MOTION PREDICTION OF DEPTH VIDEO FOR DEPTH-IMAGE-BASED RENDERING USING DON'T CARE REGIONS

G. Valenzise [#], *G. Cheung* [o], *R. Galvão* [#], *M. Cagnazzo* [#], *B. Pesquet-Popescu* [#], *A. Ortega* [+]

[#] Telecom ParisTech[*], [o] National Institute of Informatics, [+] University of Southern California

ABSTRACT

To enable synthesis of any desired intermediate view between two captured views at decoder via depth-image-based rendering (DIBR), both texture and depth maps from the captured viewpoints must be encoded and transmitted in a format known as texture-plus-depth. In this paper, we focus on the compression of depth maps across time to lower the overall bitrate in texture-plus-depth format. We observe that depth maps are not directly viewed, but are only used to provide geometric information of the captured scene for view synthesis at decoder. Thus, as long as the resulting geometric error does not lead to unacceptable synthesized view quality, each depth pixel only needs to be reconstructed at the decoder coarsely within a tolerable range. We first formalize the notion of tolerable range per depth pixel as *don't care region* (DCR), by studying the synthesized view distortion sensitivity to the pixel value—a sensitive depth pixel will have a narrow DCR, and vice versa. Given per-pixel DCRs, we then modify inter-prediction modes during motion prediction to search for a predictor block matching per-pixel DCRs in a target block (rather than the fixed ground truth depth signal in a target block), in order to lower the energy of the prediction residual for the block. We implemented our DCR-based motion prediction scheme inside H.264; our encoded bitstreams remain 100% standard compliant. We show experimentally that our proposed encoding scheme can reduce the bitrate of depth maps coded with baseline H.264 by over 28%.

Index Terms— Multiview video, depth-image-based rendering, motion estimation

1. INTRODUCTION

To enhance visual experience beyond conventional single-camera-captured video, elaborate arrays of closely spaced cameras (e.g., 100 cameras were used in one setup in [1]) are now proposed to capture a scene of interest from multiple viewing angles, so that an observer can interactively choose a specific captured viewpoint as the video is played back in time. If, in addition to texture maps (RGB images), depth maps[1] (per-pixel physical distance between scene objects and the capturing camera) are also acquired, then the observer can synthesize successive intermediate views between two camera-captured views via depth-image-based rendering (DIBR) [3] for smooth view transition, achieving free-viewpoint visual experience [4]. Transmitting both texture and depth maps of multiple viewpoints—a format known as *texture-plus-depth*—from server to client entails a large bit overhead, however. In this paper, we address the problem of temporal coding of depth maps in texture-plus-depth format for multiview video.

The key observation in our work is that depth maps are not themselves directly viewed, but are only used to provide geometric information of the captured scene for view synthesis at decoder. Thus, as long as the resulting geometric error does not lead to unacceptable synthesized view quality, each depth pixel only needs to be reconstructed coarsely at decoder, e.g., within a defined tolerable range. We first formalize the notion of this tolerable range per depth pixel as *don't care region* (DCR) using a threshold τ, by studying the synthesized view distortion sensitivity to the pixel value. Specifically, if a depth pixel's reconstructed value is inside its defined DCR, then the resulting geometric error will lead to distortion in a targeted synthesized view by no more than τ. Clearly a sensitive depth pixel (e.g., an object boundary pixel whose geometric error will lead to confusion between background and foreground) will have a narrow DCR, and vice versa.

Given per-pixel DCRs, we then modify inter-prediction modes during motion compensation in such a way that, for each pixel of a block, we find the smallest residue that brings the predicted pixel inside DCR. This is different from the conventional approach that aims at reconstructing a fixed ground-truth depth block, and results in a lower energy of the prediction residuals. SKIP mode is also similarly altered, so that code block of the same location in reference frame is evaluated against DCRs in a target block in the current frame. We implemented our DCR-based motion compensation scheme inside H.264 [5]; our encoded bitstreams remain 100% standard compliant. We show experimentally that our proposed encoding scheme can reduce the bitrate of depth maps coded with baseline H.264 by over 28%.

