TRACKING USING BAYESIAN INFERENCE WITH A TWO-LAYER GRAPHICAL MODEL

T. Rehrl, N. Theißing*, A. Bannat, J. Gast, D. Arsić, F. Wallhoff, G. Rigoll*

Institute for Human-Machine Communication
Technische Universität München
Munich, Germany

ABSTRACT

This paper introduces a new visual tracking technique combining particle filtering and Dynamic Bayesian Networks. The particle filter is utilized to robustly track an object in a video sequence and gain sets of descriptive object features. Dynamic Bayesian Networks use feature sequences to determine different motion patterns. A Graphical Model is introduced, which combines particle filter based tracking with Dynamic Bayesian Network-based classification. This unified framework allows for enhancing the tracking by adapting the dynamical model of the tracking process according to the classification results obtained from the Dynamic Bayesian Network. Therefore, the tracking step and classification step form a closed *tracking-classification-tracking* loop. In the first layer of the Graphical Model a particle filter is set up, whereas the second layer builds up the dynamical model of the particle filter based on the classification process of the Dynamic Bayesian Network.

Index Terms— particle tracking, graphical models

1. INTRODUCTION

The tracking of objects and humans is an important task in image-processing and still of high interest in ongoing research efforts. In addition, the recognition and classification of dynamic gesture patterns are still challenging tasks, especially when facing cluttered environments, changing lighting conditions, etc.

Best to our knowledge, there has been little effort made to combine tracking and classification of gestures in a unifying framework providing better recognition as well as tracking of gestures. However, some approaches heading in the same direction exist: An adaptive velocity model was introduced in [1] among other modifications for improving the particle tracking. A motion-based particle filter for head tracking was proposed in [2] and the analytical justification for its superiority over the standard condensation tracking was given in [3]. In [4], different linear dynamical models were coupled in a state-system, where the different models can be chosen according to transition probabilities deposed in a class model. A combination of condensation algorithm and Graphical Models in order to improve the tracking was presented in [5], where facial expressions were observed. In that approach, the temporal progression was subjected to the linear process of the particle filter, whereas the spatial correlation between the facial features was inferred by an undirected Graphical Model

The rest of this paper is organized as follows: In Section 2, the basic concepts of Graphical Models are given, whereas, in Section 3 the basic concepts of particle filtering are presented. Our algorithmic approach using the classification process as a substitute for the dynamical model of the particle filtering is introduced in Section 4. In Section 5, we present first promising results for real data. The paper concludes with a summary and an outlook over the next planned working steps.

*Both authors contributed equally to the work presented in this paper.

This work has partially been supported by the DFG excellence initiative research cluster *Cognition for Technical Systems – CoTeSys, www.cotesys.org.*

2. GRAPHICAL MODELS

Graphical Models (GMs) [6] are applied in many areas of research, since they provide a descriptive and illustrative way to depict problems regarding control theory, computer science, pattern recognition, etc. In general, the GMs combine probability theory and graph theory portraying the interdependences between different random variables. In this paper we consider only directed GMs, also known as Bayesian Networks (BNs). When BNs model time series data, they are called Dynamic Bayesian Networks (DBNs), which we apply to model the dynamical model of the particle filter. Hidden Markov Models (HMMs) are a sub-class of DBNs, where observations are dependent on an unobservable variable, referred as *hidden state*.

An efficient inference algorithm for GMs is the *Junction Tree Algorithm*. This algorithm uses cluster potentials (cliques ψ and separators ϕ) for describing the dependencies between random variables $(X_1, \ldots, X_n)$ by the quotient of cluster and separator potentials

$$p(X_1, \ldots, X_n) = \frac{\prod_{C \in \mathcal{C}} \psi(C)}{\prod_{S \in \mathcal{S}} \phi(S)}, \tag{1}$$

The DBN modeling the dynamical models of the different motion classes was realized with the Graphical Model Toolkit [7].

3. PARTICLE TRACKING

Reliable tracking of objects in a video sequence is still a challenging task for current research. The condensation algorithm [8] is a robust tracking method successful even under unfavorable conditions.

