Encoding Scene Structures for Video Compression

Georgios Georgiadis[a] Avinash Ravichandran[a] Stefano Soatto[a] Alessandro Chiuso[b]

[a]UCLA Vision Lab
University of California, Los Angeles, CA 90095
{giorgos,avinash,soatto}@ucla.edu

[b]Department of Information Engineering
University of Padova, 35131 Padova, Italy
chiuso@dei.unipd.it

ABSTRACT

We describe an approach to partition a video stream into *structure* regions that are temporally encoded and disjoint from *texture* regions, that are synthesized so as to preserve the statistical properties of the original data stream. Structures encode regions of an image that can be put into correspondence in different images of the same scene, and are encoded via a dictionary that takes into account spatial and temporal regularities. Textures are synthesized in a manner that preserves perceptual similarity.

1. INTRODUCTION

The growth of video consumption over Internet and wireless [1] has recently rekindled interest in video compression, specifically towards breaking the compression ceiling of existing algorithms that encode video as a stream of blocks of pixels. Our goal is to develop algorithms that achieve high compression performance relative to a perceptual metric by encoding *not* the video stream directly, but instead the *scene* that generated it, albeit without explicitly reconstructing it.

To this end, it has been noted that, under suitable assumptions, *"structures"* (regions of images that can be associated to a local reference frame) correspond to photometric or geometric characteristics of the *scene* if correspondence can be established across different images of the same scene [2]. This has been exploited in [3] to show that the domain of each image can be partitioned into two disjoint (multiply-connected) sets: Structures and "textures," with the latter corresponding to spatially stationary regions. In this paper we propose a method to improve the spatial localization of such a partition.

1.1 Related work

Our work is naturally related to video compression schemes as reflected in standards such as H.262, H.263, H.264 [4]. While standard methods perform both spatial and temporal prediction of the measured signal, they rarely model the data formation process explicitly in terms of the underlying *scene.* We do not advocate explicit reconstruction of the underlying scene; however, we describe "structures" in images as a function of corresponding structures in space, that under suitable conditions can be inferred from temporal correspondence [3]. Thus, "trackable" regions can be temporally encoded making use of optical flow whereas non-trackable regions can be encoded spatially. The residual of the encoding process can then be encoded as standard. Another approach is followed by [5] where the frames are divided into four types of regions: sketchable and trackable, non-sketchable and trackable, sketchable and non-trackable and non-sketchable and non-trackable. Each type of region is encoded individually taking advantage of its properties.

In this paper, we propose a sequential approach whereby we first detect regions of the image that contain *structures* and are *properly sampled* (Section 1.2). We then exploit the complementarity of *structures* and *textures* [3] to encode the remainder of the image by performing texture segmentation (Section 3.3). We then infer a compressed representation of the textures (Section 3.1) that enables us to synthesize a perceptually similar realization at decoding (Section 3.2). This approach will suffer in traditional evaluation metrics, but is designed to yield perceptually equivalent encoding/decoding at smaller complexity. To encode the structure regions, we create a dictionary by taking into account the spatial and temporal components of such regions. For textured regions we store the compressed representation of each texture (Section 4).

1.2 Preliminaries

To make the paper self-contained, this section summarizes the notation and definitions of [3]. Please refer to [3] for further information. Images $\{I_{ij}\}_{(i,j)=1:(N,M)} \in \mathbb{R}^{M\times N}$ are quantized versions of functions $I : D \subset \mathbb{R}^2 \to \mathbb{R};\ x \mapsto I(x)$ where $\mathcal{B}$ is a neighborhood of $x_{ij} \in D$ of size $\epsilon > 0$. Groups $g : D \subset \mathbb{R}^2 \to \mathbb{R}^2; x \mapsto g(x)$ induce actions $I \circ g \doteq I(g(x))$, for instance translation $T \in \mathbb{R}^2$, via $g(x) \doteq x + T$. Some transformations undergone by (subsets of the domain and range of) an image over time can be represented as groups, for instance translation, Euclidean, similarity, affine, projective, all the way to the diffeomorphic group. This is relevant to compression because the entire orbit of a group transformation can be encoded by a single element of the group. The process of choosing such a "canonical" element is called *canonization* and described in detail in [2]. Succinctly, this is done through the use of a *co-variant detector*, a functional $\psi : \mathcal{I} \times G \to \mathbb{R}; (I, g) \mapsto \psi(I, g)$ that has isolated extrema (loci where $\nabla\psi = 0$) that co-vary with the group, so if $g = \hat{g}(I)$ is such that $\nabla\psi(\hat{g}(I)) = 0$, then $\hat{g}(I \circ g) = (\hat{g} \circ g)(I)$. Examples of (translation-) co-variant detectors are the Laplacian-of-Gaussian (LoG), the difference-of-Gaussians (DoG), the Hessian-of-Gaussian (HoG), the Harris' corner and its variants. Because images are not differentiable, co-variant detectors are typically regularized operators with a scale parameter σ. A region $\Omega \subset D$ is *canonizable* at scale σ if there exists a co-variant detector ψ that has one and only one isolated extremum in Ω at that scale. Note that the same region may be canonizable at multiple scales. The canonization process yields a number of regions each containing exactly one "structure," that is an extremum with respect to the chosen group at the given scale. Such structures may be "real," in the sense of corresponding to some property of the *scene*, or they may be "aliases" due to sampling artifacts, noise, and other phenomena that do not depend on the scene. An image is *properly sampled* at a scale σ if any co-variant detector functional operating on the sampled image $\{I_{ij}\} \in \mathbb{R}^{N\times M}$ at scale σ yields the "same answer" (topology) that it would if run on the "original" (continuous) image $I : D \to \mathbb{R}^+$.