The outline of the paper is as follows. We first discuss related work in Section 2. We then define formally per-pixel DCR in Section 3. Given per-pixel DCRs, we discuss how different coding modes in motion compensation are modified in Section 4. Finally, we present experimental results and conclusions in section 5 and 6, respectively.

2. RELATED WORK

It was argued in [6] that since depth maps in texture-plus-depth multiview video are only used for view synthesis and not themselves directly viewed, synthesized-view-specific metrics should be used during depth map coding optimizations. [6] proposed alternative mode selection strategies in H.264 when coding depth maps, so that the distortion term reflects distortion in the synthesized view rather then distortion of the depth maps themselves.

Observing that depth maps are mostly flat surfaces with sharp edges, alternative coding schemes have also been proposed [7, 8]. [7] proposed edge-adaptive wavelets, and [8] proposed edge-adaptive transforms, where the goal in both schemes is to avoid filtering across depth edges, which would result in many hard-to-code high frequency components in the transform domain. We differ from

[*] This work has been partially funded by the ANR agency via the PERSEE project (Projet ANR-09-BLAN-0170).

[1] Depth maps can either be estimated via stereo-matching algorithms, or captured directly using time-of-flight cameras [2].

these works in that we focus on reducing the energy of the prediction residual during motion compensation, given that each depth pixel only needs to be reconstructed within a well defined tolerable range.

Don't care regions have been originally defined for finding the sparsest representation of transform coefficients in the spatial dimension in [9]. There, given per-pixel tolerable range for reconstruction (don't care regions) in a code block, the sparsest transform domain representation of depth signal is sought by minimizing the l_0-norm. In this work we extend the approach in [9] by exploiting the degrees of freedom defined in DCRs to seek coding gain in the *temporal* dimension for depth video. How to jointly optimize depth video in both spatial and temporal dimension given per-pixel DCR is left for future work.

3. DON'T CARE REGION

3.1. Background

In the texture+depth video format, each camera-captured view $n = 1, \ldots, N$ is represented by one texture and one depth map. If the images are properly rectified (i.e., they are warped so that one captured image is a pure horizontal shift of another), then depth can be easily converted to disparity information, which is proportional to the inverse of depth. In the following, we will use both "disparity" and "depth" to refer to the disparity map at each view. Given $\mathbf{v}_n$, $\mathbf{v}_{n+1}$ and $\mathbf{d}_n$, $\mathbf{d}_{n+1}$, texture and disparity maps at views n and $n+1$, respectively, it is possible to synthesize any texture map $\mathbf{v}_k$ at intermediate view k, $k \in [n, n+1]$, using a depth-image-based rendering (DIBR) algorithm such as [3]. Essentially, any DIBR algorithm synthesizes a pixel value in $\mathbf{v}_k$ by properly mapping corresponding pixels from texture maps $\mathbf{v}_n$ and $\mathbf{v}_{n+1}$, according to their disparity. If no corresponding pixels in $\mathbf{v}_n$ and $\mathbf{v}_{n+1}$ are found (due to dis-occlusion), then an inpainting technique can be used to fill in the missing pixels using neighboring pixel information. If the captured cameras are close to each other, however, then the number of dis-occluded pixels is expected to be small.

Since the disparity values are used as geometric information for pixel mapping during DIBR (and geometry of the captured scene varies greatly across space), not all the disparity pixels need be reconstructed with the same fidelity in order to guarantee a certain quality in the synthesized view. For example, depth pixels corresponding to smooth areas can be reconstructed with less accuracy than pixels at foreground object boundaries, as errors in the latter would produce large distortion when the decoder errs in mapping foreground textural pixels to background and vice versa. We formalize this concept as "don't care region" in the next paragraph.

3.2. Definition of Don't Care Regions (DCR)

We now define per-pixel DCRs for depth map $\mathbf{d}_n$, assuming target synthesized view is n. In other words, we consider the case $k = n$, since the encoder does not known which viewpoint k will be chosen at the decoder. This is generally the most difficult scenario, since the target view is the farthest from the reference. A similar procedure can be done for depth map $\mathbf{d}_{n+1}$ assuming target synthesized view $n+1$.

