Global Pose Estimation Using Non-Tree Models

Hao Jiang and David R. Martin
Computer Science Department, Boston College
Chestnut Hill, MA 02467, USA
hjiang@cs.bc.edu, dmartin@cs.bc.edu

Abstract

We propose a novel global pose estimation method to detect body parts of articulated objects in images based on non-tree graph models. There are two kinds of edges defined in the body part relation graph: Strong (tree) edges corresponding to the body plan that can enforce any type of constraint, and weak (non-tree) edges that express exclusion constraints arising from inter-part occlusion and symmetry conditions. We express optimal part localization as a multiple shortest path problem in a set of correlated trellises constructed from the graph model. Strong model edges generate the trellises, while weak model edges prohibit implausible poses by generating exclusion constraints among trellis nodes and edges. The optimization may be expressed as an integer linear program and solved using a novel two-stage relaxation scheme. Experiments show that the proposed method has a high chance of obtaining the globally optimal pose at low computational cost.

1. Introduction

Detecting poses of articulated objects is critical for applications such as gesture recognition, articulated object tracking, surveillance, and human computer interaction. Broadly speaking, there have been two approaches to this problem: One may either detect full body poses directly from full body templates, or one may first detect body parts and then assemble the parts into poses. For a small to medium number of postures, a full body matching based on Chamfer matching or shape context matching [1] can be used. For clean segmented body pose images, classification methods [2] can be used to map the shape of the body pose into body part configurations. For articulated objects, however, whole body matching requires a very large number of templates – limiting both the practicality of the approach, and the generalizability of the approach to new poses and new object classes.

In contrast, methods based on detecting parts and then optimizing the body assembly use many fewer templates and promise better generality, but face a challenging optimization problem. Due to the difficulty of detecting parts reliably in a bottom-up manner, the usually large number of body part candidates prohibits an exhaustive search of the whole body configuration space in real applications.

Part assembly methods therefore tend to focus on mitigating the hard optimization problem, either by simplifying the model or by approximating the optimization. For a simplified body part relation model using a tree structured graph, dynamic programming (DP) [5, 4, 6] can be used to globally optimize the body pose estimation. However, a tree relation graph cannot model interactions between body parts non-adjacent in the tree due to occlusion or symmetry conditions. Consequently, DP methods naturally detect both arms at the same location, and both legs also. Sequential application of DP after removing the detected arm and leg may detect the other arm and leg, but mistakes from the first stage cannot be undone.

One may instead use a non-tree part relation graph to enforce the correct configuration constraints between body parts. Unfortunately, for formulations based on a general relation graph one must resort to approximate optimization methods. One widely used approximation search scheme is based on belief propagation (BP). BP has been used to detect hand gestures [7] and body part configurations [3, 13]. Though guaranteed to converge if the part relation graph is a tree, for a general graph, BP is not guaranteed to converge at all [12]. Sampling methods, such as MCMC, have also been applied to both upper body [9] and full body [8] pose estimation, and have been widely used in tracking articulated objects [10]. These methods have the advantage of finding multiple feasible solutions, but stochastic search usually converges much more slowly than deterministic schemes.

One would like a deterministic method of pose estimation that is not only efficient, but achieves the global optimum. Mathematical programming holds promise. Integer quadratic programming [11] has been proposed for optimizing body part configuration. The formulation permits arbitrary constraints between any pair of body parts, which results in a general relation graph. Direct optimization is too

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

expensive, but a linear relaxation can obtain a lower bound on the original NP-hard problem. The bound is not tight, however, and a greedy rounding step is still required.

In this paper, we follow the strategy of part detection and assembly for body pose estimation, and propose a different approach for optimizing the pose search. Instead of trying to solve an arbitrary topology part relation graph problem, we restrict the relations between body parts to a class of augmented tree graphs that contain two kinds of edges: Strong (tree) edges corresponding to the body plan that can enforce any type of constraint, and weak (non-tree) edges that are used for expressing *exclusion constraints* that arise from inter-part occlusion and symmetry conditions. This model is capable of adding many expressive constraints between body parts, and at the same time permits an efficient search. The optimization is formulated as a constrained multiple shortest path problem, which may be expressed as an integer linear program and solved deterministically using a novel two-stage relaxation method. Experiments confirm that the proposed method has a detection error rate nearly equal to the fully integer program solution, but at a fraction of the computational cost.

Even though many different approaches have been proposed for body pose estimation, there are few practical methods available for optimizing part assembly. The contribution of this paper is an expressive non-tree model combined with an efficient mathematical programming solution for the problem of localizing the parts of an articulated object. We augment the body part graph from a tree structure in order to add the constraints necessary for detecting a correctly articulated configuration, while inheriting computational efficiency from the model's tree-structured backbone. The proposed relaxation scheme is tighter than previous fully linear relaxation schemes, and yields an efficient solution. Our method does not need a pre-segmentation process, and because it is a global search it does not require initialization.

The outline of the paper is as follows. §2 presents the pose model, the transformation to a multiple shortest paths problem, and the proposed optimization method. §3 presents detection examples, as well as an empirical evaluation of the proposed model and optimization method. We conclude in §4.

