DESIGN OF LOW-COMPLEXITY, NON-SEPARABLE 2-D TRANSFORMS BASED ON BUTTERFLY STRUCTURES

Haoming CHEN and Bing ZENG

Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology
Clearwater Bay, Kowloon, Hong Kong, China
{eehmchen, eezeng}@ust.hk

ABSTRACT: *The transform used in most image and video coding standards is the separable 2-D discrete cosine transform (DCT), which has been proven to be a robust approximation of the optimal Karhunen-Loève transform (KLT) for the 1st-order Markov sources with a large correlation coefficient. However, such separable 2-D DCT surely is not the best choice when it is applied on some residual or directional signals. Based on the butterfly architecture for DCT's fast implementation, we present in this paper a novel design of non-separable 2-D transforms that get much closer to the KLT but at the implementation cost no bigger than that of the DCT. The critical issue in our design is how to pair all node-variables in various stages of the butterfly structure. We propose a near-optimal pairing strategy to solve this problem and present some examples to demonstrate its effectiveness.*

1. INTRODUCTION

For an N-point random vector $\boldsymbol{x}=[x_0,\ldots,x_{N-1}]^T$ with covariance matrix $\boldsymbol{R}_x=[r_x(i,j)]_{N\times N}$, we know that the KLT [1, 2] can be derived from the eigenvectors of $\boldsymbol{R}_x$ to achieve the optimal R-D performance when a coding is involved. Under the 1st-order stationary Markov condition, $\boldsymbol{R}_x$ can be modeled by a Toeplitz-type matrix with

$$r_x(i,j)=\rho^{|i-j|}, i,j=0,1,\cdots,N-1, |\rho|<1. \tag{1}$$

When $\rho\to 1$ (a very strong inter-pixel correlation), it has been proven that the DCT can approximate the KLT closely [3].

In the 2-D case with $\boldsymbol{x}_{2D}=\{x_{i,j}\}_{N\times N}$, the correlation between two pixels A and B at locations (p_A,q_A) and (p_B,q_B) can be modeled as

$$r_x(A,B)=\rho^{\sqrt{(p_A-p_B)^2+(q_A-q_B)^2}}, 0\le p_A,q_A,p_B,q_B<N. \tag{2}$$

More generally, one 2-D block $\boldsymbol{x}_{2D}$ may have a dominating orientation. To model this scenario, we use a new covariance matrix with

$$r_x(A,B)=\rho^{\sqrt{d_1^2(\alpha)+\eta^2\cdot d_2^2(\alpha)}}, \tag{3}$$

where $\eta=b/a>1$ represents the ratio of long and short radius of an elliptical function (see Fig. 1) and

$$\begin{cases} d_1(\alpha)=(p_A-p_B)\cos\alpha-(q_A-q_B)\sin\alpha \\ d_2(\alpha)=(q_A-q_B)\cos\alpha+(p_A-p_B)\sin\alpha \end{cases} \tag{4}$$

Clearly, the dominating direction is along the angle α and the directionality becomes stronger as η gets bigger.

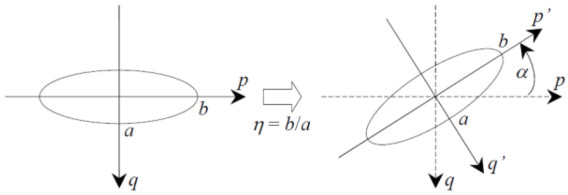

Fig. 1. Elliptical function and its rotated version used to model 2-D directional sources.

Let's use $\boldsymbol{C}$ to denote the matrix of a transform applied on $\boldsymbol{x}$ and $\boldsymbol{X}=[X_0,\ldots,X_{N-1}]^T$ the transform coefficients. The coding gain achieved by using transform $\boldsymbol{C}$ is defined as the ratio of arithmetic mean to geometric mean of the coefficients' variances [4]:

$$G_{\boldsymbol{C}}=\frac{1}{N}\sum_{n=0}^{N-1}\sigma_{X_n}^2 \Bigg/ \sqrt[N]{\prod_{n=0}^{N-1}\sigma_{X_n}^2}. \tag{6}$$