A pixel $v_n(i,j)$ in texture map $\mathbf{v}_n$, with associated disparity value $d_n(i,j)$, can be mapped to a corresponding pixel in view $n+1$ through a view synthesis function $s(i,j; d_n(i,j))$. In the simplest case where the views are captured by purely horizontally shifted cameras, $s(i,j; d_n(i,j))$ corresponds to a pixel in texture map $\mathbf{v}_{n+1}$ of view $n+1$ displaced in the x-direction by an amount proportional to $d_n(i,j)$; i.e.,

$$s(i,j; d_n(i,j)) = v_{n+1}(i, j - \gamma \cdot d_n(i,j)) \qquad (1)$$

where γ is a scaling factor depending on the camera spacing.

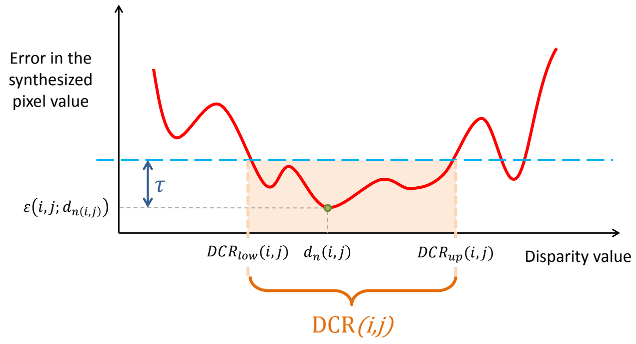


Fig. 1. Definition of DCR for a given threshold τ.

We now define *view synthesis error*, $\varepsilon(i,j;d)$, as the absolute error between the mapped-to pixel $s(i,j;d)$ in the synthesized view $n+1$ and the mapped-from pixel $v_n(i,j)$ in v_n, given disparity value d for pixel (i,j) in v_n; i.e.,

$$\varepsilon(i,j;d) = |s(i,j;d) - v_n(i,j)| \,. \qquad (2)$$

If $\mathbf{d}_n$ is compressed, the reconstructed value $\tilde{d}_n(i,j)$ employed for view synthesis may differ from $d_n(i,j)$ by an amount $e(i,j) = \tilde{d}_n(i,j) - d_n(i,j)$, resulting in a (generally larger) view synthesis error $\varepsilon(i,j; d_n(i,j) + e(i,j)) > \varepsilon(i,j; d_n(i,j))$. We define the *Don't Care Region* $\mathrm{DCR}(i,j) = [\mathrm{DCR}_{low}(i,j), \mathrm{DCR}_{up}(i,j)]$ as the *largest* contiguous interval of disparity values containing the ground-truth disparity $d_n(i,j)$, such that the view synthesis error for any point of the interval is smaller than $\varepsilon(i,j; d_n(i,j)) + \tau$, for a given threshold $\tau > 0$. The definition of DCR is illustrated in Figure 1. Note that DCR intervals are defined *per pixel*, thus giving precise information about how much error can be tolerated in the disparity maps. We also remark that the DCRs can be computed at the encoder side since both the views and the associated disparities are available.

4. MOTION PREDICTION USING DCR

The defined per-pixel DCRs give us a new degree of freedom in the encoding of disparity maps, where we are only required to reconstruct each depth pixel at the decoder to within its defined range of precision (as opposed to the original depth pixel), thus potentially resulting in further compression gain. Specifically, we change three aspects of the encoder in order to exploit DCRs: i) motion estimation, ii) residual coding, and iii) skip mode.

4.1. Motion estimation

During motion estimation for depth map encoding, the encoder searches, for each target block $\mathcal{B}$, a corresponding predictor block $\mathcal{P}$ in a reference frame which minimizes the Lagrangian cost function

$$\mathcal{P}^* = \arg\min_{\mathcal{P}} D_{\mathrm{MV}}(\mathcal{B}, \mathcal{P}) + \lambda_{\mathrm{MV}} R_{\mathrm{MV}}(\mathcal{B}, \mathcal{P}), \qquad (3)$$

