DISTRIBUTED CODING OF SHIFTS USING THE DFT PHASE

Marco Dalai [1], *Riccardo Leonardi* [1], *Pier Luigi Dragotti* [2]

[1]Department of Electronics for Automation, University of Brescia, Italy
email: {marco.dalai, riccardo.leonardi}@ing.unibs.it
[2]Electrical and Electronic Engineering, Imperial College, London SW7-2BT, UK
email: p.dragotti@imperial.ac.uk

ABSTRACT

In this paper we consider the problem of image encoding with side information at the decoder, where the side information is an integer shifted version of the image at the encoder. The encoder is asked to send the shift of its own image with respect to the side information which is only available at the decoder. We propose a solution based on the encoding of the phase sign of the DFT coefficients, taken at exponentially spaced positions. We first introduce the method under ideal hypothesis, i.e. noiseless conditions without border effects, giving a theoretical foundation to the technique. Then, we consider the more realistic case of noisy images with border effects, showing the effectiveness of the proposed method.

***Index Terms*—** Image processing, video signal processing.

1. INTRODUCTION

The field of distributed source coding (DSC), initiated by Slepian and Wolf in their famous work [1], has received a lot of attention in recent years. Many researchers have been investigating the possible application of the theoretical results to concrete problems, distributed image and video coding being probably the most studied ones. The idea of using DSC methods for single source video coding was first proposed independently in [2] and [3], see [4] for an overview of the problem. The problem of distributed coding of images was studied for example in [5] while the method proposed in [2] was applied to this problem in [6]. A different approach was proposed in [7] and further works have been studying the problem of distributed video coding in multi-camera systems, see for example [8].

A fundamental problem encountered in both the fields of distributed image and video coding is the need of performing compensations at the decoder. In the case of single camera video sequences, for example, a classic non-distributed coding technique consists in performing a motion compensation first and then encoding the prediction error. In a distributed system the motion compensation must be performed at the decoder, without having actually access to the frame to be predicted. A similar fact holds for the problem of disparity compensation at the decoder. Consider that motion and disparity compensations are combinations of simple "shift" compensations on different portions of frames. So, in order to study an efficient way to perform motion and disparity compensation at the decoder, we should first find a solid solution to the problem of performing shift compensation.

In this paper we are thus interested in a simple problem of "distributed shift encoding" which can be summarized in the following way. Two images x and y are obtained by cropping a common scene from two displaced positions. Supposing that the y image is available at the decoder, we want to find an efficient strategy for communicating to it the shift of x with respect to y.

In the whole paper we use the following notations: '$\log(\cdot)$' is the base-2 logarithm; for an integer m, '$\{\cdot\}_m$' indicates the modulo-m operation; the symbol '$\overset{2\pi}{=}$' indicates a modulo-2π congruence and we consider phases to always take values on the interval $[-\pi, \pi]$.

2. DISTRIBUTED SHIFT CODING: 1-D CASE

Suppose we have two N-point signals $x(\cdot)$ and $y(\cdot)$ which differ only by a circular shift s, with $0 \leq s < S$, $S < N$, i.e.:

$$x(n) = y(\{n-s\}_N), \quad n = 0, 1, \ldots, N-1.$$

For the sake of simplicity, let us consider the case when both N and S are powers of 2. Suppose an encoder has to communicate x to a decoder, using y as side information. If y is available to both encoder and decoder, and if s is uniformly distributed between 0 and $S-1$, then $\log(S)$ bits are needed for encoding x, as it is only necessary to specify the value of s. Suppose now y is only available to the decoder. Supported by distributed source coding theory, one may wonder whether it is still possible to encode x - or equivalently s - using only $\log(S)$ bits. We now prove that this is indeed possible and, in addition, that the solution is not even unique.

First note that if the shape of x and y is *a priori* known to both encoder and decoder, then the problem is quite trivial. It is only necessary that the encoder and the decoder agree on one particular point p of the shape and use the following strategy. Let p_x and p_y be the position of p in x and y respectively; the encoder sends the value of $\{p_x\}_S$ and the decoder estimates s as $\hat{s} = \{p_y - \{p_x\}_S\}_S$. The obtained result satisfies $0 \leq \hat{s} < S$ and it is congruent to s modulo S, so that we necessarily have $\hat{s} = s$.

