MUSIC SEGMENT SIMILARITY USING 2D-FOURIER MAGNITUDE COEFFICIENTS

Oriol Nieto, Juan Pablo Bello**

Music and Audio Research Lab
New York University
{oriol, jpbello}@nyu.edu

ABSTRACT

Music segmentation is the task of automatically identifying the different segments of a piece. In this work we present a novel approach to cluster the musical segments based on their acoustic similarity by using 2D-Fourier Magnitude Coefficients (2D-FMCs). These coefficients, computed from a chroma representation, significantly simplify the problem of clustering the different segments since they are key transposition and phase shift invariant. We explore various strategies to obtain the 2D-FMC patches that represent entire segments and apply k-means to label them. Finally, we discuss possible ways of estimating k and compare our competitive results with the current state of the art.

Index Terms— Music Segmentation, 2D-Fourier Transform, Clustering

1. INTRODUCTION

The task of music segmentation aims to automatically estimate the structure of a given audio signal by performing two subtasks: (*i*) identify the boundaries that will define a set of segments (or sections), and (*ii*) label them based on their acoustic similarity (e.g. *verse*, *chorus*). Music segmentation is relevant in many scenarios, e.g. to facilitate the navigation of large music collections, to create representative music summaries, to improve retrieval algorithms by analyzing music databases at a segment level.

One of the most standard music segmentation approaches is to compute the self-distance matrix of a set of audio features extracted from a given audio signal and find the repeated segments across its diagonals [1, 2]. Other solutions include hidden Markov models [3], matrix factorization [4, 5], and other diverse techniques (e.g. [6, 7, 8]). The two subtasks of music segmentation are often addressed separately, e.g. segment boundaries [9] and labeling of the segments [10], even though efforts combining the two have also been presented [11].

In this work we present a novel approach to label segments in Western popular music by using 2D-Fourier Magnitude Coefficients (2D-FMCs). Recently, these coefficients have proven to be an efficient solution to the task of large-scale cover song identification [12, 13] because of their interesting inherent characteristics: key transposition and phase shift invariance. By aggregating 2D-FMCs into fixed-size patches representing full tracks, the comparison between tracks becomes fast and trivial. Analogously, and as a novel process, we explore various methods to obtain a set of segment-synchronous 2D-FMCs that can be used to characterize the similarity between segments of a given track, and to group those segments using k-means clustering. This results in a simple and computationally inexpensive process (as opposed to [14] or [4]). We also discuss methods to estimate the optimal k, and systematically evaluate the main components of our approach, resulting in state of the art performance.

2. 2D-FMCS IN MUSIC SEGMENT SIMILARITY

In Western popular music, segments representing the same music section are likely to have common harmonic or melodic sequences (e.g. phrases, melodic lines, riffs, chord progressions), which are often played at different tempi, instrumentation and dynamics, and are flanked with repetitions and ornaments (that could cause phase shifts in the pattern), or even at different keys. In this section we detail how beat-synchronous 2D-FMCs are invariant to these changes and therefore can be effective to label the different segments of a given piece.

2.1. Beat-Synchronous Chroma Representation

Similarly to other works (e.g. [9, 4, 5]), we solely base the proposed algorithm on chroma representations, which have proven to be a relevant musical aspect when segmenting musical pieces, especially for Western popular music [15].

The chosen features are Pitch Class Profiles (PCP or *chromagrams*), which can be obtained from the audio signal by computing a constant-Q spectrogram and folding each pitch into a single octave. This results in a 12-pitch class vector representing the energy for each one of the classes of the chromatic scale for every time frame. In our case, starting from a mono audio signal sampled at 22050 Hz, we form a constant-Q transform using the method [16] with a frame rate of 20 Hz and compute the PCP using 8 octaves (starting at 27.5Hz, or A0) with 12 bins per octave.

Once the PCP vectors are computed, we estimate the beats of the track and average the pitch vectors within beat bound-

*This work was supported by a scholarship of Fundación Caja Madrid and the National Science Foundation under grant IIS-0844654.

aries, such that the resulting representation becomes beat-synchronous, similarly to [17]. This results not only in a local tempo invariant representation, but also in a significant reduction of the size of the PCP matrix. In our implementation, the beats are extracted using The Echo Nest API[1], since the authors are familiar with this API[2]. Then, we segment the beat-synchronous PCP vectors using the boundaries that define the sections of a given track. These boundaries can be automatically estimated using existing methods (we use the approach described in [9]).

