $P^*(X)$ of the image X is given by Equation 2.

$$P^*(X) = \underset{P(X)}{\operatorname{argmin}} d(P(H), P(X)) \qquad \text{(Equ. 2)}$$

[0098] The pixels of the patches can be represented by the engine as stacks on the vectors $\mathbf{h}$

and $\mathbf{x}$. Similarly, the pixels of D that correspond to $\mathbf{x}$ can be represented with $\mathbf{d}$, and the set of

pixels in $P(X)$ that corresponds to the ith pixel in the patch in D can be represented by D_i, as

in FIG. 4. Hence, $\mathbf{x} = \mathbf{Vd}$, where the j-th $\mathbf{v}_i$ of $\mathbf{V}$ is 1 if $x_j \in D_i$, and 0 otherwise. Then, the

above computation could be expressed as in Equation 3.

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} d(\mathbf{h}, \mathbf{x}), \ \mathbf{x} = \mathbf{Vd} \qquad \text{(Equ. 3)}$$

[0099] The SSIM index is computed by multiplying three components corresponding to

luminance, contrast, and covariance based comparisons. The widely used form of SSIM is

given by [Brunet et al. 2012] as in Equation 4, where $\mu_x = \sum w_i x_i$ denotes the mean,

$\sigma_x^2 = \sum w_i (x_i - \mu_x)^2$ the variance, and $\sigma_{xh} = \sum w_i (x_i - \mu_x)(h_i - \mu_h)$ the covariance with

weights w_i, and x_i denoting the ith component of $\mathbf{x}$.

$$SSIM(\mathbf{h}, \mathbf{x}) = \frac{(2\mu_h\mu_x + c_1)(2\sigma_{xh} + c_2)}{(\mu_h^2 + \mu_x^2 + c_1)(\sigma_h^2 + \sigma_x^2 + c_2)}, \qquad \text{(Equ. 4)}$$

[00100] The values c_1 and c_2 are small constants added to avoid instability. For the

simplicity of the expressions, and since the small values used in practice do not affect results

for the downscaling problem, the constants can be set as $c_1 = c_2 = 0$. Since x_i and h_i are in

$[0,1]$, $SSIM(\mathbf{x}, \mathbf{h}) \in [0,1]$. It is 1 when $\mathbf{x} = \mathbf{h}$, and decreases as the patches become less

similar. Herein, a dissimilarity measure $d(\mathbf{h}, \mathbf{x})$ can be defined as $1 - SSIM(\mathbf{h}, \mathbf{x})$.

[00101] The $d(\cdot, \cdot)$ is not a distance function, and not even convex. Instead of directly

trying to solve the problem in Equation 3, we thus define another problem that is easy to be

solved, by parametrizing the solution to the original problem. Specifically, we fix the mean

μ_x and variance σ_x of $\mathbf{x}$ to arbitrary values, leaving only σ_{xh} as the free term in SSIM

(Equation 4). We thus optimize for σ_{xh} under these constraints to get the optimum for this

subproblem. Finally, we find the μ_x and σ_x that gives the global optimum. As detailed

below, the global optimum can be obtained by setting $\mu_x = \mu_h$, and $\sigma_x = \sigma_h$, and solving

Equation 5.

$$\underset{\mathbf{x}}{\max} \ \sigma_{xh}, \ \mu_x = \mu_h, \sigma_x = \sigma_h, \mathbf{x} = \mathbf{Vd} \qquad \text{(Equ. 5)}$$

[00102] Note that since $\mathbf{x} = \mathbf{Vd}$, the terms μ_x, σ_x, and σ_{xh} can also be expressed in terms of

$\mathbf{d}$. For example, we can write $\mu_x = \mathbf{w}^T\mathbf{x} = (\mathbf{V}^T\mathbf{w})^T\mathbf{d} = \mathbf{m}^T\mathbf{d}$ with

$\mathbf{m} = \left[\sum_{x_i \in D_1} w_i \cdots \sum_{x_i \in D_{n_p}} w_i \right]^T$. Similarly, $\sigma_x^2 = \mathbf{d}^T \mathbf{M} \mathbf{d} - \mu_x^2$ and $\sigma_{xh} = \mathbf{a}^T \mathbf{d} - \mu_x \mu_h$, where

$\mathbf{M}$ is a diagonal matrix with $M_{ii} = m_i$, and $a_i = \sum_{h_j \in D_i} w_j h_j$. With these substitutions, the

computation in Equation 5 becomes that of Equation 6, the solution of which is provided in

Equation 7 with $l_i = a_i/m_i$, and $\sigma_l^2 = \sum_{i=1}^{n_p} m_i (l_i - \mu_h)^2$.

$$\max_{\mathbf{d}} \quad \mathbf{a}^T \mathbf{d}, \quad \mathbf{m}^T \mathbf{d} = \mu_h, \quad \mathbf{d}^T \mathbf{M} \mathbf{d} = \mu_h^2 + \sigma_h^2 \qquad \text{(Equ. 6)}$$

$$d_i^* = \mu_h + \frac{\sigma_h}{\sigma_l} (l_i - \mu_h), \qquad \text{(Equ. 7)}$$

[00103]   See below for discussion of SSIM based optimization and global optimums.
Solutions of optimization problems involving the SSIM index by fixing the mean have been
utilized for other applications, where the optimum is then searched for using iterative
methods [Channappayya et al. 2008a; Ogawa and Haseyama 2013; Shao et al. 2014].
However, closed-form solutions could only be derived for simple image models
[Channappayya et al. 2006; Chai et al. 2014], or expansions on Fourier type bases [Brunet et
al. 2010]. Although the images H and D, or basis vectors $\mathbf{v}_i$ do not satisfy the properties
required for these solutions, using techniques described herein, a closed-form solution can be
derived due to the structure of the downscaling problem.

[00104]   For each pixel in the output image D, there is an optimum value from each patch
overlapping that pixel. Each of these patches belongs to a different patch set S_k. The final
value of the pixel is found by averaging these values. The weights, w_i, are usually taken from
a Gaussian or constant window [Silvestre-Blanes 2011; Brunet 2012]. Following the latter,
the weights can assumed to be uniformly summing to 1, since patches are rather small. Then,
the value for the ith pixel in image D (the i is now defined as a global index in D) is as shown
in Equation 8 where P_k denote the n_p patches overlapping this pixel.

$$d_i^* = \frac{1}{n_p} \sum_{P_k} \mu_h^k + \frac{\sigma_h^k}{\sigma_l^k} (l_i - \mu_h^k) \qquad \text{(Equ. 8)}$$

[00105]   The form of the optimum image in Equation 8 is a non-linear filter on the input
image H. The filter adapts to the image content in a perceptually optimal way as measured by
the SSIM index. The engine includes means or programming instructions for implementing
this filter. The construction of the solution makes it clear that it preserves the local
luminance and contrast of the input image H while maximizing local structural similarity.
Although the filter is non-linear, it can be implemented with a series of linear operations as
apparent from Equation 8, as described by the pseudocode presented hereinbelow.

<u>Discussion and Analysis</u>

[00106]   We can view Equation 8 as an adaptive unsharp masking filter [Polesel et al. 1997] applied to the averaged l_i values, where the sharpening factor depends non-linearly on the local image content with the ratio σ_h^k/σ_l^k of the standard deviations of the input image, and a filtered version of it. This ratio thus adaptively adjusts the filter using H as the reference image so as to preserve the local features. Unsharp masking combined with pixel-wise contrast measures extracted from a reference image has previously generated excellent results for enhancing images generated by tone mapping [Krawczyk et al. 2007] or color to greyscale conversion [Smith et al. 2008], as well as for rendered scenes [Ritschel et al. 2008]. The SSIM-optimal filter here leads to a similar term for the downscaling problem.

[00107]   FIG. 5 illustrates post-sharpening after filtering. In the top right image, sharpening results in severe ringing and fails to capture the small-scale details in the background. The Lanczos filter (middle right image) can reduce ringing but still cannot capture the details well. The method described herein (bottom right image) utilizes the local content in the input image to avoid artifacts while preserving details.

[00108]   It is well-known that trying to get sharper results by using a post-sharpening step after filtering, or a filter that generates sharper results by better approximating the sinc filter leads to artifacts when used for image downscaling [Kopf et al. 2013]. The methods described herein avoid such problems and lead to better preservation of image features. Post-sharpening after filtering leads to severe ringing on the foreground object while failing to preserve the contrast in the background. This approach is fundamentally disadvantaged since the sharpening filter cannot use information from the original high resolution image to enhance the downscaled image. The Lanczos filter reduces the artifacts, but also fails to preserve the background. The adaptivity of the derived filter in Equation 8 ensures that all features are preserved while avoiding the ringing artifacts.

[00109]   While a number of parameters might be varied, the main free parameter is the patch size, n_p. In general, determining the patch size for SSIM to best correlate the results with the response of the human visual system is a difficult problem. However, recent works confirm that as the image complexity increases, the window size should be reduced [Silvestre-Blanes 2011]. For the downscaling problem, it is crucial to capture the local structures in the input image H as well as possible. However, as the downscaling factor s increases, the patch size $s\sqrt{n_p}$ in H also gets bigger. Thus, for our problem, it is preferred to keep the patch size n_p as small as possible, such as $n_p{=}4$, for a 2 by 2 patch. A similar

14

conclusion stems from the interpretation of the filter as an adaptive unsharp mask. The smoothed image in unsharp masking, corresponding to the averaged means μ_h^k of the patches in our case, can be made smoother to capture lower frequency bands. However, many lower bands are already captured in D. Furthermore, as the patch size gets larger, the ratio of the standard deviations decrease, leading to less enhancement.

[00110]   FIG. 6 illustrates the effect of the patch size on the downscaled images. Increasing the patch size from left to right shows a loss of small scale features. In these example, from left to right, the patch sizes are 2^2, 8^2, and 32^2. As the patch size increases, small scale features are lost. In the limit that the whole image is covered by one patch, the downscaled image approaches the filtered image given by l_t, since the contrasts σ_h and σ_l can be matched almost exactly.

[00111]   Since the values of the pixels in D are not constrained to lie in $[0, 1]$ in the optimization, some pixels might end up having values outside this dynamic range. However, since the mean and standard deviations match for the optimum solution, in practice, the percentage of these pixels and their distance to the dynamic range is negligible for natural images.

[00112]   FIG. 7 illustrates this point. For the upper plots, the percentage of pixel values outside the dynamic range for 3000 random natural images for seven different sizes is shown. The lower plots show that for each of the input images and sizes, the mean SSIM index and mean standard deviation between the downscaled image generated using all S_k by averaging and those generated using individual S_k's, are computed. FIG. 7 shows the histograms of these values over the same set of images and sizes as in the top plots. Both measures show that optimizing over different sets does not alter the solution significantly. Working with a small patch size of 2 by 2, the choice of the patch sets does not lead to a noticeable difference. The resulting optimized images for different patch sets S_k and their mean (the SSIM-optimal image) are almost identical. In FIG. 7, the distributions of the mean SSIM indices and mean standard deviations are shown computed between the mean image (our solution) and the images optimized over different S_k's, for the same set of 3000 images and seven sizes as above. Both measures indicate that the resulting images are almost identical.

