# CLUSTERING OF VIDEO OBJECTS BY GRAPH MATCHING

*JeongKyu Lee* *JungHwan Oh* *Sae Hwang*

Department of Computer Science and Engineering
University of Texas at Arlington, Arlington, TX 76019-0015 U. S. A.
e-mail: {*jelee, oh, hwang*}*@cse.uta.edu*

**ABSTRACT**

We propose a new graph-based data structure, called *Spatio Temporal Region Graph* (STRG) which can represent the content of video sequence. Unlike existing ones which consider mainly spatial information in the frame level of video, the proposed STRG is able to formulate its temporal information in the video level additionally. After an STRG is constructed from a given video sequence, it is decomposed into its subgraphs called *Object Graphs* (OGs), which represent the temporal characteristics of video objects. For unsupervised learning, we cluster similar OGs into a group, in which we need to match two OGs. For this graph matching, we introduce a new distance measure, called *Extended Graph Edit Distance* ($EGED$), which can handle the temporal characteristics of OGs. For actual clustering, we exploit *Expectation Maximization* (EM) with $EGED$. The experiments have been conducted on real video streams, and their results show the effectiveness and robustness of the proposed schemes.

## 1. INTRODUCTION

Graph is a powerful tool for pattern representation and classification in various fields [1, 2, 3], such as image processing, video analysis, and biomedical applications. The primary advantage of graph-based representation is that it can represent patterns and relationships among data easily. To take this advantage into video analysis, several studies have proposed the graph-based techniques [4, 5, 6, 7]. In Region Adjacency Graph (RAG) [4, 5], segmented regions and spatial relationships among them are expressed as nodes and edges, respectively. However, RAG cannot represent the temporal characteristic of video which is its representative feature. Also, various graph matching algorithms such as bipartite matching [6] and error-correcting matching [7] have been used in video data. However, the existing graph matching algorithms still require high computational cost, and suffer from low accuracy since they consider only the spatial feature to match video data.

To address these, we first propose a new graph-based data structure, called *Spatio-Temporal Region Graph* (STRG) representing spatial and temporal relationships among objects in a video sequence. The STRG is constructed by combining RAGs which are generated from each frame, and decomposed into its subgraphs, called *Object Region Graph*s (ORGs) representing the same corresponding regions. ORGs representing the same object are merged into an *Object Graph* (OG) which represents each semantic object in a video sequence. For unsupervised learning, we cluster similar OGs into a group, in which we need to match two OGs. For this graph matching, we introduce a new distance measure, called *Extended Graph Edit Distance* ($EGED$), which can handle temporal characteristics of OGs. Our contributions are as follows:

- We propose a new data structure, STRG based on graph for video. It can represent not only spatial features of objects in a video, but also temporal relationships among them.
- We propose a new distance measure $EGED$ which provides more accurate measurement between OGs by considering temporal characteristics.
- For unsupervised learning, we exploit a model-based clustering algorithm (EM) with $EGED$.

The remainder of this paper is organized as follows. In Section 2, we explain how to construct an STRG from RAGs, and how to decompose STRG into OGs. In Section 3, we introduce the $EGED$ for graph matching, and a model-based clustering algorithm to group similar OGs. The performance study is reported in Section 4. Finally, Section 5 presents some concluding remarks.

## 2. GRAPH-BASED DATA STRUCTURE FOR VIDEO

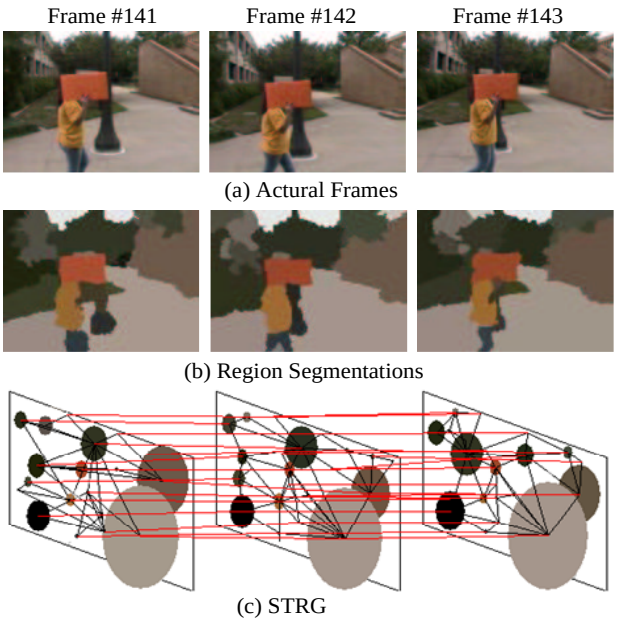
In this section, we describe STRG and OG for video.

