

# Modifying Non-Local Variations Across Multiple Views

Tal Tlusty  
Technion, Israel

taltlusty@gmail.com

Tomer Michaeli  
Technion, Israel

tomer.m@ee.technion.ac.il

Tali Dekel  
Google Research

tdedel@google.com

Lihi Zelnik-Manor  
Technion, Israel

lihi@technion.ac.il

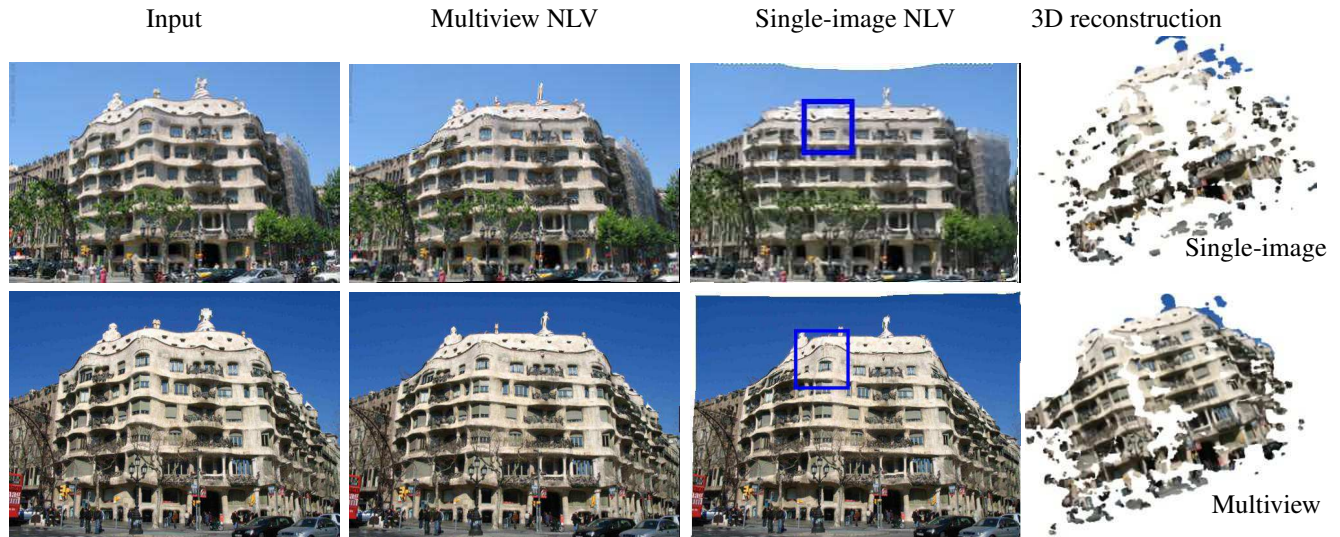


Figure 1: Gaudi’s Casa Mila is famous for its irregular shapes. Single-image NLV idealizes each image independently, resulting in utterly different geometries – the building’s facade is flat and some of the porches are straightened only in the top image. Reconstruction of a corresponding 3D structure completely fails. Conversely, our multiview NLV approach reduces the irregularities in both images in a synchronized manner and the corresponding 3D shape shows straightened verandas.

## Abstract

We present an algorithm for modifying small non-local variations between repeating structures and patterns in multiple images of the same scene. The modification is consistent across views, even-though the images could have been photographed from different view points and under different lighting conditions. We show that when modifying each image independently the correspondence between them breaks and the geometric structure of the scene gets distorted. Our approach modifies the views while maintaining correspondence, hence, we succeed in modifying appearance and structure variations consistently. We demonstrate our methods on a number of challenging examples, photographed in different lighting, scales and view points.

## 1. Introduction

Repetition of patterns and structures is a widespread phenomenon, e.g. in the leaves of plants and flowers, animal

furs or rocks and sand dunes. Recurring structures can be also found in man-made environments, for example, a row of chairs in a large stadium or a pile of boxes in a shoe store. In many cases, the recurring structures are not perfectly identical and sometimes the deviations from an ‘ideal’ structure are small and hard to notice by the naked eye. Revealing these deviations may be useful in many situations, for example, revealing deformations in a production line or detecting irregular cells in a petri dish. Modifying these variations and correcting them could be useful for beautification of images.

Recently, a method was devised for revealing the Non-Local Variations (NLV) in a single image [7]. Their method recovers a simple geometric transformation that can be applied to the input image in order to obtain an ‘ideal’ image in which the variations between repeated structures are minimal. By applying the inverse transformation, these non-local variations can be exaggerated. Their method was used for several applications such as idealizing images, revealing object properties and visualizing defects in material inspection.

tion. Their approach is based on generating an ideal image where each patch is replaced by the average of its nearest-neighbors (in terms of appearance), and then estimating the deviation of the input image from this ideal image.

In the modern media age we live in, oftentimes multiple images of the same scene are taken. Different people capture the same monuments, events, and objects, and share them publicly. In this paper, we present an approach to multiview NLV which extends the single-image NLV of [7] to a pair of images of the same scene that could have been taken under different lighting conditions and from different viewpoints. We show that applying single-image NLV independently to each image leads to inconsistent corrections, and hence, the ability to reconstruct a corresponding 3D shape is damaged (see Figure 1). In order to preserve the geometrical consistency, we add a correspondence constraint which enforces the images to be idealized together. We show that this approach reveals and corrects not only appearance variations but also non-local 3D shape variations.

We suggest two ways to guarantee consistency across multiple views. The first requires geometrical consistency between the idealizing transformations, while the second approach requires similarity between the appearances of the idealized images. Demanding similar appearance implicitly requires that the idealizing geometrical transformations that the images undergo are consistent. An advantage of the transformation-based approach is its applicability to images with different lighting and illumination while the appearance-based approach is appropriate only when the input images are very similar in their colors, e.g., a stereo pair or frames of a video. We examine multiview-NLV on pairs of images taken by different people, at different times, from different viewpoints and varying illuminations.

The rest of the paper is organized as follows. Section 2 surveys related work. Mathematical formulation with a short overview of the multi-view NLV problem is described in Section 3, and a detailed description of the algorithm is provided in Section 4. Experimental validation is presented in Section 5 and conclusions are drawn in Section 6.