[00113]   FIG. 8 illustrates an example image optimized over different S_k. The images are almost identical and differ slightly in some of the patches where the texture has large and high frequency variations.

[00114]   For some of the patches, the intensities l_i can be constant such that we get $\sigma j = 0$. For these cases, there might be no way to match the contrast, as required by the solution, and only the mean can be matched. Hence, for a patch with $\sigma_l < 10^{-6}$, we set the values of the pixels of the downscaled image in this patch to the mean μ_h of the patch.

[00115]   SSIM is defined for images with a single channel, although some works explore utilizing extracted features [Lissner et al. 2013], or working in various color spaces [Bonnier et al. 2006]. The engine can use the RGB space for all image processing and apply the downscaling to each channel independently.

Results

[00116]   We performed a large number of experiments to validate the practical value of our method with thousands of images and many different downscaling factors, a detailed analysis, comparisons to existing methods, and a formal user study.

Downscaling Results and Analysis

[00117]   Our technique generates local pixel patterns that form structures resembling those in the input image, when viewed by a human observer. This effect is most apparent when there are perceptually important features (as in FIGs. 1, 10), textures (as in FIGs. 15, 16), or other small-scale details (as in FIGs. 1, 2, 15, 16, 17) in the input images. While trying to capture as much structure as possible, it also preserves the local contrast and luminance of the input image, which makes the overall look of the downscaled image close to the input (e.g., FIGs. 1, 16).

[00118]   The downscaling process performed by the engine does not significantly alter the features that are already captured by low-pass filters. This results in less jagged edge artifacts than previous downscaling methods. For example, FIG. 9 illustrates a downscaled edge of a picture frame. The input images on the left and on the right are four output images. From top to bottom of the right side of FIG. 9, the output images are done by the original image, bicubic filtering, content-adaptive downscaling, and our process, respectively. Our process preserves the details better while leading to less jaggy edge effects. Our method performs a slight enhancement on the edge, resulting in fewer artifacts than with the content-adaptive method. If some details cannot be captured with the pixel budget in the downscaled image, they are mapped to noise-like structures that resemble those in the input image if viewed at the native resolution, as opposed to Moiré patterns, as with subsampling.

[00119]   FIG. 11 illustrates this, with the left image being bicubic filtering, the middle image being subsampling (to Moiré patterns showing), and our result without Moiré patterns.

The method is also spatio-temporally consistent, leading to accurate representation of features, as can be clearly seen in FIGs. 1, right, and FIG. 10. Classical filtering methods such as bicubic filtering are also consistent, but fail to generate crisp images. Aligning the kernels to local image features [Kopf et al. 2013] can generate crisper results, but the resulting

5   kernels can miss or distort some features as in FIG. 10, and small changes in input images are sometimes amplified, leading to flickering, as might have to be dealt with when downscaling video. In the sets of three images in the center of FIG. 10, the top is the original image, the middle is content-adaptive downscaling [Kopf et al. 2013], and the bottom is our result. The features are kept intact with our method.

10   [00120]   There are numerous studies on the correlation of the SSIM index with human perception when used as an image quality measure [Wang and Bovik 2009]. However, our particular problem of downscaling called for a tailored formal user study. The design of our user study follows that of the previous study performed by Kopf et al. [2013], including the images used and all design choices.

15   [00121]   The study is based on presenting the participants a large image and two downscaled versions of that image. The participant is then asked to select the small image that she/he thinks represents a better downscaled version of the large image, or indicate no preference. One of the small images presented for each test is computed using our process described herein and the other by a different process, such as subsampling, the classical box,

20   bicubic, Lanczos filtering, bilateral filtering, generalized sampling [Nehab and Hoppe 2011], and content-adaptive downscaling [Kopf et al. 2013]. There were 125 participants in the study.

        [00122]   The 13 natural images used in the study, originally from the MSRA Salient Object Database [Liu et al. 2011], are the same as the ones used in the previous study [Kopf et al.

25   2013]. We show some example results in FIG. 16. They cover a variety of scenes with different types and scales of structures. The images were shown at the native resolution of the display, and zooming was not provided. The long side of the large images is 400 pixels, and that of the small images is 128 pixels. The study was performed online with participants from different parts of the world, educational backgrounds, occupations, and computer experience.

30   Similar to the previous study [Kopf et al. 2013], we allowed the participants to move closer to the screen if they would like to, as would happen in practice for real-world situations. Each test for a particular participant involved a different image, and was repeated twice to check for consistency. All the results coming from subjects with consistency lower than 80% were

discarded [Kopf et al. 2013], leaving results from 64 participants (the results do not change significantly for other rejection rates). There was no time limit to finish the study.

[00123]   FIG. 12 illustrates user study results. In each group of three bars, the left bars represent how many times a user selected the downscaled image done by our process, the
5   middle bars represent how many times the user indicated no preference, and the right bars represent how many times the user indicates a preference for the other process. The study showed a clear preference for the results of our process against competing methods. The best competing downscaling method is simple subsampling, which was also the case for the previous study [Kopf et al. 2013]. Since subsampling does not involve any filtering, it
10   preserves the crisp look of the images well, of course at the cost of well-known strong aliasing artifacts. For the user study images where these artifacts are not visible, the participants could not decide which image to choose. For other images where the artifacts are noticeable, there is a clear preference for our images. Hence, our process preserves the crisp look of the images as in subsampling, but without the visible aliasing artifacts.

15   Implementation and Performance

[00124]   The methods here can be based on a non-linear filter on the input image and can be implemented very efficiently and robustly with simple convolutions and sums.

[00125]   Pseudocode for a process is provided further below. This process was implemented in Matlab with native Matlab operators, some of which use multiple CPU cores.
20   We performed a performance test with 100 randomly chosen images on a computer with the configuration Intel Core i7 3770K CPU @350GHz. The method of Kopf et al. [2013] was run as a native executable. The results of the test are reported in FIG. 13 for different input image sizes (with output image size fixed to 80 by 60), and output sizes (with input image size 640 by 480).

25   [00126]   Our process is only a few times slower than the box filter we used in the implementation of our algorithm, and 500 to 5000 times faster than the method of Kopf et al. [2013] that relies on an iterative expectation-maximization based optimization. In this test, the engine ran two box filterings followed by subsampling on the input image, and further operations on images of size proportional to the output image as can be seen in the
30   pseudocode. For smaller output sizes relative to the input size, it performs closer to the initial box filter we used, while increasing the output size slows it down a few times, as can be seen in FIG. 13, right.

Variations

[00127]   Other variations might address the indifference to scene semantics. Seeing local structures in an image without any reference to what they actually represent may lead to preservation of undesired details such as noise present in the input image, as we show in FIG. 14, which is smoothed out by non-adaptive filters. In FIG. 14, the inserts, from left to right, are the original image, bicubic filtering, and our result. Since our method lacks scene semantics, it tries to preserve the noise in the input image.

[00128]   Our results exhibit fewer jagged edges (FIG. 9) and aliasing artifacts (FIG. 11) than methods that generate crisp images. However, if the image contains very regular repeating structures with a high frequency, aliasing can happen. The SSIM index tends to not prefer patches with a constant value, since this makes the index 0. Instead, our algorithm tries to reproduce the local contrast and structure. However, for perfectly regular structures, a constant patch value might be preferred instead. For those cases, such as on standard aliasing tests, we can get artifacts similar to those produced by previous enhancement methods [Kopf et al. 2013]. Fortunately, such regular structures are rarely present in natural images. We observed that the small perturbations to regular structures that exist in most natural images can break the artifacts, as in FIG. 11.

[00129]   The SSIM index is known to not preserve the blur in the images [Chen et al. 2006]. We also observed that as opposed to thumbnail generation methods, our downscaling results do not contain the same amount of blur in the input image, especially for high downscaling ratios. We experimented with an extension of SSIM in the gradient domain, by solving for the gradients of the downscaled image, and subsequently a Poisson equation to get the actual image and with some additional steps, this might work.

Additional Variations

[00130]   We used the basic form of the SSIM index. There are numerous extensions that modify the local similarity measure, the patch averaging stage, or extend it to feature and color spaces. Although the downscaled videos exhibit less flickering due to the consistency of the filter, better downscaling results can be obtained by incorporating extensions of the SSIM index to videos, e.g. models of speed perception [Wang and Li 2007]. Other perceptual measures might be be utilized to improve image scaling results.

[00131]   The SSIM index sees the image at the level of patches, and cannot by itself adapt to scene semantics. This leads to problems such as the noise amplification in FIG. 14. Scene semantics such as background/foreground separation, properties of the objects in the scene, or

saliency maps can be integrated into our algorithm by adaptively weighting the patches, or adjusting the parameters (α, γ) and patch size locally.

<u>Conclusions</u>

[00132]   A novel method for image downscaling is provided that aims to optimize for the perceptual quality of the downscaled results. Extensive tests involving hundreds of images and the user study clearly indicate that it generates perceptually accurate and appealing downscaling results, outperforming previous techniques. Despite its effectiveness and non-linear nature, it has a very simple, robust, efficient, and parallelizable implementation, making the algorithm a practical addition to the arsenal of image filters.

[00133]   FIG. 15 illustrates that the process of downscaling described here is able to capture small-scale details and textures while preserving local contrast and luminance to produce a perceptually accurate downscaled image. FIG. 16 illustrates examples of images used for the user study. For each of FIGs. 16A and 16B, the original image is on the left, and the four smaller images on the right are subsampling (top-left), bicubic filtering (top-right), content-adaptive downscaling (bottom left), and our perceptual downscaling (bottom-right).

[00134]   The mean SSIM(X,Y) computed over two images X and Y is a metric that measures the similarity between the two images. The higher the value of the mean SSIM, the more similar the two images are. Mean SSIM has been shown to correlate well with human perception, meaning that when mean SSIM(X,Y) is high, humans perceive X and Y as very similar images and when mean SSIM(X,Y) is low, humans perceive X and Y as dissimilar images. Mean SSIM has been used for some image processing tasks. It is in general computationally demanding to optimize for an image X, given an input image Y, by maximizing SSIM(X,Y). The function SSIM(X,Y) can be defined between two corresponding image patches, one from X, and the other from Y. This function can then be averaged over the images to get the mean SSIM(X,Y).

[00135]   For downscaling, simpler metrics such as the least squares norm, i.e., $||X - Y||^2$, for some representation of the images, have been used to measure the difference between the images X and Y. A familiar example is the "bicubic filter", which generates a smooth downscaled image by removing the details in the original high resolution image. As explained herein, measuring the difference between the high resolution image H and the downscaled image D using SSIM can provide better results.

[00136]   As an example, consider an input high resolution image, H, comprising 1000 x 1000 pixels and an output downscaled image, D, comprising 100 x 100 pixels. From D, an upscaled D, called X (1000 x 1000 pixels) is generated for use in calculating SSIM values. In

X, each pixel of D is repeated in a 10 x 10 area in X. This is illustrated in FIG. 3, top row, and FIG. 4. Then, for each patch pair (patch(H), patch(X)), with a patch from X and the corresponding patch from H, the image processor will attempt to maximize the value of SSIM(patch(H),patch(X)) by changing the pixel values in patch(X), with the constraint that

5   each 10 x 10 area in X should have the same pixel value (which corresponds to a single pixel value in D).