### 2.1. Spatio-Temporal Region Graph

For a given video, each frame is segmented into a number of regions using region segmentation technique. Then, Region Adjacency Graph (RAG) is obtained by converting each region into node, and spatial relationships among regions into edges [4, 5]. RAG is good for representing spatial relationships among the nodes indicating the segmented regions. However, it cannot represent temporal characteristics of video. We propose a new graph-based data structure for video, *Spatio Temporal Region Graph* (STRG) which is temporally connected RAGs. The STRG can handle both temporal and spatial characteristics of video, and defined as follows:

**Definition 1** *Given a video segment $S$, a Spatio-Temporal Region Graph, $Gst(S)$, is a six-tuple graph, $Gst(S) = \{V, E_S, E_T, \nu, \xi, \tau\}$, where*

- *$V$ is a finite set of nodes for segmented regions from $S$,*
- *$E_S \subseteq V \times V$ is a finite set of spatial edges of $S$,*
- *$E_T \subseteq V \times V$ is a finite set of temporal edges of $S$,*
- *$\nu : V \rightarrow A_V$ is a set of functions generating node attributes,*
- *$\xi : E_S \rightarrow A_{E_S}$ is a set of functions generating spatial edge attributes,*
- *$\tau : E_T \rightarrow A_{E_T}$ is a set of functions generating temporal edge attributes.*




(a) Actual Frames

(b) Region Segmentations

(c) STRG

**Fig. 1**. Example of STRG for frame #141 − #143

In STRG, the node attributes ($A_V$) represent size (i.e., number of pixels), dominant color and location of corresponding region, the spatial edge attributes ($A_{E_S}$) represent the relationships between two adjacent nodes such as spatial distance and orientation, and the temporal edge attributes ($A_{E_T}$) represent the relationships between corresponding nodes in two consecutive frames such as velocity and moving direction. Fig. 1 (a) and (b) are actual frames in a sample video and their region segmentation results, respectively. Fig. 1(c) shows a part of STRG for frames #141 − #143 constructed by adding temporal edges which are horizontal lines between the frames.

An STRG is an extension of RAGs by adding temporal edges ($E_T$) to them. $E_T$ represents temporal relationships between corresponding nodes in two consecutive RAGs. The main procedure of building STRG is therefore, how to construct $E_T$, which is similar to the problem of objects tracking in a video sequence. To find the corresponding nodes in two consecutive RAGs, we use a *graph isomorphism* and *maximal common subgraph* [3]. These algorithms are conceptually simple, but have a high computational complexity. To address this, we decompose a RAG into its neighborhood graphs ($G_N(v)$) which are subgraphs of RAG as follows:

**Definition 2** *$G_N(v)$ is the neighborhood graph of a given node $v$ in a RAG, if for any nodes $u \in G_N(v)$, $u$ is the adjacent node of $v$, and has one edge such that $e_S = (v, u)$.*

Let $\mathbb{G}_N^m$ and $\mathbb{G}_N^{m+1}$ be sets of the neighborhood graphs in $m^{th}$ and $(m+1)^{th}$ frames respectively. For each node $v$ in $m^{th}$ frame, the goal is to find the corresponding target node $v'$ in $(m+1)^{th}$ frame. To decide these corresponding nodes, we use the neighborhood graphs in Definition 2. For each neighborhood graph $G_N(v)$ in $\mathbb{G}_N^m$, the goal is converted to finding the corresponding target graph $G_N(v')$ in $\mathbb{G}_N^{m+1}$, which is an isomorphic or the most similar graph to $G_N(v)$. First, we find the neighborhood graph in $\mathbb{G}_N^{m+1}$, which is isomorphic to $G_N(v)$. Second, if we cannot find any isomorphic graph in $\mathbb{G}_N^{m+1}$, we find the most similar neighborhood graph to $G_N(v)$ using a similarity measure, $SG(G_N(v), G_N(v'))$, which is defined as follows:

