

CFA '18 LE HAVRE ■ 23-27 avril 2018
14^{ème} Congrès Français d'Acoustique



Analyse et catégorisation du geste: application à la harpe de concert

G. Rocheman^a, J.-L. Le Carrou^b et B. David^c

^aInstitut Jean le Rond d'Alembert, 4 place Jussieu, 75252 Paris Cedex 05, France

^bSorbonne Université, CNRS, Institut Jean Le Rond d'Alembert, Equipe LAM, F-75005 Paris, France

^cLTCI, Télécom ParisTech, Université Paris-Saclay, 46 rue Barrault, 75013 Paris, France

gabrielrocheman@gmail.com

Au cours de la phase d'apprentissage de leur instrument, les musiciens développent une expertise gestuelle importante, qui se traduit en particulier par la très grande reproductibilité des trajectoires de leurs pincements [1]. Cette reproductibilité confine même à l'émergence de familles de pincements distinctes, reconnaissables à l'œil nu lorsqu'on observe la forme des trajectoires. Pour l'analyse de la boucle d'interaction instrumentiste-instrument, il est essentiel de tester l'influence de ces familles sur le son produit, et de trier ces familles selon les harpistes qui les utilisent. Il convient alors de procéder à une catégorisation systématique des trajectoires de pincements. Nous explorons ici l'application d'une technique de catégorisation non supervisée à ce problème : la factorisation en matrices non négatives. Nous montrons que, dans son application la plus simple, cette technique permet effectivement d'extraire des familles de pincements et leurs caractéristiques. Notamment, elle est capable de différencier la forme d'un pincement produisant un accord de celle d'un pincement produisant un arpège. Toutefois, il est nécessaire de l'adapter aux contraintes du problème pour une catégorisation plus fine, et de modifier le paramétrage des données pour extraire les conditions initiales de vitesse typiques de chaque famille.

1 Introduction

Au cours de la phase d'apprentissage de leur instrument, les musiciens développent une expertise gestuelle importante. Dans le cas de la harpe de concert, cette expertise se traduit par une grande reproductibilité des conditions initiales de mise en vibration des cordes, qui confère musicalement un grand contrôle sur le son. La harpe étant un instrument peu transportable, les différences sonores d'un harpiste à l'autre dans une même salle et sur un même instrument ne peuvent être expliquées par des différences de lutherie ou de salle. Des études ont déjà été faites pour isoler les caractéristiques propres au processus de mise en vibration des cordes de harpes par des harpistes professionnels et amateurs, et leur influence sur le son [1, 2]. Un résultat frappant de ces deux études est l'observation, chez les joueurs experts, de familles de pincements distinctes et reconnaissables à l'œil nu sur le tracé des trajectoires des doigts. L'étude de ces familles et de leurs caractéristiques pourrait bien mener à une compréhension fine du rôle mécanique du musicien dans la production sonore.

L'intérêt pour l'excitation dans les instruments à cordes libres, outre celui de l'analyse de cette boucle d'interaction, peut être pris dans l'optique de la conception d'une synthèse sonore plus réaliste. En effet, malgré les avancées croissantes d'efficacité et de réalisme dans les techniques de synthèse sonore, aucun moteur de synthèse de sons d'instruments à cordes pincées n'a encore inclus l'influence du geste musicien dans ses modèles. Mais pour pouvoir générer des pincements réalistes, il est impératif de comprendre ces familles de pincements observées de manière informelle, leur rôle dans la production sonore et leurs caractéristiques mécaniques. Pour cela, nous proposons de formaliser cette observation par l'application d'une technique de catégorisation automatique non supervisée à une base de données de pincements mesurés.

Plusieurs techniques de catégorisation non supervisées existent dans la littérature. L'Analyse en Composantes Principales (ACP), Multiples (ACM) ou Indépendantes (ACI), par exemple, sont souvent utilisées pour représenter les données dans un seul plan principal pour en analyser graphiquement les groupements, générer des modèles de répartition statistique par des algorithmes Espérance-Maximisation (EM), ou pour résoudre par des méthodes statistiques des problèmes de séparation aveugle des sources [3].