[00137]   Normally, this is a computationally demanding and complex optimization, but using the techniques presented herein, a closed-form solution can be derived in various ways, such as by matching the means and standard deviations, and maximizing the covariance, as

10   illustrated in part by Equation 5. Equation 7 illustrates a solution. The image processor does this for all patches in the images X and H. The set of patches can be divided into sets S_i, with non-overlapping patches, as in FIG. 3, bottom row. Since a pixel in D only belongs to a single patch in S_i, its value can optimized only over the unique patch that it belongs to in S_i. This gives an optimum downscaled image D for this S_i. Finally, we average over all the

15   resulting Ds optimized over different S_is to treat all patches equally. The result of this averaging, and hence the final value for a pixel is in Equation 8. If the patch sizes are kept as small as possible (e.g., 2 x 2 in D, and hence 2s x 2s in H and X), details are preserved well.

[00138]   FIG. 17 illustrates examples of our downscaling method adaptively adjusting local details such that downscaled images perceptually close to the original image are generated.

20   <u>SSIM based Optimization and Global Optimums</u>

[00139]   We parameterize the solution of the optimization problem by setting $\mu_x = \alpha\mu_h$, and $\sigma_x = \gamma\sigma_h$, for arbitrary (α, γ). Then, to maximize $SSIM(\mathbf{h}, \mathbf{x})$ for this particular (α, γ), maximize σ_{xh}. This leads to the following constrained optimization problem of Equation 9.

$$\max_{\mathbf{d}} \quad \mathbf{a}^T\mathbf{d}, \quad \mathbf{m}^T\mathbf{d} = \alpha\mu_h, \quad \mathbf{d}^T\mathbf{M}\mathbf{d} = \alpha^2\mu_h^2 + \gamma^2\sigma_h^2 \qquad \text{(Equ. 9)}$$

25   [00140]   This problem can be solved by standard methods, such as the method of Lagrange multipliers as we show below. The solution is given by Equation 10.

$$d_i^*(\alpha, \gamma) = \alpha\mu_h + \gamma\frac{\sigma_h}{\sigma_l}(l_i - \mu_h). \qquad \text{(Equ. 10)}$$

[00141]   For each (α, γ), the $\mathbf{d}^*$ with the components d_i^* thus maximizes the covariance σ_{hx} and hence SSIM. If we plug in this expression for d_i^* into the expression for SSIM in

30   Equation 4, we get the following maximum SSIM.

$$SSIM(\mathbf{h}, \mathbf{d}^*(\alpha, \gamma)) = 4\frac{\sigma_l}{\sigma_h}\frac{\alpha\gamma}{(1+\alpha^2)(1+\gamma^2)} \qquad \text{(Equ. 11)}$$

[00142]   This expression is maximized if we select $\alpha = \gamma = 1$, giving us the global

optimum $\mathbf{d}^*$. Hence, the solution of the problem in Equation 9 with the choice $(\alpha, \gamma) = (1,1)$

coincides with the solution of the original problem in Equation 3.

[00143]   For simplicity of the equations, we make the following definitions $\mathbf{e} := \mathbf{M}^{1/2}\mathbf{d}$,

$\mathbf{b} := \mathbf{M}^{-1/2}\mathbf{m}$, $c^2 := \alpha^2\mu_h^2 + \gamma^2\sigma_h^2$, $\mathbf{f} := \mathbf{M}^{-1/2}\mathbf{a}$. Then, the problem in Equation 5 above can

be rewritten as in Equation 12.

$$\max_{\mathbf{e}} \quad \mathbf{f}^T\mathbf{e}, \mathbf{b}^T\mathbf{e} = \alpha\mu_h, \quad ||\mathbf{e}||^2 = c^2 \tag{Equ. 12}$$

[00144]   We solve this problem with the method of Lagrange multipliers. Hence, we

optimize the function of Equation 13.

$$F(\mathbf{e}, \lambda_1, \lambda_2) = \mathbf{f}^T\mathbf{e} - \lambda_1(\mathbf{b}^T\mathbf{e} - \alpha\mu_h) - \lambda_2(||\mathbf{e}||^2 - c^2) \tag{Equ. 13}$$

[00145]   Taking the derivatives with respect to $\mathbf{e}$, λ_1, and λ_2 gives us Equations 14-16.

$$\mathbf{e} = \frac{-\mathbf{f} - \lambda_1\mathbf{b}}{2\lambda_2} \tag{Equ. 14}$$

$$-(\mu_h + \lambda_1) = 2\alpha\mu_h\lambda_2 \tag{Equ. 15}$$

$$\mathbf{a}^T\mathbf{1} + 2\lambda_1\mu_h + \lambda_1^2 = 4c^2\lambda_2^2. \tag{Equ. 16}$$

[00146]   Combining the last two equations, we can solve for λ_1 and λ_2 as in Equation 17.

$$\lambda_1 = \frac{-\mu_h \pm \alpha\mu_h\sqrt{\mathbf{a}^T\mathbf{1} - \mu_h^2}}{\gamma\sigma_h} \tag{Equ. 17}$$

$$\lambda_2 = \mp\frac{1}{2}\frac{\sqrt{\mathbf{a}^T\mathbf{1} - \mu_h^2}}{\gamma\sigma_h}. \tag{Equ. 18}$$

[00147]   Substituting these into the expression for $\mathbf{e}$ gives us

$$\mathbf{e} = \frac{-\mathbf{f} - (-\mu_h \pm \frac{\alpha\mu_h\sigma_l}{\gamma\sigma_h})\mathbf{b}}{\mp\frac{\sigma_l}{\gamma\sigma_h}}. \tag{Equ. 19}$$

[00148]   Hence, we get the solution of Equation 20 where $\mathbf{1}$ denotes the vector of ones.

$$\mathbf{d} = \alpha\mu_h\mathbf{1} \pm \frac{\gamma\sigma_h}{\sigma_l}(\mathbf{1} - \mu_h\mathbf{1}) \tag{Equ. 20}$$

[00149]   In order to decide on the sign, maximize the covariance and hence $\mathbf{a}^T\mathbf{d}$.

Substituting the expression for $\mathbf{d}$, we can see that this dot product is maximized for the

positive sign.

Pseudocode for Operations

[00150]   In the algorithm below, implementable in hardware and/or software, operations are

element-wise on the single channel images, denoted with big letters. The function

convValid($X, P(y)$) convolves image X with an averaging filter of size y by y for the valid

range of the image such that the kernel stays within the image limits. The function convFull

is similar but the image is assumed to be padded with zeros to allow the kernel go out of the image limits. The function subSample(X, y) subsamples image X at intervals of y, I_X produces an image of the size of X with all ones, $X(C)$ gets all entries of the image X for which the corresponding entry in the image C returns true, and $\varepsilon = 10^{-6}$. The inputs to the process are an input image H, a downscaling factor s and a patch size n_p. The output is a downscaled image D. The steps are:

[00151]   1: $L \leftarrow$ subSample(convValid($H, P(s)$), s)

[00152]   2: $L_2 \leftarrow$ subSample$\left(\text{convValid}\left(H^2, P(s)\right), s\right)$

[00153]   3: $M \leftarrow$ convValid($L, P(\sqrt{n_p})$)

[00154]   4: $S_l \leftarrow$ convValid$\left(L^2, P\left(\sqrt{n_p}\right)\right) - M^2$

[00155]   5: $S_h \leftarrow$ convValid($L_2, P(\sqrt{n_p})$) $- M^2$

[00156]   6: $R \leftarrow \sqrt{S_h/S_l}$

[00157]   7: $(S_l < \varepsilon) \leftarrow 0$

[00158]   8: $N \leftarrow$ convFull($I_M, P(\sqrt{n_p})$)

[00159]   9: $T \leftarrow$ convFull($R \times M, P(\sqrt{n_p})$)

[00160]   10: $M \leftarrow$ convFull$\left(M, P\left(\sqrt{n_p}\right)\right)$

[00161]   11: $R \leftarrow$ convFull$\left(R, P\left(\sqrt{n_p}\right)\right)$

[00162]   12: $D \leftarrow (M + R \times L - T)/N$

[00163]   According to one embodiment, the techniques described herein are implemented by one or generalized computing systems programmed to perform the techniques pursuant to program instructions in firmware, memory, other storage, or a combination. Special-purpose computing devices may be used, such as desktop computer systems, portable computer systems, handheld devices, networking devices or any other device that incorporates hard-wired and/or program logic to implement the techniques.

[00164]   For example, FIG. 18 is a block diagram that illustrates a computer system 1800 upon which an embodiment of the invention may be implemented. Computer system 1800 includes a bus 1802 or other communication mechanism for communicating information, and a processor 1804 coupled with bus 1802 for processing information. Processor 1804 may be, for example, a general purpose microprocessor.

[00165] Computer system 1800 also includes a main memory 1806, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 1802 for storing information and instructions to be executed by processor 1804. Main memory 1806 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 1804. Such instructions, when stored in non-transitory storage media accessible to processor 1804, render computer system 1800 into a special-purpose machine that is customized to perform the operations specified in the instructions.

[00166] Computer system 1800 further includes a read only memory (ROM) 1808 or other static storage device coupled to bus 1802 for storing static information and instructions for processor 1804. A storage device 1810, such as a magnetic disk or optical disk, is provided and coupled to bus 1802 for storing information and instructions.

[00167] Computer system 1800 may be coupled via bus 1802 to a display 1812, such as a computer monitor, for displaying information to a computer user. An input device 1814, including alphanumeric and other keys, is coupled to bus 1802 for communicating information and command selections to processor 1804. Another type of user input device is cursor control 1816, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 1804 and for controlling cursor movement on display 1812. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

[00168] Computer system 1800 may implement the techniques described herein using customized hard-wired logic, one or more ASICs or FPGAs, firmware and/or program logic which in combination with the computer system causes or programs computer system 1800 to be a special-purpose machine. According to one embodiment, the techniques herein are performed by computer system 1800 in response to processor 1804 executing one or more sequences of one or more instructions contained in main memory 1806. Such instructions may be read into main memory 1806 from another storage medium, such as storage device 1810. Execution of the sequences of instructions contained in main memory 1806 causes processor 1804 to perform the process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions.

[00169] The term "storage media" as used herein refers to any non-transitory media that store data and/or instructions that cause a machine to operation in a specific fashion. Such storage media may comprise non-volatile media and/or volatile media. Non-volatile media

includes, for example, optical or magnetic disks, such as storage device 1810. Volatile media includes dynamic memory, such as main memory 1806. Common forms of storage media include, for example, a floppy disk, a flexible disk, hard disk, solid state drive, magnetic tape, or any other magnetic data storage medium, a CD-ROM, any other optical data storage

5   medium, any physical medium with patterns of holes, a RAM, a PROM, an EPROM, a FLASH-EPROM, NVRAM, any other memory chip or cartridge.