Since a unitary transform is variance-preserving, a simplified formula can be used to compute the coding gain as follows:

$$G_{\boldsymbol{C}}=-\frac{1}{N}\sum_{n=0}^{N-1}\log_2\left(\sigma_{X_n}^2\right). \tag{7}$$

Meanwhile, the energy packing efficiency (EPE) [5] is defined as the energy portion contained in the M largest ones out of all N transform coefficients, i.e.,

$$EPE_{\boldsymbol{C}}=\sum_{n=0}^{M-1}\sigma_{X_{(n)}}^2 \Bigg/ \sum_{n=0}^{N-1}\sigma_{X_n}^2. \tag{8}$$

In the following, we present examples to show the difference between the DCT and the KLT in two popular scenarios appeared in practice.

A. Directional source (no intra-prediction)

Suppose that a 4×4 image block $\{x_{i,j}\}_{4\times 4}$, $i,j=1,\ldots,4$ has the diagonal down-left (DDL) direction, i.e., $\alpha=45^{\circ}$. The 4×4 2-D separable DCT can be converted into an equivalent 16-point 1-D transform. Assuming $\eta=5$ and $\rho=0.95$, we use Eq. (3) to calculate the covariance matrix of the 16-point vector and derive the corresponding KLT [6]. Applying the conventional DCT and the 16-point non-separable KLT respectively, we obtain the coding gain and EPE as: $G_{\text{DCT}}\approx 2.0404$ and $EPE_{\text{DCT(3)}}\approx 0.8547$ for DCT and $G_{\text{KLT}}\approx 2.4112$ and $EPE_{\text{KLT(3)}}\approx 0.8929$ for KLT ($M=3$).

B. Residual source (after intra-prediction)

Consider the 4×4 image block again, but now with $\alpha = 90^{\circ}$, $\eta = 5$, and $\rho = 0.95$. Clearly, the dominating direction is vertical so that we apply the Mode-0 (in H.264) intra-prediction. After the intra-prediction, we calculate the covariance matrix for each column (all columns have the same result) and then derive the corresponding KLT. The coding gain and EPE incurred by the DCT are $G_{\mathrm{DCT}} \approx 3.1169$ and $EPE_{\mathrm{DCT(2)}} \approx 0.9147$, respectively; whereas applying the KLT results in $G_{\mathrm{KLT}} \approx 3.3232$ and $EPE_{\mathrm{KLT(2)}} \approx 0.9237$ ($M = 2$).

C. Observations and what to solve?

In both scenarios (with and without intra-prediction), one can see clearly that the KLT performs much better than the DCT. However, one also knows that the computation complexity of the DCT is significantly lower than that of the KLT, thanks to its butterfly implementation [7]. Now, the question we would like to solve in this paper is: can we maintain a similar complexity as the DCT while achieving a performance asymptotically approaching to that of the KLT? In the remaining part of this paper, we will formulate this problem into an optimization and present an efficient solution so as to yield some "good" 2-D (non-separable) transforms.

2. From Best Rotation Angles to Unconstrained Optimization

In principle, a unitary transform matrix can be factorized into a product of multiple Givens rotations [8], defined as

$$G(j,k,\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ & \ddots & & \\ 0 & \cos\theta & \sin\theta & 0 \\ & & \ddots & \\ 0 & -\sin\theta & \cos\theta & 0 \\ & & & \ddots \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where j and k are the indices of the two selected nodes among N candidates. Subsequently, one implementation of any given unitary transform is a cascade of multiple Givens rotations: $G(j_l,k_l,\theta_l)$, $l = 1,\ldots,L$. From this perspective, our problem can be stated as the maximization of the coding gain over all $G(j_l,k_l,\theta_l)$, $l = 1,\ldots,L$, for a given L (that is selected to control the implementation cost).