## 2. Related Work

**Non-Local Variations in a Single Image.** In a recent work, Dekel *et al.* [7] presented a non-parametric algorithm for detecting and visualizing small variations between repeating structures within a single image. Their method can handle complex deformations that cannot be modeled by a parametric model. Their algorithm iteratively alternates between two steps: first, computing an ‘ideal’ average image, in which every patch is replaced by the average of its nearest neighbors, and second, computing the ‘idealizing’ transform between the ideal image and the original input image. Lukáč *et al.* [26] proposed generating a symmetry feature set, and then using it as input to the NLV algorithm of [7], thus, extending the idealizing transformation

to work with rounded symmetry repeating structures. Wadhwa *et al.* [37] presented a different approach for revealing tiny deformations within a single image using a local parametric method. In their algorithm, a parametric model is fitted to objects in the input image. Then, the residual between the ideal model and the fitted model is calculated. In our work, we extend the non-parametric approach of [7], to handle non-parametric, complex deformations consistently between multiple images of similar content.

**Texture Manipulation.** Also related to our problem are works that estimate the geometric deformation between multiple images for texture replacement in videos [19][33][39]. These methods first find a texture deformation, and then compute the correspondence in order to track the motion between frames. Another approach for texture replacement in multi-view images uses depth information to recover near-regular texture deformation [34]. Somewhat related is also the work of Yücer *et al.* [42], that propose a method for consistent manipulation of multiple images of a common object based on an adaptation of the Lucas-Kanade [25] framework.

Methods for single-image texture manipulation have various applications, such as texture symmetrization [16][24], texture replacement [8][23][24], or shape from texture [5][27]. Another widely-discussed application is texture synthesis, where high quality texture is synthesized from a small texture patch or image [8][16].

A different method for single-image texture synthesis [1] is synthesizing time-varying weathered textures by estimating an “age map” for the input image, which is based on prevalence analysis of image patches. Dekel *et al.* [7] differs from previous texture manipulation works in three principle aspects: (i) Most texture manipulation methods are based on a single texon repetition to form a lattice and rely on finding the deformation in the lattice. In the NLV the repeating structures can be located in different locations in the image. (ii) Some of the methods [16] do not rely on near-regular textures, however they cannot treat images with multiple textures. (iii) The NLV [7] algorithm works in different scales, hence locates even small scale variations in a near-regular texture. Since our image editing algorithm is based on [7], all the above are relevant to us as well.

**Motion Magnification.** A series of studies on revealing and magnifying tiny variations in video-sequences has received great acknowledgement inside and outside the computer vision community [21][38][40]. These articles present several approaches to reveal and magnify temporal variations that are invisible to the naked eye using frequency analysis of the video. Our goal, however, is finding variations within each of the input images while preserving the correspondence between the modified images.

**Co-Saliency.** Identifying unique patterns in recurring structures correspondingly in a set of images is also used for co-saliency detection [4][6][9][14][15][17][18][30]. Most of these methods integrate two kinds of cues: inner image saliency and multi-image correspondence which enforces similar objects to be salient together [6][9][15][17][18]. Other methods, use the correspondence information to force physical constraints on the mutual saliency maps [4][30]. Unlike the co-saliency methods, our algorithm locates variations between very similar recurrent structures correspondingly in multi-view images.

**Co-Segmentation.** Co-segmentation algorithms are related to our work since they also deal with synchronous analysis of multiple images of the same scene, albeit their goal is segmentation. Rother *et al.* [31] co-segment a pair of images using a graph based solution for generative model estimation while forcing histogram matching of the foreground pixels. Other unsupervised algorithms for co-segmentation [36][28][29][13][32][41] use variations of optimization methods and different sets of features for correspondence. Unlike co-segmentation methods, our approach requires dense pixel-wise matching between the images to obtain consistent images manipulation.

### 3. Problem Formulation

To outline the problem formulation we start by briefly reviewing the single-image NLV formulation of Dekel *et al.* [7] in Section 3.1. Then we lay out our extension to multiple views in Section 3.2.

#### 3.1. Single-Image NLV

Given an input image  $I$ , the NLV method seeks to determine a smooth geometric deformation  $\mathcal{T}$  corresponding to a dense flow field  $(u(x, y), v(x, y))$ , which maximizes the resemblance between recurring patches in  $I$ . This is done by introducing an auxiliary ‘ideal’ image  $J$ , which is restricted to: (i) have strong patch repetitions, and (ii) be similar to  $\mathcal{T}\{I\}$  for some deformation  $\mathcal{T}$ . More specifically, the ‘ideal’ image  $J$  and the idealizing deformation  $\mathcal{T}$  are obtained through the minimization of the energy

$$E_{\text{NLV}}(\mathcal{T}, J, \text{DB}) = E_{\text{rec}}(J, \text{DB}) + \lambda E_{\text{data}}(\mathcal{T}\{I\}, J) + \alpha_r E_{\text{reg}}(\mathcal{T}) \quad (1)$$

over both  $J$  and  $\mathcal{T}$ . Here, the first term is a ‘recurrence’ energy, which measures the dissimilarity between each patch  $p \in J$  and its nearest neighbor patches  $\{q\}$  from the database DB of all overlapping patches in  $J$ . It is defined as

$$E_{\text{rec}}(J, \text{DB}) = - \sum_{p \in J} \log \left( \sum_{q \in \text{DB}} \exp \left\{ -\frac{1}{2h^2} \|p - q\|^2 \right\} \right), \quad (2)$$

where  $h$  is some bandwidth parameter. The second term in (1) aims for similarity between the ideal image  $J$  and the deformed input image  $\mathcal{T}\{I\}$ . Specifically,

$$E_{\text{data}}(\mathcal{T}\{I\}, J) = \iint \psi(\|J(x, y) - \mathcal{T}\{I\}(x, y)\|^2) dx dy, \quad (3)$$

where  $\psi(a^2) = \sqrt{a^2 + \varepsilon^2}$ , for some small  $\varepsilon$ . Finally, the last term in (1) is a regularizer on the transformation  $\mathcal{T}$ , which directs the flow field to be smooth,

$$E_{\text{reg}}(\mathcal{T}) = \iint \psi(\|\nabla u(x, y)\|^2 + \|\nabla v(x, y)\|^2) dx dy. \quad (4)$$

To minimize the energy (1), the NLV algorithm alternates between updating the transformation  $\mathcal{T}$ , the ideal image  $J$ , and the patch database DB. The updates of  $\mathcal{T}$  and  $J$  are performed with an Iterative Reweighted Least Squares (IRLS) type algorithm [2].