2.2. Computing the 2D-FMC Segments

By computing the magnitude 2D-Fourier transform of a sequence of beat-synchronous PCP, we achieve three main characteristics: (i) key transposition invariance, (ii) phase shift invariance, and (iii) local tempo invariance.

The 2D-Fourier transform, applied to the 2D signal $x_i \in \mathbb{R}^{M \times N}$, is defined as follows:

$$X_i(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_i(m,n) e^{-2\pi i \left(\frac{mu}{M} + \frac{nv}{N}\right)} \quad (1)$$

where x_i is the i-th PCP segment of a given track, M is the dimensionality of the PCP vector (i.e. 12), and N is determined based on one of the strategies described below.

The goal of this stage of the process is to produce segment-synchronous feature vectors of the same dimensionality $M \times N$. However, different segments of a given track will have different lengths, requiring some form of segment length normalization in our analysis. We explore three different strategies:

- **Maximum Window Size**: In this setup, N is set to the maximum length of the set of PCP segments that constitute a track. Since most of the segments will be less than N, we will zero-pad the segments before obtaining the 2D-FMC. The zero-pad operation is performed across the time dimension, resulting in an interpolated version of the patch of length N, which makes the comparison with other patches possible.
- **Minimum Window Size**: Another approach is to set N to the smallest segment size of all the PCP segments of a given track. The majority of the PCP segments will be greater than N, so we need to group the longer segments into this smaller N. To do so, we divide the segments into 2D-FMC patches of size N with a hop size of one beat and aggregate them into a single patch of length N. We consider three different types of aggregation: mean, median, and maximum.
- **Fixed Window Size**: In this case, we choose a specific size for N and then compute as follows: If the PCP segment size is less than N, then zero-pad as in the maximum segment type. On the other hand, if the PCP segment size is greater than N, then we divide the longer segment into smaller patches and aggregate them using the mean, median or maximum as in the minimum segment type.

[1] http://developer.echonest.com

[2] A study on beat trackers and their impact on music segmentation could be a good future contribution to the field.

2.3. Clustering the 2D-FMC Segments

Before clustering, we take the logarithm of the patch such that the weight of the DC component is alleviated and the higher frequencies are emphasized, as it empirically showed to yield better results in our experiments. We also exploit the symmetry of the 2D-FMC by removing half of the coefficients.

We use k-means clustering with Euclidean distance on the segment-synchronous 2D-FMC patches. Further, to validate the quality of each partition, we use the Bayesian Information Criterion (BIC_k), which is defined as follows:

$$\text{BIC}_k(S) = L - (p/2)\log(N) \quad (2)$$

where $S \in \mathbb{R}^{B \times M \times N}$ is the set of B 2D-FMC segment-synchronous patches, p is the number of free parameters of the system (which in our case is the sum of k classes, $N \times k$ centroid coordinates and the variance estimate σ^2 of the partition), and L is the log-likelihood of the data when using k. Formally:

$$L = -(N/2)\log(2\pi) - (NM/2)\log(\sigma^2) - (N-k)/2 \quad (3)$$

More information on this model can be found in [18]. We run k-means with various k and use the knee point detection method [19] in BIC_k in order to estimate the most optimal k.

2.4. Illustrating the Process

In Figure 1 an example of our method is depicted with the song "And I Love Her" by The Beatles. The beat-synchronous PCP matrix (top-left), the segment-synchronous 2D-FMC patches (bottom-left), and the normalized Euclidean distance between each pair of 2D-FMC patches (right) are shown. The segments S (solo) and V4 (verse 4) are key-modulated versions of segments V1, V2, and V3. This modulation is marked with an arrow in the beat-synchronous PCP matrix, but disappears in the 2D-FMC representation, which successfully makes these five segments close to each other as shown by the self-distance matrix. The bridge (B) is harmonically different to the rest of the segments, which is also captured in the self-distance matrix, while the intro (I) and outro (O) share harmonic parts and, even though they have different time lengths, are grouped closer to each other.