$$SG(G_N(v), G_N(v')) = \frac{|G_C|}{min(|G_N(v)|, |G_N(v')|)} \quad (1)$$

where $|G|$ denotes the number of nodes of $G$, and $G_C$ is the maximal common subgraph of $G_N(v)$ and $G_N(v')$. $G_C$ can be computed based on maximal clique detection [8]. The higher the value of $SG$, the more similarity between $G_N(v)$ and $G_N(v')$. For $G_N(v) \in \mathbb{G}_N^m$, $G_N(v')$ is the corresponding neighborhood graph in $\mathbb{G}_N^{m+1}$, whose $SG$ with $G_N(v)$ is the largest among neighborhood graphs in $\mathbb{G}_N^{m+1}$, and greater than a certain threshold value. In this way, we find all pairs of corresponding neighborhood graphs (eventually corresponding nodes) from $\mathbb{G}_N^m$ to $\mathbb{G}_N^{m+1}$.

### 2.2. Object Graph

An STRG constructed in the previous subsection is decomposed into *Object Region Graphs* (ORGs). We consider a temporal subgraph that can be defined as a set of sequential nodes connected to each other by a set of temporal edges ($E_T$) as follows:

**Definition 3** *Given a graph $Gst = \{V, E_S, E_T, \nu, \xi, \tau\}$, a temporal subgraph of Gst is a graph, $Gst' = \{V', E'_S, E'_T, \nu', \xi', \tau'\}$ such that*

- *$V' \subseteq V$, $E'_S = E_S \cap (V' \times V')$ and $E'_T = E_T \cap (V' \times V')$*
- *$\nu'$, $\xi'$ and $\tau'$ are the restrictions of $\nu$, $\xi$ and $\tau$ to V, $E_S$ and $E_T$, respectively, i.e.*

$$\nu'(v) = \begin{cases} \nu(v) & \text{if } v \in V', \\ undefined & \text{otherwise,} \end{cases}$$

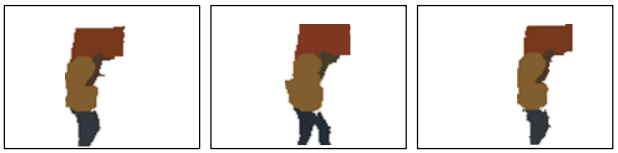
$$\xi'(e_S) = \begin{cases} \xi(e_S) & \text{if } e_S \in E'_S, \\ undefined & \text{otherwise,} \end{cases}$$

$$\tau'(e_T) = \begin{cases} \tau(e_T) & \text{if } e_T \in E'_T, \\ undefined & \text{otherwise.} \end{cases}$$

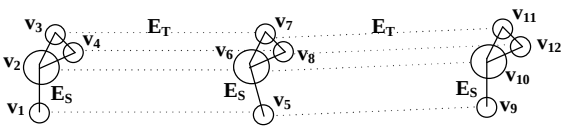
An ORG is a special case of temporal subgraph of STRG when the spatial edge set $E_S$ is empty. However, due to the limitations of region segmentation techniques, different color regions belonging to a single object cannot be detected as a single region. For instance, a body of person may consist of several regions such as head, upper body and lower body. Fig. 2 (a) shows an object (a person) which is segmented into four regions over three frames. Since there are four regions in each frame, we build four ORGs, i.e. $(v_1, v_5, v_9)$, $(v_2, v_6, v_{10})$, $(v_3, v_7, v_{11})$, and $(v_4, v_8, v_{12})$ like Fig. 2 (b). Since they belong to a single object, it is better to merge those ORGs into one. For convenience, we refer to the merged ORGs as *Object Graph* (OG). In order to merge two ORGs which belong to a single object, we consider the attributes (i.e. velocity and moving direction) of temporal edge ($E_T$). If two ORGs have same moving direction and same velocity, these can be merged into one. In Fig. 2 (c), four ORGs are merged into a single OG, i.e. $(v_2, v_6, v_{10})$.