The observed sequence $\mathcal{Z}_T = \{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$ is related to the information the observer is interested in, which is referred to as the state sequence $\mathcal{X}_T = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ of the pattern.

In each frame t, the state $\mathbf{x}_t$ of the observed object influences the observation $\mathbf{z}_t$, which is therefore exclusively dependent on $\mathbf{x}_t$:

$$p(\mathcal{Z}_t|\mathcal{X}_t) = \prod_{i=1}^{t} p(\mathbf{z}_i|\mathbf{x}_i). \tag{2}$$

Additionally, in the classical condensation algorithm the object states $\mathbf{x}_t$ are assumed to be subject to the Markov property, i.e. dependent only on their immediate temporal predecessor:

$$p(\mathbf{x}_t|\mathcal{X}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{3}$$

The density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ expresses the dependency of the current state on its predecessor, i.e. the change of the state vector over time. Thus, the term can be interpreted as the dynamical model describing the motion of object.

The condensation algorithm can be subdivided into two steps: prediction of the current state $\mathbf{x}_t$ having the observed sequence $\mathcal{Z}_{t-1}$, and a measurement step having the entire observation sequence $\mathcal{Z}_t$. Combining both steps and regarding equation 2 and the fact that given a state $\mathbf{x}_t$ the observation sequence $\mathcal{Z}_t$ has no more information, the recursive step from one frame to its successor is given by:

$$p(\mathbf{x}_t|\mathcal{Z}_t) = k_t p(\mathbf{z}_t|\mathbf{x}_t) \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathcal{Z}_{t-1})\, d\mathbf{x}_{t-1}, \quad (4)$$

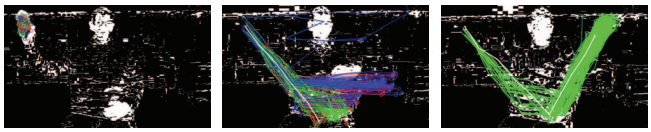
with $k_t = 1/p(\mathbf{z}_t|\mathcal{Z}_{t-1})$.

A sampling with the Monte Carlo Method is used to approximate the computationally infeasible integration over the state $\mathbf{x}_{t-1}$.

4. TRACKING USING BAYESIAN INFERENCE

The new approach presented in this paper is a fusion of a DBN classification and condensation tracking – a first impression can be seen in Figure 1.

The tracking result of an object obtained via the condensation algorithm can be used by a DBN to classify the performed motion, according to that result the dynamical model of the tracking process can be adapted, forming a closed *tracking-classification-tracking* loop. The interface between tracking and classification is the dynamical model in the particle filter: The term $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ serves as an estimation of the object motion, approximated with a linear filter. If this dynamical behavior is, instead, computed using inference on a DBN describing the dynamics more exactly, a better tracking is obtained.



(a) 1. frame (b) 4. frame (c) 10. frame

Fig. 1. In the 1. frame, the particles are placed on the hand (initialization) for the eight gesture classes. In the 4. and 10. frame, the classification results and their related tracking paths are indicated with their corresponding colors. The most probable path is indicated in white.

4.1. Conditional Density Propagation

The objective of this tracking and classification algorithm is to deduce the pattern class $\hat{m} \in \mathcal{M}$ from the observation sequence $\mathcal{Z}_T$.

The set of all discrete observation steps is denoted as $\mathcal{Z}_T = \{z_1, \ldots, z_T\}$, where T is the number of discrete time steps. The pattern class of interest is given by

$$\hat{m} = \arg\max_m p(m|\mathcal{Z}_T). \quad (5)$$

The observation model employed in the condensation algorithm assumes the observation $\mathcal{Z}_T$ to be a cluttered, noise-affected offspring of the actual, hidden state sequence $\mathcal{X}_T$ which describes all relevant properties of the pattern. A marginalization over the set of all possible states $\mathcal{X}_T$ yields

$$p(m|\mathcal{Z}_T) = \int_{\mathcal{X}_T} p(m|\mathcal{X}_T) p(\mathcal{X}_T|\mathcal{Z}_T)\, d\mathcal{X}_T \quad (6)$$

with $p(m|\mathcal{X}_T, \mathcal{Z}_T) = p(m|\mathcal{X}_T)$, since according to the observation model, $\mathcal{Z}_T$ does not contain any additional information given $\mathcal{X}_T$.