[00170]   Storage media is distinct from but may be used in conjunction with transmission media. Transmission media participates in transferring information between storage media. For example, transmission media includes coaxial cables, copper wire and fiber optics,

10   including the wires that comprise bus 1802. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[00171]   Various forms of media may be involved in carrying one or more sequences of one or more instructions to processor 1804 for execution. For example, the instructions may

15   initially be carried on a magnetic disk or solid state drive of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a network connection. A modem or network interface local to computer system 1800 can receive the data. Bus 1802 carries the data to main memory 1806, from which processor 1804 retrieves and executes the instructions. The instructions received by main memory

20   1806 may optionally be stored on storage device 1810 either before or after execution by processor 1804.

[00172]   Computer system 1800 also includes a communication interface 1818 coupled to bus 1802. Communication interface 1818 provides a two-way data communication coupling to a network link 1820 that is connected to a local network 1822. For example,

25   communication interface 1818 may be an integrated services digital network (ISDN) card, cable modem, satellite modem, or a modem to provide a data communication connection to a corresponding type of telephone line. Wireless links may also be implemented. In any such implementation, communication interface 1818 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

30   [00173]   Network link 1820 typically provides data communication through one or more networks to other data devices. For example, network link 1820 may provide a connection through local network 1822 to a host computer 1824 or to data equipment operated by an Internet Service Provider (ISP) 1826. ISP 1826 in turn provides data communication services through the world wide packet data communication network now commonly referred to as

the "Internet" 1828. Local network 1822 and Internet 1828 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 1820 and through communication interface 1818, which carry the digital data to and from computer system 1800, are example forms of transmission media.

[00174]   Computer system 1800 can send messages and receive data, including program code, through the network(s), network link 1820 and communication interface 1818. In the Internet example, a server 1830 might transmit a requested code for an application program through Internet 1828, ISP 1826, local network 1822 and communication interface 1818. The received code may be executed by processor 1804 as it is received, and/or stored in storage device 1810, or other non-volatile storage for later execution.

[00175]   FIG. 19 is a simplified functional block diagram of a storage device 1948 having an application that can be accessed and executed by a processor in a computer system. The application can one or more of the applications described herein, running on servers, clients or other platforms or devices. Storage device 1948 can be one or more memory devices that can be accessed by a processor and storage device 1948 can have stored thereon application code 1950 that can be configured to store one or more processor readable instructions. The application code 1950 can include application logic 1952, library functions 1954, and file I/O functions 1956 associated with the application.

[00176]   Storage device 1948 can also include application variables 1962 that can include one or more storage locations configured to receive input variables 1964. The application variables 1962 can include variables that are generated by the application or otherwise local to the application. The application variables 1962 can be generated, for example, from data retrieved from an external source, such as a user or an external device or application. The processor can execute the application code 1950 to generate the application variables 1962 provided to storage device 1948.

[00177]   One or more memory locations can be configured to store device data 1966. Device data 1966 can include data that is sourced by an external source, such as a user or an external device. Device data 1966 can include, for example, records being passed between servers prior to being transmitted or after being received.

[00178]   Storage device 1948 can also include a log file 1980 having one or more storage locations 1984 configured to store results of the application or inputs provided to the application. For example, the log file 1980 can be configured to store a history of actions.

[00179]   FIG. 20 illustrates an example of a downscaling engine 2002 that takes in an input image file 2004 and outputs an output image file 2006 using the processes described herein. Internal image storage 2008 is used to hold image data while being processed and program code 2010 represents program instructions to perform the downscaling described herein.

[00180]   Operations of processes described herein can be performed in any suitable order unless otherwise indicated herein or otherwise clearly contradicted by context. Processes described herein (or variations and/or combinations thereof) may be performed under the control of one or more computer systems configured with executable instructions and may be implemented as code (e.g., executable instructions, one or more computer programs or one or more applications) executing collectively on one or more processors, by hardware or combinations thereof. The code may be stored on a computer-readable storage medium, for example, in the form of a computer program comprising a plurality of instructions executable by one or more processors. The computer-readable storage medium may be non-transitory.

[00181]   Conjunctive language, such as phrases of the form "at least one of A, B, and C," or "at least one of A, B and C," unless specifically stated otherwise or otherwise clearly contradicted by context, is otherwise understood with the context as used in general to present that an item, term, etc., may be either A or B or C, or any nonempty subset of the set of A and B and C. For instance, in the illustrative example of a set having three members, the conjunctive phrases "at least one of A, B, and C" and "at least one of A, B and C" refer to any of the following sets: {A}, {B}, {C}, {A, B}, {A, C}, {B, C}, {A, B, C}. Thus, such conjunctive language is not generally intended to imply that certain embodiments require at least one of A, at least one of B and at least one of C each to be present.

[00182]   The use of any and all examples, or exemplary language (e.g., "such as") provided herein, is intended merely to better illuminate embodiments of the invention and does not pose a limitation on the scope of the invention unless otherwise claimed. No language in the specification should be construed as indicating any non-claimed element as essential to the practice of the invention.

[00183]   In the foregoing specification, embodiments of the invention have been described with reference to numerous specific details that may vary from implementation to implementation. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. The sole and exclusive indicator of the scope of the invention, and what is intended by the applicants to be the scope of the invention, is the literal and equivalent scope of the set of claims that issue from this application, in the specific form in which such claims issue, including any subsequent correction.

[00184]   Further embodiments can be envisioned to one of ordinary skill in the art after reading this disclosure. In other embodiments, combinations or sub-combinations of the above-disclosed invention can be advantageously made. The example arrangements of components are shown for purposes of illustration and it should be understood that combinations, additions, re-arrangements, and the like are contemplated in alternative embodiments of the present invention. Thus, while the invention has been described with respect to exemplary embodiments, one skilled in the art will recognize that numerous modifications are possible.

[00185]   For example, the processes described herein may be implemented using hardware components, software components, and/or any combination thereof. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. It will, however, be evident that various modifications and changes may be made thereunto without departing from the broader spirit and scope of the invention as set forth in the claims and that the invention is intended to cover all modifications and equivalents within the scope of the following claims.

[00186]   All references, including publications, patent applications, and patents, cited herein are hereby incorporated by reference to the same extent as if each reference were individually and specifically indicated to be incorporated by reference and were set forth in its entirety herein.

WHAT IS CLAIMED IS:

1. A method of downscaling images, the method comprising:

receiving a first image defined by a first set of pixels establishing a first image size at a
    first resolution;

generating a second image defined by a second set of pixels establishing a second image
    size at a second resolution by downscaling the first image to form the second image,
    wherein values of the second set of pixels are defined by a function of the first set of
    pixels and wherein the second image size is smaller than a first image size;

upscaling the second image to a third image, wherein the third image is defined by a
    third set of pixels derived from the second set of pixels establishing a third image
    size substantially the same size as the first image size with a third resolution;

associating individual pixels in the second image with a corresponding group of pixels
    from the third set of pixels;

sampling a first image area at a first location of the first set of pixels to generate a first
    image sample;

sampling a second image area of the third set of pixels corresponding to the first image
    area location and size to generate a second image sample;

measuring the similarity between the first image area and the second image area by
    processing the first image sample and the second image sample to generate a
    perceptual image value;

recursively adjusting the values of third set of pixels until the image perception value
    matches a perceptual standard value within a pre-defined threshold; and

adjusting individual pixel values in the second image to a representative pixel value of
    each of the corresponding group of pixels.

2. The method of claim 1, implemented using a computer system that has
inputs for receiving an electronically-readable representation of the first image and outputs
for outputting an electronically-readable representation of the second image, and a processor
with program instructions stored in memory for processing image data according to the
method.
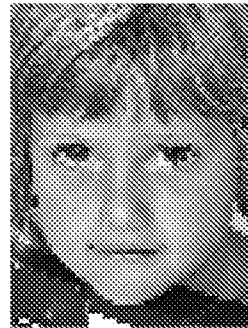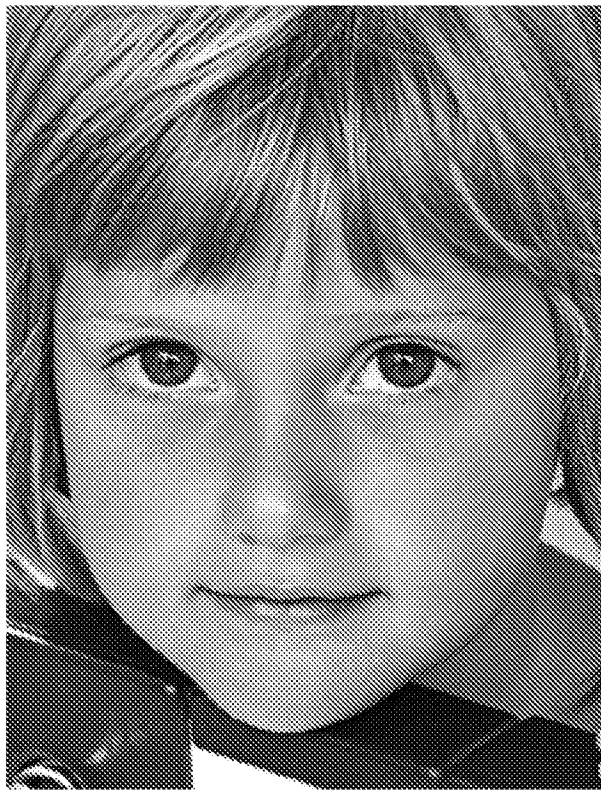
Subsampling   Bicubic
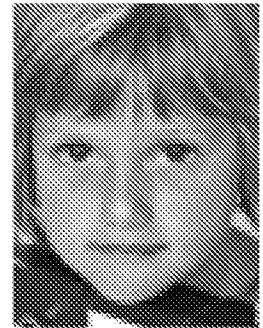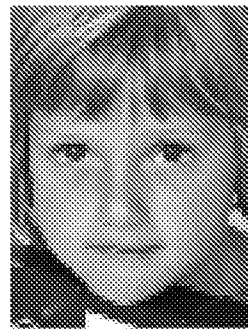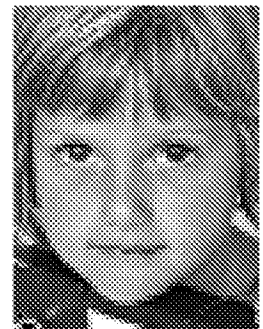
Content-Adaptive   Perceptual

FIG. 1A
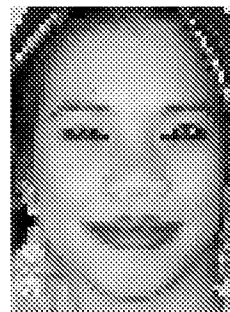
Subsampling   Bicubic
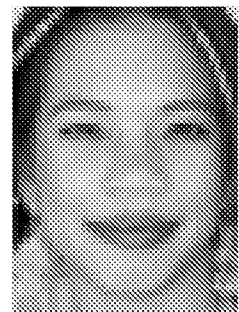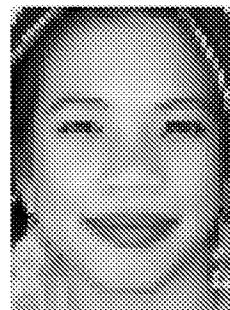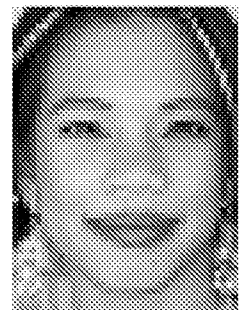
Content-Adaptive   Perceptual