A. Optimal angle for a standing-alone Givens rotation

Let's start from the simplest case - one Givens rotation standing alone. Referring to Fig. 2(a), the node-variables before and after the rotation is related as

$$\begin{bmatrix} X_0 \\ X_1 \end{bmatrix} = \boldsymbol{C} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}, \; \boldsymbol{C} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}, \theta \in [0, \pi/2] \tag{9}$$

Let's denote the covariance matrix of $[x_0\; x_1]^T$ as $\boldsymbol{r} = [r(i,j)]$ and the covariance matrix of $[X_0\; X_1]^T$ as $\boldsymbol{R} = [R(i,j)]$, i, $j = 0$ or 1. Then, we have $\boldsymbol{R} = \boldsymbol{C}\boldsymbol{r}\boldsymbol{C}^T$. According to (7), the coding gain can be calculated as:

$$G_{\boldsymbol{C}} = -\frac{1}{2}\sum_{i=0}^{1}\log_2 R(i,i). \tag{10}$$

It can be shown that maximizing $G_{\boldsymbol{C}}$ is equivalent to minimizing the product of $R(0,0)$ and $R(1,1)$ which are

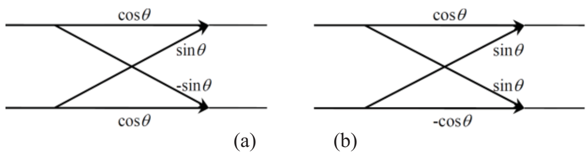

Fig. 2. Two butterfly structures to implement one Givens rotation

$$\begin{cases} R(0,0) = r(0,0)\cos^2\theta + r(1,1)\sin^2\theta + \left(r(0,1) + r(1,0)\right)\sin\theta\cos\theta \\ R(1,1) = r(0,0)\sin^2\theta + r(1,1)\cos^2\theta - \left(r(0,1) + r(1,0)\right)\sin\theta\cos\theta \end{cases}$$

Since $R(0,0) + R(1,1) = r(0,0) + r(1,1) = \text{constant}$ (the variance-preserving property of a unitary transform), the minimal product of $R(0,0)$ and $R(1,1)$ can be obtained when $R(0,0)$ or $R(1,1)$ reaches its extreme value. We omit the details and directly present the "best" angle θ as follows:

$$\theta = \begin{cases} \phi/2 & \text{if } \left(r(0,0) - r(1,1)\right)\cdot\left(r(0,1) + r(1,0)\right) \geq 0 \\ (\pi - \phi)/2 & \text{otherwise} \end{cases} \tag{11}$$

where

$$\phi = \cos^{-1}\frac{\left|r(0,0) - r(1,1)\right|}{\sqrt{\left(r(0,0) - r(1,1)\right)^2 + \left(r(0,1) + r(1,0)\right)^2}}. \tag{12}$$

The coding gain of the transform with the "best" angle is

$$\begin{aligned} G_{\boldsymbol{C}} &= -\frac{1}{2}\log\left(R(0,0)\cdot R(1,1)\right) \\ &= -\frac{1}{2}\log\left(r(0,0)\cdot r(1,1) - r(0,1)\cdot r(1,1)\right) \\ &\geq -\frac{1}{2}\log\left(r(0,0)\cdot r(1,1)\right) = G_{\mathrm{orig}} \end{aligned} \tag{13}$$

where the G_{orig} is the coding gain before transform. It is interesting that, after applying the transform $\boldsymbol{C}$ with the best angle θ, we obtain $R(0,1) = R(1,0) = 0$, which means two nodes are de-correlated completely and this is exactly the same as what the KLT can achieve. Subsequently, we have reached the optimal solution for a standing-alone butterfly structure.

In practice, one may find that some modified rotation structure actually appear in the butterfly implementation of a transform matrix. For instance, the modified one shown in Fig. 2(b) is often encountered. Fortunately, our analysis tells that the "best" angle θ and G_C are exactly the same as determined above in Eqs. (11) and (13), respectively.

B. Unconstrained optimization

Notice that the analytical and closed-form solution has been derived in Eqs. (11) and (12) to determine the "best" angle so as to maximize the coding gain for a standing-alone Givens rotation. For a generic butterfly structure with more than 2 input node-variables, the optimization problem is much more complicated, as stated as follows:

$$\begin{aligned} &\underset{j_l,k_l,\theta_l}{\text{maximize}} \quad G_{\boldsymbol{C}} \\ &\text{subject to} \quad \boldsymbol{C} = \prod_{l=1}^{L} G(j_l,k_l,\theta_l) \end{aligned} \tag{14}$$

that is, we should determine, at each stage l (L stages in total), a "best" pair of indices (indicating two node variables among N candidates) and the "best" angle. In our previous work [9], we proposed a hybrid strategy to solve the problem with some constraints in which the structure of the conventional 2-D DCT's flow-graph is retained (as constraints) so that the optimization is only over θ_l of some $G(j_l,k_l,\theta_l)$. In this current work, we relax all variables, including the indices, to study the problem formulated in (14). Clearly, selecting a pair (j_l,k_l) among multiple node-variables leads to a much more complicated problem. A pairing strategy will be presented in the next section to solve this problem.