Inserting this into (5) yields

$$\hat{m} = \arg\max_m \int_{\mathcal{X}_T} p(m|\mathcal{X}_T) p(\mathcal{X}_T|\mathcal{Z}_T)\, d\mathcal{X}_T. \quad (7)$$

To avoid a high computational load, the recursive concept of the condensation algorithm is held (see Section 3), to track the pattern states in two steps: First, predict an a-priori-density $p(\mathcal{X}_t|\mathcal{Z}_{t-1})$ from the frame $t-1$. Then, measure the predicted density (using the observation) to generate the a-posteriori-density $p(\mathcal{X}_t|\mathcal{Z}_t)$.

The first-order process approximating $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ in equation 4 is replaced here by inference in a DBN taking all other RVs into account, thus the Markov property is not valid anymore constituting an important difference between our approach and the classical condensation.

In the prediction step, the a-priori-density is created as

$$p(\mathcal{X}_t|\mathcal{Z}_{t-1}) = p(\mathcal{X}_{t-1}|\mathcal{Z}_{t-1}) p(\mathbf{x}_t|\mathcal{X}_{t-1}), \quad (8)$$

since with given state vectors in the past, the past observation vectors do not provide any new information, thus
$p(\mathbf{x}_t|\mathcal{X}_{t-1}, \mathcal{Z}_{t-1}) = p(\mathbf{x}_t|\mathcal{X}_{t-1})$.

$p(\mathcal{X}_{t-1}|\mathcal{Z}_{t-1})$ is the a-posteriori-density of the preceding frame and $p(\mathbf{x}_t|\mathcal{X}_{t-1})$ is the prediction of the state vector in frame t from the state vectors in frames $1 \ldots t$. The latter term is to be computed by inference within the DBN.

Using the independence of an observation $\mathbf{z}_t$ on any RV except of its corresponding state $\mathbf{x}_t$ yielding $(\mathbf{z}_t|\mathcal{X}_t, \mathcal{Z}_{t-1}) = p(\mathbf{z}_t|\mathbf{x}_t)$, the a-posteriori-density in the measurement step can be expressed as

$$p(\mathcal{X}_t|\mathcal{Z}_t) = k_t\, p(\mathbf{z}_t|\mathbf{x}_t) p(\mathcal{X}_t|\mathcal{Z}_{t-1}), \quad (9)$$

with $k_t = 1/p(\mathbf{z}_t|\mathcal{Z}_{t-1})$, and where $p(\mathbf{z}_t|\mathbf{x}_t)$ can be evaluated by computing the value of a weight function.

4.2. The Graphical Model

The process generating the observed feature vectors of a motion can be modeled by the DBN in Figure 2. The prolog and the $T-1$ chunks constituting the DBN consist of four nodes:

The *motion class* m_t represents the class of the observed pattern indicating the kind of motion an observed object performs. An observation sequence is assumed to consist of one complete motion, from its beginning to its end. The class of motion hence remains unchanged throughout the sequence.

The *temporal progression state* q_t is similar to the state variable in a HMM. It represents the temporal progression of the motion as each state represents a discrete time step.

The *state vector* $\mathbf{x}_t$ denotes, in this case, the position of the tracked object.

The observation vector $\mathbf{z}_t$ is the actual observation. In this application, it is an array of image pixels.

For inference and prediction, only the subgraph (crosshatched) of the DBN in Figure 2 is used which does not contain the observation vectors $\mathbf{z}_t$. From $\mathcal{X}_{t-1}$ inference is applied to predict the next state vector x_t by creating 50 particles utilizing the learned transition probabilities from the DBN and measure their correlation with the observation $\mathbf{z}_t$.