In other words, proper sampling indicates *topological equivalence* of the response of co-variant detectors between the "original" and the sampled image. Unfortunately, we do not have the "original" image. However, under three assumptions (co-visibility, Lambertian reflection and constant illumination), topological equivalence of co-variant detectors responses between the scene and the image can be replaced by topological equivalence of co-variant detector responses between *different images of the same scene* [2]. Testing for proper sampling entails establishing correspondence of covariant selection trees [6].

2. TEMPORAL REDUNDANCY

Proper sampling yields as a byproduct a partition of the image(s) into two regions. Those for which unique correspondence can be established, and the rest. We call the former ones *trackable* regions. They are both canonizable *and* properly sampled. Trackable regions are characterized by the "signature" of each region at the finest scale at which it is tracked, for instance the actual pixel values in a neighborhood of the origin of the tracked frame, as well as the frame itself, for instance position, orientation and scale for the case a similarity reference frame.

To determine trackable regions, we use [6], that requires a detection threshold. The effects of such a threshold are visible in Figure 1, where the number of tracks decreases by increasing the threshold. The ones that persist are usually the most accurate. One can see that almost all trajectories on the sea are eliminated. In later sections we will argue that those regions are spatially stationary and will exploit this property for compression. Trackable regions exhibit temporal redundancy and we would like to exploit that instead.

Co-variant detector functionals can be chosen to canonize a variety of groups, from the simplest (translation) to the most complex (homeomorphisms). The larger the group, the more costly it is to encode, the larger the region that can be encoded. The optimal choice of group depends on the statistics of the images being compressed, and there is no choice that is best for any sequence. For the purpose of illustration, in what follows we will focus on the similarity group of translations, rotations and isotropic scaling. In many cases one can assume that (planar) rotation is negligible and focus on the location-scale group.

Tracking then provide a (moving) reference frame, relative to which one can encode a portion of the region of the image. If the image is undergoing a similarity transformation, no change will be observed in the moving

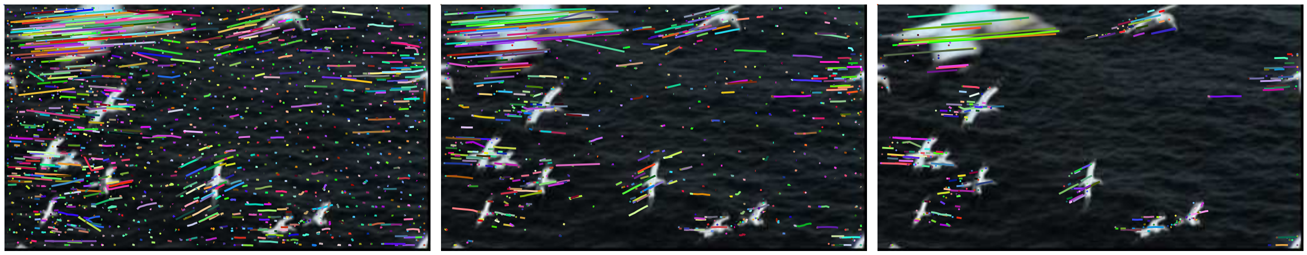

Figure 1: *Varying the co-variant detection threshold produces different numbers of trackable regions. By looking at the number of tracks produced as a function of the threshold it can be observed that there are essentially three regions of interest. The first region corresponds to the tracked points shown in the leftmost image. These tracks contain highly unstable trajectories that provide limited compression benefit. The second region corresponds to the central image Here the quality of the tracks is higher as extended tracks provide improved temporal redundancy. Finally, we have the third case where the number of tracks is significantly lower, but the tracks are much more stable and of higher quality.*

frame. Most typically, however, similarities are not sufficient to explain the complexity of the image even in a small neighborhood and therefore the region will progressively change over time.