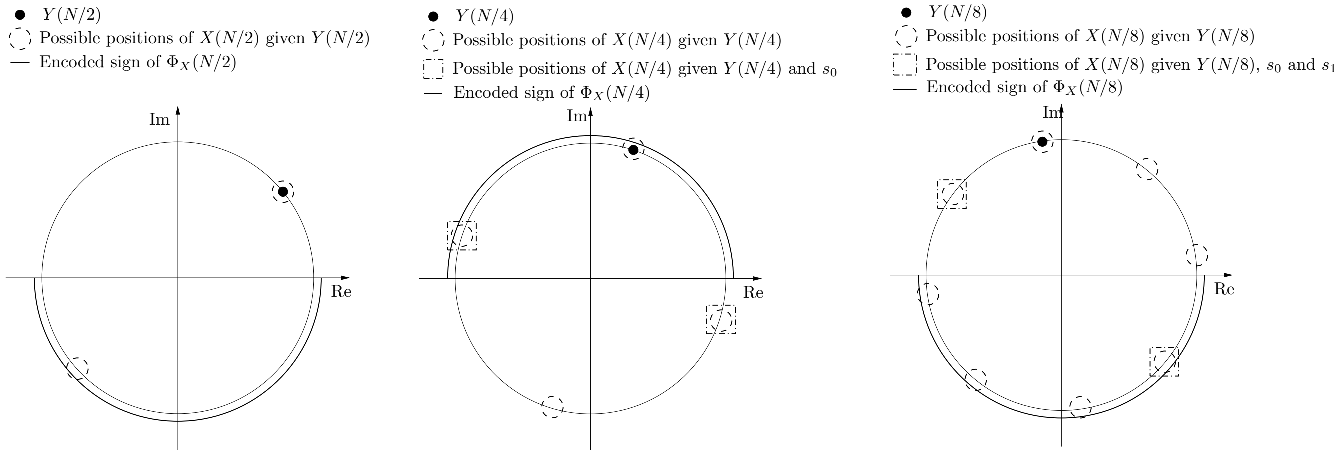
On the contrary, if the shape of x is not known *a priori*, the problem becomes more interesting and it must be treated in a different way. An immediate idea is to work in the DFT phase domain. Let $X(\cdot)$ and $Y(\cdot)$ be the DFT of x and y respectively, and, for every k, let $\Phi_X(k)$ and $\Phi_Y(k)$ be the phase of the coefficient $X(k)$ and $Y(k)$ respectively. From the shift hypothesis on x and y, the phases of the DFT are related by the following equation

$$\Phi_X(k) \overset{2\pi}{=} -\frac{2\pi sk}{N} + \Phi_Y(k). \tag{1}$$

In this section we show how it is possible to extract few bits from the DFT phase so as to communicate the shift from encoder to decoder.

First note that, if we take $k = 1$, we have

$$\Phi_X(1) \overset{2\pi}{=} -2\pi\frac{s}{N} + \Phi_Y(1). \tag{2}$$


● $Y(N/2)$
Possible positions of $X(N/2)$ given $Y(N/2)$
— Encoded sign of $\Phi_X(N/2)$
Im
Re
● $Y(N/4)$
Possible positions of $X(N/4)$ given $Y(N/4)$
Possible positions of $X(N/4)$ given $Y(N/4)$ and s_0
— Encoded sign of $\Phi_X(N/4)$
Im
Re
● $Y(N/8)$
Possible positions of $X(N/8)$ given $Y(N/8)$
Possible positions of $X(N/8)$ given $Y(N/8)$, s_0 and s_1
— Encoded sign of $\Phi_X(N/8)$
Im
Re

(a) Decoding of bit $s_0 = 1$

(b) Decoding of bit $s_1 = 1$

(c) Decoding of bit $s_2 = 0$

Fig. 1. Procedure for the decoding of the first three bits of s. In this case we had $s = 3$.