Our method estimates $k = 3$ unique labels for this track (I+O, V+S, and B). However, the ground truth indicates 5 unique labels (I, V, B, S, and O). Harmonically, it makes sense to have only 3 unique labels, but the timbre (e.g. for the guitar solo) and the placement of the segments (e.g. intro and outro) also have a relevant role in music segmentation. This, plus the inherent subjectiveness of this task, makes this problem remarkably difficult. In fact, it has been shown that it is unlikely that two people would manually annotate a specific dataset identically [9, 20, 21]. Efforts towards improv-

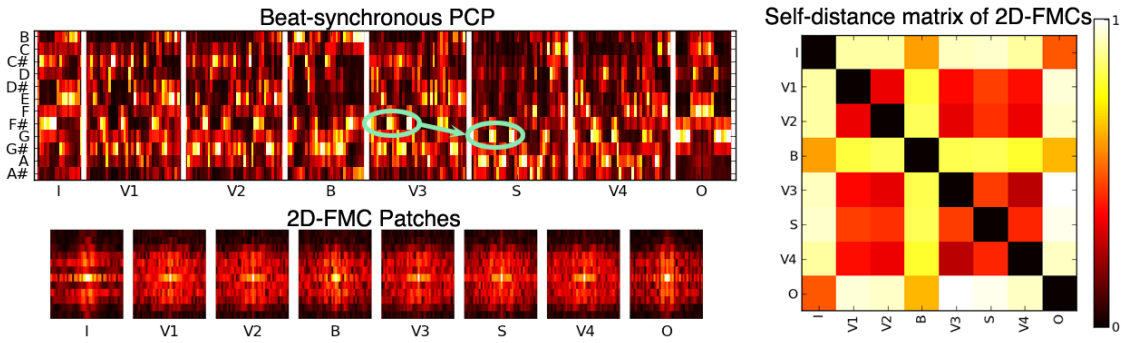

Fig. 1. Example of the similarity between 2D-FMC patches representing sections of the song "And I love Her" by The Beatles. The beat-synchronous PCP features are on the top-left, segmented with the ground truth segments by vertical white lines. The key transposition between V3 and S is marked. On the bottom-left the 2D-FMC patches are shown for each of the segments. On the right, the similarity between 2D-FMC patches is shown using the normalized Euclidean distance.

ing these annotation issues in music segmentation are being carried out by the authors at the moment.

3. EXPERIMENTS

In this section we aim to find, via experimentation, the optimal parameters of our system: the segment-synchronization strategy, and the number of unique labels k. To do so, we make use of The Beatles dataset annotated by the Tampere University of Technology[3], which is a hand annotated dataset of 180 tracks including boundaries and segment labels, thus facilitating the evaluation of automatic tasks like music segmentation. Even though we considered other datasets such as SALAMI [20], it was easier for us to collect the audio data for The Beatles (12 CDs).

3.1. Evaluation

To evaluate the results, we used the pairwise clustering analysis introduced in [3] —which yields three values: F-measure (P_F), Precision (P_P) and Recall (P_R)— and the entropy metrics defined in [22] —which result in three scores: F-measure (S_F), over-segmentation (S_o) and under-segmentation (S_u). The former evaluation is more sensitive to boundary positions, while the latter strongly penalizes randomly labeled clusters (more details on the differences between these two metrics are found in [22]). In any case, we keep the former metric (i.e. pairwise clustering) for comparison purposes. Each presented result is the average of 10 different runs, since k-means is sensitive to initialization.

3.2. Optimal Segment-Synchronization Strategy

We make use of the annotated boundaries and the real k (i.e. the number of unique labels from the ground truth for each track) to experimentally determine the best segment-synchronization strategy. We run our algorithm with the three different strategies discussed in section 2.2: maximum, minimum and fixed. For the minimum and fixed types, we also explored three different types of aggregation: median, mean, and max. Finally, for the fixed strategy, we used a window size of $N = 32$ (i.e. 8 bars at $\frac{4}{4}$ time signature, which is most common in popular music), since it empirically yielded better results when compared to other multiples of 4.

