Visual Graphs from Motion (VGfM): Scene Understanding with Object Geometry Reasoning

Paul Gay[1(✉)], James Stuart[2], and Alessio Del Bue[1]

[1] Visual Geometry and Modelling (VGM) Lab, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova, Italy
{paul.gay,alessio.delbue}@iit.it

[2] Center for Cultural Heritage Technology, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova, Italy
stuart.james@iit.it

Abstract. Recent approaches on visual scene understanding attempt to build a scene graph – a computational representation of objects and their pairwise relationships. Such rich semantic representation is very appealing, yet difficult to obtain from a single image, especially when considering complex spatial arrangements in the scene. Differently, an image sequence conveys useful information using the multi-view geometric relations arising from camera motions. Indeed, object relationships are naturally related to the 3D scene structure. To this end, this paper proposes a system that first computes the geometrical location of objects in a generic scene and then efficiently constructs scene graphs from video by embedding such geometrical reasoning. Such compelling representation is obtained using a new model where geometric and visual features are merged using an RNN framework. We report results on a dataset we created for the task of 3D scene graph generation in multiple views.

Keywords: Scene graph · 3D object detection · Scene understanding

1 Introduction

The ability to automatically generate semantic relationships between objects in a scene is useful in numerous fields. As such, in recent years there has been a significant amount of research toward this goal [7,18,19,22,33] leading to the proposal of encoding relationships using a scene graph [13].

Common approaches for constructing scene graphs utilize visual appearance to guide the process, relying mainly on extracted Convolutional Neural Network (CNN) features. However, CNN visual features fail to encode spatial relationships

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-20893-6_21) contains supplementary material, which is available to authorized users.

C. V. Jawahar et al. (Eds.): ACCV 2018, LNCS 11363, pp. 330–346, 2019.
https://doi.org/10.1007/978-3-030-20893-6_21

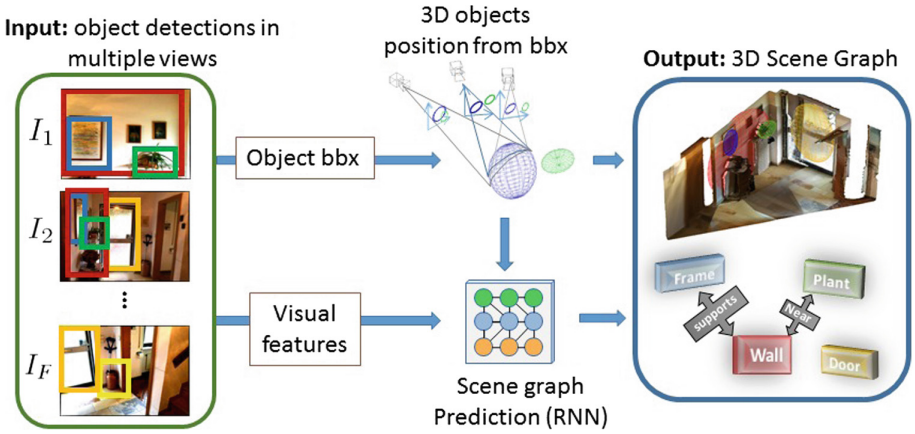

Fig. 1. Overview of the Visual Graphs from Motion (VGfM) approach for 3D scene graph generation from multiple views. As input, an object detector extracts and matches 2D bounding boxes from objects in multiple images. The 3D position and occupancy of each object are estimated and, in parallel, visual features are extracted from each bounding box. These elements are then used to predict a 3D graph where each edge defines the semantic relationship between a pair of ellipsoids

due to their invariance properties [25]. This is further compounded when considering complex 3D scenes where relationship predicates can become ambiguous and not easily solvable from a single view. This is exemplified in Fig. 1, considering the image I_2 the plant could be, '*near* the wall' or '*supported by* the wall'. This ambiguity can be rectified through the understanding provided by adjacent images, resolving for the camera pose and predicting the *near* predicate – 'plant *near* the wall', as the *support* predicate would be related to the shelf. In this paper we aim to encode the required information using the knowledge of the 3D geometry of the objects in the scene.

We therefore propose to combine the advantages of both visual and geometric information to efficiently predict spatial relations between objects as shown in Fig. 1. Given a set of 2D object bounding box detections matched across a sequence of images, using multi-view relations we compute the 3D locations and occupancies of objects described as a set of 3D ellipsoids. At the same time, we extract visual features from each object detection to model their visual appearance. These two representations are given as input to a Recurrent Neural Network (RNN) which has learned to predict a coherent scene graph where the objects are vertices and their relationships are edges. Overall, our Visual Graphs from Motion (VGfM) approach is appealing as it combines geometric and semantic understanding of an image, which has been a long term goal in computer vision [1,4,12,23,26,28,30]. We demonstrate the effectiveness of such a representation by creating a new dataset for 3D scene graph evaluation that is derived from the data provided in ScanNet [5]. To summarize, our contributions in this paper are:

- To define the problem and the model related to the computation of 3D scene graph representations across multiple views;
- To extract reliable geometric information in multiple views, we propose an improved geometric method able to estimate objects position and occupancy in 3D, modelled as a set of quadrics;
- Finally, to provide a new real world dataset, built over ScanNet, which can be used to learn and evaluate 3D scene graph generation in multiple views[1].

The paper is structured as follows. Relevant literature to the VGfM approach is reviewed in Sect. 2. We outline our refined strategy for object 3D position and occupancy in Sect. 3, then present VGfM and its learning procedure in Sect. 4. The dataset is described in Sect. 5 with detailed evaluation of VGfM performance and the benefit of geometry refinement. We then conclude the paper in Sect. 6.