A simple approach to encode such changes is to create a dictionary element for each region of the same size as its neighborhood that will be time invariant i.e. constant. The best possible representation would be the average of the pixel values over time. In mathematical notation, if $F(t) \doteq \{I(x,t), x \in \mathcal{B}_\sigma(t)\}$ are the pixel values in an neighborhood $\mathcal{B}$ at scale σ at time t, we represent a trackable region in a dictionary as $T = \frac{1}{N_2 - N_1}\sum_{t=N_1}^{N_2} F(t)$ where N_1, N_2 are the first and last frames that track appears. Hence the scale of the dictionary element is naturally selected to be the finest scale at which the track was detected and its temporal appearance is averaged out to further improve compression rates. Hence the dictionary is composed of elements of different spatial resolutions resulting in a multi scale representation of the trackable regions.

3. SPATIAL REDUNDANCY

We exploit spatial redundancy by encoding stationary regions through their sample statistics. We consider three regions: $\omega, \bar{\omega}$ and Ω. The generator $\omega \subset \mathbb{Z}^2$, with cardinality $|\omega|$, is the smallest region where some statistic ϕ_ω (*"features"*) is defined. A probability distribution $dP(I)$ on the set of images induces a distribution on the feature via the map ϕ_ω: $dP(\phi_\omega) = dP(\phi(I_{|_\omega}))$, that is *G-stationary* if there exists $g \in G$ such that $\mathbb{E}(\phi_{g(\omega)})$ is translation-invariant:

$$\mathbb{E}(\phi_{g(\omega)}) = \mathbb{E}(\phi_{g(\omega)+T}), \quad T \in \mathbb{R}^2 \tag{1}$$

where $g(\omega) = \{g(x) \mid x \in \omega\} \cap \mathbb{Z}^2$ and $g(\omega) + T = \{g(x) + T \mid x \in \omega\} \cap \mathbb{Z}^2$. In order to perform an empirical test, we need a larger region $\bar{\omega} \supset \omega$ where to aggregate the statistics. I is locally stationary in $\bar{\omega}$ if (1) is satisfied *not* for all $T \in \mathbb{R}^2$, but only for those such that $g(\omega) + T \subset \bar{\omega}$. We call such T's *admissible*, and $\sigma = |\bar{\omega}|$ the stationarity scale. The region Ω is the *largest* admissible region where the stationarity assumption is satisfied. In order to perform a stationarity test based on empirical data we rely on the assumption that the underlying process is ergodic, that is

$$\frac{1}{N}\sum_{i=1}^{N} \phi_{g(\omega)}(I_{|_{g(\omega)+T_i}}) \xrightarrow{\text{a. s.}} \mathbb{E}(\phi_{g(\omega)}(I_{|_{g(\omega)}})) \tag{2}$$

for all $T_i \in \mathbb{R}^2$. Assuming that the conditions above are satisfied, stationarity can be tested by considering samples of the image I in $\bar{\omega}$, and comparing them to admissible samples in a transformed domain $\bar{\omega} + \bar{T}$.

Once established that a process is stationary, hence spatially predictable, we can inquire on the existence of a statistic that is *sufficient* to perform the prediction. We say that a process is Markovian if every position $x \in \Omega$ admits a neighborhood $\omega \ni x$, such that a statistic ϕ_ω computed "inside" ω (after excluding x, lest the statement would be a tautology) makes $I(x)$ independent of the "outside" $I_{|_{\omega^c}}$, where ω^c is the complement of ω in Ω:

$$I(x) \perp I_{|_{\omega^c}} \mid \phi_{\omega \setminus x}. \tag{3}$$

Equation (3) establishes $I_{|\omega}$ as a *sufficient statistic.* In general, there will be many regions ω that satisfy this condition; the one with the smallest area $|\omega| = r$, is a *minimal sufficient statistic.* From now on, we will refer to ϕ_ω as the minimal *Markov sufficient statistic.*

With all the notions introduces thus far, we can define a texture as *a region of an image that can be rectified into a sample of a stochastic process of a planar lattice that is locally stationary, ergodic and Markovian [3].*

More precisely, assuming for simplicity the trivial (translation) group, a region $\Omega \subset D \subset \mathbb{R}^2$ of an image is a texture at scale $\sigma > 0$ if there exist regions $\omega \subset \bar{\omega} \subset \Omega$ such that I is a realization of a stationary (Eq. 1), ergodic (Eq. 2), Markovian (Eq. 3) process locally within Ω, with $I_{|\omega}$ a Markov sufficient statistic and $\sigma = |\bar{\omega}|$ the stationarity scale. In this manuscript we will restrict our attention to the case where G is the location-scale group.