Nous proposons ici d'appliquer une technique appelée *factorisation en matrices non négatives* (NMF) : un algorithme stable, robuste, facile d'implémentation,

et qui permet d'envisager de nombreuses pistes de complexification du modèle pour l'étude du mouvement [4–6]. C'est un modèle de mélange linéaire qui consiste à exhiber un jeu de vecteurs générateurs et une famille de combinaisons linéaires, permettant de reconstruire au mieux un jeu de données. La particularité de ce système est une contrainte de positivité appliquée à tous les éléments considérés. Cette contrainte de positivité des données garantit l'efficacité de la catégorisation par rapport – par exemple – à d'autres algorithmes tels que l'ACP, parce que la positivité des vecteurs générateurs qu'elle produit et de leurs combinaisons linéaires empêchent que certains vecteurs soient utilisés seulement pour en compenser d'autres [3]. Couplé à la parcimonie que la NMF permet, cette contrainte permet non seulement de catégoriser les données, mais d'exhiber les caractéristiques essentielles à chaque catégorie, comme le montre le succès de la NMF dans divers problèmes de catégorisation non supervisée et d'extraction de caractéristiques, notamment pour la reconnaissance faciale et pour la séparation aveugle de sources audio [3, 7].

L'objectif de notre travail est donc la constitution d'un dictionnaire de trajectoires-types à partir de données expérimentales de pincements. Ce dictionnaire servira à classer les pincements d'un côté, et à générer des pincements réalistes de l'autre.

Nous détaillerons d'abord la base de données de pincements, les caractéristiques connues des trajectoires qu'elle contient, puis nous développerons la technique de NMF dans un cadre général, avant de préciser ce que cette technique suppose pour notre base de données. Enfin, nous exposerons quelques résultats obtenus par cette catégorisation, et discuterons des acquis et des pistes d'améliorations possibles de ce système.

2 Trajectoires

2.1 Descriptions de la base de données

Nous avons appliqué nos algorithmes à des trajectoires obtenues lors de la thèse de D.Chadefaux [1]. Ces trajectoires incluent les informations tri-dimensionnelles, acquises en suivant le déplacement d'un point sur le doigt du harpiste par une caméra rapide (5167 images par seconde) et un miroir, tous deux fixés à la harpe. Au terme de cette étude, un total de 177 pincements produits par 9 harpistes ont été enregistrés dans divers contextes musicaux : accord ou arpège, différents doigts, index ou annulaire, etc. Les durées

Harpistes		
H1 : 7	H2 : 17	H3 : 21
H4 : 20	H5 : 23	H6 : 19
H7 : 19	H8 : 20	H9 : 31
total : 177		

		Annulaire			Index		
(b) Accord		4	3	3	3	3	4
		3	4	3	4	3	3
		3	7	8	2	0	8
		38			30		
(b) Arpège		0	6	8	0	5	6
		6	8	7	7	8	6
		8	8	7	6	5	8
			58			51	

TABLEAU 1 – Tableau récapitulatif des pincements de la base de donnée : (a) pincements par harpiste, (b) pincements par type, doigt, et par harpiste

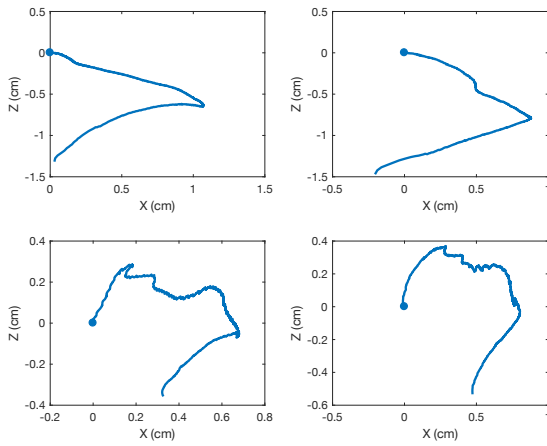


FIGURE 1 – Exemples de trajectoires reconnaissables. Le point de départ est indiqué par un point bleu en (0, 0).

totales d'interaction entre doigt et corde varient au cours de l'expérience entre 50 ms et 870 ms.

Cette base de donnée est étiquetée suivant plusieurs paramètres, notamment type de pincement (accord ou arpège), doigt utilisé (annulaire ou index), et harpiste (numérotés de 1 à 9). Le tableau 1 récapitule la répartition des différents pincements dans la base de données.