The results are shown in Table 1. As we can see, the best performance is given by the maximum window size type, with a P_F of 81.96% and S_F of 87.18%, which defines the upper bound of our system's performance. This strategy clearly outperforms all other strategies tested. We hypothesize that, by bypassing aggregation and including all information within each segment, this strategy captures important low-frequency periodicities that are characteristic of the segments, e.g. subsequence repetitions. To put these results in context, we also show the results reported by Kaiser & Sikora [10] that also tested their method with the ground truth boundaries. Our method finds a better balance between precision and recall, as opposed to Kaiser's, which tends to over-detect the number of unique labels. In the rest of the paper, we use the maximum length synchronization strategy, in accordance with these results.

Ntype	Aggr.	P_F	P_P	P_R	S_F	S_o	S_u
Max	–	**81.96**	84.35	81.3	**87.18**	86.27	89.14
Min	median	67.67	68.43	70.06	75.8	74.61	77.73
	mean	67.74	70.47	67.93	76.01	74.09	78.03
	max	67.42	69.49	74.23	75.96	74.23	77.33
Fixed	median	68.12	70.61	68.73	76.12	74.71	78.38
	mean	68.23	70.78	68.61	76.27	74.40	78.52
	max	69.80	72.48	69.80	77.32	75.34	79.54
Kaiser [10]		80.0	87.0	76.6	–	–	–

Table 1. Results of our system when using the boundaries and the real k from the ground truth.

3.3. Estimating k

In this subsection we aim to estimate k (number of unique segments per track) in the most optimal way, while still us-

[3] http://www.cs.tut.fi/sgn/arg/paulus/beatles_sections_TUT.zip

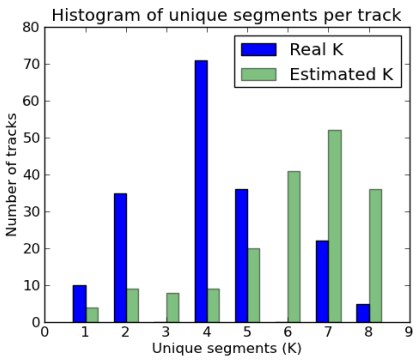

Fig. 2. Histogram of the unique number of segments and the estimated ones in The Beatles dataset.

k	P_F	P_P	P_R	S_F	S_o	S_u
3	68.20	55.94	95.03	71.46	94.54	59.66
4	76.12	70.18	88.60	81.20	89.60	76.29
5	**76.83**	80.47	77.93	**83.28**	82.68	85.82
6	72.26	85.14	66.11	81.68	76.14	90.30
auto	71.50	83.93	68.76	80.35	83.39	85.65

Table 2. Results of our system when using different k (fixed and auto) while using ground truth boundaries.

ing the ground truth boundary annotations. By examining the Beatles dataset, we observe that the median k is 5, with a histogram peak at $k = 4$ (see Figure 2). We run our algorithm with four different $k = \{3, 4, 5, 6\}$. Unsurprisingly, the results in Table 2, show that best performance is reached when $k = 5$, closely followed by $k = 4$. Note that as k increases, the metrics related to under-segmentation P_P and S_u increase, and the metrics related to over-segmentation P_R and S_o decrease, as expected.

In order to estimate k, we use the knee point detection method on the BIC, described in subsection 2.3. The histogram of estimated k can be seen in Figure 2 in green, with results shown in Table 2. The approximated k tends to find more labels than the ones existing in the dataset (the median of the estimated k is 7 instead of 5). A way of alleviating this might be by using x-means [18], which uses a tree structure for increasingly large partitions, and only increases it if the difference between the BIC value of the new partition (with $k + 1$ clusters) and the current one is greater than a certain threshold. We leave this as future work. The results show how fixing $k = 5$ yields better F-measures, illustrating the difficulty of estimating k. Note that this estimation is made with a small number of 2D-FMC segment-synchronous patches (it is uncommon for a track to have more than 15 segments), which likely has a negative effect on clustering. One idea is to obtain 2D-FMC patches for every beat (with a fixed number of beats for each patch and a hop size of one) in order to have a greater number of patches. Even though fixing k can also be interpreted as overfitting the dataset, it is not an uncommon practice [10, 5], and therefore in the last experiments we use both fixed and automatic k.