#### 3.2. Multiview NLV

When more than a single view of the scene is available, we want the detection and correction of the non-local variations to be consistent across views. In this section we present our formulation for revealing non-local variations in a pair of images. This formulation can be easily extended to more than two images. Due to space limitations the extension to  $k$ -views is not included in this version (go to our website <http://cgm.technion.ac.il/ComputerGraphics-Multimedia/Software/MultiViewNLV> for an extended version of this work).

Let  $I_1$  and  $I_2$  be two images of the same scene which contain inherent recurring structures with small variations. The images could be a stereo pair, or taken from different views and differently illuminated. The aim of our formulation is to recover two idealized images  $J_1$  and  $J_2$  with reduced structural variance, which are consistent with each other such that they correspond to a plausible valid scene. For example, in Figure 1, we want the porches on Gaudi’s Casa Mila to deform consistently across the two views, corresponding to a 3D building with fewer depth variations. As can be seen in the figure, idealizing each image separately destroys the geometrical structure of the scene since each image is manipulated differently and converges to a different local minimum.

Denote by  $\mathcal{T}_1$  and  $\mathcal{T}_2$  the idealizing transformations with corresponding flow-fields  $(u_1(x, y), v_1(x, y))$  and  $(u_2(x, y), v_2(x, y))$ , so that the corrected input images are

$$\begin{aligned} \mathcal{T}_1\{I_1\}(x, y) &= I_1(x + u_1(x, y), y + v_1(x, y)), \\ \mathcal{T}_2\{I_2\}(x, y) &= I_2(x + u_2(x, y), y + v_2(x, y)). \end{aligned} \quad (5)$$

A naive way to determine  $\mathcal{T}_1$  and  $\mathcal{T}_2$ , would be to minimize the single-view energy (1) for each of the views independently (with two auxiliary ‘ideal’ images  $J_1$  and  $J_2$ ,



respectively). However, in order to obtain consistent transformations, we also incorporate a geometrical consistency constraint through a correspondence loss  $E_{\text{corr}}$ . Our multi-view energy is thus defined as:

$$E_{\text{MV-NLV}}(\mathcal{T}_1, J_1, \text{DB}_1, \mathcal{T}_2, J_2, \text{DB}_2) = E_{\text{NLV}}(\mathcal{T}_1, J_1, \text{DB}_1) + E_{\text{NLV}}(\mathcal{T}_2, J_2, \text{DB}_2) + \alpha_c E_{\text{corr}}(\mathcal{T}_1, \mathcal{T}_2, J_1, J_2), \quad (6)$$

where  $\alpha_c$  is a parameter tuning the importance of multiview consistency. We examine two options for the correspondence loss, described below.

Before we define the loss, we note that it is important to solve our optimization problem in a single coordinate system. To do so, we set one of the images as the anchor view and the second image is warped to the anchor image coordinates. Without loss of generality, we choose  $I_1$  as the anchor and  $I_2$  is warped to the anchor coordinates. To make our formulation as simple as possible, we refer to  $I_2$  as the warped input image. The recovered ideal images  $J_1$  and  $J_2$ , and transformations  $\mathcal{T}_1$  and  $\mathcal{T}_2$ , are obtained in the anchor view coordinate system. To get back to the original coordinate system, we unwarped the recovered transformation and apply it to the original input image.

**Transformation consistency** The first option for the correspondence loss is to constrain the idealizing transformations  $\mathcal{T}_1$  and  $\mathcal{T}_2$  to be similar. This can be done by demanding similarity between their corresponding flow-fields. Specifically, letting  $w_x(x, y) = u_1(x, y) - u_2(x, y)$  and  $w_y = v_1(x, y) - v_2(x, y)$ , we define the transformation consistency loss to be

$$E_{\text{corr}}(\mathcal{T}_1, \mathcal{T}_2) = \iint (w_x^2(x, y) + w_y^2(x, y)) dx dy. \quad (7)$$

This loss penalizes large deviations between the recovered transformations of the two views.

**Appearance consistency** A second option for the correspondence loss is to constrain the idealized images to have similar appearance, while constraining the transformation differences to be smooth. For this penalty to be meaningful, we work in the coordinates of the input images, thus

$$E_{\text{corr}}(\mathcal{T}_1, J_1, \mathcal{T}_2, J_2) = \frac{\lambda}{\alpha_c} \iint \psi (\|\mathcal{T}_1^{-1}\{J_1\}(x, y) - \mathcal{T}_2^{-1}\{J_2\}(x, y)\|^2) dx dy + \iint \psi (\|\nabla w_x(x, y)\|^2 + \|\nabla w_y(x, y)\|^2) dx dy. \quad (8)$$

Here  $\psi$  is as in (3) and (4),  $(w_x, w_y)$  is as in (7), and the integrals are only over valid pixels (where  $I_2$  was mapped to  $I_1$  with high confidence).

---

#### Algorithm 1 Multi-View NLV

---

**Input:** Images  $I_1, I_2$ ; Correspondence field  $D_x, D_y$ .

**Output:** Ideal images  $J_1, J_2$ , Idealizing transformations  $\mathcal{T}_1, \mathcal{T}_2$ .

Down-sample to coarsest scale

Warp  $I_2$  to  $I_1$  coordinates using  $D_x, D_y$ .

Initialize  $\mathcal{T}_1$  and  $\mathcal{T}_2$  to be identity mapping,  $J_1 = I_1$  and  $J_2 = I_2$ .

**repeat**

1. **Database Update:**

Set  $\text{DB}_1, \text{DB}_2$  as all overlapping patches from  $J_1$  and  $J_2$ , respectively.

2. **Image Update:**

Minimize 6 w.r.t.  $J_1, J_2$

3. **Transformation Update:**

Minimize 6 w.r.t.  $\mathcal{T}_1, \mathcal{T}_2$ ,

**Upscale images and transformations**

**until** Fine scale

---

Each of the proposed formulations has its advantages. The transformation consistency formulation of (7), enables processing images with completely different illuminations, or even more extreme appearance differences. The appearance consistency formulation of (8), assumes visual similarity between the images. This is more limiting, on one hand, but in scenarios when this assumption holds (e.g., video frames or a stereo pair) the additional constraint could lead to more accurate results. In practice, we have found that the transformation consistency provided satisfactory results in all our experiments. Therefore, we henceforth provide the detailed algorithm description only for the transformation consistency option, and leave the appearance consistency algorithm to the supplementary.