Now, given that $s < N$, for every value of s the value on the right hand side of the eq. (2) determines a different point in the range $[-\pi, \pi]$, and the phases obtained for different values of s differ by integer multiples of $2\pi/N$. So, in theory, if a quantization $\hat{\Phi}_X(1)$ of $\Phi_X(1)$ into $2\pi/N$-width intervals is known at the decoder, then by using the value of $\Phi_Y(1)$ it is possible to recover the value of s. Of course, in this case, $\hat{\Phi}_X(1)$ takes on N different values; anyway, given that $s < S$, only the value of $\{\hat{\Phi}_X(1)\}_S$ is really needed at the decoder. So, only $\log(S)$ bits are required in order to quantize $\Phi_X(1)$ so that the decoder can recover the value of s. A careful analysis shows that this method is not substantially different, from a theoretical point of view, from the previously mentioned technique involving the use of p_x and p_y. The advantage is that this second method can be used unaltered independently from the shape of x.[1]

This strategy based on the quantization of $\Phi_X(1)$, even if it is theoretically valid under the assumed ideal hypothesis, has some disadvantages in terms of robustness, because it is based on an arbitrarily precise evaluation of the phase of one coefficient. In the presence of noise, or in the more concrete case where "non-circular" shifts are involved, some phase "errors" are usually introduced, and in the above scheme even a small error can cause a wrong extraction of the value of s.

Here we propose a different method to extract the shift value, which is based on a coarse quantization of more coefficients, rather than on a fine quantization of only one coefficient. Let us consider the phase of DFT coefficients taken at exponentially spaced positions, i.e.

$$\Phi_X(N/2), \Phi_X(N/4), \Phi_X(N/8), \ldots, \Phi_X(N/S).$$

We show that a 1-bit quantization of the above phases - for a total amount of $\log(S)$ bits - suffices to recover the value of s at the decoder.

Let us write the binary representation of s as $s_q s_{q-1} \cdots s_1 s_0$, $s_i \in \{0, 1\}, i = 0, \ldots, q$. First consider the $N/2$-th DFT coefficient. For this coefficient eq. (1) becomes

$$\begin{aligned}\Phi_X(N/2) &\overset{2\pi}{=} -\pi s + \Phi_Y(N/2)\\ &\overset{2\pi}{=} -\pi s_0 + \Phi_Y(N/2).\end{aligned}$$

It is clear that when $\Phi_Y(N/2)$ is known, the sign of $\Phi_X(N/2)$ uniquely determines the value of s_0. So, one bit extracted from $\Phi_X(N/2)$ (i.e., the sign) allows to determine the least significative bit of s at the decoder (see fig. 1(a) for an example). Now, by using an iterated procedure, we show by induction that the binary representation of s can be reconstructed from the signs of the considered coefficients see fig. 1(b) and 1(c) for an example). In fact, supposing that the bits $s_0, s_1, \ldots, s_{h-1}$ has been determined using the signs of $\Phi_X(N/2), \Phi_X(N/2^2), \ldots, \Phi_X(N/2^h)$, and consider the coefficient $X(N/2^{h+1})$, we have that

$$\begin{aligned}\Phi_X(N/2^{h+1}) &\overset{2\pi}{=} -\pi\frac{s}{2^h} + \Phi_Y(N/2^{h+1})\\ &\overset{2\pi}{=} -\pi s_h - \frac{\pi}{2^h}\{s\}_{2^h} + \Phi_{\hat{Y}}(N/2^{h+1}).\end{aligned}$$

Now, clearly $\{s\}_{2^h} = s_{h-1} \cdots s_1 s_0$ is known to the decoder, so that the only unknown term in the right hand side of the above equation is s_h. So, again, the sign of $\Phi_X(N/2^{h+1})$ uniquely determines s_h. This proves that the $\log(S)$ bits that represent the signs of the phases $\Phi_X(N/2^i), i = 1, \ldots, \log(S)$, allow the decoder to reconstruct the value of s.