k	P_F	P_P	P_R	S_F	S_o	S_u
4	53.93	47.57	67.18	58.76	69.00	53.37
5	54.41	53.83	58.75	63.01	65.82	62.48
6	57.34	64.07	54.49	68.09	65.26	72.95
7	58.31	71.74	51.15	**71.15**	65.01	80.19
auto	57.31	66.68	52.75	68.95	65.99	76.39
Grohganz [23]	68.0	71.4	68.8	–	–	–
Kaiser [10]	60.8	61.5	64.6	–	–	–
Mauch [14]	66	61	77	69.48	76	64
Nieto [5]	59.3	48.9	83.2	47.78	49.8	47.8
Serrà [24] *	**71.8**	65.1	80	–	–	–
Weiss [4]	60	57	69	58.84	62	56

Table 3. Results of our system when using different k (fixed and auto) and estimated boundaries. *: reported in [23]

3.4. Estimated Boundaries

To the best of our knowledge, the highest results published on boundary detection using chroma representations on The Beatles are found in [9]. We implemented their method to estimate the boundaries that our algorithm will employ.

In table 3, our method is compared with a number of state of the art techniques in the literature. In this case we need to use a higher k in order to obtain better F-measures, which might be due to false-positives in estimated boundaries. These false boundaries likely result in shorter segments that need to be labeled differently in order to maximize the scores. Imprecise boundary estimations also make the 2D-Fourier transform not to capture the lower frequencies caused by the longer periodicities of the segment, which worsen the results as we saw in subsection 3.2. Lukashevich showed that poorly estimated boundaries greatly penalize P_F compared to S_F [22], which may explain why the differences between the two increase for our method compared with the previous experiments. This illustrates a drawback of our method: its high sensitivity to good boundary estimation, as clearly illustrated by a lower P_F than those of the other approaches in the table. On the other hand, when contemplating the entropy scores, we see that our algorithm obtains the best result for $k = 7$ with an S_F of 71.15%. Since the S_F measure is particularly sensitive to random clustering, this demonstrates state of the art performance for segment labeling when compared to previous approaches.

4. CONCLUSIONS

We have presented a novel algorithm to capture the similarity between music segments using 2D-FMCs. This representation is invariant to key transpositions and phase shifts, making similarity computations on chroma features both robust and efficient. In addition we have introduced novel solutions for computing segment-synchronous features, and for automatically learning the number of unique segment labels. Furthermore, we have tested the optimal combinations of strategies and parameters by running a series of experiments that show how the algorithm is competitive when compared with the state of the art.

5. REFERENCES

[1] Meinard Müller, "Audio Structure Analysis," in *Information Retrieval for Music and Motion*, chapter 7, pp. 141–168. Springer-Verlag, Berlin, 2007.

[2] Jouni Paulus, Meinard Müller, and Anssi Klapuri, "Audio-Based Music Structure Analysis," in *Proc of the 11th International Society of Music Information Retrieval*, Utrecht, Netherlands, 2010, pp. 625–636.

[3] Mark Levy and Mark Sandler, "Structural Segmentation of Musical Audio by Constrained Clustering," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 318–326, Feb. 2008.

[4] Ron Weiss and Juan Pablo Bello, "Unsupervised Discovery of Temporal Structure in Music," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1240–1251, 2011.

[5] Oriol Nieto and Tristan Jehan, "Convex Non-Negative Matrix Factorization For Automatic Music Structure Identification," in *Proc. of the 38th IEEE International Conference on Acoustics Speech and Signal Processing*, Vancouver, Canada, 2013, pp. 236–240.

[6] Ewald Peiszer, *Automatic Audio Segmentation: Segment Boundary and Structure Detection in Popular Music*, Master's thesis, Vienna University of Technology, 2007.

[7] Ilias Theodorakopoulos, George Economou, and Spiros Fotopoulos, "Unsupervised Music Segmentation Via Multi-scale Processing of Compressive Features' Representation," in *Proc. of the 18th IEEE International Conference on Digital Signal Processing*, Fira, Greece, July 2013, pp. 1–6.

[8] Jouni Paulus and Anssi Klapuri, "Music Structure Analysis Using a Probabilistic Fitness Measure and a Greedy Search Algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1159–1170, Aug. 2009.