FIG. 1B

FIG. 2A
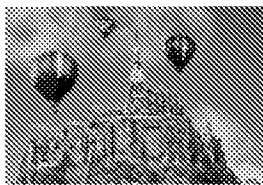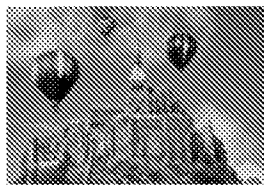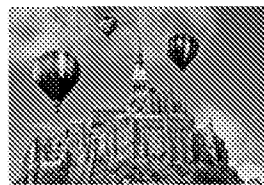
Subsampling   Bicubic   Sharpened   Lanczos
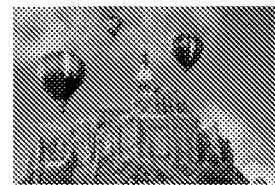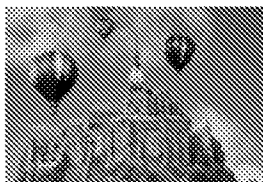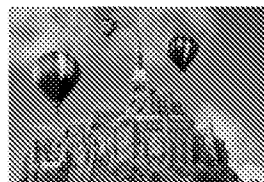
Box   Generalized   Content-Adaptive   Perceptual
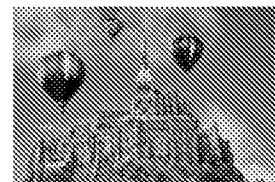
FIG. 2B
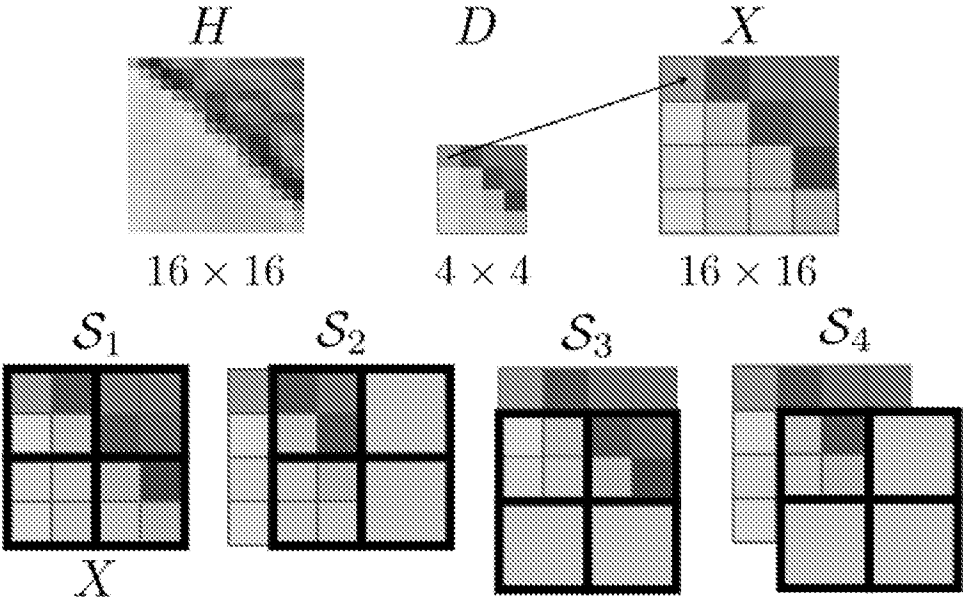
FIG. 3
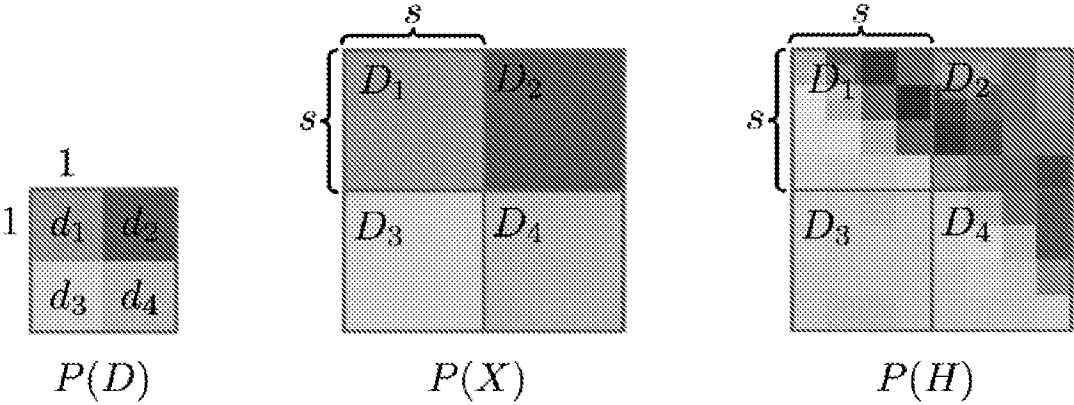
FIG. 4

FIG. 5

FIG. 6
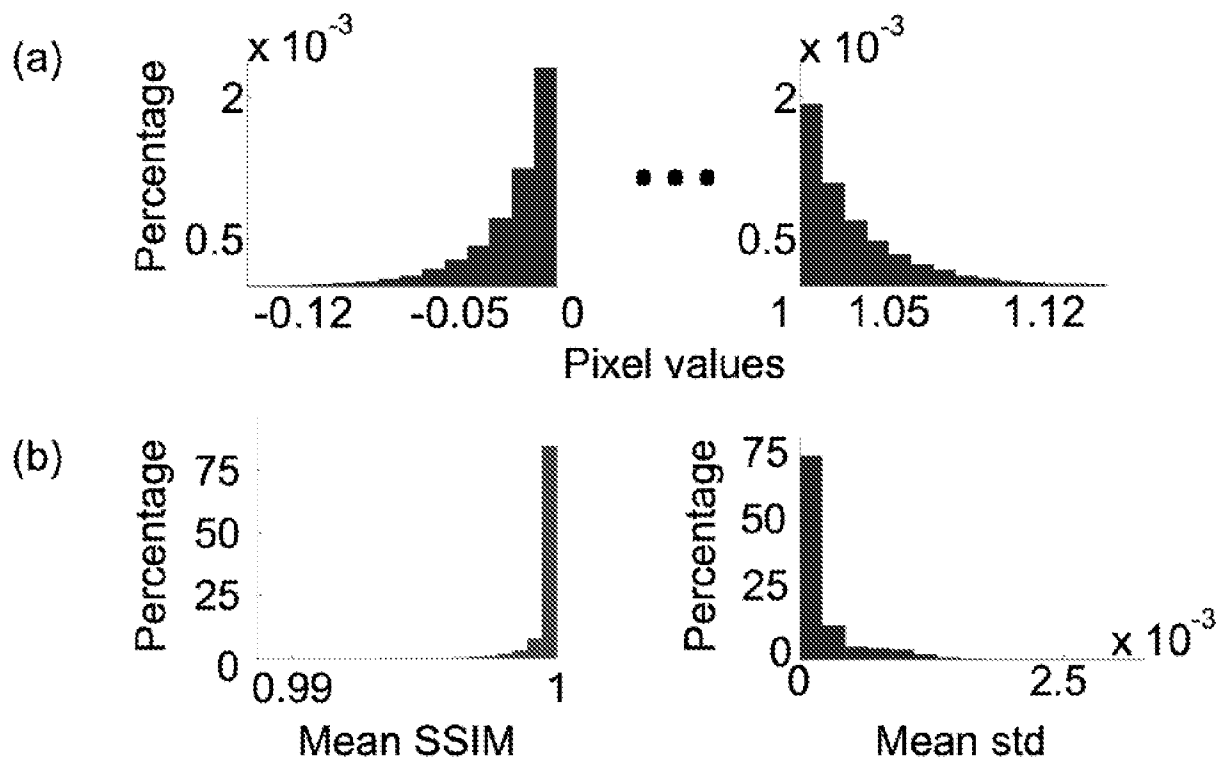
FIG. 7
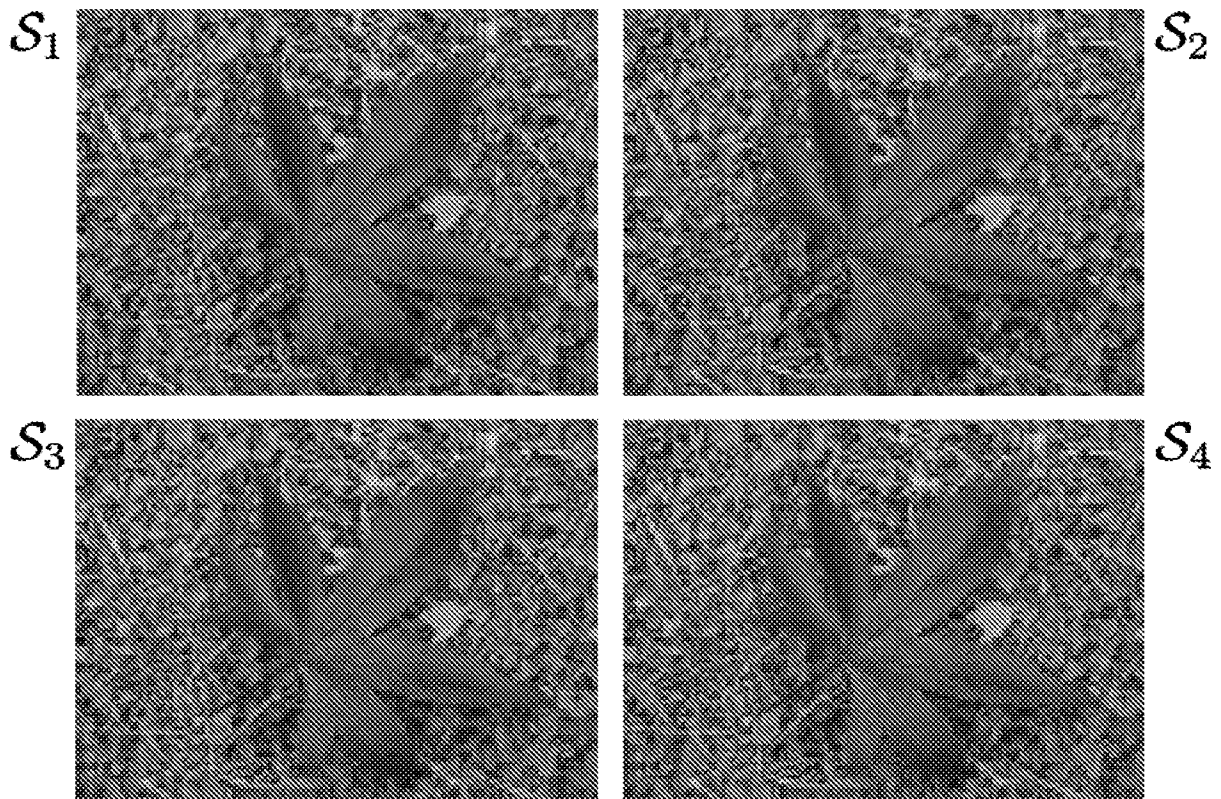
FIG. 8