In order to infer the description of a texture, we therefore look for Markov sufficient statistics. Without a complexity constraint, there are many regions ω that satisfy the conditions; we therefore seek for the *smallest* one, by solving

$$\hat{\omega} = \arg\min_{\omega} H(I(x)|\omega - \{x\}) + \frac{1}{\beta}|\omega|. \tag{4}$$

Note that this is a consequence of the Markovian assumption and the resulting Markov sufficient statistic satisfies the Information Bottleneck principle [7] with $\beta \to \infty$. In practice, we do not know the probabilistic description of the random field, so the best we can do is to approximate the entropy in (4) from sample data:

$$H(I(x)|\omega - \{x\}) \simeq -\frac{1}{|\Omega|}\sum_{i=1}^{|\Omega|} \log p(I(x_i)|\omega_i - \{x_i\}) \tag{5}$$

where ω_i is a neighborhood of x_i. Here p can either be finitely parameterized or specified in a non-parametric fashion by samples in a region $\bar{\omega}$, with $\omega \subset \bar{\omega} \subset \Omega$. For instance, given $\bar{\omega}$ we can draw a number K of regions of size $r = |\omega|$. The larger r, the smaller K, so we can write $K = K(r, \sigma)$ with $\sigma = |\bar{\omega}|$. For instance, if $\bar{\omega}$ is a square neighborhood of side σ, then $K = 4r\sigma - 4r^2 + 1$.

To compute $\log p(I(x)|\omega - \{x\})$, we search among $k = 1, \dots, K(r, \sigma)$ for the region ω_k that best matches a neighborhood ω of x:

$$\hat{k} = \arg\min_{k=1:K} d(I(\omega_k - \{x_k\}), I(\omega - \{x\})) \tag{6}$$

where d is some distance among images restricted to their neighborhoods. If we call $\hat{I}(x) = I(x_{\hat{k}})$, we approximate $-\log p(I(x)|\omega - \{x\}) \simeq d(I(x) - \hat{I}(x))$ which yields an estimator of the entropy:

$$\hat{H}(I(x)|\omega - \{x\}) \doteq \frac{1}{|\Omega|}\sum_{i=1}^{|\Omega|} d(I(x_i), \hat{I}(x_i)). \tag{7}$$

Note that this function depends on both $r = |\omega|$ as well as $\sigma = |\bar{\omega}|$. The larger $\bar{\omega}$ the better the estimate, so we must trade off σ. However, the size of ω is automatically traded off in $K(r, \sigma)$: Choosing $r = \sigma$ will yield only one sample $K = 1$, and therefore the prediction error $d(I(x) - \hat{I}(x))$ will be large. Similarly, too small a σ will cause many false matches of $I(\omega - \{x\}))$ with poor predictive power for $I(x)$. The tradeoff will naturally settle for $1 < r < \sigma$. Therefore, we can simultaneously infer both σ and r by minimizing the sample version of (4) with a complexity cost on $\sigma = |\bar{\omega}|$:

$$\hat{\omega}, \sigma = \arg\min_{\omega, \sigma = |\bar{\omega}|} \hat{H}(I(x)|\omega - \{x\}) + \frac{1}{\beta}|\bar{\omega}|. \tag{8}$$

Note that both ω and $\bar{\omega}$ will be necessary for extrapolation: ω defines the Markov neighborhood used for comparing samples, and $\bar{\omega}$ defines the region where such samples are sought to approximate the probability distribution $p(I(x)|\omega - \{x\})$.

Eq. (8) provides means to infer both the Markov sufficient statistic $I(\hat{\omega})$ as well as the scale $\sigma = |\bar{\omega}|$ of a texture, assuming that the stationarity and ergodicity assumptions are satisfied. Testing for stationarity (ergodicity must be assumed and cannot be validated) amounts to inferring Ω, a *texture segmentation* problem. An extrapolation algorithm is also described. Below we provide the algorithms developed for these procedures.

3.1 Compression

Given $\{I(x), x \in \Omega\}$, compression is achieved by inferring the (approximate) minimal sufficient statistic ω and the stationarity scale σ by solving (8). Then $I(\bar{\omega})$, for any $\bar{\omega} \subset \Omega$ with $|\bar{\omega}| = \sigma$ is stored. To infer the unknowns, we parametrize both ω and $\bar{\omega}$ to be square neighborhoods. In addition, we approximate $d(I(x_i), \hat{I}(x_i))$ with the KL-divergence between the distributions of pixel values around the neighborhoods of x_i and $x_{\hat{k}_i}$ (Alg. 1 [3]).