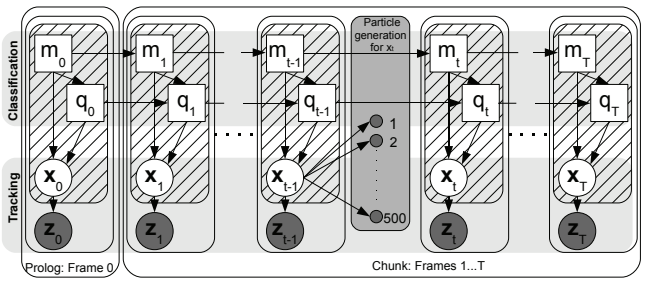


Fig. 2. The DBN used to combine tracking and classification.

4.3. Bayesian Inference

The prediction term $p(\mathbf{x}_t|\mathcal{X}_{t-1})$ is evaluated by Bayesian inference by marginalizing over each RV in the path between $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$:

$$p(\mathbf{x}_t|\mathcal{X}_{t-1}) = \sum_{m_{t-1}} \sum_{q_{t-1}} \sum_{m_t} \sum_{q_t} p(\mathbf{x}_t, m_{t-1}, q_{t-1}, m_t, q_t|\mathcal{X}_{t-1}) \tag{10}$$

Splitting the conditional probability term and considering the mutual independences of the RVs yields

$$p(\mathbf{x}_t|\mathcal{X}_{t-1}) = \sum_{m_{t-1}} \sum_{q_{t-1}} \sum_{m_t} \sum_{q_t} p(m_{t-1}, q_{t-1}|\mathcal{X}_{t-1}) \; p(m_t|m_{t-1}) p(q_t|m_t, q_{t-1}) p(\mathbf{x}_t|m_t, q_t). \tag{11}$$

The motion class m is assumed to remain constant throughout the observed motion, i.e. $m_t =: m$ for any time-step t. With this, the probability function term simplifies to

$$p(m_t|m_{t-1}) = \delta(m_t, m_{t-1}), \tag{12}$$

with the Kronecker delta δ.

Inserting this term into equation 11 results in

$$p(\mathbf{x}_t|\mathcal{X}_{t-1}) = \sum_{m} \sum_{q_{t-1}} \sum_{q_t} p(m, q_{t-1}|\mathcal{X}_{t-1}) \; p(q_t|m, q_{t-1}) p(\mathbf{x}_t|m, q_t). \tag{13}$$

By splitting the leftmost term and shifting the innermost sum according to the distributive law, the conditional probability mass function

$$p(\mathbf{x}_t|\mathcal{X}_{t-1}) = \sum_{m} p(m|\mathcal{X}_{t-1}) \sum_{q_{t-1}} p(q_{t-1}|m, \mathcal{X}_{t-1}) \sum_{q_t} p(q_t|m, q_{t-1}) p(\mathbf{x}_t|m, q_t) \tag{14}$$

describes intuitively which operations have to be performed by an algorithm in order to calculate $p(\mathbf{x}_t|\mathcal{X}_{t-1})$:

For each possible motion class m and motion state q_{t-1} in the previous time-step, their respective probability has to be determined by the term $p(m, q_{t-1}|\mathcal{X}_{t-1})$, given the knowlegde of all previous state vectors $\mathcal{X}_{t-1}$. Then, the transition probability to each current motion state q_t from its predecessor is calculated by $p(q_t|m, q_{t-1})$. Finally, the term $p(\mathbf{x}_t|m, q_t)$ predicts the current state vector $\mathbf{x}_t$ as a multi-dimensional mixture of Gaussian components whose parameters are learned in advance. This density function provides the Gaussian means and covariances for each tracked state, while the other densities can be seen as weighting factors. Thus in each time step the algorithm samples from each pattern class, from each preceding state and each current state. Using these values, it tracks the pattern state by sampling from a weighted set of Gaussian curves.