## 3. DISTANCE FUNCTION AND CLUSTERING OF OBJECT GRAPHS

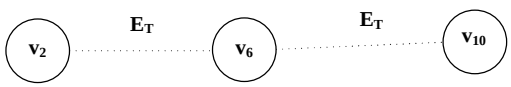
For unsupervised learning, we cluster similar OGs into a group, in which we need to match two OGs. For this graph matching, we introduce a new distance measure, called *Extended Graph Edit Distance* ($EGED$), which can handle temporal characteristics of OGs, then employ a model-based clustering algorithm using *Expectation Maximization* (EM).



(a) Sample object segmented several parts




(b) Example of ORGs for (a)




(c) Merged OG

**Fig. 2**. The example of OG merging

### 3.1. Extended Graph Edit Distance

The purpose of the edit distance for graphs is to compute the minimum cost of graph edit operations such as adding, deleting, and changing nodes, to transform one graph to the other. However, a graph edit distance uses a simple edit cost function, which may cause very low accuracy in computing the distance between OGs because it does not consider time. To address it, we consider a temporal characteristic of OG to compute the distance (dissimilarity) between OGs. Since the main operations to edit graphs deal with nodes and their attributes rather than edges, we consider only the nodes and their attributes. Let $OG^s_m$ and $OG^t_n$ be $s^{th}$ and $t^{th}$ OGs with $m$ and $n$ number of nodes, respectively.

$$OG^s_m = \{v^s_1, \ldots, v^s_m, \nu^s\}, \quad OG^t_n = \{v^t_1, \ldots, v^t_n, \nu^t\}$$

The distance function $EGED$ between $OG^s_m$ and $OG^t_n$ can be defined as follows.

**Definition 4** *The Extended Graph Edit Distance ($EGED$) between two moving object graphs $OG^s_m$ and $OG^t_n$ is defined as:*

$$EGED(OG^s_m, OG^t_n) = \begin{cases} \sum_{i=1}^{m} |v^s_i - g_i| & \text{if } n = 1, \\ \sum_{i=1}^{n} |v^t_i - g_i| & \text{if } m = 1, \\ min[EGED(OG^s_{m-1}, OG^t_{n-1}) + dist(v^s_m, v^t_n), \\ \quad EGED(OG^s_{m-1}, OG^t_n) + dist(v^s_m, gap), \\ \quad EGED(OG^s_m, OG^t_{n-1}) + dist(gap, v^t_n)] \\ & \text{otherwise.} \end{cases}$$

*where $gap$ is an added, deleted or changed node, and $g_i$ is a gap for $i^{th}$ node. And,*

$$dist(v^s_i, v^t_j) = \begin{cases} |v^s_i - v^t_j| & \text{if } v^s_i, v^t_j \text{ are not a gap} \\ |v^s_i - g_j| & \text{if } v^t_j \text{ is a gap} \\ |v^t_j - g_i| & \text{if } v^s_i \text{ is a gap.} \end{cases}$$

For better readability, let $v$ indicate a value $\nu(v)$ of node attribute. $dist$ is the cost function for editing nodes. Depending on how to select a gap ($g_i$), various distance functions are possible. In our case, $g_i = \frac{v_{i-1}+v_i}{2}$ is used for $dist$, which can handle local time shifting for the distance function accurately.

### 3.2. Clustering with EM + $EGED$

In order to group similar OGs, we employ EM clustering algorithm. First, OGs are selected randomly from the population. Let $Y_j$ be the $j^{th}$ OG with a dimension $d$. Each OG is assigned to a cluster $k$ with a probability of $w_k$ such that $\sum_{k=1}^{K} w_k = 1$, which is the sum of the membership probabilities of all the measurements for $Y_j$ to a cluster. A finite Gaussian mixture model is chosen to cluster OGs since it is widely used and easy to implement [9]. Also, $EGED$ in Definition 4 is applied for the distance measure. The $d$-dimensional Gaussian mixture density using $EGED$ is given by

$$p(Y_j|\Theta) = \sum_{k=1}^{K} \frac{w_k}{2\pi^{1/2}|\sigma_k|} e^{-\frac{1}{2\sigma^2} EGED(Y_j, \mu_k)^2} \tag{2}$$

Equation (2) is $one$-dimensional Gaussian mixture density function with the $EGED$ for OGs. This mixture model provides some benefits to handling OGs as follows. It can reduce the dimension, deal with various time lengths of OGs, and give an appropriate distance function for OGs in each cluster. Suppose that $Y$s are mutually independent, the log-likelihood ($\mathcal{L}$) of the parameters ($\Theta$) for a given data set $Y$ can be defined from Equation (2) as follows.