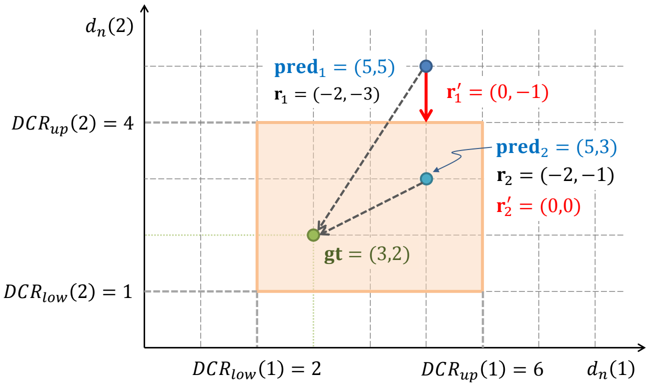


Fig. 2. Coding the residuals using DCR with a toy example with just two pixels ($d_n(1)$ and $d_n(2)$). In conventional coding, given predictor (**pred**), one aims to reconstruct the original ground truth (**gt**). However, considering DCR, it is sufficient to encode a generally smaller residual, i.e. one that enables to reconstruct a value inside or on the border of the DCR (shaded area in the picture).

where $R_{\text{MV}}(\mathcal{B}, \mathcal{P})$ is the bit overhead required to code the motion vector from position of $\mathcal{P}$ to $\mathcal{B}$, and λ_{MV} is a Lagrange multiplier. The term $D_{\text{MV}}(\mathcal{B}, \mathcal{P})$ is a measure of the energy of the prediction residual $r(i,j) = \mathcal{P}(i,j) - \mathcal{B}(i,j)$ for each pixel (i,j) in the target block $\mathcal{B}$ and the corresponding pixel in the predictor block $\mathcal{P}$. Typical choices for measuring the energy of residuals include the sum of absolute or squared differences — SAD or SSD, respectively.

For a given predictor block $\mathcal{P}$, we can reduce the energy of the prediction residuals using defined per-pixel DCRs as follows. We first define a per-block *DCR space* for a target block $\mathcal{B}$ as the feasible space containing depth signals with each pixel falling inside its per-pixel DCR. As an example, Figure 2 illustrates the DCR space for a two-pixel block with per-pixel DCR $[2, 6]$ and $[1, 4]$. For a given predictor block, to minimize the energy of the prediction residuals, we *identify a signal in DCR space closest to the predictor signal in Euclidean distance*. In Figure 2, if the predictor is $(5, 5)$, we identify $(5, 4)$ in DCR space as the closest signal in DCR space, with resulting residuals $(0, -1)$. If the preditor is $(5, 3)$, we identify $(5, 3)$ in DCR space as the closest signal with residuals $(0, 0)$.

In mathematical terms, we compute a prediction residual $r'(i,j)$ for each pixel (i,j) given predictor pixel value $\mathcal{P}(i,j)$ and DCR $[\text{DCR}_{low}(i,j), \text{DCR}_{up}(i,j)]$ according to the following soft-thresholding function:

$$r'(i,j) = \begin{cases} \mathcal{P}(i,j) - \text{DCR}_{up}(i,j) & \text{if } \mathcal{P}(i,j) > \text{DCR}_{up}(i,j), \\ \mathcal{P}(i,j) - \text{DCR}_{low}(i,j) & \text{if } \mathcal{P}(i,j) < \text{DCR}_{low}(i,j), \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

We then use the residuals $r'(i,j)$ with respect to DCR to calculate D_{MV} in (3). If SAD is used as distortion metric, we get:

$$D'_{\text{MV}} = \sum_{(i,j)\in\mathcal{B}} |r'(i,j))|. \tag{5}$$

Since the distortion D_{MV} is now zero for any motion vector which points to a predictor inside DCR, the encoder can select from a potentially larger set of zero-distortion candidate predictors. Among them, the one with the smallest rate term R_{MV} will result in a small Lagrangian cost.

4.2. Coding of prediction residuals

Once the optimal predictor $\mathcal{P}^*$ for a given target block has been found, we encode r' with respect to the per-block DCR, in place of the residuals r computed with respect to ground truth depth signal. Notice that this applies also to INTRA coding modes as well. Although the prediction technique is different from the case of INTER modes (spatial prediction is used instead of temporal prediction), we still encode the residue that enables to reconstruct a value inside the DCR which is as close as possible to the predictor. Since this criterion is applied to any pixel in a block, we are in fact coding the residuals with respect to DCR having minimum energy (ℓ_2 norm). In general, since both rate and distortion terms are computed using minimum-energy residuals r' for inter and intra modes, the actual selected mode for a given target block will be different from the one selected when coding residuals with respect to the ground truth signal.