[9] Joan Serrà, Meinard Müller, Peter Grosche, and Josep Lluís Arcos, "Unsupervised Detection of Music Boundaries by Time Series Structure Features," in *Proc. of the 26th AAAI Conference on Artificial Intelligence*, Toronto, Canada, 2012, number 2009, pp. 1613–1619.

[10] Florian Kaiser and Thomas Sikora, "Music Structure Discovery in Popular Music Using Non-Negative Matrix Factorization," in *Proc. of the 11th International Society of Music Information Retrieval*, Utrecht, Netherlands, 2010, pp. 429–434.

[11] Meinard Müller, Nanzhu Jiang, and Peter Grosche, "A Robust Fitness Measure for Capturing Repetitions in Music Recordings With Applications to Audio Thumbnailing," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 21, no. 3, pp. 531–543, 2013.

[12] Thierry Bertin-Mahieux and Daniel P. W. Ellis, "Large-Scale Cover Song Recognition Using The 2D Fourier Transform Magnitude," in *Proc. of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, 2012, pp. 241–246.

[13] Eric J. Humphrey, Oriol Nieto, and Juan P. Bello, "Data Driven and Discriminative Projections for Large-scale Cover Song Identification," in *Proc. of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil, 2013.

[14] Matthias Mauch, Katy Noland, and Simon Dixon, "Using Musical Structure to Enhance Automatic Chord Transcription," in *Proc. of the 10th International Society of Music Information Retrieval*, Kobe, Japan, 2009, pp. 231–236.

[15] Jordan B L Smith, Ching-hua Chuan, and Elaine Chew, "Audio Properties of Perceived Boundaries in Music," *IEEE Transactions on Multimedia*, vol. (upcoming), 2013.

[16] Christian Schörkhuber and Anssi Klapuri, "Constant-Q Transform Toolbox for Music Processing," in *Proc. of the 7th Sound and Music Computing Conference*, Barcelona, Spain, 2010, pp. 56–64.

[17] Daniel P. W. Ellis and Graham E. Poliner, "Identifying 'Cover Songs' with Chroma Features and Dynamic Programming Beat Tracking," in *Proc. of the 32nd IEEE International Conference on Acoustics Speech and Signal Processing*, Honolulu, HI, USA, 2007, pp. 1429–1432.

[18] Dan Pelleg and Andrew Moore, "X-means: Extending K-means with Efficient Estimation of the Number of Clusters," in *Proc. of the 17th International Conference on Machine Learning*, Stanford, CA, USA, 2000, pp. 727–734.

[19] Qinpei Zhao, Ville Hautamaki, and Pasi Fränti, "Knee Point Detection in BIC for Detecting the Number," in *Advanced Concepts for Intelligent Vision Systems*, Nice, France, 2008, pp. 664–673.

[20] Jordan B. Smith, J. Ashley Burgoyne, Ichiro Fujinaga, David De Roure, and J. Stephen Downie, "Design and Creation of a Large-Scale Database of Structural Annotations," in *Proc. of the 12th International Society of Music Information Retrieval*, Miami, FL, USA, 2011, pp. 555–560.

[21] Michael J Bruderer, Martin F Mckinney, and Armin Kohlrausch, "Perception of structural boundaries in popular music," in *Proc. of the 9th International Conference on Music Perception and Cognition*, Bologna, Italy, 2006, number 1983, pp. 157–162.

[22] Hanna Lukashevich, "Towards Quantitative Measures of Evaluating Song Segmentation," in *Proc. of the 10th International Society of Music Information Retrieval*, Philadelphia, PA, USA, 2008, pp. 375–380.

[23] Harald Grohganz, Michael Clausen, Nanzhu Jiang, and Meinard Müller, "Converting Path Structures into Block Structures using Eigenvalue Decomposition of Self-Similarity Matrices," in *Proc. of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil, 2013.

[24] Joan Serrà, Meinard Müller, Peter Grosche, and Josep Lluís Arcos, "The Importance of Detecting Boundaries in Music Structure Annotation," in *Music Information Retrieval Evaluation eXchange*, Porto, Portugal, 2012.