2 Related Work

We now review the 3 topics related to our approach: scene graph generation from images, classification from videos and 3D object occupancy estimation.

Early works on visual relation detection were training classifiers to detect each relation in an image independently from each other [9,19]. However, a scene graph often contains chains of relationships for instance: *A man HOLDING a hand BELONGING TO a girl*. Intuitively, a model able to leverage on this fact should obtain more coherent scene graphs. To account for this, Xu *et al.* [31] proposed a model which explicitly defines a 2D scene graph. The framework naturally deals with chains of relations because inference is performed globally over all the objects and their potential relations. To this end, a message passing framework was developed using standard RNNs. This is in line with current approaches which combine the strengths of graphical models and neural networks [17,34]. Each object and relation represents a node in a two layer graph and is modelled by the hidden state of an RNN. The state of each node is refined by the messages sent from adjacent nodes. This architecture has the flexibility of graphical models and thus can be used to merge heterogeneous sources of information such as text and images [16]. We utilize this mechanism in our model while extending it to incorporate the 3D geometry. To the best of our knowledge, this is the first time that geometric reasoning is exploited for scene graph generation.

Object detection within a sequence (video) is largely still reliant on temporal confidence aggregation across image detections or applying RNN for temporal memory [29]. With the difficulty of predicting confidence within a CNN [21] these approaches rely on detection consistency. Alternatively, more advanced video tubelets in T-CNN [14] are optimized for the detection confidence. In a similar way, we exploit the multiple view information within our model by including a fusion mechanism based on message passing across images.

[1] Code and data can be found at: https://github.com/paulgay/VGfM.

Recently, new techniques have emerged to estimate the 3D spatial layout of the objects as well as their occupancy [2,10,24]. These techniques rely on the quality of deep learning object detectors [10,24] or the use of additional range data [2]. Similarly volumetric approaches have been used to construct the layout of objects in rooms, or construct objects and regress their positioning [30]. These strategies provide alternative representations for scene graph generation since they associate object labels to the 3D structure of the scene, but lack the relationships required to construct a scene graph. In particular the approach localization from Detection (LfD) [24] leverages 2D object detector information to obtain the 3D position and the occupancy of a set of objects represented through quadrics. Although ellipsoids are an approximation of the region occupied by an object, they provide the necessary support for spatial reasoning in a closed form which can be efficiently computed. However, in the current methods [11,24], there is no explicit constraint to enforce the quadric to be a valid ellipsoid. As a consequence, low baselines and inaccurate bounding boxes might result in degenerate quadrics. In the next section, we present an extension named LfD with Constraints (LfDC) which is based on linear constraints on the quadric centers. It has the advantage of being a fast closed-form solution while being more robust than LfD [24].

3 Robust Object Representation with 3D Quadrics

Even if they are an approximate representation of objects, a representation based on ellipsoids (or formally quadrics) can be embedded in the graph effectively with multiple views (as described in Sect. 4). In this section, we briefly consider the prior work for generating quadrics from multi-view images, then resolve for their limitations so making the approach more suitable for scene graph construction.

Let us consider a set of image frames $f = \{1 \dots F\}$ representing a 3D scene under different viewpoints. A set of $i = \{1 \dots N\}$ rigid objects is placed in arbitrary positions. We assume that each object is detected in at least 3 images. Each object i in each image frame f is given by a 3×3 symmetric matrix $\mathtt{C}_{if}$ which represents an ellipse inscribed in the bounding box as shown in Fig. 1 (left & top middle). The aim is to estimate the 4×4 matrix $\mathtt{Q}_i$ representing the 3D ellipsoid whose projection onto the image planes best fit the measured 2D ellipses $\mathtt{C}_{if}$. The relationship between $\mathtt{Q}_i$ and their reprojected conics $\mathtt{C}_{if}$ is defined by the 3×4 perspective camera matrices $\mathtt{P}_f$ which are assumed to be known (i.e. the camera is calibrated). The LfD method described in [24] solves the problem in the dual space where it can be linearized as:

$$\beta_{if}\mathbf{c}_{if} = \mathtt{G}_f\mathbf{v}_i, \tag{1}$$

where β_{if} is a scaling factor, the 6-vector $\mathbf{c}_{if}$ is the vectorised conic of the object i in image f, the 10-vector $\mathbf{v}_i$ is the vectorised quadric and the matrix $\mathtt{G}_f$ contains the elements of the camera projection matrix after linearization[2]. Then, stacking

[2] The supplemental material provides more mathematical details about this step.

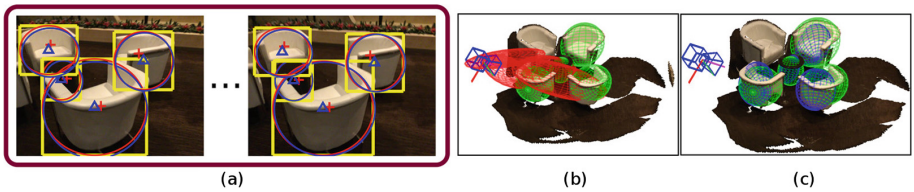

Fig. 2. (a) are example ellipse reprojections of LfD [24] in red and LfDC in blue with cross and triangle respectively for the centers. (b, c) is the point cloud, 3D quadrics (using same color labelling), camera poses for two camera, the ground truth is shown in green. It can be seen that the proposed solution overcomes the limitation of [24]. (Color figure online)

column-wise Eq. (1) for $f = 1 \dots F$, with $F \geq 3$, we obtain:

$$\mathtt{M}_i \mathbf{w}_i = \mathbf{0}_{6F}, \tag{2}$$

where $\mathbf{0}_x$ denotes a column vector of zeros of length x, and the matrix $\mathtt{M}_i \in \mathbb{R}^{6F \times (10+F)}$ and the vector $\mathbf{w}_i \in \mathbb{R}^{10+F}$ are defined as follow:

$$\mathtt{M}_i = \begin{bmatrix} \mathtt{G}_1 & -\mathbf{c}_{i1} & \mathbf{0}_6 & \mathbf{0}_6 & \dots & \mathbf{0}_6 \\ \mathtt{G}_2 & \mathbf{0}_2 & -\mathbf{c}_{i2} & \mathbf{0}_2 & \dots & \mathbf{0}_2 \\ \vdots & & & & \ddots & \\ \mathtt{G}_F & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 & \dots & -\mathbf{c}_{iF} \end{bmatrix}, \quad \mathbf{w}_i = \begin{bmatrix} \mathbf{v}_i \\ \boldsymbol{\beta}_i \end{bmatrix}, \tag{3}$$

where $\boldsymbol{\beta}_i = [\beta_{i1}, \beta_{i2}, \cdots, \beta_{iF}]^\top$ contains the scale factors of the object i for the different frames.

Since the object detector can be inaccurate, it makes sense to find the quadric $\tilde{\mathbf{w}}_i$ by solving the following minimization problem:

$$\tilde{\mathbf{w}}_i = \arg\min_{\mathbf{w}} \|\mathtt{M}_i \mathbf{w}\|_2^2, \quad s.t. \|\mathbf{w}\|_2^2 = 1, \tag{4}$$

where the equality constraint $\|\mathbf{w}\|_2^2 = 1$ avoids the trivial zero solution. The solution of this problem consists in computing the `SVD` on the $\mathtt{M}_i$ matrix and taking the right singular vector associated to the minimum singular value.

However, the algebraic minimization in Eq. (4) does not enforce the obtained quadric to be a valid ellipsoid. As can be seen in Fig. 2, fitted ellipsoids can be inaccurate despite giving a reasonable 2D projection. The proposed LfDC solution generates ellipsoids as Fig. 2c, and in turn improves overall performance.

A common indication of the estimated quadric being degenerate can be fairly guessed by checking where the estimated ellipse center is located. If the center is outside the boundaries of the estimated ellipse contour, this clearly points out to a degenerate configuration. Given this last observation, rather than constraining directly the solution to lie in a valid ellipsoid parameter space, we include a set of equations imposing the reprojection of the center of the 3D ellipsoid being

closer to the centers of the ellipses. This can be done by adding an additional set of rows in the matrix $\tilde{\mathtt{M}}_i$ used in Eq. (2).

This constraint can be added by observing that the center parameters of the vectorised dual quadric $\mathbf{v}$ appear separately in linear terms[3] at position 4, 7 and 9 in the vector. The same fact holds for the vectorised conic $\mathbf{c}$ at positions 3 and 5 (we omit indexes to simplify the notation):

$$\mathbf{c}^* = \mathbf{c}_{3,5} = \begin{bmatrix} -t_1^c & -t_2^c \end{bmatrix}, \quad \mathbf{v}^* = \mathbf{v}_{4,7,9} = \begin{bmatrix} -t_1 & -t_2 & -t_3 \end{bmatrix}, \tag{5}$$

where $\mathbf{c}^*$ and $\mathbf{v}^*$ contain the centers of the ellipse and the ellipsoid respectively. We can use this fact to directly include the equations which enforce the ellipsoid center to be projected in the centers of the ellipses. Given a frame f and an object i, the constrained equations are:

$$\mathtt{G}_f^c \mathbf{v}_i^* = \mathbf{c}_{if}^* \beta_{if}, \tag{6}$$

with the 2×10 matrix $\mathtt{G}_f^c$ defined as:

$$\mathtt{G}_f^c = \begin{bmatrix} 0 & 0 & 0 & p_{11} & 0 & 0 & p_{12} & 0 & p_{13} & p_{14} \\ 0 & 0 & 0 & p_{21} & 0 & 0 & p_{22} & 0 & p_{23} & p_{24} \end{bmatrix}, \tag{7}$$

where each value p_{ij} corresponds to an element of the camera matrix $\mathtt{P}_f$. These equations are included in the system of Eq. (2) by replacing the matrix $\mathtt{M}_i$ by $\tilde{\mathtt{M}}_i$ such that:

$$\tilde{\mathtt{M}}_i \mathbf{w}_i = \mathbf{0}_{8F}, \tag{8}$$

where the matrix $\tilde{\mathtt{M}}_i \in \mathbb{R}^{8F \times (10+F)}$ is defined as follow:

$$\tilde{\mathtt{M}}_i = \begin{bmatrix} \mathtt{G}_1 & -\mathbf{c}_{i1} & \mathbf{0}_6 & \mathbf{0}_6 & \dots & \mathbf{0}_6 \\ \mathtt{G}_1^c & -\mathbf{c}_{i1}^* & \mathbf{0}_2 & \mathbf{0}_2 & \dots & \mathbf{0}_2 \\ \vdots & & & & \ddots & \\ \mathtt{G}_F & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \dots & -\mathbf{c}_{iF} \\ \mathtt{G}_F^c & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 & \dots & -c_{iF}^* \end{bmatrix}. \tag{9}$$

The solution of this new system can then be obtained with the SVD of the $\tilde{\mathtt{M}}_i$ matrix as done for the minimization problem described in Eq. (4). This method, named LfDC, has both the effect of reducing the number of degenerated quadrics (i.e. to localize more objects in the scene) and to improve the quality of object localizations and occupancy estimation as it will be shown in the experimental section. For these reasons, LfDC also enables to improve the performances when estimating the scene graphs using multi-view relations.