## 4. Detailed Description of the Algorithm

To solve the optimization problem (6), we alternate between updating the patch databases  $\text{DB}_1, \text{DB}_2$ , the ideal images  $J_1, J_2$ , and the transformations  $\mathcal{T}_1, \mathcal{T}_2$ , as summarized in Algorithm 1. We run the algorithm in a coarse-to-fine pyramid structure in order to find variations in different scales. The scale of the coarsest pyramid level is selected by the user according to the size of the repeating structures of interest.

**Database update** In this step we update the databases  $\text{DB}_1, \text{DB}_2$  by extracting all overlapping patches from the current ideal images  $J_1, J_2$ , respectively. These databases are then held fixed throughout the other algorithm steps.

**Image update** In this step, we minimize the objective (6) with respect to the ideal images  $J_1, J_2$  while holding

all other variables fixed. This step drives the patches in  $J_1, J_2$  to be similar to those in the current databases  $DB_1, DB_2$ , while constraining the images to remain close to the geometrically corrected versions of the input images,  $\mathcal{T}_1\{I_1\}, \mathcal{T}_2\{I_2\}$ , and to each other. This has the effect of strengthening the patch repetitions within  $J_1, J_2$ . To simplify the exposition, we denote the geometrically corrected input images as  $I_1^c = \mathcal{T}_1\{I_1\}$  and  $I_2^c = \mathcal{T}_2\{I_2\}$ .

Substituting the transformation consistency loss (7) in (6), and retaining only the terms that depend on  $J_1, J_2$ , we obtain the objective

$$E_{\text{rec}}(J_1, DB_1) + E_{\text{rec}}(J_2, DB_2) + \lambda E_{\text{data}}(\mathcal{T}_1\{I_1\}, J_1) + \lambda E_{\text{data}}(\mathcal{T}_2\{I_2\}, J_2).$$

This objective is separable in  $J_1$  and  $J_2$ , and each sub-problem is identical to the one in the image update objective of the single-image NLV algorithm. We thus use the same solution technique as in [7]. Specifically, setting the gradient w.r.t. to  $J_1$  to zero, we get that

$$J_1(x, y) = \beta_1(x, y)Z_1(x, y) + (1 - \beta_1(x, y))I_1^c(x, y), \quad (9)$$

where  $Z_1$  is an image obtained by replacing each patch in  $J_1$  by a weighted combination of its  $K$  Nearest Neighbor (NN) patches from the database  $DB_1$ , and

$$\beta_1(x, y) = \frac{W_{\text{data}}^1(x, y)}{W_{\text{data}}^1(x, y) + \frac{h^2}{M^2}}. \quad (10)$$

Here,  $M$  denotes the patch width,  $h$  is the bandwidth parameter in (2), and  $W_{\text{data}}^1(x, y) = \frac{1}{\lambda} \psi(\|J_1(x, y) - I_1^c(x, y)\|^2)$ . Since  $Z_1$  and  $\beta_1$  are both functions of the unknown  $J_1$ , we alternate between updating  $J_1$  according to (9) and computing  $\beta_1$  using (10). The image  $J_2$  is updated similarly.

**Transformation update** In this step, we update the idealizing transformations  $\mathcal{T}_1, \mathcal{T}_2$  while keeping the images  $J_1, J_2$  fixed. We force  $\mathcal{T}\{I\}$  to be similar to  $J$  and the transformation to be piece-wise spatially smooth. The objective to be minimized is

$$\lambda E_{\text{data}}(\mathcal{T}_1\{I_1\}, J_1) + \lambda E_{\text{data}}(\mathcal{T}_2\{I_2\}, J_2) + \alpha_r E_{\text{reg}}(\mathcal{T}_1) + \alpha_r E_{\text{reg}}(\mathcal{T}_2) + E_{\text{corr}}(\mathcal{T}_1, \mathcal{T}_2). \quad (11)$$

Substituting (3), (4), and (7), this objective becomes

$$\begin{aligned} & \lambda \iint \psi(\|J_1(x, y) - I_1(x + u_1, y + v_1)\|^2) dx dy \\ & + \lambda \iint \psi(\|J_2(x, y) - I_2(x + u_2, y + v_2)\|^2) dx dy \\ & + \alpha_r \iint \psi(|\nabla u_1(x, y)|^2 + |\nabla v_1(x, y)|^2) dx dy \\ & + \alpha_r \iint \psi(|\nabla u_2(x, y)|^2 + |\nabla v_2(x, y)|^2) dx dy \\ & + \alpha_c \iint (w_x^2(x, y) + w_y^2(x, y)) dx dy, \end{aligned} \quad (12)$$

where  $(w_x, w_y)$  is the difference between the flow fields, as in (7). This formulation is very similar to that of conventional optical flow estimation. Only here we need to simultaneously determine two flow fields, which conform to two data terms, two smoothness terms, and a consistency term between them. We solve this optimization problem similarly to the IRLS optical flow method of [20]. The (lengthy) detailed derivations are provided in the Supplementary.

**Implementation details** To perform the initial coordinate warping to the anchor view's coordinates system, we compute the correspondence field  $D_x, D_y$  between  $I_1$  and  $I_2$ . When the input is a stereo pair, we calculate the disparity field using Semi-Global Matching (SGM) [12]. For unconstrained multi-view images that could have similar content, but with different scales and illuminations, we used either the robust correspondenc of [35] or that of SIFT-flow [22].

## 5. Empirical Evaluation

In this section we assess the capabilities of our proposed multiview NLV algorithm and compare it to the single-image NLV algorithm of [7]. Our results exhibit the main contribution of the proposed approach that produces geometrically consistent idealized image pairs that preserve the original 3D structure properties, only idealized.

Our experiments were performed on natural images with a variety of textures and different kinds of repeated objects. Some images were captured by us, and some were found online. In addition we also tested several images that were rendered from a 3D model using [11], in order to examine specific phenomena under controlled conditions. More results are presented in the supplementary.