3. PAIRING STRATEGY

The results for a standing-alone Givens rotation derived earlier suggests an iterative approach to the problem formed above when the number of nodes comes to $N>2$: in each stage l, we choose a pair of two nodes in such a way that its best angle θ_l (analytically determined from Eq. (11) for any selection) leads to the *maximum* coding gain after performing the Givens rotation (with angle θ_l) on two selected node variables. Following our earlier notations, we also use $\boldsymbol{r}$ and $\boldsymbol{R}$ to represent the covariance matrices of the node-variables before and after the l-th Givens rotation, respectively. Notice that there are N node-variables so that both $\boldsymbol{r}$ and $\boldsymbol{R}$ have size $N\times N$. The coding gain achieved after the l-th Givens rotation is defined as follows:

$$G_C = -\frac{1}{N}\sum_{i=0}^{N-1}\log_2 R(i,i). \tag{15}$$

As the l-th Givens rotation takes place on two nodes (j_l,k_l), only two diagonal elements $R(i,i)$ would change after the transform, i.e., $i=j_l$ and $i=k_l$. Upon this change, the coding gain now becomes:

$$\begin{aligned} G_C &= -\frac{1}{N}\log_2\left(R(j_l,j_l)\cdot R(k_l,k_l)\times \prod_{\substack{i=0\\ i\neq j_l \text{ and } i\neq k_l}}^{N-1} R(i,i)\right) \\ &= -\frac{1}{N}\log_2\left(\left(1-\frac{r(j_l,k_l)\cdot r(k_l,j_l)}{r(j_l,j_l)\cdot r(k_l,k_l)}\right)\times \prod_{\substack{i=0\\ i\neq j_l \text{ and } i\neq k_l}}^{N-1} R(i,i)\right). \end{aligned} \tag{16}$$

Based on Eq. (16), we now present an iterative procedure to design transforms to approach the best coding gain (that is only achievable by the KLT) as follows,

Pairing strategy algorithm: Given the initial covariance matrix $\boldsymbol{r}=[r(i,j)]_{N\times N}$ of the input random vector $\boldsymbol{x}=[x_0,\ldots,x_{N-1}]^T$, set $l=0$, and initialize the transform matrix with $\boldsymbol{C}=\boldsymbol{I}$ (identity matrix).

1) For $l=1:L$, search over $\Pi=\{0,1,\ldots,N-1\}$ to identify two integers (j_l,k_l) that lead to the largest ratio of $r(j_l,k_l)\times r(k_l,j_l)$ w.r.t. $r(j_l,j_l)\times r(k_l,k_l)$.
2) Calculate the best angle for the selected pair θ_l using Eq. (14). Stop if the best angle derived is equal to 0; otherwise, update the transform matrix $\boldsymbol{C}$ by $G(j_l,k_l,\theta_l)$ (with all parameter determined): $\boldsymbol{C}\Leftarrow\boldsymbol{C}\cdot G(j_l,k_l,\theta_l)$.
3) Calculate the covariance matrix of the output node-variables as follow:

   $$\boldsymbol{R}=G(j_l,k_l,\theta_l)\cdot\boldsymbol{r}\cdot G^t(j_l,k_l,\theta_l)$$

   and use it to replace $\boldsymbol{r}$, i.e., $\boldsymbol{r}\leftarrow\boldsymbol{R}$. Go to the next iteration.

Remark: When a modified butterfly structure appears, our analysis, once again, tells that the same result can be derived.