The conditional probability mass function $p(q_t|m, q_{t-1})$ can be expressed by the corresponding clique and separator potentials (see Figure 3):

$$p(\mathbf{x}_t|\mathcal{X}_{t-1}) = \sum_{m} \sum_{q_{t-1}} p(m, q_{t-1}|\mathcal{X}_{t-1}) \sum_{q_t} \frac{\psi_{\mathcal{X}_{t-1}}(m_t, q_t, q_{t-1})}{\phi_{\mathcal{X}_{t-1}}(m_t, q_{t-1})} \cdot \frac{\psi_{\mathcal{X}_{t-1}}(m_t, q_t, \mathbf{x}_t)}{\phi_{\mathcal{X}_{t-1}}(m_t, q_t)}. \tag{15}$$

The clique potential $\psi_{\mathcal{X}_{t-1}}$ and the separator potential $\phi_{\mathcal{X}_{t-1}}$ have immediate dependencies only towards their parameters. However, since their mapping is determined by global message passing at each time step, they are dependent from the whole set $\mathcal{X}_{t-1}$. For this reason, the Markov property $p(\mathbf{x}_t|\mathcal{X}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$ of the condensation algorithm does *not* apply in this case. Instead, all observed features of the past are taken account for an optimal prediction based on the maximum of available information.

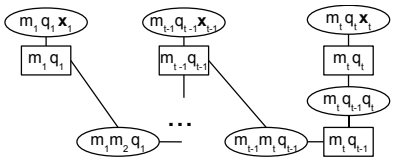


Fig. 3. The Junction Tree of the DBN, depicting the clusters (circles) and their shared separators (rectangles).

4.4. Tracking-Classification-Tracking Loop

After initialization of the tracker, in this case, we assumed the knowledge about the location of the first state $x_{t=0}$, the *tracking-classification-tracking* loop starts. The following steps are performed until the end of the sequence $t = T$ is reached:

Variable prediction: The motion class m_t and motion state q_t of the current frame are sampled with knowledge of the particle sequence.

Particle prediction: The state vector x_t has to be predicted by the sampled movement class and current state.

Measurement: verification of the validity of the predicted sample by measurement.

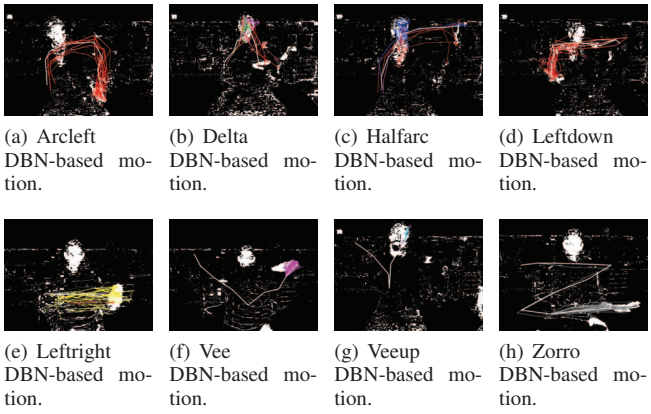
Resampling: The new particles are sampled out of the set of preceding samples considering the result of a weight function.

Classification: applying Bayesian inference from Section 4.3.

5. EXPERIMENTS

We tested the system with RGB image sequences (ten frames per sequence, ten sequences per class) for eight different gesture classes, see Figure 4. The tracking framework used this raw data to extract skin-color matrices. The skin color was used as the weighting factor for the particle filtering process. Tracking was then performed using our Bayesian Inference as the dynamical propagation model. As a comparison, a tracker using Brownian molecule motion as a representative of purely stochastical particle propagation was also used.

For evaluating the performance of the algorithm in comparison to existing methods, the measures Tracker Detection Rate (TRDR) and Object Tracking Error (OTE) presented in [9] are applied. For the initialization, a fixed number of 50 particles was used. The particles were set on the relevant object, i.e. the human hand. In each frame, the centroid of the tracked object is calculated and compared



(a) Arcleft DBN-based motion. (b) Delta DBN-based motion. (c) Halfarc DBN-based motion. (d) Leftdown DBN-based motion.

(e) Leftright DBN-based motion. (f) Vee DBN-based motion. (g) Veeup DBN-based motion. (h) Zorro DBN-based motion.

Fig. 4. Tracking results for the eight sequences (one for each class) by the Bayesian Inference Tracker.

to that of the Ground Truth. The average Cartesian distance between them in each frame constitutes the OTE:

$$\text{OTE} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(x_i^g - x_i)^2 + (y_i^g - y_i)^2} \tag{16}$$

(x_i^g, y_i^g) is the object position of Ground Truth, (x_i, y_i) the tracked one. N is the number of frames.