FIG. 9
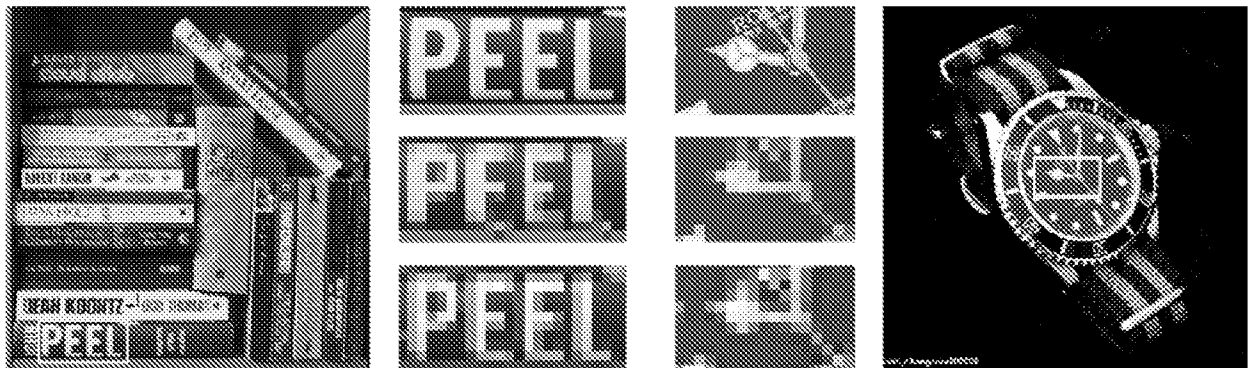
FIG. 10
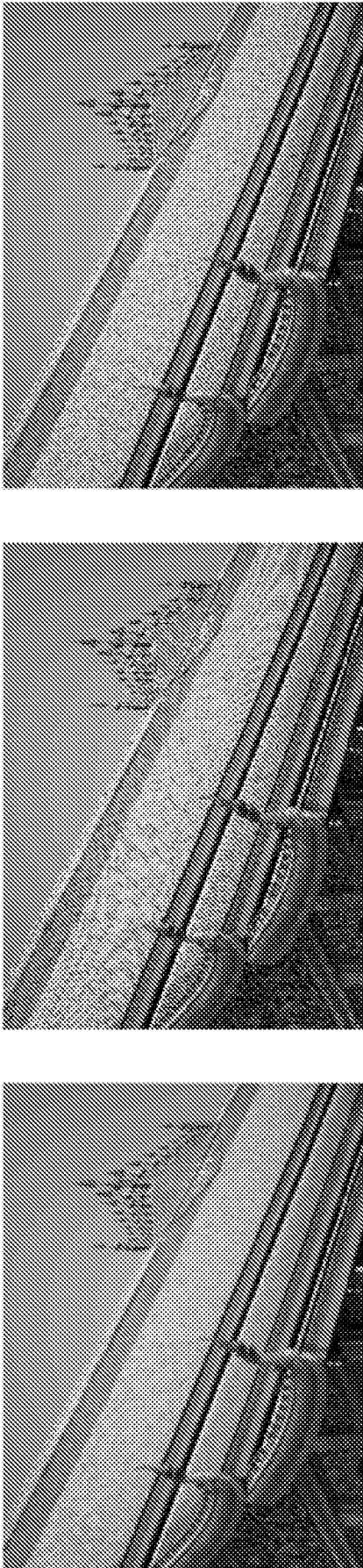
FIG. 11
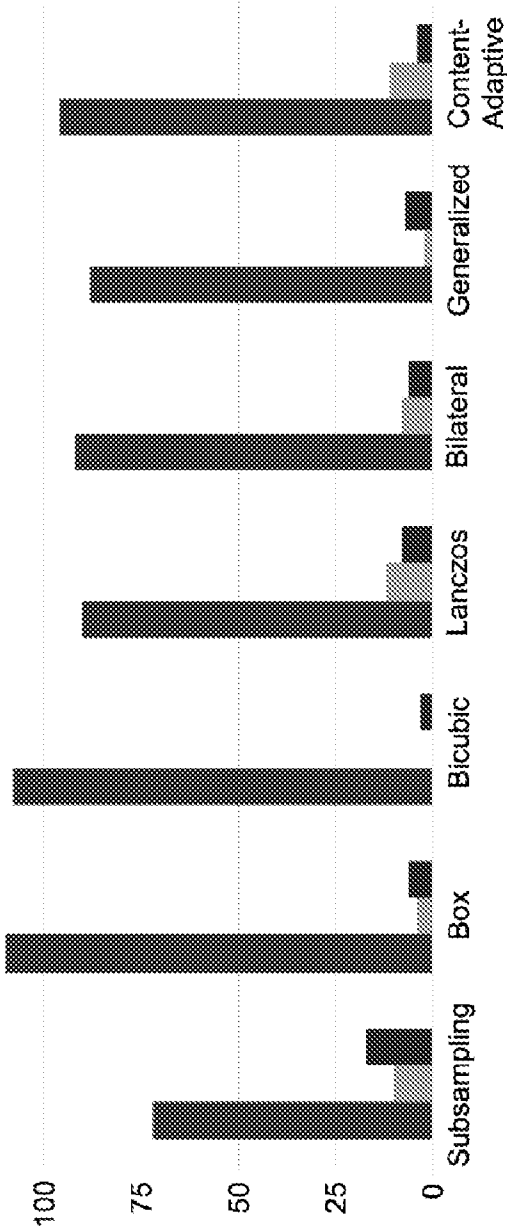
FIG. 12
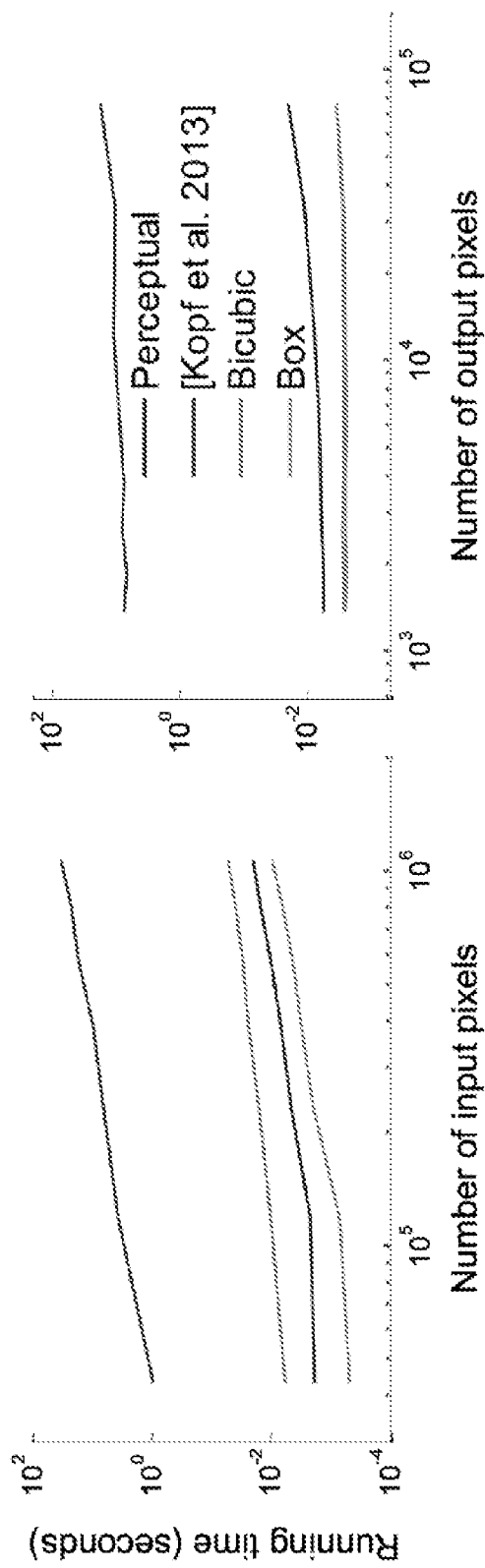
FIG. 13
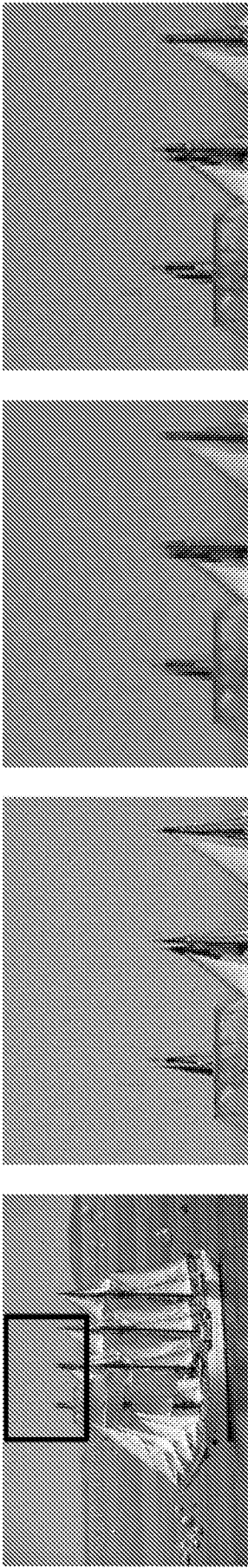
FIG. 14
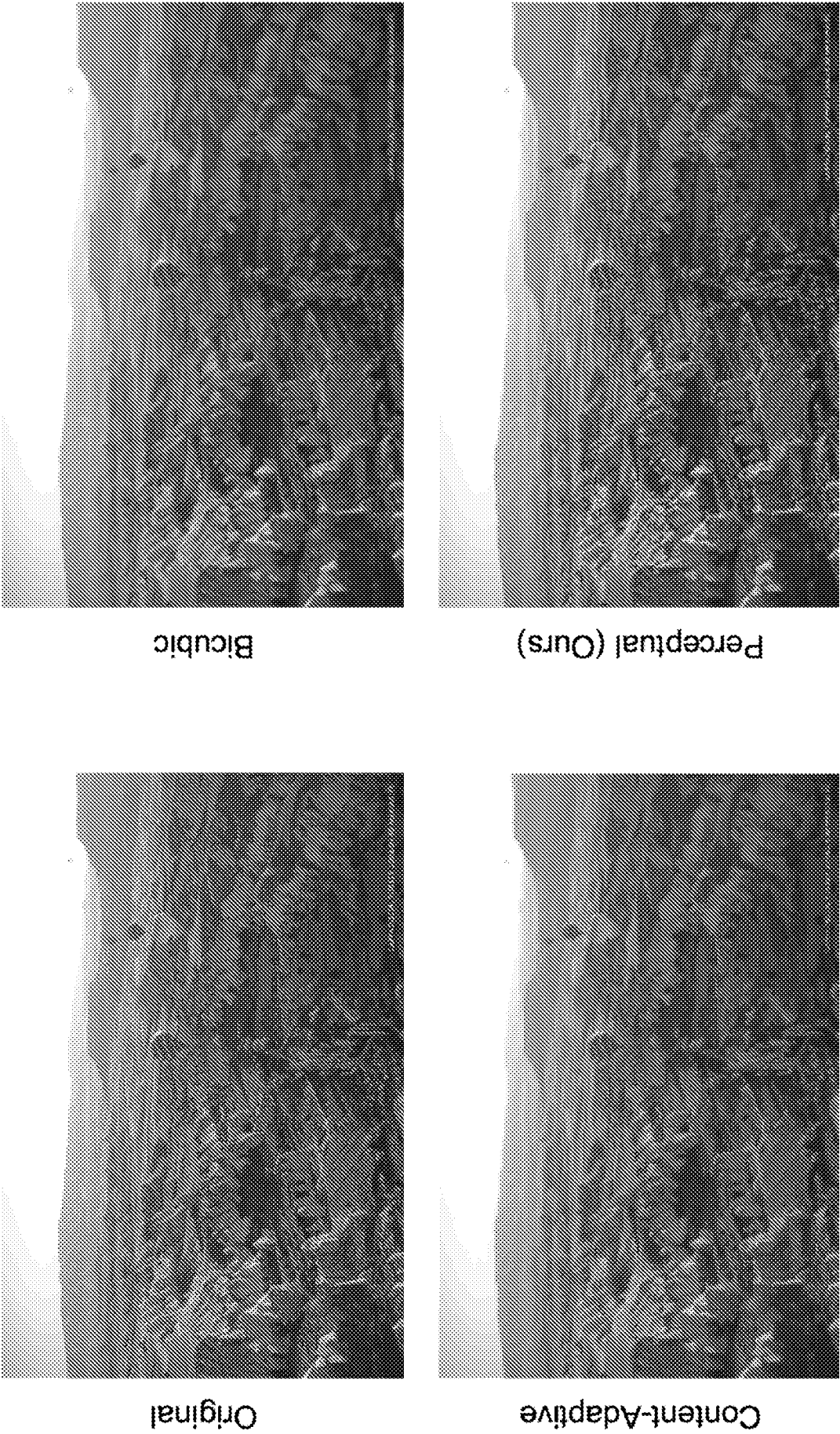
FIG. 15