In all our experiments we applied the algorithms to the original input images, using 3–4 pyramid levels, such that in the coarsest level the patch size covers the largest repeating structure in the input image. A patch size of  $M = 15 \times 15$  was fixed for all scales, so that we cover different sizes of repeating objects. For example, in the corn example in Figure 4, we can create uniform corn kernels (fine scale) as well as aligned kernels rows (coarse scale). The results obtained with appearance consistency and transformation consistency were pretty much similar (when appearance consistency could be applied), hence, we show here results for the latter. The parameters were set to  $\lambda = 3$ ,  $\alpha_r = 0.03$ ,  $\alpha_c = 0.06$ ,  $h = 0.1$ ,  $\epsilon = 1e^{-6}$ , and  $K = 30$  NNs.

**Multiview:** Figures 1, 2 and 3 show example results for images photographed from different views. In the case of Figures 1 and 3 also at different occasions and by different people, resulting in severe scale and illumination changes. It can be seen that our multiview approach corrects the variations in a consistent manner across views, resulting in a corresponding idealized 3D shape, whereas, the single-view NLV leads to different transformations and distorted shapes. The visualization of the transformation flow-fields

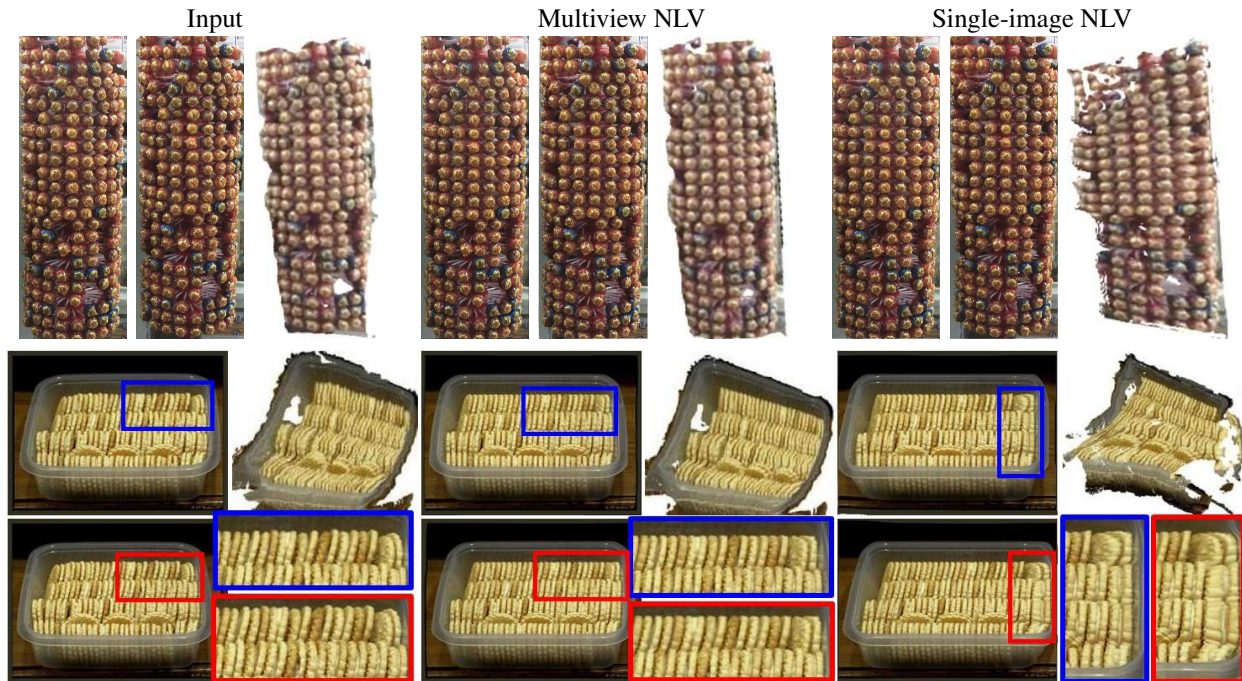


Figure 2: **Multiview vs. Single-image:** NLV can reveal and correct the disorganization in the lollipop stand and cookie box. However, “idealizing” each image independently with Single-image NLV results in inconsistencies, e.g., the lollies on the top-left of the stand and rightmost cookies (we zoom-in on regions to highlight this). It is easier to see that in the 3D reconstructions that have holes in those areas and show overall awkward geometries. Conversely, our multiview NLV approach reduces the irregularities in both images in a synchronized manner resulting in straightened candies and cookies, a cylindrical stand and rectangular box.

in Figure 3 illustrates the importance of multiview correspondence. The variations we wish to discover are those inherent to the structures in the scene. However, when each image is processed independently, a local minima solution could be found. Enforcing the geometric correspondence drives the optimization to find a solution that complies with both images, and hence, tends to better comply with the actual shape variations in the scene. We show the actual modified images in the supplementary.

**Stereo:** Figure 4 shows results when the input images are either taken by a stereo pair, or rectified images (we used the ‘Epipolar Rectification Toolkit’ [10]). It is evident that single-image NLV results in inconsistent transformations, ruining the correspondence and leading to a 3D shape with holes and distortions. Conversely, our multiview NLV algorithm produces idealized images, and a corresponding idealized 3D shape. For example, the corn kernels are regular and the cob’s shape is a smooth cylinder.

**Correction vs. exaggeration:** In Figure 5 we exemplify how multiview NLV can both correct and exaggerate the variations within the images. It is interesting to see how both the color and the shape of the berries become more (or less) regular when processed by our algorithm.

**Different illuminations:** We further tested our approach on pairs of images taken from the same viewpoint, but with

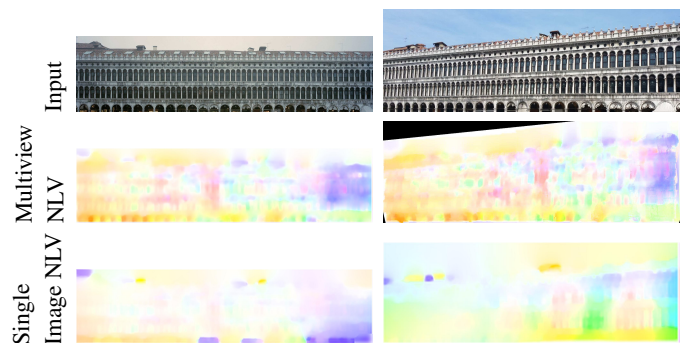


Figure 3: **Flow correspondence:** Given two images of the colonnade in Piazza San Marco we can reveal the variations between the seemingly identical colons and arches. Here we show the transformation field between each input image and its version with exaggerated variations via NLV. In Multiview NLV the transformation fields of the two images correspond, e.g., arches were expanded or shrunk consistently in both views. Conversely, in Single-image NLV the transformations are quite different and inconsistent, e.g., the right region is purple (moves up and right) in one image and green and yellow (moves down and left) in the other. This indicates breaking the geometrical consistency.