[3] We refer to supplemental material for further mathematical details.

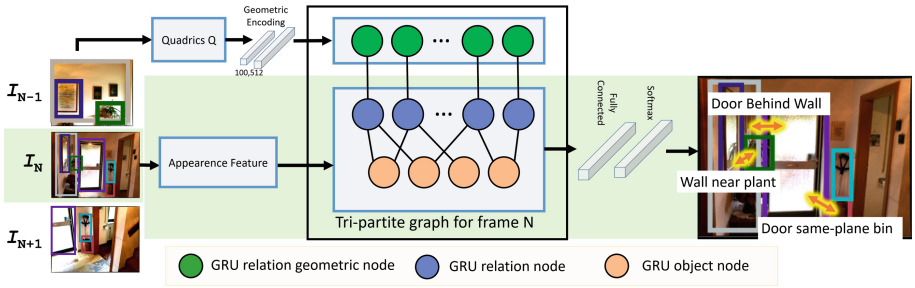

Fig. 3. Our scene graph generation algorithm takes as input a sequence of images with a set of object proposals (as ellipsoids). In addition, visual features are extracted for each of the bounding boxes, these features are then fed to initialise the GRU object and relation nodes. A tri-partite graph connects the object, relation and geometric nodes and iterative message passing updates the hidden states of the object and relation nodes. At the conclusion of the message passing the scene graph is predicted by the network and then the next image of the sequence is processed.

4 Scene Graphs from Multiple Images

The VGfM approach models the scene graph within a tri-partite graph which takes as input the features, both visual and geometric (from Sect. 3), and outputs the prediction of the object labels and predicates, as illustrated on Fig. 3. The graph merges geometric and visual information, as well as refining jointly the state of all the objects and their relationships. This process is performed iteratively over each of the F images of the sequence.

Therefore, let $G = (\vartheta, E)$ denotes the tri-partite graph of a current image. We define ϑ as the set of nodes that corresponds to attributes, defined as $\vartheta = \{\vartheta^g, \vartheta^o, \vartheta^r\}$ related to geometry, objects and relationships respectively, while E refers to pairwise edges which connect each object with its relation. The set of object nodes is denoted as $\vartheta^o = \{\vartheta^o_i, i = 1 \dots O\}$ and models their semantic states. Similarly, ϑ^r models the semantic states of the relationships and is defined as $\vartheta^r = \{\vartheta^r_{i\to j},\ i = 1 \dots O,\ j = 1 \dots O,\ i \neq j\}$. Finally, $\vartheta^g = \{\vartheta^g_{i\to j},\ i = 1 \dots O,\ j = 1 \dots O,\ i \neq j\}$ is the set of geometric nodes constructed over the quadrics previously computed expressing the geometric state of each relation (see Sect. 4.1 for construction).

The states of the graph are then iteratively refined by message passing among the nodes, exchanging information about their respective hidden states (see Sect. 4.2). The hidden states $h_{i\to j}$ (resp. h_i) of each relation node $\vartheta^r_{i\to j}$ (and resp. object node ϑ^o_i) are modelled with Gated Recurrent Units [3] (GRU). This allows each node to refine its state by exploiting incoming messages from its neighbors. Differently from the object and relation nodes, each geometric node $\vartheta^g_{i\to j}$ is considered as an observation and its state $g_{i\to j}$ is fixed, this allows the reliability of the geometric information to be enforced. If the geometric nodes are removed from the graph, we obtain the framework of [31].

After K iterations of message passing the hidden states from the object and relation nodes are used to compute the classification decision, i.e. object and relation labels, as provided by the final fully connected layer. This layer takes as input the hidden state of a relation node and produces a distribution over the relation labels through a softmax, this step is performed to compute the object labels as well. We treat predicate labels as in the multi-label scenario where a predicate is detected for a given relation if the softmax score is higher than the label indicating its absence. We further outline the training specifics of the model in Sect. 4.4. With the creation of the scene graph the next image in the sequence is then processed.

As our goal is to share information between images, we can encourage sharing beyond object and relation nodes and pass messages between images within the sequence. This can be simply performed by connecting tri-partite graph nodes ϑ^r, ϑ^o among images and this process is explained in Sect. 4.3.

4.1 Construction of the Geometric Nodes

As described in Sect. 3, we obtain a set of ellipsoids $Q = \{\mathtt{Q}_i, i = 1 \dots O\}$ from the object detections. We then extract the 3D coordinates of the center of each quadric $\mathtt{Q}_i$ and the six points at the extremities of its main axis. Finally, the geometric encoder takes as input the coordinates extracted from the ellipsoids $\mathtt{Q}_i$ and $\mathtt{Q}_j$ in order to produce the state of the geometric node $\vartheta^g_{i\to j}$. This encoder consists of a multi-layer perceptron with two fully connected layers of sizes $100, 512$. These values were identified empirically to give enough capacity to the network to link both the quadric positions and occupancies and the given complexity of the final labels. We additionally experimented with a bag-of-word based encoding, proposed in [22], and found similar performances.

4.2 Message Passing Between Nodes

The refinement of the hidden states is carried out via message passing. At each inference iteration, messages are sent along the graph edges. Each relation node $\vartheta^r_{i\to j}$ is linked by undirected edges to the object state nodes ϑ^o_i, ϑ^o_j and the corresponding geometric node $\vartheta^g_{i\to j}$. We use the message pooling scheme proposed in [31].