$$\mathcal{L}(\Theta|Y) = log \prod_{j=1}^{M} p(Y_j|\Theta) = \sum_{j=1}^{M} \log \sum_{k=1}^{K} w_k p_k(Y_j|\theta_k) \tag{3}$$

To find appropriate clusters we estimate the optimal values of the parameters ($\theta_k$) and the weights ($w_k$) in Equation (3) using the EM algorithm which is a common procedure used to find the Maximum Likelihood Estimates (MLE) of the parameters iteratively. The EM algorithm produces the MLE of the unknown parameters iteratively. Each iteration consists of two steps: E-step and M-step.
**E-step**: It evaluates the posterior probability of $Y_j$, belonging to each cluster $k$. Let $h_{jk}$ be the probability of $j^{th}$ OG for a cluster $k$, then it can be defined as follows:

$$h_{jk} = P(k|Y_j, \theta_k) = \frac{w_k}{p_k(Y_j|\theta_k)}$$

**M-step**: It computes the new parameter value that maximizes the probability using $h_{jk}$ in E-step as follows:

$$w_k = \frac{1}{M} \sum_{j=1}^{M} h_{jk}, \quad \mu_k = \frac{\sum_{j=1}^{M} h_{jk} Y_j}{\sum_{j=1}^{M} h_{jk}}$$

$$\sigma_k = \frac{\sum_{j=1}^{M} h_{jk} EGED(Y_j, \mu_k)^2}{\sum_{j=1}^{M} h_{jk}}$$

The iteration of E and M steps is stopped when $w_k$ is converged for all $k$. After the maximum likelihood model parameters ($\hat{\Theta}$) in Equation (3) are decided, each OG is assigned to a cluster.

## 4. EXPERIMENTAL RESULTS

To access the proposed method for clustering OGs, we performed the experiments with the real video streams captured by a video camera. Table 1 shows the description of the video and results of the experiments. The first two videos (Room1 and Room2) were taken from a room in a building, and the other two (Car1 and Car2)

from outside, which have some traffic scenes. The third and the fourth columns of Table 1 are the number of actual video objects and the number of correctly detected OGs, respectively. As seen in the fifth column, the accuracy of graph-based object tracking reaches up to 94.7% on average.

**Table 1**. Results of graph-based object detection and clustering for real video streams

| Video | Duration | OG performance | | | Clustering Error Rate | | |
|---|---|---|---|---|---|---|---|
| | | Actual OGs | Found OGs | Accu-racy | EM | KM | KHM |
| Room1 | 40h 30m | 438 | 411 | 93.8% | 16.8% | 33.6% | 29.1% |
| Room2 | 4h 12m | 159 | 147 | 92.5% | 14.4% | 28.9% | 22.7% |
| Car1 | 15m | 202 | 195 | 96.5% | 8.8% | 17.6% | 13.0% |
| Car2 | 12m | 210 | 203 | 96.7% | 9.5% | 17.7% | 13.3% |
| Total | 45h 7m | 1009 | 956 | 94.7% | 12.4% | 24.5% | 19.5% |

We compare the performance of EM clustering algorithm with K-means (KM) and K-harmonic means (KHM). To be fair, all of the clustering algorithms use $EGED$ distance measure defined in Definition 4. In order to evaluate the clustering algorithm, we use the clustering error rate defined as:

$$Clustering\ Error\ Rate\ (\%)\ = (1 - \frac{Number\ of\ Correctly\ Clustered\ OGs}{Number\ Of\ Total\ OGs}) \times 100$$

Table 1 also shows that EM is around two times better than KM and KHM for all videos in terms of the clustering error rate. Fig. 3 shows the example of clustering result for the first video (Room1). As seen in this figure, OGs are grouped into 8 clusters. The first column indicates the number of clustered OGs, and the second column is the visualization of each cluster by plotting its members (OGs). Two sample OGs of each cluster are shown in the third column by some selected frames. The different clusters have different characteristics: for example, Cluster 2 has the objects moving bottom to top-right corner, and Cluster 3 has a similar pattern but with an opposite direction to Cluster 2. The interesting results are observed in Cluster 7 such that it has the noise data such as unexpected illumination changes at night. The algorithm clusters even those noise data into separated groups correctly.