We note that, although computing minimum-energy prediction residuals from (4) is computationally convenient (in fact, its complexity grows linearly with the number of pixels), this is not necessarily the best possible strategy in terms of rate-distortion performance. This is because the minimum-energy residual may not lead to the lowest transform coding rate; e.g., lower-energy residuals $\{1, 0, 1, 1\}$ leads to coding of more non-zero transform coefficients (thus higher rate) than higher-energy residuals $\{1, 1, 1, 1\}$. Therefore, the best motion vector and the best coding residuals for a given block with defined per-block DCR is a joint optimization problem, whose solution is not trivial. We leave the investigation of this problem for future work.

4.3. Skip mode

The coding of prediction residuals for INTER/INTRA modes described in the previous section guarantees that the reconstructed block will be within DCR (up to quantization errors). If the SKIP mode is selected instead, the prediction residuals are not coded. Thus, the reconstructed pixels could be potentially far away from DCR. This is potentially harmful since, by construction, there is no upper bound to the distortion in the synthesized view when a depth pixel is reconstructed outside DCR. This requires SKIP mode to be handled differently from INTER/INTRA.

In order to be sure that distortion in the synthesized view will be bounded in SKIP macroblocks, we prevent the SKIP mode to be selected from the encoder if *any* reconstructed pixel of that macroblock violates DCR. More formally, we alter the distortion term D_{MD} in the Lagrangian function used for mode decision according to the following barrier penalty function:

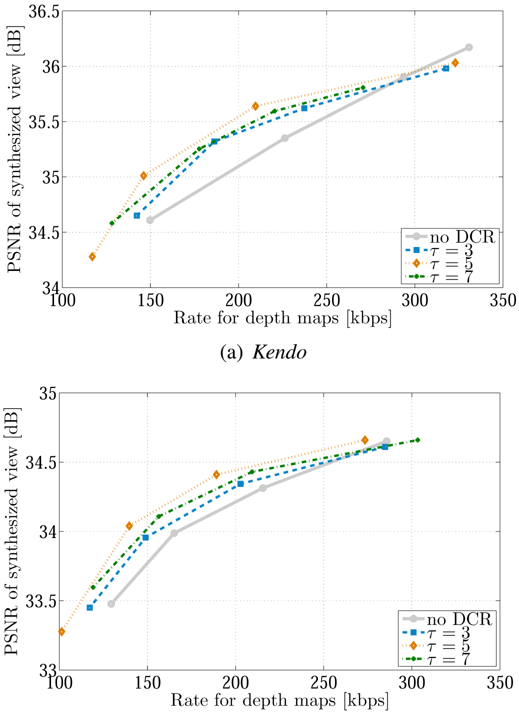
$$D'_{\text{MD}} = \begin{cases} 0 & \text{if } r'(i,j) = 0 \quad \forall i,j; \\ +\infty & \text{otherwise.} \end{cases} \tag{6}$$

Although this could be conservative in terms of rate optimization, it guarantees that the distortion in the synthesized view for SKIP macroblocks will be bounded by τ.

5. EXPERIMENTATION

We modified an H.264/AVC encoder (JM reference software v. 18.0) in order to include DCR in the motion prediction and coding of residuals. Our test material includes 100 frames of two multiview video sequences, *Kendo* and *Balloons*,[2] with spatial resolution of

[2] Available at http://www.tanimoto.nuee.nagoya-u.ac.jp/



(a) *Kendo*

(b) *Balloons*

Fig. 3. RD curves for *Kendo* and *Balloons*

1024×768 pixels and frame rate equal to 30 Hz. For both sequences we coded the disparity maps $\mathbf{d}_3$ and $\mathbf{d}_5$ of views 3 and 5 (with IPP…GOP structure), using either the original H.264/AVC encoder or the modified one. In the latter case, we computed per-pixel DCRs with three values of τ, namely $\tau = \{3, 5, 7\}$. Given the reconstructed disparities in both cases (with/without DCR), we synthesize view $\mathbf{v}_4$ using the uncompressed views $\mathbf{v}_3$ and $\mathbf{v}_5$ and the compressed depths $\tilde{\mathbf{d}}_3$ and $\tilde{\mathbf{d}}_5$. Finally, we evaluate the quality of the reconstructed view $\hat{\mathbf{v}}_4$ w.r.t. ground-truth center view $\mathbf{v}_4$.