Proposition (convergence): Denoting the coding gain after l-th iteration as $G^{(l)}$, we have

$$G^{(l)}\geq G^{(l-1)} \text{ and } \lim_{l\to\infty}G^{(l)}=G_{\text{KLT}}=-\frac{1}{N}\sum_{i=0}^{N-1}\log_2\lambda_i.$$

where the G_{KLT} is the coding gain achieved by the KLT and the λ_i are the eigenvalues of Σ_x - the covariance matrix of the original input random vector $\boldsymbol{x}=[x_0,\ldots,x_{N-1}]^T$.

Proof: The inequality follows directly from Eq. (13). Considering the fact that G_C can never exceed G_{KLT}, the pairing transform algorithm will converge to a stable point. We can verify that the stable point is exactly the optimal G_{KLT}. From the pairing strategy algorithm, one can find that, without any termination conditions, the coding gain $G^{(l)}$ keeps growing until all $r(i,j)$ $(i\neq j)$ becomes zero, which means:

$$\boldsymbol{C}\cdot\Sigma_x\cdot\boldsymbol{C}^t=diag[\gamma_0,\cdots,\gamma_{N-1}].$$

Since the transform matrix $\boldsymbol{C}$ obtained after each iteration is always a product of multiple Givens rotations (unitary matrices), $\boldsymbol{C}$ itself is also unitary all the time. Considering the fact that Σ_x is positive-definite and its eigenvalues are unique, the set of $\{\gamma_i\}$, $i=0,\ldots,N-1$, must be the same as the set of eigenvalues of Σ_x. □

4. RESULTS ON 4×4 NON-SEPARABLE TRANSFORMS

We focus on the 4×4 block-size (commonly used in H.264). First, let us assume that the model with $\alpha=45^\circ$, $\eta=5$, and $\rho=0.95$ in Eq. (3) is selected, leading to Mode-3 or diagonal-down-left (DDL) mode. We can code such a 4×4 block directly (without any intra-prediction) or follow the H.264 standard to do the Mode-3 intra-prediction. In either way, a 16×16 covariance matrix can be obtained so that we can derive its KLT. Since the separable 4×4 2-D DCT (or the equivalent 16-point 1-D DCT) needs 32 butterflies totally, we set $L=32$ when applying our iterative pairing strategy to design a non-separable transform. Some comparative results (in terms of the coding gain and EPE) among the traditional DCT, the optimal KLT, and our new transforms are presented in Table 1.

It is clear that our new transforms do offer a better performance than the separable 2-D DCT in both scenarios (without or with intra-prediction). More interesting results can be observed from the asymptotic performances over the whole iterative procedure as shown in Fig. 3: the performance of our new transforms in either case has actually surpassed that of the DCT way before the 32nd iteration is finished. For instance, only 14 or 6 iterations are needed to deliver a performance better than that of the DCT, which means a BIG saving in the implementation cost.

5. CONCLUDING REMARKS

We all know that the KLT and DCT stand for two good representatives in the area of image and video coding: the first offers the best R-D performance and the second can be implemented very efficiently. In this paper, we attempted to design some non-separable transforms (for 2-D data) with two goals: (1) approaching as closely as possible to the KLT's R-D performance and (2) maintaining the DCT's implementation complexity.

TABLE 1. Comparative results for different transforms under two scenarios

	4×4 block without intra prediction (α=45°, η=5, and ρ=0.95)			4x4 block with DDL intra prediction (α=45°, η=5, and ρ=0.95)		
	KLT	DCT	Proposed	KLT	DCT	Proposed
Coding Gain	2.4112	2.0404	2.3852	2.8956	2.5173	2.8748
EPE	0.8929	0.8570	0.8915	0.8245	0.7431	0.8191