3. 2-D AND APPLICATION TO IMAGES

3.1. Ideal case

In this section we apply the theoretical development presented in the previous section to the practical problem of encoding the relative shift between images. We first consider the ideal case where an image x is obtained by applying a 2-dimensional circular shift to an N by N image y. If $\mathbf{v} = (r, c)$ is the shift vector, where $0 \leq r < R$ and $0 \leq c < C$, with $R < N$ and $C < N$, the relation between the images is

$$x(n, m) = y(\{n - r\}_N, \{m - c\}_N), \quad n, m = 0, 1, \ldots, N - 1,$$

[1] Actually this is not true in some pathological cases we are not interested in. E.g., if the $X(1)$ coefficient is exactly zero the method cannot be applied.

and the relation between the 2-dimensional DFT's is now given by

$$\Phi_X(k,l) = -j\frac{2\pi kr}{N} - j\frac{2\pi lc}{N} + \Phi_Y(k,l). \tag{3}$$

It is easy to see from the above equation that the problems of determining r and c can be separated. In fact, by taking for example $l = 0$, we cancel the term including c, and we reduce eq. (3) to an equivalent of eq. (1), where r plays the role of s. So, by taking respectively $l = 0$ and $k = 0$, we can solve the problem of encoding/decoding r and c independently. By applying the same technique exposed in the previous section, the only required bits can thus be extracted from the DFT of x as the signs of the phases of vertical and horizontal frequencies, i.e.,

$$\Phi_X(N/2,0), \Phi_X(N/4,0), \ldots, \Phi_X(N/R,0),$$
$$\Phi_X(0,N/2), \Phi_X(0,N/4), \ldots, \Phi_X(0,N/C).$$

In this case, the total amount of required bits is $\log(R)+\log(C)$. So, the 2-dimensional problem in the ideal situation of noiseless circular shifts is optimally solved exactly in the same way as in the 1-D case.

3.2. A more realistic scenario: adding redundancy

Now we apply the above insight to a more realistic situation where the two images x and y are obtained by cropping a common scene from two shifted positions. In this case, with respect to the ideal setting considered before, the shift between x and y is not a circular one; moreover, the two images are affected by noise. We model this fact by saying that there is a scene $z(n,m)$ and independent noises n_x, n_y such that

$$y(n,m) = z(n,m) + n_y(n,m), \tag{4}$$
$$x(n,m) = z(n-r, m-c) + n_x(n,m). \tag{5}$$

An important element to clarify is that, under these different assumptions, we are not anymore interested in using exactly $\log(R) + \log(C)$ bits in order to encode the shift. In fact, due to the noise and to border effects, it is reasonable to allow the use of more bits in order to encode the shift. Moreover, in this case, the values of R and C are assumed to be much smaller than N, because when R and C get comparable with N the overlap between the x and y image gets smaller and smaller. Finally, it is reasonable to assume that the number of required bits to encode the shift may depend on the strength of the additive noise on the x and y images. So, for this practical situation, we relax the problem to more informal constraints and we aim at finding a robust strategy in order to use a small number of bits to encode the shift between the images.[2]

The main idea for encoding the shift in this practical situation, then, is to use the insight given by the theoretical development proposed for the ideal case and "extend" the technique by increasing its robustness. In order to do this, it is necessary to add redundancy to the encoded data, as is usually done in channel coding. In our scheme, when we considered the phase relation expressed by eq. (3), we noted that it is possible to solve the problem separately for r and c by putting $l = 0$ and $k = 0$ respectively, so as to use a minimum number of bits. Now, given that we are looking for robustness, it is very useful to go in the opposite direction and note the fact that when l and k are both different form zero the value of the resulting phase is affected by both r and c. So, if instead of using only the coefficients associated to vertical and horizontal frequencies, as in eq.'s (4), (4), we also consider "diagonal" frequency phases of the form $\Phi_X(N/2^i, N/2^j)$, we actually add some sort of "parity-check" to the code.