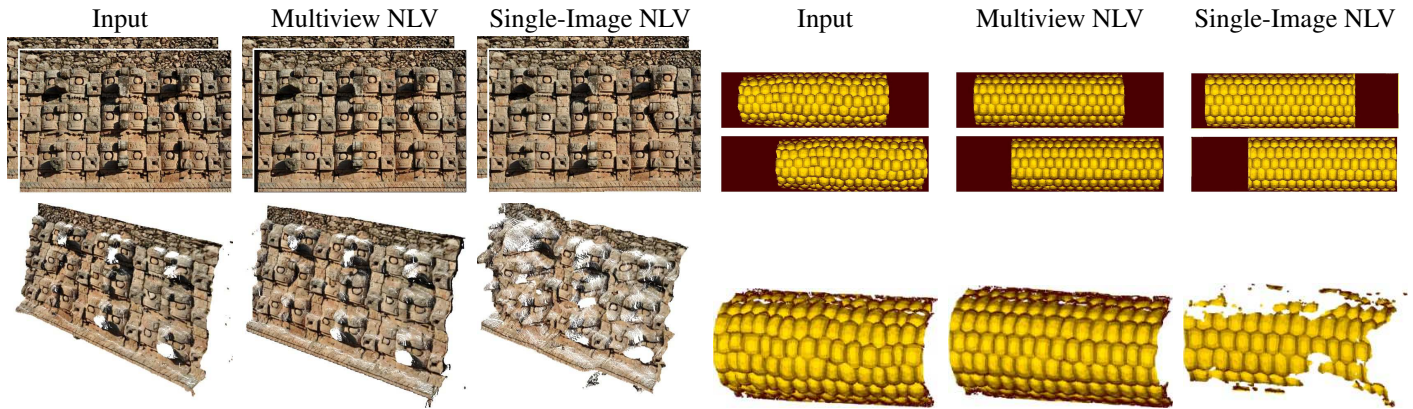


Figure 4: **Stereo:** In these stereo-pairs NLV can straighten the wall relief patterns and regularize the corn kernels. When using multiview NLV the corresponding 3D shapes are regularized as well. Conversely, single-image NLV ruins the geometrical consistency, implying no corresponding correct 3D shape exists, hence the holes in the reconstruction.

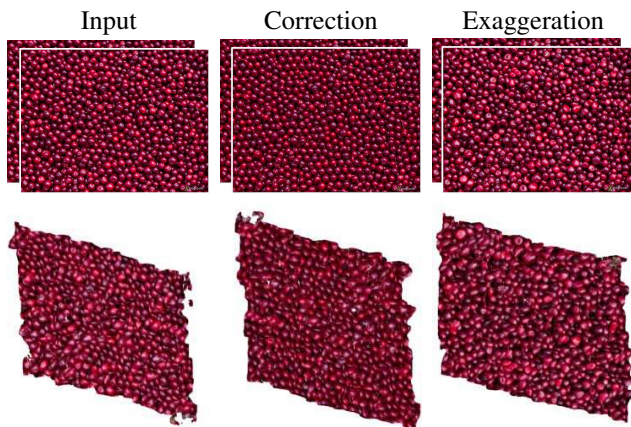


Figure 5: **Correction vs. exaggeration:** NLV can reorganize the berries to be of more similar color and shape, or vice versa, it can extenuate the variations between them.

illumination and style differences (from the MIT-Adobe FiveK Dataset [3]). Figure 6 presents one example result (more are provided in supplementary). It can be seen that single-view NLV produces inconsistent transformation hence blending the two images produces in a blurred result, whereas our multiview NLV yields corresponding idealized images and hence the blended result in sharp.

**Parameter tuning:** In Figures 7, 8 we analyze the effect of tuning the parameter  $\alpha_c$ , which controls the correspondence term. In Figure 7 the frames of a video of can-can dancers were idealized together. For small  $\alpha_c$  the correspondence between the input frames is weak, which, in this case, results in unnatural deformation of the dancers' legs. Conversely, for large  $\alpha_c$  the correlation between the frames is enforced, and therefore, the deformations do not occur. It is important to note, however, that applying single-image

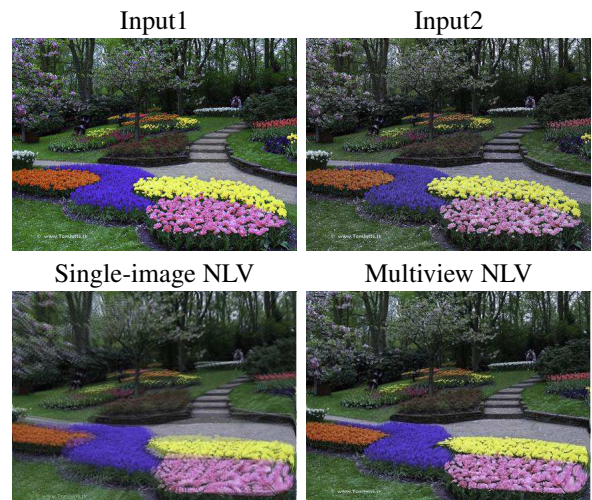


Figure 6: **Different illuminations.** Here the input images were taken from the same viewpoint, but were manipulated to have different illuminations. Blending images that were idealized with single-image NLV yields a blurry image due to the inconsistencies across views. On the contrary, our multiview NLV idealizes both images correspondingly, resulting in a sharp blended image with straightened flower beds and, nicely organized flowers.

NLV to each frame produces much worse results, evident from the corresponding flow-field. The flow-field should capture the dancers' motion, which is the case for both multiview results, and not for the single-image NLV result.