Si on trace les trajectoires dans le plan orthogonal à l'axe de la corde, des similitudes sont observées entre certaines trajectoires, qui semblent faire apparaître des familles de pincements (voir Figure 1). Outre ces familles, les pincements en arpège et les pincements en accord sont reconnaissables par la complexité de leur tracé et leur durée. En effet, les pincements en accord sont plus courts et plus simple que les trajectoires en arpège, comme illustré sur la Figure 1.

3 Catégorisation par factorisation en matrices non négatives

3.1 Cas général

La NMF est un algorithme de catégorisation qui repose sur un principe de réduction de dimension de l'espace des données. Chaque point de donnée (numéroté $n \in \llbracket 1, n_{tot} \rrbracket$) est représentée comme le vecteur colonne \mathbf{V}_n d'une matrice, que l'on veut approximer par un modèle de mélange linéaire :

$$\mathbf{V}_n \approx \sum_{k=1}^K H_{nk} \mathbf{W}_k \quad (1)$$

où chaque vecteur de données est représenté par une combinaison linéaire d'un petit nombre K de *vecteurs générateurs* \mathbf{W}_k . Dans l'équation (1), les H_{nk} sont les coefficients de la combinaison linéaire de \mathbf{W}_k permettant de reconstruire \mathbf{V}_n ; on les appelle *coefficients d'activation*.

L'algorithme consiste à optimiser les vecteurs générateurs \mathbf{W}_k , tout en optimisant leurs coefficients d'activation H_{nk} , en minimisant itérativement une distance entre données reconstruites (côté droit de l'équation (1)) et données réelles (côté gauche de l'équation (1)). Plusieurs méthodes existent pour cette itération [8, 9]; nous utiliserons l'itération multiplicative de [8]. Le point commun de ces méthodes est qu'elles requièrent toutes une représentation vectorielle de nos données comme les colonnes d'une matrice dont tous les éléments sont positifs.

La NMF permet de contrôler la parcimonie du modèle. En particulier, on peut imposer aux coefficients d'activation H_{nk} d'être les plus parcimonieux possibles. Cela revient à s'assurer qu'un minimum de vecteurs \mathbf{W}_k soit nécessaire à la reconstruction de chaque colonne, et donc d'obtenir que chaque \mathbf{W}_k représente à lui seul une catégorie de pincement. Dans le cas des trajectoires de pincements, les éléments de la famille génératrice $\{\mathbf{W}_k\}_{k \in K}$ seront des trajectoire-types $\mathbf{W}_k(t)$, recombinaison par les coefficients H_{nk} pour reconstituer la $n^{\text{ième}}$ trajectoire de l'ensemble de données $\mathbf{V}_n(t)$.

3.2 Paramétrage des données

Les présupposés de la NMF – en particulier la contrainte de positivité des données – nécessitent un paramétrage judicieux de nos données. En effet, nous voulons des données positives, mais qui conservent une structure d'espace vectoriel adaptée au modèle de mélange linéaire défini par l'équation (1), ainsi que le sens physique du point (0, 0) comme position au repos de la corde. Cela suppose, en particulier, de ne pas imposer de décalage arbitraire à nos données.

Dans un premier temps, ces contraintes de paramétrage nous ont amenés à une simplification des trajectoires : pour une trajectoire en deux dimensions, écrite $(x(t), z(t))$, nous ne prendrons que la valeur d'écartement à l'origine : $\rho(t) = \sqrt{x^2(t) + z^2(t)}$. Cette simplification peut être vue comme une réduction des trajectoires 2-D à un pincement en 1-D, en négligeant l'influence de l'angle de polarisation (par opposition à une projection suivant l'un des deux axes).

Avant de pouvoir appliquer l'algorithme, il faut de plus pouvoir ranger ces valeurs positives dans les colonnes d'une matrice pouvant être combinées linéairement par une addition point-à-point. Cela suppose en particulier d'avoir

le même nombre de points N pour toutes les trajectoires. Pour ce faire, nous utilisons une interpolation cubique, pour rééchantillonner chaque trajectoires $\rho_n(t)$ sur une nouvelle échelle de temps $\rho_n(\tau)$, $\tau \in \llbracket 1, N \rrbracket$; la valeur de la nouvelle période d'échantillonnage $\delta_n\tau$ est stockée pour conserver les données de vitesse. La fréquence d'échantillonnage ainsi trouvée est importante : c'est elle qui nous permet de retrouver les données de vitesse. Nous l'avons donc accolée au vecteur de trajectoire comme un caractère additionnel, à utiliser pour la catégorisation.