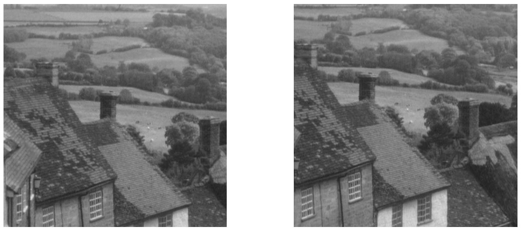

(a) Image x (b) Image y

Fig. 2. Example of 256x256 x and y images cropped from the 512x512 *"goldhill"* image. Here we have $r = 21$, $c = 36$ and n_x and n_y are independent white gaussian noises with $\sigma_{n_x} = \sigma_{n_y} = 2$. In this case 69 bits suffice to correctly detect the shift with the computationally light decoder (in less than 1 second), while 39 bits suffice in the case of the complex decoder (in more than 300 seconds).

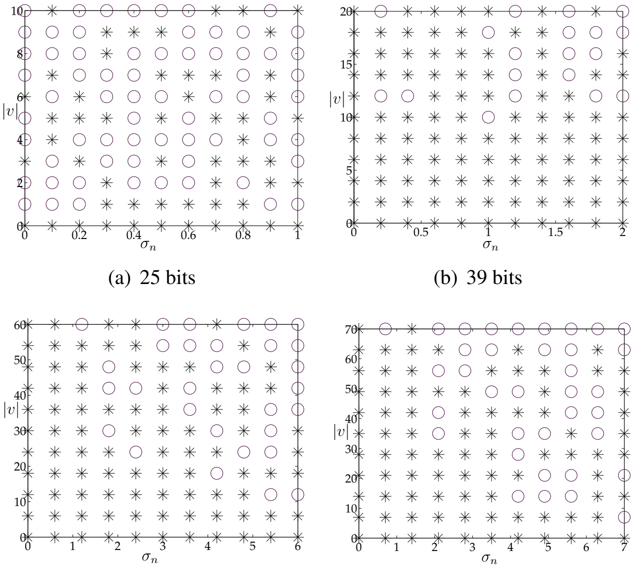
So, we need to extend the initial idea and to consider the general case where we encode the sign of the phases of coefficients $\Phi_X(k,l)$ for values of k and l that are either 0 or powers of 2. In this case anyway the procedure for the decoding of the bits of the shift becomes much more involved and it is not possible to use a decoding technique as the one described for the ideal case. Here we actually find that the performance of the coding technique is strongly related to the computational complexity of the decoder.

In our problem, we consider full search methods where all possible values of r and c are tested so as to find the most plausible shift, doing the equivalent operation of a minimum distance decoding in channel coding. Here we propose two different full-search methods which are associated to two different computational complexities for the decoder, the more complex method having of course better performance.

The main idea, which is common to both decoding techniques, is that, theoretical discussions apart, we can see the bits extracted from the phase of the x image as a hash of the image. At the decoder, what we want to do is to estimate the shift that, applied to y, gives an image with a hash similar to that of x. Actually, the image x and the shift-compensated y will always coincide only in the central part, as we cannot recreate at the decoder the portion of the x image located on the disappeared border. So, in order to smooth the border effects we can apply smoothing windows on the x and y images. For the x image, the way the windowing operation is performed is not an issue; we simply multiply the x image by the window before performing the DFT operation. The way this smoothing window is used at the decoder, instead, makes the difference between the complex and the light methods proposed here.

We start by describing the more complex technique, which is somehow also the most obvious one. The decoder consider all possible pairs of (r,c) values; for every one of them a circular shift by a (r,c) vector is applied to y. The resulting image is multiplied by the window so at to remove the border effects, it is transformed, and the signs of the phase of the DFT coefficients are extracted. The hamming distance of the obtained code from the one extracted from x is then computed, and the values of r and c that minimize this distance are kept as best estimate of the true shift components. Note that with this technique, when the correct value of r and c are checked, the

[2] We point out that, for e.g. a 256x256 image, reasonable values of R and C would need to be smaller than 128, and thus $\log(R) + \log(C)$ bits would mean less than 14 bits.



(a) 25 bits

(b) 39 bits

(c) 61 bits

(d) 81 bits

Fig. 3. Success/failures in the computationally light decoding of v, depending on the amplitude of the shift and on the noise strength, for different number of used bits. Images x and y were obtained here by cropping the image *"goldhill"* in random positions. An asterisk indicates a success while a circle indicates a failure. Note that the axis scale is different for different number of bits used.

shifted and windowed image y differs from the windowed x mainly only for the noise, the border effects being smoothed by the window. This gives to the technique a great robustness. On the other hand, the main disadvantage is that for every (r, c) pair a DFT must be computed for the y image. This lead to a very high computational complexity that may be considered as an intolerable drawback of this method.