FIG. 16A
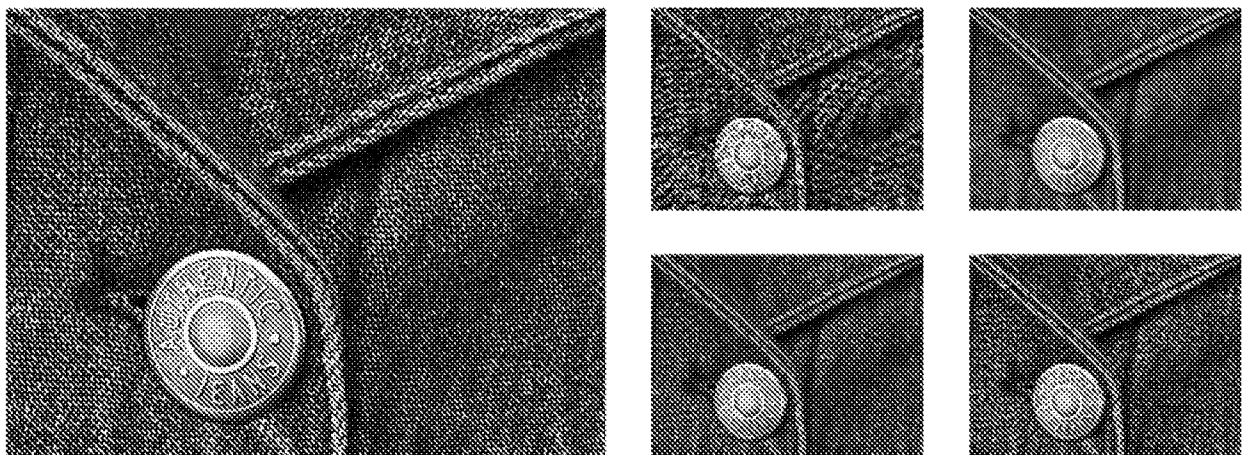
FIG. 16B

FIG. 17

FIG. 18

FIG. 19

2004

Input Image File

2008

INTERNAL
IMAGE
STORAGE

2010

Program Code

2002

DOWNSCALING ENGINE

2006

Output Image File
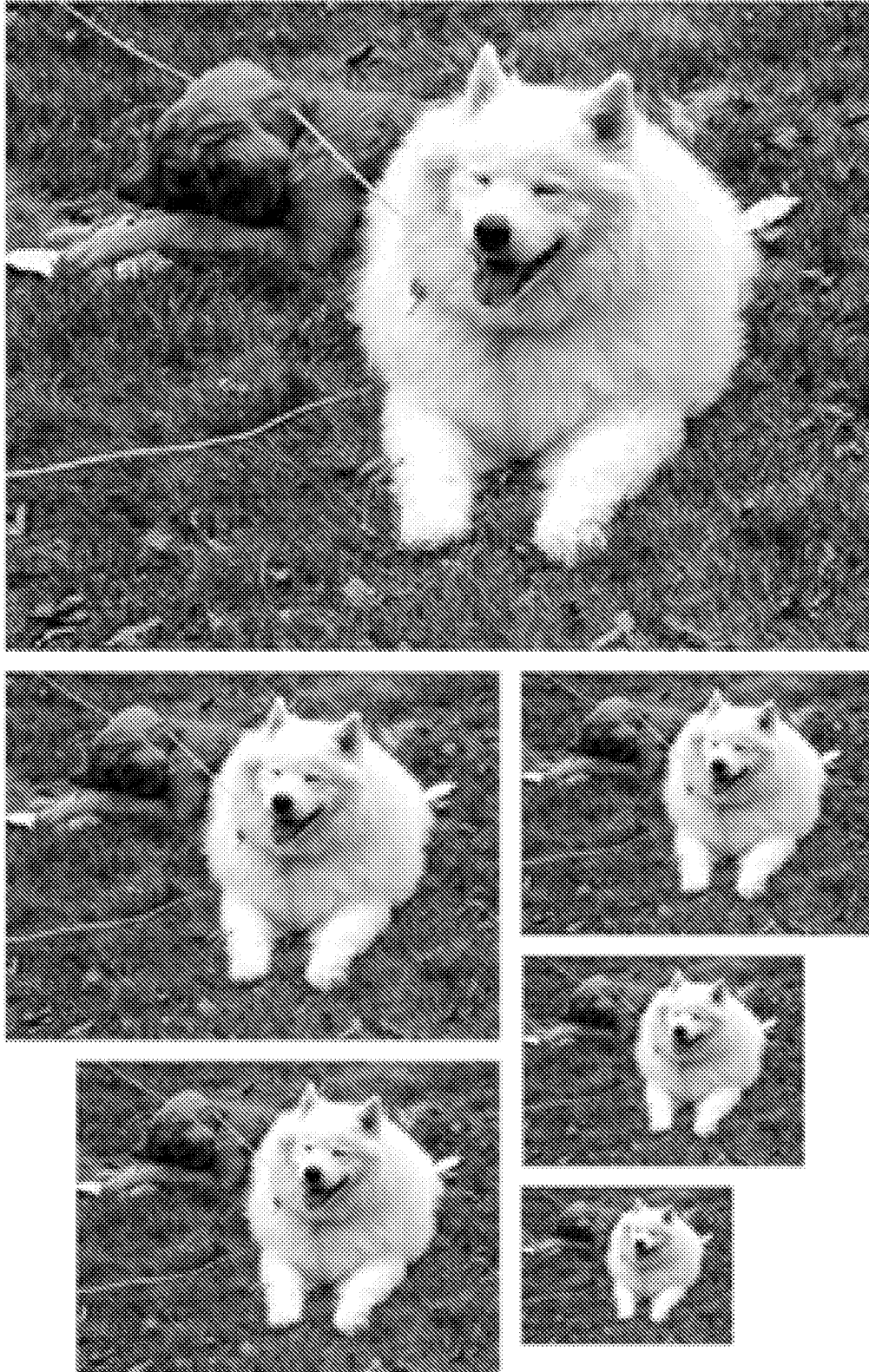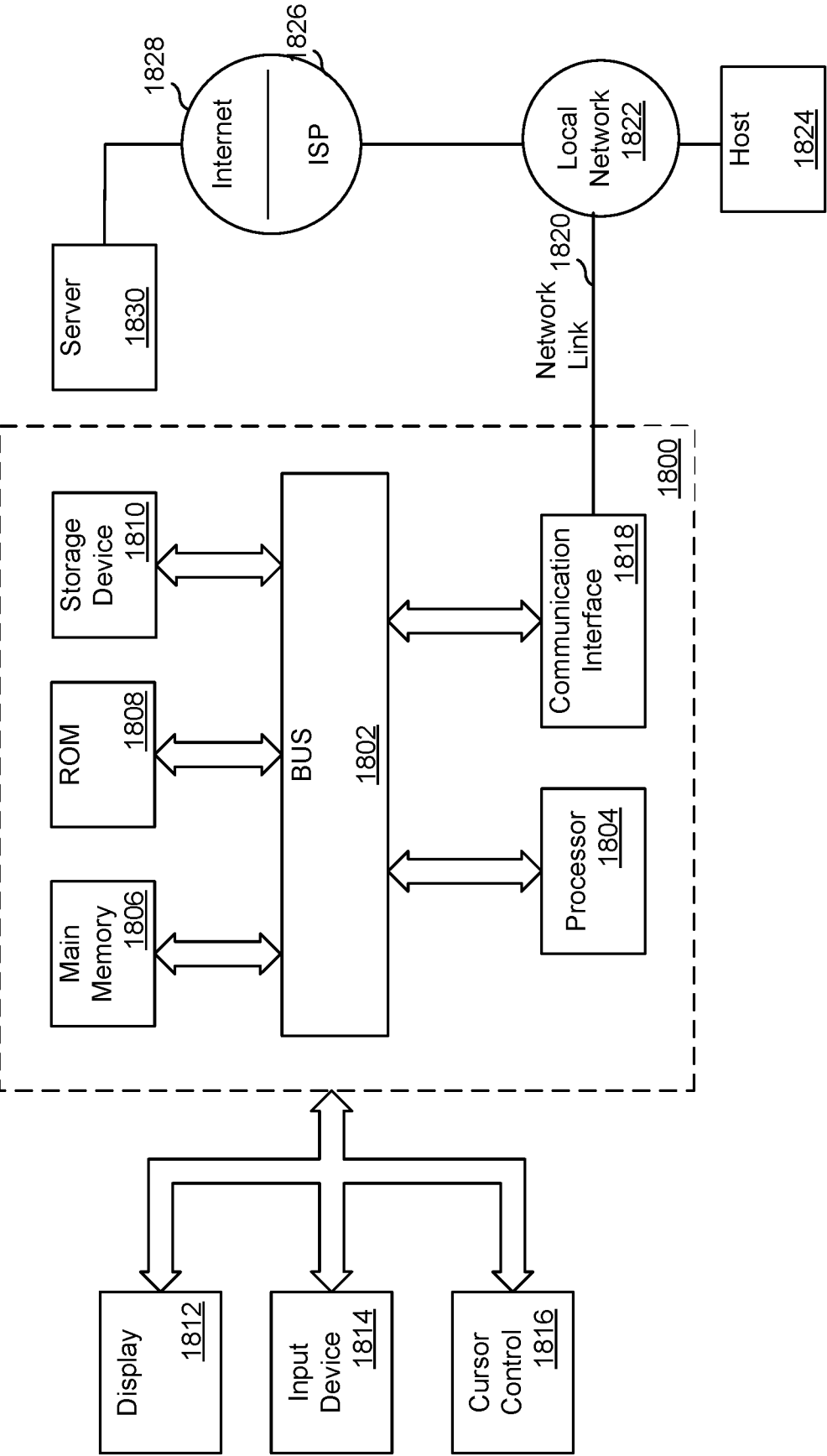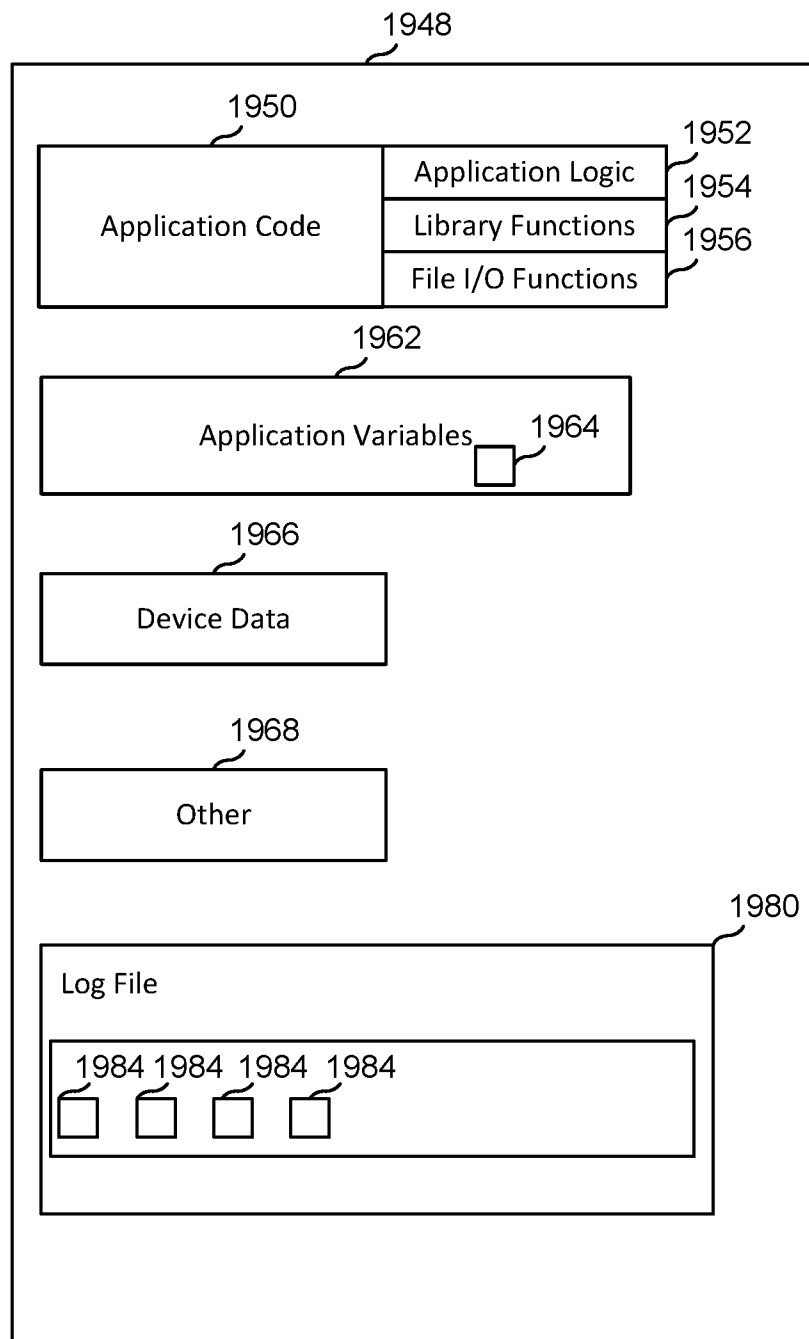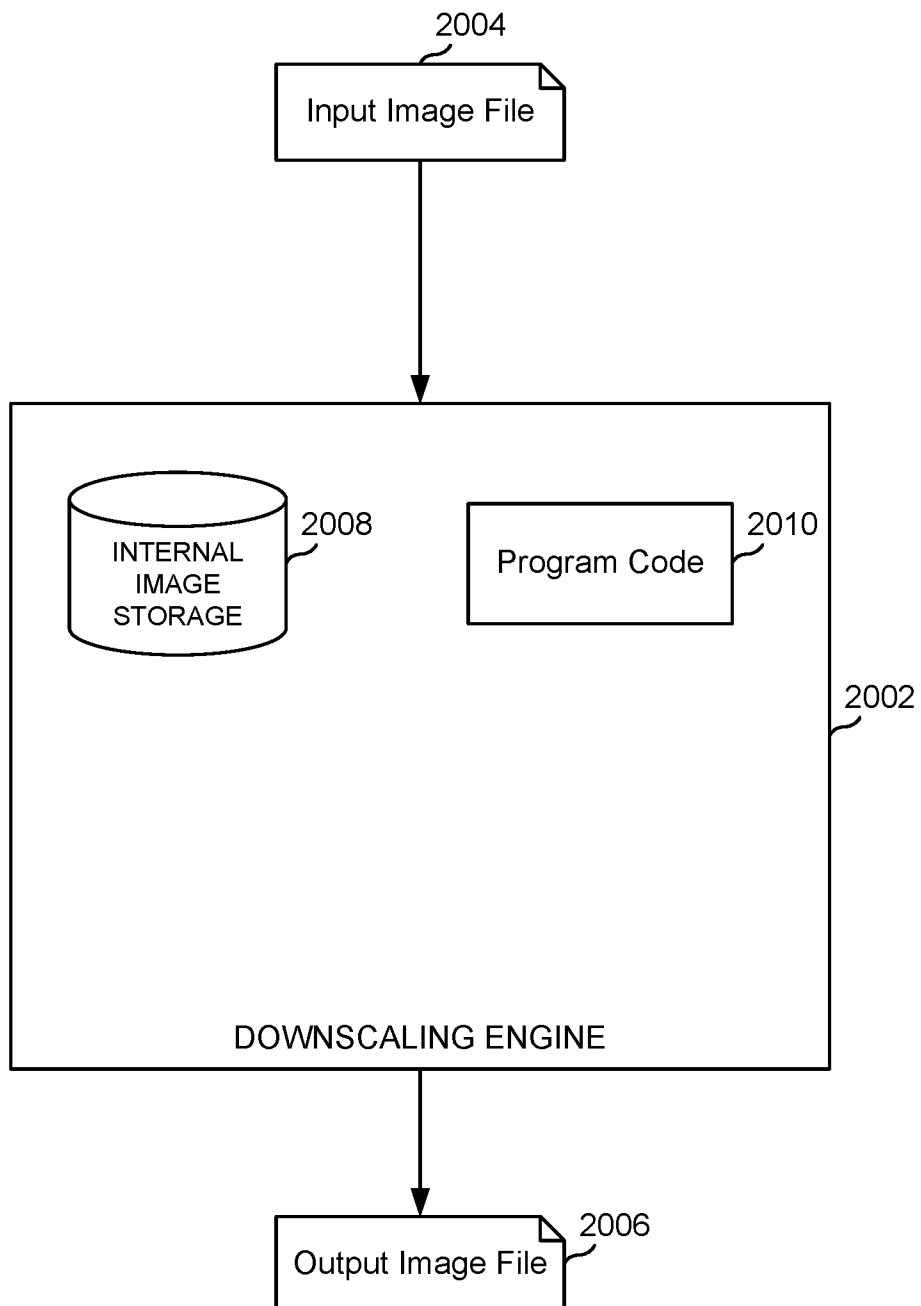
FIG. 20

INTERNATIONAL SEARCH REPORT

International application No

PCT/IB2016/054404

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06T3/00   G06T3/40
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2011/160225 A1 (WORLDPLAY BARBADOS INC [BB]; LANCASTER GREGORY K [CA]) 29 December 2011 (2011-12-29) abstract page 4, line 10 - page 5, line 18 figure 1 -----	1,2
X	WO 2013/079516 A1 (ELETTRA SINCROTRONE TRIESTE S C P A [IT]) 6 June 2013 (2013-06-06) abstract page 2, line 15 - page 3, line 25 page 4, line 28 - page 5, line 13 page 7, line 6 - line 29 -----	1,2

-/--

| X | Further documents are listed in the continuation of Box C. | | X | See patent family annex. |

Date of the actual completion of the international search	Date of mailing of the international search report
27 October 2016	16/11/2016

| Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016 | Authorized officer Rusu, Alexandru |

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