At each iteration, the node ϑ^o_i receives the following message:

$$m_i = \sum_{j:i\to j} \sigma(\mathbf{a}_1[h_i, h_{i\to j}])h_{i\to j} + \sum_{j:j\to i} \sigma(\mathbf{a}_2[h_i, h_{j\to i}])h_{j\to i}, \tag{10}$$

where $[,]$ denotes the concatenation operator, σ is the sigmoid function, $\{j : i \to j\}$ is the set of all the relations where object j is present at the right of the predicate, and the weights $\mathbf{a}_1$ and $\mathbf{a}_2$ are learned. The relationship nodes are also updated, where each node $\vartheta^r_{i\to j}$ receives the following message:

$$m_{i\to j} = \sigma(\mathbf{b}_1[h_i, h_{i\to j}])h_i + \sigma(\mathbf{b}_2[h_j, h_{i\to j}])h_j + \sigma(\mathbf{b}_3[g_{i\to j}, h_{i\to j}])g_{i\to j}, \tag{11}$$

where b_1, b_2 and b_3 are learned parameters.

As with loopy belief propagation, this can be seen as an approximation of an exact global optimization, enabling the refinement of each hidden state based on its context. Conversely to a classic message passing scheme, the last inference decision on the label values is not performed within the tri-partite graph but by using a last fully connected layer. On average in our experiments, the inference time is 0.25 second per image on a Tesla K80.

4.3 Sharing Information Among Multiple Images

We now extend the proposed single image model to fuse information among the images of the sequence. In this case, the visual features can be shared where the network benefits from taking into account potential appearance changes as well as aiding consistency among the views. To this end, we rely on the message passing mechanism and include cross-image links which connect the tri-partite graphs for each image. As shown in Fig. 4, each relation node receives messages from all the nodes modelling the same relation in the other images. The same principle is applied for the object nodes.

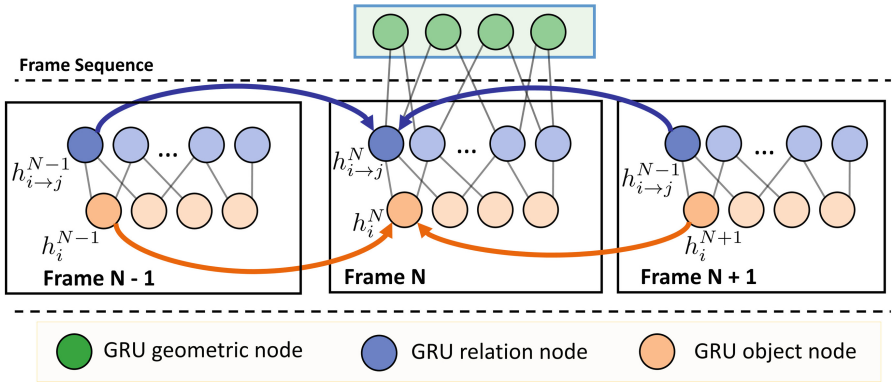

Fig. 4. This figure displays how the graphs operating on a single frame (same as shown on Fig. 3) are linked by the fusion mechanism.

We extend the notation so that it refers to nodes and messages image by image. Let us denote by $m_{i\to j}^f$ (resp. m_i^f) the message that the relation node $h_{i\to j}^f$ (resp. the object node h_i^f) appearing in the image f receives from the other images. Then, we compute the messages with the following equations:

$$m_{i\to j}^f = \sum_{l, f\neq l} \sigma(\mathbf{c}_1[h_{i\to j}^f, h_{i\to j}^l])h_{i\to j}^l, \tag{12}$$

$$m_i^f = \frac{1}{F}\sum_{l, f\neq l} \sigma(\mathbf{c}_2[h_i^f, h_i^l])h_i^l, \tag{13}$$

where $\mathbf{c}_1$ and $\mathbf{c}_2$ are learned weights. This formulation can be seen as weighted average of the visual features were the weights are learned as an attention mechanism. This new cross-image message is then added to the local one described in Sect. 4.3 to form the final message.

4.4 Learning

Our model is trained with cross-entropy loss. We also use similar hyperparameters to Xu et al. [31] with a learning rate of $1e^{-3}$ and $K = 2$ iterations of message passing. Batches of 8 images were used for the single image system. For the multi-image approach described in Sect. 4.3, each batch corresponds to one image sequence. We reduce the sequence to 10 images selected uniformly to save memory space. In contrast to [31], we retain all region proposals as we are considering the ellipsoid proposal that already prunes the per-frame object proposals. We extract visual features from VGG-16 [27] pretrained on MS-COCO and use the FC_7 layer to initialize the hidden states of the RNNs. The RNNs are trained while keeping the weights of the visual features fixed. Two sets of shared weights are optimized during training: one for the objects and one for the relations. The state of the GRU for both input and output has a dimension of 512.

5 Dataset Description and Experimental Evaluation

Prior datasets for the scene graph generation problem are based on singular images with relationship annotations, but in general they do not have multi-view image sequences necessary to exploit the proposed model. We thus create GraphScanNet by manually extending and upgrading the ScanNet dataset [5] with relationships between the annotated objects. The ScanNet dataset provides 2.5 million views in more than 1500 scans annotated with semantic and instance level segmentation. 3D camera poses are also provided as estimated from an online 3D reconstruction pipeline (BundleFusion [6]) algorithm run on the RGB-D images. Since VGfM does not require depth, we also tried a visual SLAM algorithm [20], but we found that the results were not accurate enough.