In Figure 8 we show images of a rocky cliff in Yellowstone that presents wavy texture and irregular structure. Applying an idealizing transformation straightens the wavy patterns. When  $\alpha_c$  is small each image is manipulated almost independently, and the corresponding disparity ex-

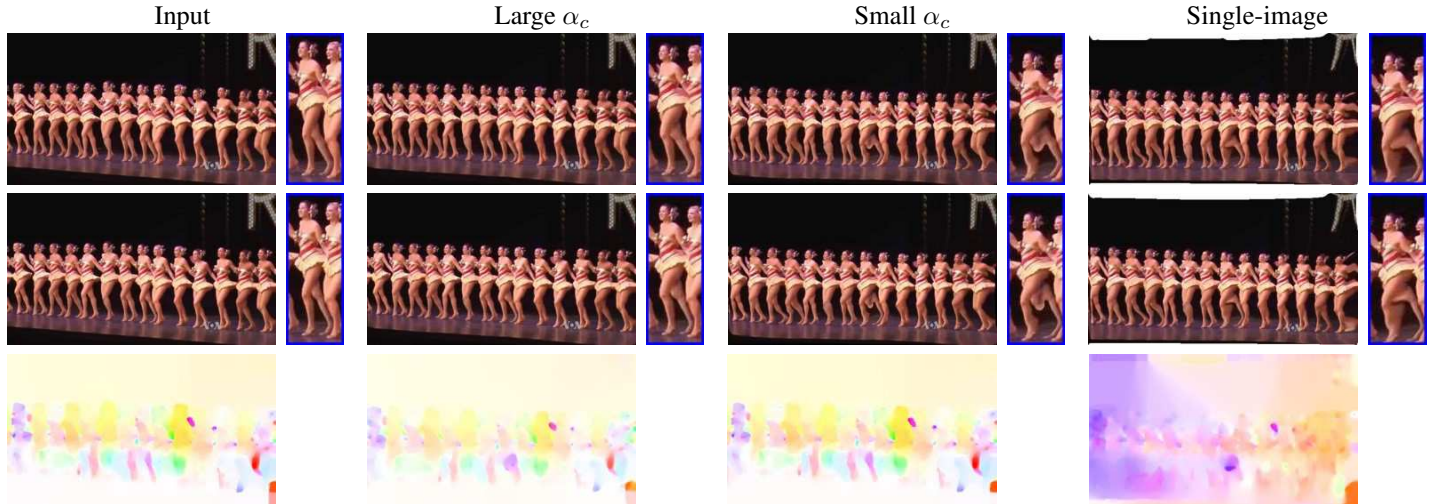


Figure 7: **The effect of parameter  $\alpha_c$ .** Each column shows a pair of images, the flow field between them, and a zoom-in on one of the dancers. Applying NLV to a pair of frames taken from a video of the Rocket Dancers aligns the dancer’s height and pose. Enforcing correspondence between the frames through our multiview NLV approach with small  $\alpha_c$  results in some artifacts, while large value of  $\alpha_c$  constrains the transformation, and hence the dancers are not distorted. In single-image NLV the transformation each image undergoes is utterly different, hence, not only are there distortions within each frame, but also the flow-field between the frames does not match the motion in the scene.

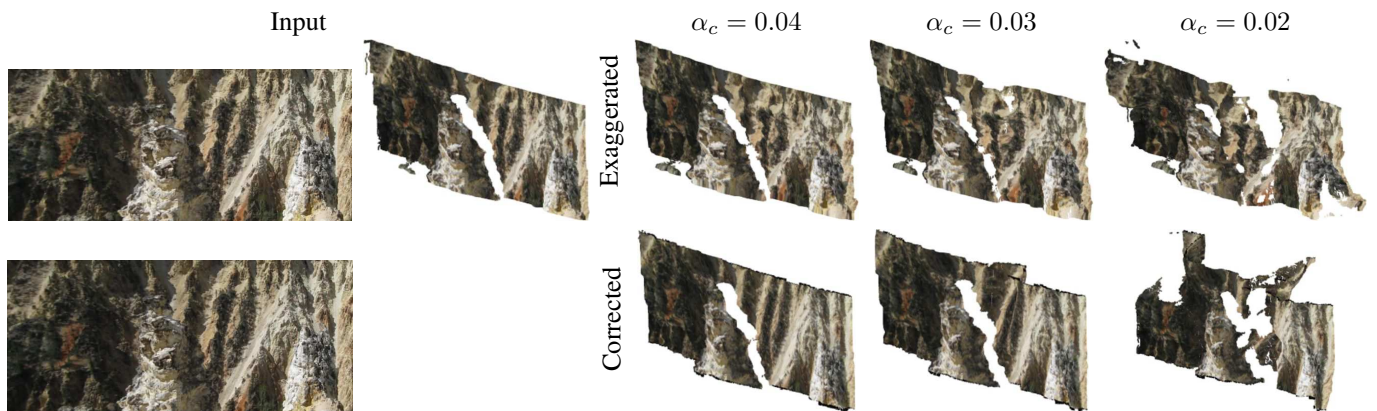


Figure 8: **The effect of parameter  $\alpha_c$ .** When  $\alpha_c$  is small the constraint between the two views is loose, allowing the views to deform differently from each other, which results in a less consistent 3D shape, in both correction and exaggeration.

hibits large depth variations. On the contrary, when  $\alpha_c$  is large the images are manipulated in a synchronous manner, and the corresponding disparity implies a smooth regular shape. Similarly, when exaggerating the variations, the larger  $\alpha_c$  the larger are the depth variations.

## 6. Conclusions

We proposed an approach to revealing, correcting and exaggerating Non-Local-Variations, in multiple views. Our extensive experiments show the necessity of the multiview approach for cases where the same scene is pictured more than once. The multiview correspondence constraint re-

duces artifacts and modifies both color and shape variations in a consistent manner. A future direction we intend to follow is seeking variations in feature space, i.e., rather than modifying image patches we intend to modify their deep features. This could allow us to process images of similar structure but different objects.

**Acknowledgements** This research was supported in part by an Alon Fellowship, the OMEK foundation, the Israel Science Foundation under Grants 1089/16, 852/17, and by the Ollendorf foundation.