Algorithm 1: Compression algorithm.

```
Initialize $\sigma = [\sigma_1, \sigma_2, \dots, \sigma_N], r = [r_1, r_2, \dots, r_N]$
foreach $\sigma$ do
    Sample square patches $\bar{\omega}$ of size $\sigma$ from $\Omega$
    foreach $\bar{\omega}$ do
        foreach $r$ do
            Sample square patches $\omega$ of size $r$ from $\bar{\omega}$
            Compute $\hat{H}(I(x)|\omega - \{x\}) + \frac{1}{\beta}|\bar{\omega}|$
Let $\hat{\omega}, \sigma = \arg\min_{\omega, \sigma = |\bar{\omega}|} \hat{H}(I(x)|\omega - \{x\}) + \frac{1}{\beta}|\bar{\omega}|$
```

3.2 Extrapolation

Given a compressed representation $I(\bar{\omega})$, we can in principle synthesize novel instances of the texture by sampling from $dP(I_{|\omega})$ within $\bar{\omega}$. In a non-parametric setting this is done by sampling directly neighborhoods $I(\omega)$ within $\bar{\omega}$. To extrapolate the texture from a given sample $I(\bar{\omega})$ compatibility conditions have to be ensured at the boundaries of $\bar{\omega}$. Hence, to satisfy both appearance and compatibility conditions, we minimize the following energy function [8]:

$$E(I_s; I(\bar{\omega})) = \sum_{\omega \in N(B)} \|I_\omega - \hat{I}_\omega\|^2 + \mu \sum_{\omega \in N(B)} \|DI_\omega - D\hat{I}_\omega\|^2 \tag{9}$$

where I_s is the textured region to be synthesized and $I(\bar{\omega})$ is the input texture. The vectors I_ω and $\hat{I}_\omega$ are neighborhoods of the input and synthesized textures respectively, centered at the central pixel of ω's neighborhood. D is the differentiation operator. In principle, $N(B)$ can be all possible ω neighborhood's within $\bar{\omega}$. For computational complexity reasons, we only consider a subset of them. In [8] neighborhoods are sampled on a grid. We sample at randomly selected locations similar to [3]. The energy above is minimized alternating direct differentiation with respect to $\hat{I}_\omega$ (that yields a linear system of equations) and using the result to determine the set $\{I_\omega\}$ using nearest-neighbor (NN) search. The process is iterated until the energy decrease is negligible, as customary. In addition, a re-weighting scheme is used to improve outlier rejection (similar to iteratively re-weighted Least Squares (IRLS)). Note that in [8] the neighborhood sizes were set manually. In our scheme we select them automatically based on the size of $\bar{\omega}$ calculated using Alg. 1 [3]. We repeat the procedure over 3 neighborhood sizes: $\{|n_i| : i = 1, 2, 3\} = [r, 2r, 3r]$. Finally, the minimization procedure is repeated over a number of different output image sizes. The relative weight parameter is set to $\mu = 10$.

3.3 Segmentation

Given an image I we want to find Ω_i for the different textures in the image. The algorithm (Alg. 2, [3]) requires knowledge of the number of regions to be segmented (model selection) that can be determined using Agglomerative Information Bottleneck (AIB) [9]. Note that it is possible for a texture region to not have a well-defined boundary. In that case, boundary compatibility conditions have to be imposed with other structures in the image, as we discuss in Section 4.

In order to perfect the encoding, we rely on the partition of the image into texture regions and structure regions, that relies on the analysis in [3]. Consider a point $x \in D$ and its neighborhood. If it is canonizable at a scale ϵ, there is a co-variant detector with support ϵ (a statistic) that has an isolated extremum. This implies that at a scale $|\bar{\omega}| = \epsilon$, it is not possible to capture the variability of the texture and, as such, it is not possible

Algorithm 2: Segmentation algorithm.

```
Initialize $N = 12, M = 30, K$
Sample (overlapping) square patches $\omega$ of sides $N$ on a dense grid from the image
Let $\{\omega\}$ be the set of sampled $\omega$'s from previous step
foreach $\omega$ do
    Calculate a histogram of intensity values with $M$ bins for each patch
Calculate an empirical probability $p(M|\{\omega\})$
Use AIB [9] to sequentially merge $\omega$'s that that decrease $I(\{\omega\}, M)$ the least
Cut the tree built by AIB in $K$ clusters to obtain K-clustering of the $\omega$'s
foreach pixel $x_i$ do
    Calculate in which $\omega$'s, $x_i$ falls in
    Find which cluster $K$, $x_i$ belongs to, using a max vote of the memberships of the $\omega$'s it falls in
Form regions $\Omega_i$ based on clustering of pixels
```

to model it as a texture. It also implies that any region ω of size $\epsilon = |\omega|$ is not sufficient to predict the image outside that region.