Although one thousand object categories are present, we refine the list of objects to resolve for annotator errors and the frequency of object occurrences in sequences resulting in a refined list of 34 object categories. Our annotations are a set of 8762 view-independent compositional relationships between couples of 3D objects. Our proposed predicates are inspired by Visual Genome [15], but we opt for a concise set that is loosely aimed to encompass many relationships that can occur within the sequences. It can be seen from the ScanNet class labels that when annotators are given expressive freedom in labels, cultural or personal bias can make annotations implausible for learning systems where many objects are synonyms or localized vernaculars. Our predicates are as follows:

Part-of: A portion or division of a whole that is separate or distinct; piece, fragment, fraction, or section, e.g. shelf is *part-of* a bookcase.

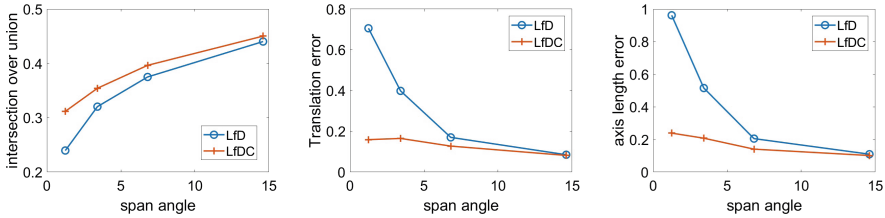

Fig. 5. Evaluation in terms of O_{3D} accuracy, translation and axes length errors for the LfD [24] method and our proposed approach LfDC.

Support: To bear or hold up (a load, mass, structure, part, etc.); serve as a foundation for. Where hypernyms could be considered support from *behind, below, hidden*; in our case 'below' is most prevalent.

Same-plane: Belonging to the same or near similar vertical plane in regards to the ground normal. As an example, a table might be *same-plan* as a chair.

Same-set: Belonging to a group with similar properties of function. The objects could define a region, e.g. in Fig. 6 the table, chair and plate belong to the same set whereas the shoes on the floor are separated. This is similar to the concept of scenario recently studied in [8], and where they proved this being a powerful clue for scene understanding.

As relationships are derived from images, there are differences in terms of number of instances for each predicate. *Same-set* and *Same-plane* appear about 30,000 times in the images, whereas *Support* 3,000 times and *Part-of* only 600 times. This has an impact on the performances as explained in the evaluation.

The 3D object segmentation enables us to construct a 3D ground-truth (GT) by fitting ellipsoids to each object mesh. Object bounding box in 2D are also computed by projecting each object point cloud into the image. Such bounding boxes are created by fitting a rectangle that encloses the set of 2D points. We then automatically extracted 2000 sequences coming from 700 different rooms with at least 4 objects in each of them. These sequences are challenging for 3D reconstruction, since the recording of the rooms was done by rotating the camera with limited translation motions. On average, the angle spanned by the camera trajectory is 4.3°.

5.1 Evaluation of the Quadric Estimation

We first evaluate how accurate are the quadrics obtained from the different methods. We run the original LfD method [24] on the extracted sequences and compared with the ones from our LfDC approach. On the 1979 sequences, we measured that only 48% of the quadrics estimated by LfD are valid ellipsoids. This number rises up to 60% when we use our LfDC method. This validates our initial hypothesis that the additional equations are useful to avoid non-valid quadrics. In the following, we evaluate the accuracy of the ellipsoids by considering only the ones who are found valid by all methods.

One of the main limitations of LfD is the sensibility when the image sequences have a short baseline (i.e. short camera path and/or very few image frames). To

study this effect, the error and accuracy values are plotted in function of the maximum angle spanned by the camera during the sequence where the object is recorded. In Fig. 5, we compare the methods according to three metrics: O_{3D}, which is the intersection over union between the proposed and the GT quadrics, the translation and the axis length errors.

We can see that the LfDC outperforms the previous LfD method in terms of the three metrics: volume overlap, translation and axis length error. The constraints on the centers are beneficial to improve on these three aspects since the solution is still computed globally for all the quadric parameters. Secondly, we observe that, although relatively small in average, the improvements are important in case of a low baseline.

5.2 Evaluation on the Scene Graph Classification Task

We evaluate our systems on the tasks of object and relation classification i.e. given the bounding boxes of a pair of objects, the system must predict both the object classes and the predicate of their relation. These two tasks encompass the problem of scene graph generation when performed recursively over the image where annotations are performed in terms of multi-label fashion i.e. presence and absence. We selected 400 rooms for training, 150 as a validation set and 150 for testing.

We first study the influence of the quadric estimation algorithms. We run our VGfM and use as input the ellipsoids provided by LfD, LfDC and GT quadrics. Results are reported in Table 1. We can see that the differences between the different methods are relatively small, but still coherent with the accuracy reported in Fig. 5. The LfD obtains the worst results and the best performing method is LfDC. Overall, the use of GT quadrics brings an additional improvement, but the accuracy remains relatively close to the other methods.

Table 1. Comparison of the use of different quadrics to classify the scene graphs. The numbers in bold are related to the best results LfD and LfDC.

Object label	GT	LfD [24]	LfDC
	76%	75%	75%
Same-plane	75%	72%	**74**%
Same-set	62%	59%	**61**%
Support	69%	64%	**67**%
Part-of	69%	65%	**69**%

We now study the influence of the different components of the system in an ablation study in Table 2. The baseline [31] uses only visual appearance. The method VGfM-2D corresponds to a variation of our method without 3D information where we computed the geometric states from the coordinates of the 2D

Table 2. This table shows the accuracy for the prediction of each predicate and the object labels. The numbers in bold are the best performing methods.