En conclusion, un vecteur de données s'écrira

$$\mathbf{V}_n = \begin{pmatrix} \delta_n\tau \\ \rho_n(1) \\ \vdots \\ \rho_n(N) \end{pmatrix} \quad (2)$$

et la matrice composée de ces colonnes \mathbf{V}_n , représentant la base de donnée de pincements, sera donnée en entrée de nos algorithmes de NMF dans la section suivante.

Remarque 1. Le paramétrage de l'équation (2) présuppose une certaine hypothèse non triviale.

En effet, les vecteurs générateurs obtenus par l'algorithme seront de la même forme

$$\mathbf{W}_k = \begin{pmatrix} \delta_k\tau \\ W_k(\tau) \end{pmatrix}$$

où $\delta_k\tau$ jouera le même rôle de période d'échantillonnage, pilotant le temps de parcours de la trajectoire-type dont la forme est décrite par $W_k(t)$.

Pendant, le modèle décrit par l'équation (1) suppose que le même coefficient d'activation H_{nk} multiplie W_k et $\delta_k\tau$, lors la reconstruction de \mathbf{V}_n . Dès lors, on impose une relation de proportionnalité entre la dilatation d'une trajectoire et son temps de parcours. Nous développerons et questionnerons la validité de cette hypothèse dans la section 4.2.

4 Résultats

4.1 Trajectoire-types et catégories de pincement

Dans un premier temps, nous voulons vérifier la validité de la catégorisation par cette méthode. La première étape est donc de tester le programme sur un problème de catégorisation *supervisée* : nous allons comparer le résultat de la catégorisation automatique à des catégories connues, données par l'étiquetage de la base de donnée. Nous avons donc appliqué la NMF à un jeu de données de 177 pincements produits par 9 harpistes, séparées en 2 fois 2 catégories : Arpège ou accord, et Annulaire ou Index (voir section 2.1). Ces trajectoires sont réduites à la seule valeur d'écartement de la corde au cours du temps, et rééchantillonnées (voir section 3.2).

Les différences entre arpèges et accords sont plus simple à distinguer par l'algorithme. Il est également envisageable de tenter de distinguer le doigt utilisé, entre index et annulaire, car cela ne rajoute que deux dimensions au modèle. Nous allons donc tenter une classification selon 4 étiquettes connues. En d'autres termes, nous avons choisi $K = 4$ dans le modèle décrit par l'équation (1), pour refléter les quatre catégories connues : Arpège / Annulaire, Arpège / Index, Accord / Annulaire et Accord / Index.

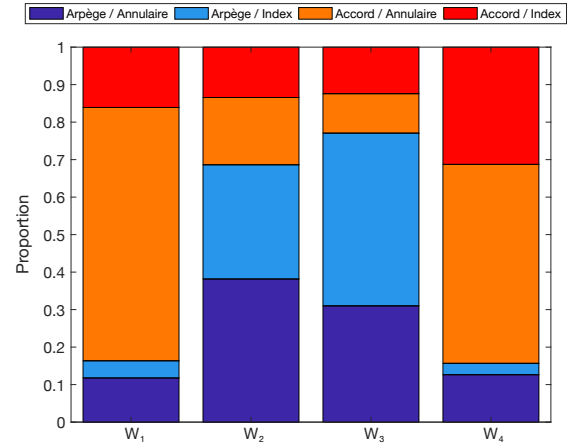


FIGURE 2 – Répartition des activations de chaque trajectoire-type en fonction de nos catégories

On obtient alors quatre vecteurs générateurs $\mathbf{W}_k = \begin{pmatrix} \delta_k\tau \\ W_k(\tau) \end{pmatrix}$, représentant quatre trajectoire-types et leur temps d'échantillonnage ; ces trajectoires génèrent toute la base de donnée.