This of course does not prevent a region that is canonizable at ϵ to be a texture at a scale $\sigma >> \epsilon$. Within a region σ there may be multiple frames of size ϵ, spatially distributed in a way that is stationary/Markovian. Vice-versa, if a region of an image is a texture with $\sigma = \bar{\omega}$, it cannot have a unique (isolated) extremum within $\bar{\omega}$, lest it would not be a sample of a stationary process. Of course, it could have multiple extrema, each isolated within a region of size $\epsilon << \sigma$.

Thus for any given scale of observation σ, a region $\bar{\omega}$ with $|\bar{\omega}| = \sigma$ is either a structure or a texture. Hence one can detect textures for each scale, as the residual of the canonization process described in Sect. 1.2. One may have to impose boundary conditions so that the texture regions fill around structure regions seamlessly. In the next section we describe how this is done in our framework.

4. EVALUATION

Once we perform co-variant detection and proper sampling we encode the trackable regions by their corresponding dictionary element as discussed in Section 2. A typical result is shown in the first image in Figure 2. The rest of the domain of the frame is a candidate for texture, following the complementarity condition of Theorem 1 in [3]. We perform texture segmentation using Algorithm 2 and we obtain the second image of Figure 2. For AIB we use the implementation from [10]. Structured regions are excluded from the segmentation and the segmentation algorithm is restricted to the rest of the domain.

It can be observed that the tracking mechanism fails to track some of the structure regions. This is unavoidable as the texture/structure partition is an early commitment based on low-level statistics. It is therefore important that the subsequent stages of processing can compensate for such unavoidable errors. In particular, in the example above, trying to synthesize those regions by the algorithm described in Section 3.2 will fail. Therefore, a stationarity test needs to be applied to verify the validity of the texture assumption before texture synthesis. This test rejects the regions within the blue boundaries, that are therefore excluded from the synthesis process, and encoded as structures instead. Since the tracking mechanism often fails to detect small structures, but small structures can be perceptually salient, such a *repechage* mechanism is critical to the successfully encode complex scenes. In the third image of Figure 2 we show the domain of the regions that have failed the stationarity test and that were initially not detected by the co-variant detection mechanism.

The collection of trackable regions is then updated, and the final partition is shown in the fourth panel in Figure 2. We then reiterate the texture segmentation routine to obtain the different textured regions in the frame. We compress each texture region using Algorithm 1. A typical result is shown in the sixth panel of Figure 2. Finally we store the dictionary and the locations of the trackable regions and the compressed representations of the textures. When more than one texture is detected in a video frame, we also store the texture boundaries.