Object label	[31]	VGfM-2D	Geometric encoder	VGfM	VGfM + Fusion
	74%	74%	58%	75%	**76**%
Same-plane	74%	74%	70%	74%	**78**%
Same-set	58%	59%	55%	61%	**62**%
Support	62%	64%	**85**%	67%	64%
Part-of	68%	69%	**80**%	69%	59%

bounding boxes instead of using the ellipsoids. To evaluate the potential of using geometry alone, we also report results while using only the geometric encoder described in Sect. 4.1. A softmax layer is appended to this encoder in order to use it as classifier. The resulting network is then trained from scratch for the tasks of predicting predicates and object labels. VGfM + Fusion corresponds to the addition of the fusion mechanism over multiple images described in Sect. 4.3. The results of the baseline method do not exceed 75% of accuracy, which suggests that this task is difficult especially for the high level *Same-set* predicate. As shown on the second column, augmenting the appearance with the 2D coordinates allows VGfM-2D to obtain an improvement of 1–2% as it is commonly observed in computer vision for this kind of feature augmentation.

The results of the geometric encoder shows large differences between the tasks. The low performance for the task of object classification is not surprising as a 3D bounding box alone carries little information about the object label. Regarding the results for the predicate prediction, we tested the same architecture but providing the GT ellipsoids and found only a difference of 1–5% depending on the predicates. It is thus possible that some errors are due to partly segmented objects in the annotations, resulting in inaccurate bounding boxes. We also observe that results are higher than any other method for the predicates *Support* and *Part-of*. One explanation is that these two predicates are less frequent in the dataset (respectively 3000 and 600 instances compared to around 30000 for the other classes). In these cases with less training data, having a more simple, shallower architecture with a reduced number of parameters helps. Unfortunately, standard data augmentation techniques such as cropping or shifting cannot be directly applied to augment the number of samples as they would introduce incoherences with the 3D geometry.

The proposed single image VGfM method has a better or similar accuracy than the methods which do not use 3D information. This suggests that the information contained in the ellipsoids is beneficial for predicting relationships and that our model is able to use it. We can draw similar conclusions for the fusion mechanism. Indeed the fusion mechanism shows improvements for the predicates which are common on the dataset. For these cases, the model successfully manages to leverage on the different sources of information to reach an improvement of accuracy of 4% with respect to the initial baseline. However, it fails to improve

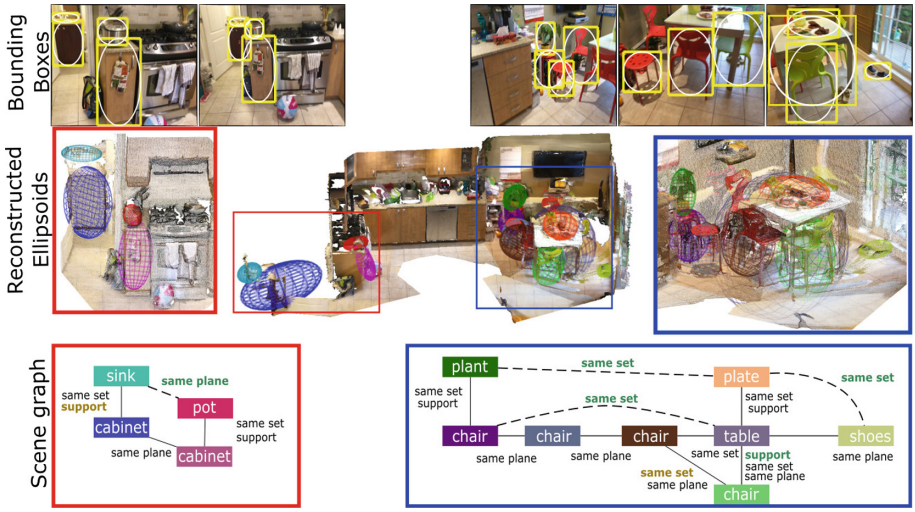

Fig. 6. The top row shows images extracted from two sequences together with the bounding box detections in yellow and in white the conics used to estimate the ellipsoids. The second row shows the resulting ellipsoids of these two sequences as well as the global object layout of the room. The third row shows the corresponding scene graphs obtained with our proposed approach. We did not display all the relations to ease the visualization. Predicates in bold brown font are miss-detections and bold green font with dashed line are false alarms (best viewed in color). (Color figure online)

for the ones which contain only a few training examples. This effect should be more important for the fusion mechanism since in this case, one training sample corresponds to a sequence of 10 images. Thus the number of instances in the training data is roughly divided by 10.

Figure 6 shows some qualitative results of two image sequences coming from the same room. On the left, the model successfully identified the two sets of objects, and it detects that the two cabinets are on the same plane. Since perspective effects are strong, this reasoning would be difficult with 2D features only. The right part is a complex scene with many overlapping objects. Although some errors are still present, leveraging over multiple views provides, as a 3D graph, a rich description of the scene which could enable further high level reasoning.

6 Conclusions

We addressed the problem of generating a 3D scene graph from multiple views of a scene. The VGfM approach leverages both geometry and visual appearance and it learns to refine globally the features and to merge the different sources of information through a message passing framework. We have evaluated on a new dataset which focuses on the relationships in 3D and show that our method outperforms a 2D baseline method.

The problem of creating a scene graph in both 2D and 3D from multiple views has been addressed for the first time in this paper, however there are many areas to be explored that can enhance performances. First, other sources of knowledge could be used. In particular, [32] shows that the manifold of the scene-graph is rather low dimensional as many of them contain recurrent patterns. This suggests that a strong prior could be built to encode this topology. Secondly, the knowledge about the visual appearance and the semantic relationships could be used to refine the geometric nodes by refining the quality of the ellipsoids. Last but not least, the case of dynamic scene could be investigated. As the predictions of our model are done per image, it can be readily applied on this setting.