## 5. CONCLUSIONS

In this work we propose a new graph-based data structure, spatio-temporal region graph (STRG) representing spatial and temporal relationships among objects in a video. After an STRG is construed, it is decomposed into its subgraphs called object graphs (OGs), which represent each semantic object in a video sequence. Since an STRG provides not only spatial view of individual frame but also temporal relationships between consecutive frames, we can detect video objects more accurately. For unsupervised learning, we cluster similar OGs into a group, in which we match two OGs. For this graph matching, we introduced a new distance measure, extended graph edit distance ($EGED$) which can handle temporal characteristics of OGs. For actual clustering, we employed a model-based EM clustering with $EGED$. It can cluster video objects semantically. The experimental results on real video streams show the effectiveness and robustness of the proposed schemes.

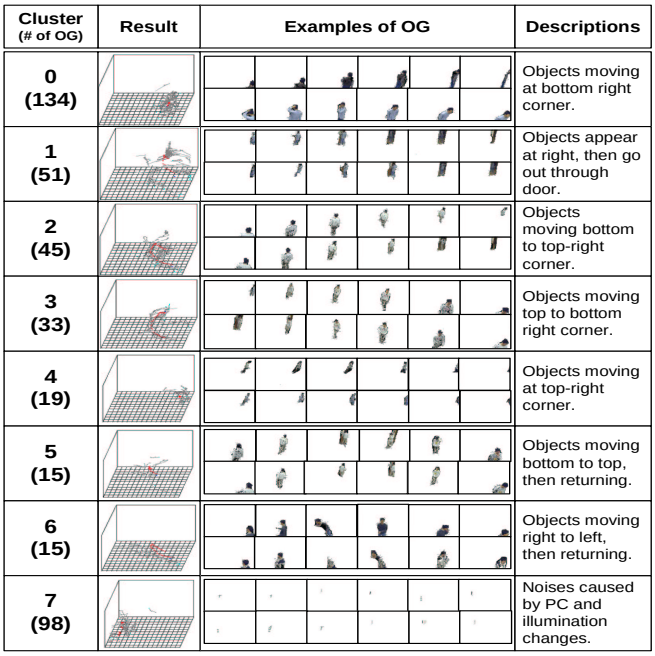



**Fig. 3**. Results of EM clustering with $EGED$ for video (Room1)

## 6. REFERENCES


[1] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Graph matching applicaations in pattern recognition and image processing," *Proc. of ICIP '03*, pp. 14–17, 2003.

[2] S. Lu, M.R. Lyu, and I. King, "Video Summarization by Spatial-Temporal Graph Optimization," in *Proceedings of the 2004 International Symposium on Circuits and Systems*, Vancouver, Canada, May 2004, vol. 2, pp. 197–200.

[3] H. Bunke and K. Shearer, "A Graph Distance Metric based on the Maximal Common Subgraph," *Pattern Recognition Letters 19*, pp. 255–259, 1998.

[4] O. Miller, E. Navon, and A. Averbuch, "Tracking of Moving Objects based on Graph Edges Similarity," in *Proc. of the ICME '03*, 2003, pp. 73–76.

[5] Chang Yuan, Yu-Fei Ma, and Hong-Jiang Zhang, "A Graph-Theoretic Approach to Video Object Segmentation in 2D+t Space," Tech. Rep., MSR, March 2003.

[6] H. T. Chen, H. Lin, and T. L. Liu, "Multi-object tracking using dynamical graph matching," *Proc. of the 2001 IEEE Conf. on CVPR*, pp. 210–217, 2001.

[7] C. Gomila and F. Meyer, "Tracking Objects by Graph Matching of Image Partition Sequences," *Proc. of 3rd IAPR-TC15 Workshop on GRPR*, pp. 1–11, 2001.

[8] G. Levi, "A Note on the Derivation of Maximal Common Subgraphs of Two Directed or Undirected Graphs," *Calcols 9*, pp. 341–354, 1972.

[9] R. Chandramouli and Vamsi K. Srikantam, "On Mixture Density and Maximum Likelihood Power Estimation via Expectation-Maximization," in *Proc. of the 2000 Conference on Asia South Pacific Design Automation*, Yokohama, Japan, 2000, pp. 423–428.