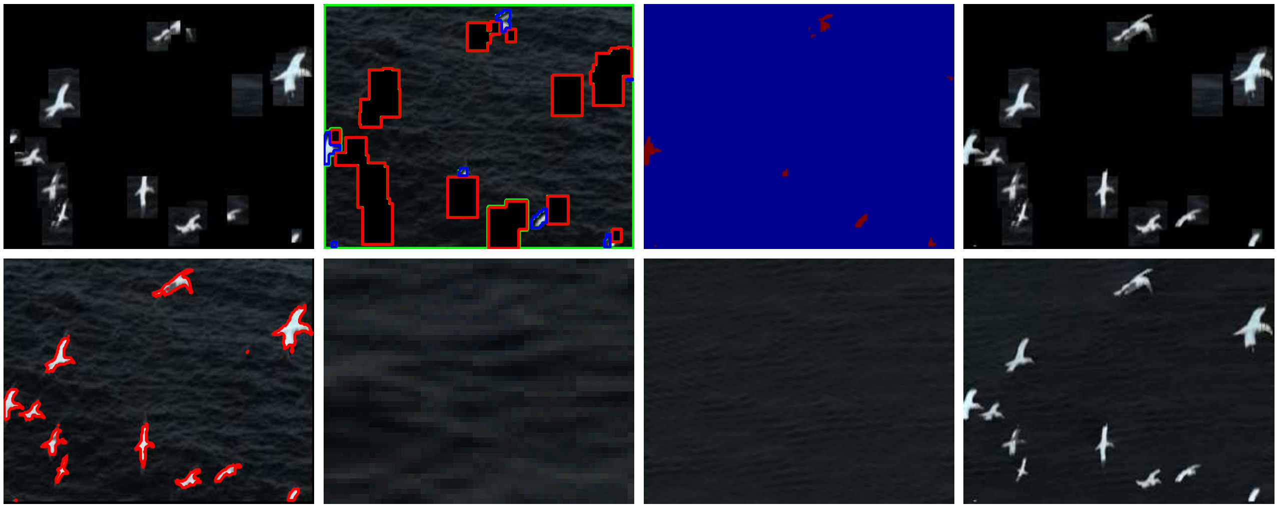

Figure 2: *Pipeline. From left to right, top to bottom: (1) Tracked regions map. (2) Initial segmentation (red indicates boundaries of tracked regions, blue and green indicate boundaries of texture region candidates). (3) Regions of the image that did not satisfy the stationarity assumption underlying the texture hypothesis. (4) Updated tracked region map. (5) Final structure / texture partition. (6) Compressed textured region. (7) Synthesized textured region. (8) Natural blending of textures and structures.*

At decoding time, we synthesize the textures using Algorithm 2 (seventh image of Figure 2) and overlay the trackable regions using the representation stored in the dictionary. Representative results of this procedure are shown in the central column of Figure 3. It is immediate to see that boundary conditions are not matched across the texture/structure partition, resulting in salient perceptual artifacts. In order to enforce compatibility at the boundary in a way that is consistent with the statistics of natural images, we determine pixel-level boundaries of the textured regions, and restrict the domain of the texture representation to this region, leaving the texture synthesis algorithm to explain the remainder. To determine the boundary, we perform texture synthesis on the entire domain, and restrict structures to regions that exhibit large residuals. Hence we can build accurate boundaries for the texture as shown in the fifth panel of Figure 2. To enforce prior knowledge we have of the statistics of natural images, we exploit boundary gradients to produce a gradient field that is (approximately) integrable, a typical property of natural images [11]. We use the algorithm proposed by [11] and typical results are shown in the right column of Figure 3.

We also compare our method with [5] in Figure 4. It can be seen that their method misses important structures that are detected by our approach.

5. CONCLUSIONS

We have described an encoding of structure regions using a dictionary representation, and exploited the partition of the image domain into structures and textures to exploit both spatial and temporal redundancy for video compression. Our approach is an extension of [3], but improves the results therein by considering a tight partition respecting object boundaries and the statistics of natural images.

ACKNOWLEDGMENTS

Research supported by ONR N000141110863 and DARPA MSEE FA8650-11-1-7156.

References

[1] CISCO, “Entering the zettabyte era, visual networking index.” CISCO VNI, 2011.

[2] S. Soatto, *Steps Toward a Theory of Visual Information*, ArXiv http://arxiv.org/abs/1110.2053, Technical Report UCLA-CSD100028, September 13, 2010.