The detection rate is determined by defining a distance threshold and checking whether each single distance between the centroids falls below it. Since in this application the object of interest (the hand) has a diameter of approximately $d = 140$ px, the threshold is defined as $d/2 = 70$px. Each frame with a distance lower than $d/2$ is rated as a positive detection and increments the counter N_p. The detection rate is then defined as

$$\text{TRDR} = \frac{N_p}{N} \tag{17}$$

In Table 1, the results of the proposed tracking method are compared to those of a condensation tracking using Brownian Particle Motion (Gaussian Normal Distribution: mean value $\mu = 0$, standard deviation $\sigma = 100$px) [10] to predict the particles' dynamic behavior.

Classification result using GM-based approach				
	DBN		Brownian	
Class	OTE	TRDR	OTE	TRDR
Arcleft	96.67px	41.00%	98.63px	36.00%
Delta	94.75px	50.00%	96.71px	41.00%
Halfarc	45.44px	84.00%	134.34px	33.00%
Leftdown	19.36px	99.00%	94.63px	35.00%
Leftright	57.41px	71.00%	78.21px	56.00%
Vee	142.65px	34.00%	86.07px	61.00%
Veeup	58.55px	72.00%	66.01px	63.00%
Zoro	165.47px	39.00%	164.72px	27.00%
Total	85.04px	61.25%	102.42px	44.00%

Table 1. Tracking performance results: This table provides an overview for the eight classes for the achieved OTE and TRDR. In addition the achieved total performance is given.

The arithmetic mean over the eight gestures was computed to retrieve an overall OTE and TRDR. In general, the results of the presented approach are significantly better than the tracking results using Brownian Motion. The weak performance of the Brownian Motion tracking is due to the frequent clutter in the data sets.

The most probable path is depicted via a white line in Figure 4. Nevertheless, the major drawback of the current system is its lack of real-time capability due to the frequent use of Bayesian Inference for each particle and the way of integration of the GMTK for inference in the tracking system.

6. CONCLUSION AND FUTURE WORK

A new approach for enhancing tracking by fusing tracking with classification in a unifying GM was presented. A closed *tracking-classification-tracking* loop improves the tracking by adapting the dynamical model of the tracking process according to the classification results obtained from the Dynamic Bayesian Network capable of discriminating between a fixed set of motion patterns. First evaluations show the potential of the approach, however, there is still room for improvement and optimization left. The bottle-neck for the processing speed is the integration of the GMTK-based classification results. In addition, the number of motion class can be extended and the number of particles can be reduced until a optimal amount for tracking and classification process is obtained.

7. REFERENCES

[1] S. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Transactions on Image Processing*, vol. 13, pp. 1434–1456, 2004.

[2] N. Bouaynaya and D. Schonfeld, "A complete system for head tracking using motion-based particle filter and randomly perturbed active contour," 2005, vol. 5685, pp. 864–873, SPIE.

[3] N. Bouaynaya and D. Schonfeld, "On the optimality of motion-based particle filtering," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 19, no. 7, pp. 1068–1072, 2009.

[4] B. North, A. Blake, M. Isard, and J. Rittscher, "Learning and classification of complex dynamics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[5] C.Y. Su and L. Huang, "Spatio-temporal graphical-model-based multiple facial feature tracking," vol. 2005, no. 13, pp. 2091–2100, 2005.

[6] M. I. Jordan, Ed., *Learning in graphical models*, MIT Press, Cambridge, MA, USA, 1999.

[7] J. Bilmes, "Gmtk: The graphical models toolkit," 2002.

[8] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, pp. 5–28, 1998.

[9] F. Bashir and F. Porikli, "Performance evaluation of object detection and tracking systems," in *In PETS*, 2006.

[10] M. Smoluchowski, "Zur kinetischen theorie der brownschen molekularbewegung und der suspensionen," *Annalen der Physik*, , no. 21, pp. 756–780, 1906.