The resulting rate-distortion curves are reported in Figure 3. For the *Kendo* sequence, using $\tau = 5$ we obtain an average gain in PSNR of 0.34 dB and an average rate saving of about 28.5%, measured through the Bjontegaard metric. Notice that the proposed method enables a significant amount of bit saving by reducing *selectively* the fidelity of the reconstructed depths where this is not bound to affect excessively the synthesized view. On the other had, to achieve an equivalent bitrate reduction, a conventional decoder should quantize prediction residuals much more aggressively, and the quantization error can affect *all* the synthesized pixels.

In order to show the impact of the proposed method on the choice of motion vectors and optimal modes at the encoder, we show in Table 1 the coding statistics of two RD points in Figure 3(a). We notice that most of the rate savings are obtained through a more efficient use of SKIP mode (which increases by over 18% in this case), and by a more efficient prediction of motion and coding of residuals. Observe that in the current setting, we are not taking into account the effect of quantization error, which could make reconstructed values lie outside DCR. We will investigate how to push the de-quantized and reconstructed values inside DCR in future work.

Table 1. Coding statistics for two RD points of *Kendo*

	bitrate [kbps]	**PSNR [dB]**	**% SKIP**	**Motion info. [bit/frame]**	**Residuals [bit/frame]**
no DCR	230.4	33.99	80.20	582.10	522.41
DCR ($\tau = 5$)	179.5	34.04	92.25	253.48	240.62

6. CONCLUSION

Depth maps need not be reproduced with high fidelity at the decoder in order to synthesize novel views with acceptable quality. In this paper we have formalized this intuition by defining per-pixel don't care regions. DCRs provide new degrees of freedom to the encoder, which can result in a higher coding efficiency of depth. Specifically, we demonstrated that DCR-aware motion compensation and coding of residuals can lead to substantial coding gains with respect to state-of-the-art video coding paradigms.

In fact, motion compensation and coding of residuals is a joint estimation problem, since any value inside DCR is a feasible reconstruction point which could entail a different RD cost. Also, quantization may move reconstructed values outside DCR, causing a deterioration of the synthesized video's quality. Solving both these two issues is the focus of our current research.

7. REFERENCES

[1] T. Fujii, K. Mori, K. Takeda, K. Mase, M. Tanimoto, and Y. Suenaga, "Multipoint measuring system for video and sound—100 camera and microphone system," in *IEEE International Conference on Multimedia and Expo*, Toronto, Canada, July 2006.

[2] S. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor—system description, issues and solutions," in *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, Washington, DC, June 2004.

[3] W. Mark, L. McMillan, and G. Bishop, "Post-rendering 3D warping," in *Symposium on Interactive 3D Graphics*, New York, NY, April 1997.

[4] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, and C. Zhang, "Multi-view imaging and 3DTV," in *IEEE Signal Processing Magazine*, November 2007, vol. 24, no.6.

[5] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," in *IEEE Transactions on Circuits and Systems for Video Technology*, July 2003, vol. 13, no.7, pp. 560–576.

[6] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map distortion analysis for view rendering and depth coding," in *IEEE International Conference on Image Processing*, Cairo, Egypt, November 2009.

[7] M. Maitre, Y. Shinagawa, and M.N. Do, "Wavelet-based joint estimation and encoding of depth-image-based representations for free-viewpoint rendering," in *IEEE Transactions on Image Processing*, June 2008, vol. 17, no.6, pp. 946–957.

[8] G. Shen, W.-S. Kim, S.K. Narang, A. Ortega, J. Lee, and H. Wey, "Edge-adaptive transforms for efficient depth map coding," in *IEEE Picture Coding Symposium*, Nagoya, Japan, December 2010.

[9] G. Cheung, A. Kubota, and A. Ortega, "Sparse representation of depth maps for efficient transform coding," in *IEEE Picture Coding Symposium*, Nagoya, Japan, December 2010.