Avant d'illustrer ces trajectoires, on peut visualiser ce qu'elles représentent. Sur la Figure 2 on peut observer la proportion des activations d'un vecteur générateur \mathbf{W}_k allouée à la reconstruction d'une catégorie de pincement donnée C . La formule de cette proportion est donnée par l'équation suivante

$$p(k, C) = \frac{\sum_{n \in C} H_{nk}}{\sum_n H_{nk}}, \quad (3)$$

où les H_{nk} sont, pour rappel, les coefficients d'activation de la trajectoire-type $W_k(t)$ dans la reconstruction de la trajectoire mesurée $V_n(t)$.

On remarque immédiatement une séparation nette entre les catégories de pincements et les trajectoire-types qu'elles activent. En effet, sur la Figure 2, les trajectoires de types 1 et 4 sont majoritairement activées pour des accords (à 84% chacun), et les trajectoires de type 2 et 3 sont utilisées majoritairement pour des arpèges (à 67% et 77% respectivement). Par contre, la différenciation par doigt ne semble pas possible par cette technique. Ces proportions sont liées à des coefficients d'activation dont la parcimonie est estimée à 55%. Une parcimonie des activations plus grande peut permettre une meilleure reconnaissance, mais une reconstruction des données de moins bonne qualité.

On peut donc nommer ces trajectoire-types $W_k(\tau)$ en fonction des catégories qu'elles représentent, et les tracer. La Figure 3 représente les $W_k(\tau)$ comme une fonction du temps représentant l'écartement de la corde à sa position initiale. Les vecteurs étant normalisés, la distance est représentée sans unités. En effet, avant leur activation par les coefficients H_{nk} , ces trajectoires représentent une forme de taille arbitraire, qui doit être dilatée ou contractée par les coefficients d'activation, puis additionnée à d'autres pour reconstruire une certaine trajectoire du jeu de donnée.

On remarque sur la Figure 3 que les pincements d'accord sont plus brefs que les pincement d'arpège, chose qui a déjà été observée dans [1]. On peut également remarquer que chaque couple de trajectoire-types (accord ou arpège) possède des caractéristiques communes. Notamment, dans

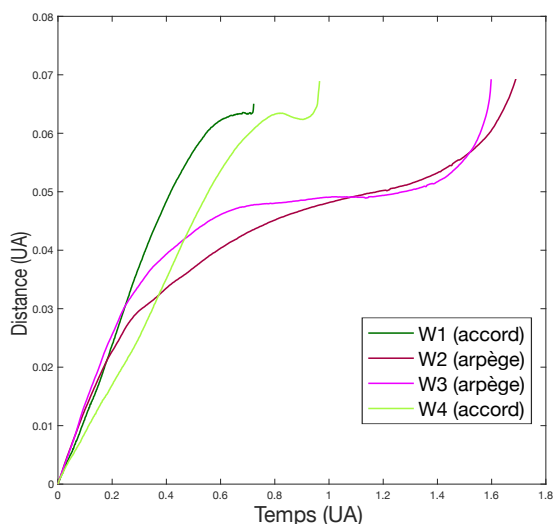


FIGURE 3 – trajectoire-types

le cas des accords, l'écartement de la corde atteint son point d'inflexion plus rapidement : la vitesse initiale est gardée plus longtemps, et les changements de direction sont plus nets et plus brefs en proportion.

4.2 Echelles temporelles des trajectoire-types

Il convient de s'attarder sur l'échelle temporelle de nos résultats.

Les temps estimés par NMF sont en unités arbitraires pour la même raison que les distance. En effet, comme l'indique l'équation 1, les périodes d'échantillonnage dans les vecteurs $\mathbf{W}_k = \begin{pmatrix} \delta_k \tau \\ W_k(\tau) \end{pmatrix}$ ne sont que des préfacteurs, activés plus ou moins fortement par les coefficients H_{nk} pour reconstruire un pincement particulier \mathbf{V}_n . Comme indiqué dans la remarque 1, nous avons donc imposé une relation de proportionnalité entre la dilatation d'une trajectoire-type et son temps de parcours. La validité de notre paramétrage est donc soumise à une condition vérifiable : bien que les trajectoire-types n'aient pas de temps de parcours absolu, les rapports de durée doivent être conservés entre pincement en accord et pincement en arpège.