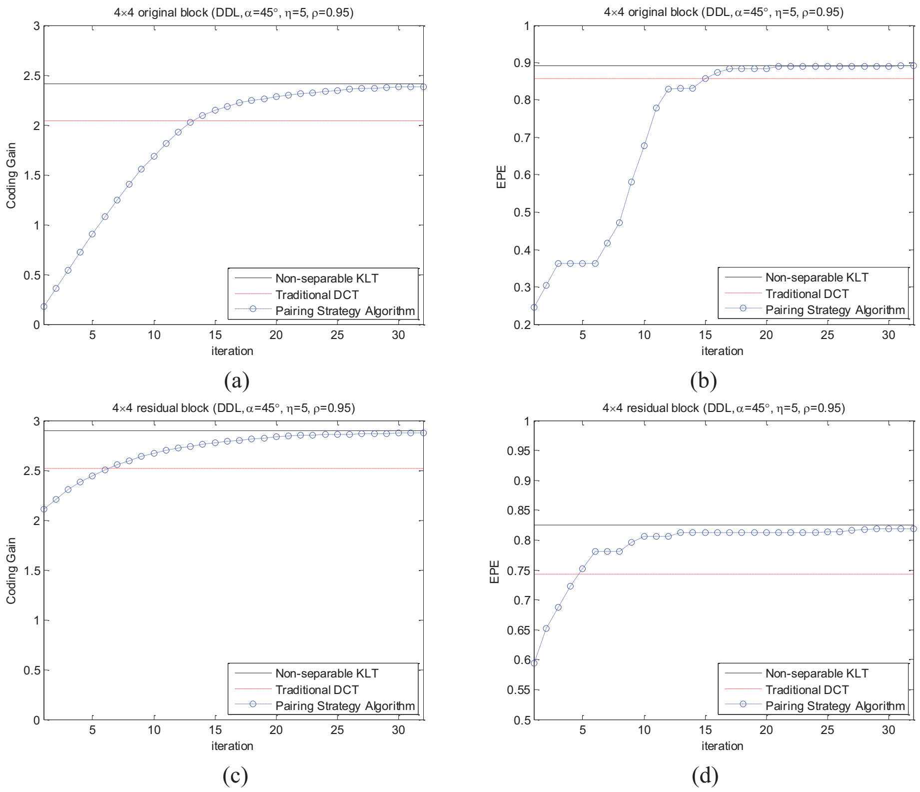

Fig. 3. Asymptotic performances: (a) and (b) without intra-prediction; (c) and (d) with the DDL intra-prediction.

We started the analysis from the standing-alone 2-point Givens rotation to derive the optimal close-form (analytical) solution. However, when the number of points comes to be more than 2, the global optimality is extremely hard to get analytically. To solve this problem, we developed an iterative pairing strategy algorithm: in each iteration, the problem can be simplified as finding a best pair of two nodes and then determining the corresponding rotation angle to get the maximum coding gain improvement. This strategy is particularly meaningful for cases where the traditional DCT is far from the KLT, e.g. directional or predicted sources.

Some theoretical results have been provided to secure the algorithm's convergence to the optimal KLT. Design examples have also been presented to show the superiority over the DCT. One of our future works is to try to prove that the convergence (to the KLT) will be reached in a finite number of iterations and further to determine this finite number.

REFERENCES

[1] K. Karhunen, "Über lineare Methoden in derWahrscheinlichkeitsrechnung," *Ann. Acad. Sci. Fenn., Ser. A.1.: Math.-Phys.*, vol. 37, pp. 3–79, 1947.

[2] M. Loève, "Fonctions aléatoires du second ordre," in *Processus stochastiques et mouvements Browniens*, P. Lévy, Ed. Paris, France: Gauthier-Villars, 1948.

[3] A. K. Jain, "A sinusoidal family of unitary transforms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 1, no. 4, pp. 356-365, Oct. 1979.

[4] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Practice and Standards*. Massachusetts: Kluwer Academic Publishers, 2002.

[5] H. Kitajima, "Energy packing efficiency of the hadamard transform," *IEEE Trans. on Communi.,* , vol. 24, no. 11, pp. 1256–1258, 1976.

[6] S. Zhu, S. Au Yeung, and B. Zeng, "R-D performance upper bound of transform coding for 2-D directional sources," *IEEE Signal Processing Letters*, vol. 16, no. 10, pp. 861-864, Oct. 2009.

[7] P. Yip and K. R. Rao, "Fast discrete transforms," in *Handbook of Digital Signal Processing*, ed. by D. F. Elliott, Academic Press, San Diego, CA, 1987.

[8] P. Dita, "Factorization of unitary matrices," *J. of Physics A: Mathematical and General*, vol. 36, p. 2781, 2003.

[9] H. Chen, S. Zhu, and B. Zeng, "Design of non-separable transforms for directional 2-D sources," in *IEEE Int. Conf. on Image Processing (ICIP-2011),* Brussels, Belgium, Sept. 11-14, 2011.