## References

- [1] R. Bellini, Y. Kleiman, and D. Cohen-Or. Time-varying weathering in texture space. *ACM Transactions on Graphics (TOG)*, 35(4):141, 2016. 2
- [2] C. S. Burrus. Iterative re-weighted least-squares. *OpenStax-CNX Web site*. <http://cnx.org/content/m45285/1.12>, 2012. 3
- [3] V. Bychkovsky, S. Paris, E. Chan, and F. Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 97–104. IEEE, 2011. 7
- [4] H. Cheng, J. Zhang, Q. Wu, P. An, and Z. Liu. Stereoscopic visual saliency prediction based on stereo contrast and stereo focus. *EURASIP Journal on Image and Video Processing*, 2017(1):61, 2017. 3
- [5] M. Clerc and S. Mallat. The texture gradient equation for recovering shape from texture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):536–549, 2002. 2
- [6] R. Cong, J. Lei, H. Fu, Q. Huang, X. Cao, and C. Hou. Co-saliency detection for rgbd images based on multi-constraint feature matching and cross label propagation. *IEEE Transactions on Image Processing*, 2017. 3
- [7] T. Dekel, T. Michaeli, M. Irani, and W. T. Freeman. Revealing and modifying non-local variations in a single image. *ACM Transactions on Graphics (TOG)*, 34(6):227, 2015. 1, 2, 3, 5
- [8] C. Eisenacher, S. Lefebvre, and M. Stamminger. Texture synthesis from photographs. *Computer Graphics Forum*, 27(2):419–428, 2008. 2
- [9] H. Fu, X. Cao, and Z. Tu. Cluster-based co-saliency detection. *IEEE Transactions on Image Processing*, 22(10):3766–3778, 2013. 3
- [10] A. Fusiello and L. Irsara. Quasi-euclidean uncalibrated epipolar rectification. In *19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008. 6
- [11] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4295–4304, 2015. 5
- [12] H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 807–814. IEEE, 2005. 5
- [13] D. S. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 269–276. IEEE, 2009. 3
- [14] D. E. Jacobs, D. B. Goldman, and E. Shechtman. Cosaliency: Where people look when comparing images. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, pages 219–228. ACM, 2010. 3
- [15] D.-j. Jeong, I. Hwang, and N. I. Cho. Co-salient object detection based on deep saliency networks and seed propagation over an integrated graph. *arXiv preprint arXiv:1706.09650*, 2017. 3
- [16] V. G. Kim, Y. Lipman, and T. A. Funkhouser. Symmetry-guided texture synthesis and manipulation. *ACM Transactions on Graphics (TOG)*, 31(3):22–1, 2012. 2
- [17] H. Li, F. Meng, and K. N. Ngan. Co-salient object detection from multiple images. *IEEE Transactions on Multimedia*, 15(8):1896–1909, 2013. 3
- [18] H. Li and K. N. Ngan. A co-saliency model of image pairs. *IEEE Transactions on Image Processing*, 20(12):3365–3375, 2011. 3
- [19] P. Li, H. Sun, C. Huang, J. Shen, and Y. Nie. Efficient image/video retexturing using parallel bilateral grids. In *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, pages 131–140. ACM, 2011. 2
- [20] C. Liu et al. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009. 5
- [21] C. Liu, A. Torralba, W. T. Freeman, F. Durand, and E. H. Adelson. Motion magnification. *ACM transactions on graphics (TOG)*, 24(3):519–526, 2005. 2
- [22] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *European Conference on Computer Vision*, pages 28–42. Springer, 2008. 5
- [23] X. Liu, L. Jiang, T.-T. Wong, and C.-W. Fu. Statistical invariance for texture synthesis. *IEEE transactions on visualization and computer graphics*, 18(11):1836–1848, 2012. 2
- [24] Y. Liu, W.-C. Lin, and J. Hays. Near-regular texture analysis and manipulation. *ACM Transactions on Graphics (TOG)*, 23(3):368–376, 2004. 2
- [25] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981. 2
- [26] M. Lukáč, D. Šykora, K. Sunkavalli, E. Shechtman, O. Jamriška, N. Carr, and T. Pajdla. Nautilus: recovering regional symmetry transformations for image editing. *ACM Transactions on Graphics (TOG)*, 36(4):108, 2017. 2
- [27] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *International journal of computer vision*, 23(2):149–168, 1997. 2
- [28] L. Mukherjee, V. Singh, and C. R. Dyer. Half-integrality based algorithms for cosegmentation of images. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2028–2035. IEEE, 2009. 3
- [29] L. Mukherjee, V. Singh, and J. Peng. Scale invariant cosegmentation for image groups. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1881–1888. IEEE, 2011. 3
- [30] Y. Niu, Y. Geng, X. Li, and F. Liu. Leveraging stereopsis for saliency analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 454–461. IEEE, 2012. 3
- [31] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching—incorporating a global constraint into mrfs. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 993–1000. IEEE, 2006. 3
- [32] J. C. Rubio, J. Serrat, A. López, and N. Paragios. Unsupervised co-segmentation through region matching. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 749–756. IEEE, 2012. 3

- [33] V. Scholz and M. Magnor. Texture replacement of garments in monocular video sequences. In *Proceedings of the 17th Eurographics conference on Rendering Techniques*, pages 305–312. Eurographics Association, 2006. 2
- [34] D. Tal, I. Shimshoni, and A. Tal. Image-based texture replacement using multiview images. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 185–192. ACM, 2008. 2
- [35] M. Tau and T. Hassner. Dense correspondences across scenes and scales. *IEEE transactions on pattern analysis and machine intelligence*, 38(5):875–888, 2016. 5
- [36] S. Vicente, V. Kolmogorov, and C. Rother. Cosegmentation revisited: Models and optimization. In *European Conference on Computer Vision*, pages 465–479. Springer, 2010. 3
- [37] N. Wadhwa, T. Dekel, D. Wei, F. Durand, and W. T. Freeman. Deviation magnification: revealing departures from ideal geometries. *ACM Transactions on Graphics (TOG)*, 34(6):226, 2015. 2
- [38] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman. Phase-based video motion processing. *ACM Transactions on Graphics (TOG)*, 32(4):80, 2013. 2
- [39] R. White, K. Crane, and D. A. Forsyth. Capturing and animating occluded cloth. *ACM Transactions on Graphics (TOG)*, 26(3):34, 2007. 2
- [40] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM transactions on graphics (TOG)*, 2012. 2
- [41] H. Yu and X. Qi. Unsupervised cosegmentation based on superpixel matching and fastgrabcut. In *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, pages 1–6. IEEE, 2014. 3
- [42] K. Yücer, A. Jacobson, A. Hornung, and O. Sorkine. Transfusive image manipulation. *ACM Transactions on Graphics (TOG)*, 31(6):176, 2012. 2