Sur le Tableau 2, on peut observer les périodes d'échantillonnage moyennes de chaque catégorie connue, et celles estimée par le programme (les unités arbitraires sont normalisées pour favoriser la comparaison). On voit que le présupposé de proportionnalité de notre paramétrage nous amène à surestimer la longueur d'un pincement d'arpège par rapport à celle un pincement d'accord.

L'hypothèse de proportionnalité entre le temps de parcours et la dilatation des trajectoires, présupposée par le paramétrage de la section (3.2), ne semble permettre qu'une estimation grossière des rapports de temps entre catégories. La précision sur les données de temps conditionnant celle des données de vitesses, essentielles dans le calcul des conditions initiales, il serait donc judicieux de construire un système de catégorisation qui permette une estimation indépendante de l'échelle temporelle des trajectoire-types.

	Trajectoire	Période d'échantillonnage
Moyenne dans la base de données	Accord	0.28 ± 0.01 ms
	Arpège	51 ± 0.11 ms
Estimation par NMF	\mathbf{W}_1 (accord)	0.28 UA
	\mathbf{W}_4 (accord)	0.25 UA
	\mathbf{W}_2 (arpège)	0.67 UA
	\mathbf{W}_3 (arpège)	0.61 UA

TABLEAU 2 – Périodes d'échantillonnage caractéristique des trajectoires types trouvées par l'algorithme, et moyennées sur les catégories de la base de donnée. Les unités arbitraires sont normalisées de manière à obtenir une échelle comparable (les nombres encadrés servent de référence).

5 Conclusion et perspectives

Nous avons montré que la NMF est capable de distinguer des familles de trajectoires dans notre base de données, et d'en exhiber des représentants. Cet algorithme permet de catégoriser les trajectoires, d'identifier des catégories en lien avec le contexte de jeu, accord ou arpège, et de construire des trajectoire-types qui permettent d'étudier ses caractéristiques. Toutefois, cette catégorisation reste trop grossière pour distinguer plus que la différence entre arpège et accord, et ne permet pas d'identifier efficacement la période d'échantillonnage, nécessaire au bon calcul des données de vitesse initiale de la corde. Plusieurs perspectives s'offrent à nous pour continuer. Tout d'abord, une intégration de la trajectoire en 2-D est souhaitable pour mieux appréhender les différences entre trajectoires et obtenir une catégorisation plus fine. On pourra alors également résoudre le problème d'estimation de période d'échantillonnage en procédant directement sur les données de vitesse des trajectoires, incluant le pas de temps.

Références

- [1] Delphine Chadefaux. *Interaction musicien / instrument : le cas de la harpe de concert*. PhD thesis, Université Pierre et Marie Curie, Paris 6, 2012.
- [2] Montserrat Pàmies-Vilà. *Caractérisation du pincement de la harpe de concert en contexte musical*. Mémoire de stage de Master 2, Université Pierre et Marie Curie, Paris 6, 2015.
- [3] Nancy Bertin. *Les factorisations en matrices non-négatives. Approches contraintes et probabilistes, application à la transcription automatique de musique polyphonique*. PhD thesis, Télécom ParisTech, 2009.
- [4] Christian Vollmer, Julian P. Eggert, and Horst-Michael Gross. Modeling Human Motion Trajectories by

- Sparse Activation of Motion Primitives Learned from Unpartitioned Data. In *KI 2012 : Advances in Artificial Intelligence*, Lecture Notes in Computer Science, pages 168–179. Springer, Berlin, Heidelberg, September 2012.
- [5] Sven Rebhan, Julian Eggert, Horst-Michael Groß, and Edgar Körner. Sparse and Transformation-invariant Hierarchical NMF. In *Proceedings of the 17th International Conference on Artificial Neural Networks, ICANN'07*, pages 894–903, Berlin, Heidelberg, 2007. Springer-Verlag.
- [6] Cédric Févotte and Jérôme Idier. Algorithms for nonnegative matrix factorization with the β -divergence. *Neural computation*, 23(9) :2421–2456, 2011.
- [7] Paris Smaragdis and Judith C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, pages 177–180. IEEE, 2003.
- [8] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.
- [9] Chih-Jen Lin. Projected Gradient Methods for Nonnegative Matrix Factorization. *Neural Computation*, 19(10) :2756–2779, October 2007.