Thus, a possible different choice that we consider is a method which has a much lower computational complexity but, on the other hand, cannot reach the same performance of the previous one. In this second scheme, the y image is multiplied by the window only once at the beginning of the process, it is transformed, and the subset of meaningful frequency coefficients is extracted. Then, for every (r, c) pair, a circular shift on y by a (r, c) vector is implemented in this subset of frequencies by multiplying the coefficients by appropriate exponential factors. This lead to a different result than the the full search method presented before, as the implementation of the shift in the frequency domain is applied after the windowing operation, and thus also the window is in this case shifted. The phase signs of the the so obtained coefficients are then extracted and the hamming distance from the code of x is computed. Again, the (r, c) pair that gives the minimum distance from the x code is kept as estimate of the shift vector. Note that in this case only one DFT is computed, and the operations required for every (r, c) pair have much lower computational complexity than in the previous method.

4. EXPERIMENTAL RESULTS

In order to show the effectiveness of the proposed method and to evaluate the performance in a practical situation, we have run some experiments on test images, and we report here one of these tests. We have only performed extensive simulations using the computationally-light proposed scheme, as the computationally complex scheme requires too many operation to extensively study the performance for different noise strength and shift amplitudes (see Fig. 2 for an example of difference of performance of the two methods).

So, we have taken the 512x512 *"goldhill"* image, and we have constructed the 256x256 x and y images by cropping portions of *"goldhill"* and by adding independent white gaussian noise to them. We have then applied the proposed computationally simple method and we have checked whether it gave the right result or not. The experiment was performed by testing, for different number of bits used for the code, various shift vector lengths and increasing noise amplitudes. The results are shown in Fig. (3), where we can see that by increasing the number of bits of the code progressively from 25 to 81 we are able to correctly detect shift vectors with increasing amplitudes and for increasing strength of the noise.

5. CONCLUSION AND FUTURE WORK

In this paper we have studied the problem of distributed encoding of the shift between two signals, and we have applied the theoretical discussion in order to construct a robust method for the distributed encoding of the shift between images. The method is based on the encoding of the sign of the phase of certain meaningful coefficients of the DFT. Further work will be devote to the study of more general phase sampling techniques, and to the application of the underlying ideas to the problem of motion compensation and disparity compensation at the decoder for single-camera and multi-camera distributed video coding systems respectively.

6. ACKNOWLEDGMENTS

This work is partially funded under European Project DISCOVER FP6-IST-015314 and under Italian Ministry of Education, PRIN project 2005092199.

7. REFERENCES

[1] S. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Information Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

[2] R. Puri and K. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles," *40th Allerton Conf. on Comm., Control and Comput.*, October 2002.

[3] A. Aaron, R. Zhang, and B. Girod, "Wyner-ziv coding for motion video," *Asilomar Conference on Signals, Systems and Computers*, November 2002.

[4] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE, Special Issue on Video Coding and Delivery*, vol. 93, no. 1, pp. 71–83.

[5] A. Liveris, Z. Xiong, and C. Georghiades, "A distributed source coding technique for highly correlated images using turbo codes," in *Proc. ICASSP'02*, Orlando, FL, May 2002.

[6] G. Toffetti, M. Tagliasacchi, M. Marcon, A. Sarti, S. Tubaro, and K. Ramchandran, "Image compression in a multi-camera system based on a distributed source coding approach," in *Proc. Europ. Signal Proc. Conf.*, Antalya, September 2005.

[7] Nicolas Gehrig and Pier Luigi Dragotti, "Distributed compression of the plenoptic function," in *Proc. of ICIP'04*, Singapore, October 2004, pp. 529–532.

[8] B. Song, O. Bursalioglu, A. K. Roy-Chowdhury, and E. Tuncel, "Towards a multi-terminal video compression algorithm using epipolar geometry," in *Proc. ICASSP'06*, Tolouse, May 2006.