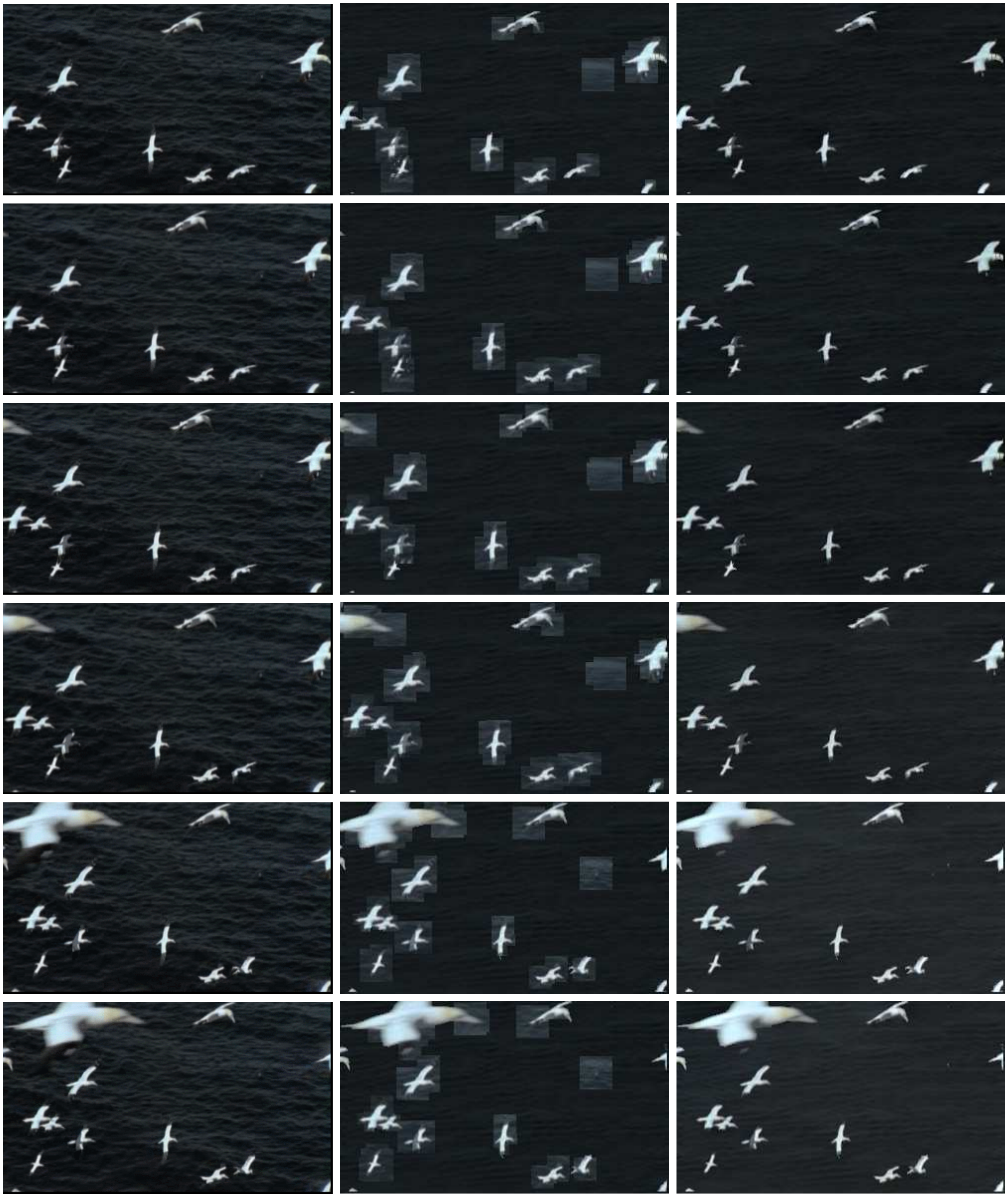

Figure 3: *Samples from the "bird-sea" sequence. Left: Input sequence, Center: Overlaid tracked regions and synthesized textures. Right: Natural blending of synthesized textures and tracked regions.*

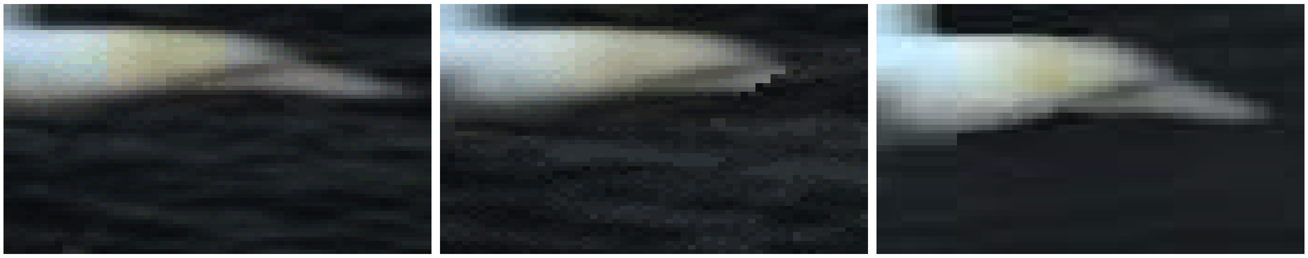

Figure 4: *Reconstruction of structured regions. Left: Input frame, Center: Video Primal Sketch, Right: Our method. Our method successfully preserves salient regions.*

[3] G. Georgiadis, A. Chiuso, and S. Soatto, "Texture, structure and visual matching," *Submitted to NIPS* , 2012.

[4] ITUT-Recommendations, "http://www.itu.int/itu-t/recommendations/," 2011.

[5] H. Zhi, Z. Xu, and S. Zhu, "Video primal sketch: A generic middle-level representation of video," *ICCV* , 2011.

[6] T. Lee and S. Soatto, "Learning and matching multiscale template descriptors for real-time detection, localization and tracking," *CVPR* , 2011.

[7] N. Tishby, F. Pereira, and W. Bialek, "The information bottleneck method," *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing* , 1999.

[8] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," *Proc. of ACM SIGGRAPH* , 2005.

[9] N. Slonim and N. Tishby, "Agglomerative information bottleneck," *NIPS* , 1999.

[10] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," 2008.

[11] M. Tao, M. Johnson, and S. Paris, "Error-tolerant image compositing," *ECCV* , 2010.