References

1. Bao, S.Y., Bagra, M., Chao, Y.W., Savarese, S.: Semantic structure from motion with points, regions, and objects. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2703–2710. IEEE (2012)
2. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3D object detection network for autonomous driving. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 211–219. IEEE (2017)
3. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: encoder-decoder approaches. arXiv preprint arXiv:1409.1259 (2014)
4. Choy, C.B., Xu, D., Gwak, J.Y., Chen, K., Savarese, S.: 3D-R2N2: a unified approach for single and multi-view 3D object reconstruction. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 628–644. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_38
5. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3D reconstructions of indoor scenes. In: Computer Vision and Pattern Recognition (CVPR), pp. 2075–2084. IEEE (2017)
6. Dai, A., Nießner, M., Zollöfer, M., Izadi, S., Theobalt, C.: BundleFusion: real-time globally consistent 3D reconstruction using on-the-fly surface re-integration. Trans. Graph. (TOG) **36**, 76a (2017)
7. Dai, B., Zhang, Y., Lin, D.: Detecting visual relationships with deep relational networks. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3298–3308 (2017)
8. Daniels, Z.A., Metaxas, D.N.: Scenarios: a new representation for complex scene understanding. arXiv preprint arXiv:1802.06117 (2018)
9. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for static human-object interactions. In: Computer Vision and Pattern Recognition Workshops (CVPR), pp. 9–16. IEEE (2010)
10. Dong, J., Fei, X., Soatto, S.: Visual inertial semantic scene representation for 3D object detection. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 782–790. IEEE (2017)
11. Gay, P., Rubino, C., Bansal, V., Del Bue, A.: Probabilistic structure from motion with objects (PSfMO). In: International Conference on Computer Vision (ICCV), pp. 3075–3084. IEEE (2017)
12. Hane, C., Zach, C., Cohen, A., Pollefeys, M.: Dense semantic 3D reconstruction. Pattern Anal. Mach. Intell. (PAMI) **39**(9), 1730–1743 (2017)

13. Johnson, J., et al.: Image retrieval using scene graphs. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3668–3678. IEEE (2015)
14. Kang, K., Ouyang, W., Li, H., Wang, X.: Object detection from video tubelets with convolutional neural networks. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1744–1756. IEEE (2016)
15. Krishna, R., et al.: Visual genome: connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vis. (IJCV) **123**(1), 32–73 (2017)
16. Li, Y., Ouyang, W., Zhou, B., Wang, K., Wang, X.: Scene graph generation from objects, phrases and region captions. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1261–1270. IEEE (2017)
17. Liang, X., Shen, X., Feng, J., Lin, L., Yan, S.: Semantic object parsing with graph LSTM. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 125–143. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_8
18. Liao, W., Shuai, L., Rosenhahn, B., Yang, M.Y.: Natural language guided visual relationship detection. arXiv preprint arXiv:1711.06032 (2017)
19. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 852–869. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_51
20. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Rob. **31**(5), 1147–1163 (2015)
21. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: high confidence predictions for unrecognizable images. In: Computer Vision and Pattern Recognition (CVPR), pp. 1544–1556. IEEE (2015)
22. Peyre, J., Laptev, I., Schmid, C., Sivic, J.: Weakly-supervised learning of visual relations. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 5189–5198 (2017). https://doi.org/10.1109/ICCV.2017.554. ISSN 2380-7504
23. Reddy, N.D., Singhal, P., Chari, V., Krishna, K.M.: Dynamic body VSLAM with semantic constraints. In: International Conference on Intelligent Robots (ICIR), pp. 1897–1904. IEEE (2015)
24. Rubino, C., Crocco, M., Del Bue, A.: 3D object localisation from multi-view image detections. Pattern Anal. Mach. Intell. (PAMI) **40**(6), 1281–1294 (2018)
25. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Neural Information Processing Systems NIPS, pp. 3856–3866 (2017)
26. Sengupta, S., Greveson, E., Shahrokni, A., Torr, P.H.S.: Urban 3D semantic modelling using stereo vision. In: International Conference on Robotics and Automation, pp. 580–585. IEEE (2013)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
28. Sung, M., Kim, V.G., Angst, R., Guibas, L.: Data-driven structural priors for shape completion. ACM Trans. Graph. (TOG) **34**(6), 175 (2015)
29. Tripathi, S., Lipton, Z.C., Belongie, S.J., Nguyen, T.Q.: Context matters: refining object detection in video with recurrent neural networks. In: British Machine Vision Conference (BMVC), pp. 1723–1731. BMVA (2016)
30. Tulsiani, S., Gupta, S., Fouhey, D., Efros, A.A., Malik, J.: Factoring shape, pose, and layout from the 2D image of a 3D scene. In: Computer Vision and Pattern Regognition (CVPR). IEEE (2018)
31. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3097–3106. IEEE (2017)

32. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: scene graph parsing with global context. In: Conference on Computer Vision and Pattern Recognition CVPR, pp. 3294–3304. IEEE (2018)
33. Zhang, H., Kyaw, Z., Chang, S.F., Chua, T.S.: Visual translation embedding network for visual relation detection. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4–12. IEEE (2017)
34. Zheng, S., et al.: Conditional random fields as recurrent neural networks. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1529–1537 (2015). https://doi.org/10.1109/ICCV.2015.179. ISSN 2380-7504