C(Continuation).   DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	DONG JIE ET AL: "Adaptive Downsampling for High-Definition Video Coding", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE SERVICE CENTER, PISCATAWAY, NJ, US, vol. 24, no. 3, 1 March 2014 (2014-03-01), pages 480-488, XP011542202, ISSN: 1051-8215, DOI: 10.1109/TCSVT.2013.2278146 [retrieved on 2014-03-03] abstract figure 1 Section II. Optimal Ratio for Video Downsampling; page 481 - page 482 -----	1,2
X	YONGBING ZHANG ET AL: "Interpolation-Dependent Image Downsampling", IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE SERVICE CENTER, PISCATAWAY, NJ, US, vol. 20, no. 11, 1 November 2011 (2011-11-01), pages 3291-3296, XP011411918, ISSN: 1057-7149, DOI: 10.1109/TIP.2011.2158226 abstract figure 1 second full paragraph; page 3292, right-hand column Section III. Content-Dependent IDID; page 3292 - page 3295 -----	1,2
A	JOHANNES KOPF ET AL: "Content-adaptive image downscaling", ACM TRANSACTIONS ON GRAPHICS (TOG), ACM, US, vol. 32, no. 6, 1 November 2013 (2013-11-01), pages 1-8, XP058033866, ISSN: 0730-0301, DOI: 10.1145/2508363.2508370 the whole document -----	1,2

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
WO 2011160225	A1	29-12-2011	US	2011317773 A1	29-12-2011
			WO	2011160225 A1	29-12-2011
WO 2013079516	A1	06-06-2013	EP	2786341 A1	08-10-2014
			US	2014328546 A1	06-11-2014
			WO	2013079516 A1	06-06-2013