2. Pose Model and Optimization

We model an articulated body configuration as an augmented tree relation graph on the body parts. The graph contains *strong edges*, which form a tree, and *weak edges*, which are the non-tree edges. We can enforce any constraint between parts connected by strong edges. At the same time, certain parts are further constrained using weak edges. The constraints enforced by weak edges are very general exclusion constraints that cause two part candidates to exclude

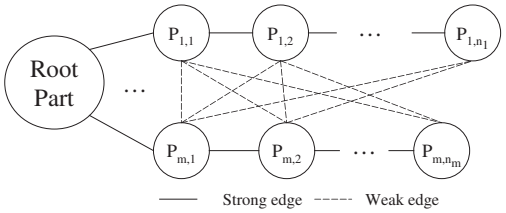

Figure 1. Generic articulated object graph model. Solid lines denote *strong edges*, which form a tree and encode arbitrary constraints. Dashed lines denote *weak edges*, which have no topological restrictions but can encode only exclusion constraints.

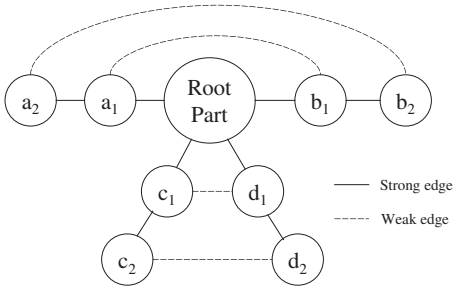

Figure 2. Human body graph model. Parts a and b form the arms; parts c and d the legs. The root part is the torso. Weak edges in this model may be used to enforce positional exclusion (i.e. occlusion), symmetry of position, symmetry of appearance, etc. Although the strong edges form a star graph in this model, the methods of this paper apply also when the strong edges form a general tree.

one another from a configuration; this mechanism is useful for implementing a variety of configural constraints. Optimizing the body part configuration embedded in the graph is formulated as an integer linear program and solved with a two-stage search scheme. Each of these elements of our model is elaborated below.

2.1. Part Detectors

The focus of this paper is optimizing body configuration, even when using a sub-optimal part detector that generates a very large number of false positives. We use simple binary part templates with Chamfer matching; low threshold Canny edges provide image boundary evidence. The torso template is extracted manually from one exemplar image; all other part templates consist simply of two parallel lines that denote the boundaries of the part on each side of its main symmetry axis. For each frame, we keep the top 10 torso candidates and the top 300 candidates for each other part. These part detectors are not selective; in this regime, which is typical, it is crucial to have a nearly globally optimal body part assembly method.

2.2. Part Model for Articulated Objects

The generic structure of our non-tree graph model is shown in Fig. 1. The root node of the graph denotes the root part of an articulated structure. The solid *strong edges* form

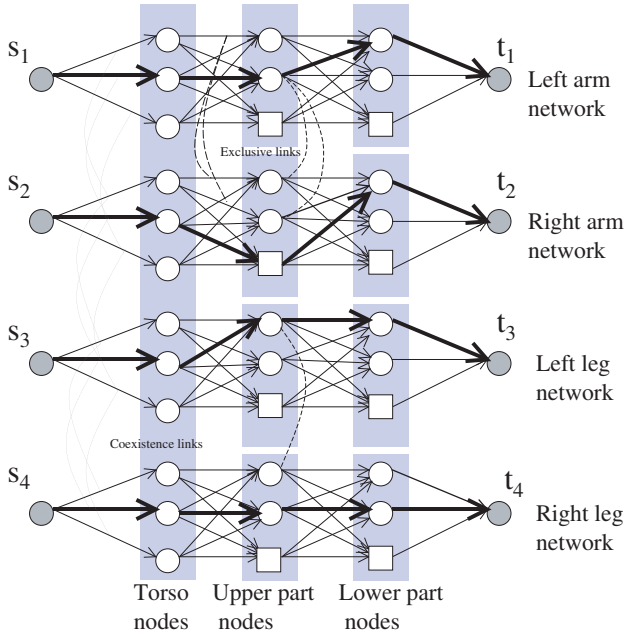

Figure 3. The multi-trellis graph structure generated from the body graph model in Fig. 2. Each linear structure in the graph generates a trellis. Each column in the trellis corresponds to a part; note that every trellis contains root part nodes. White round nodes correspond to part hypothesis generated by the part detectors; square nodes represent the occlusion state. The directed (solid) intra-trellis edges are generated by strong model edges; the undirected dashed inter-trellis arcs represent constraints introduced by weak model edges; the undirected dotted inter-trellis arcs set up the single root part constraint. Body configurations correspond to paths through the solid edges on trellises from the source (s) nodes to the sink (t) nodes; one such path is shown in bold. The constraint arcs ensure that the body configuration is admissible.

a tree, and may enforce any constraint between connected parts. The dashed *weak edges* can be used in our model to enforce expressive *exclusion constraints* between any two parts. The connection pattern of the weak edges is not constrained. Fig. 2 illustrates an example human body model, in which parts a and b are the arms and parts c and d the legs. The strong edges enforce the basic topology, while the weak edges permit occlusion reasoning and enforce symmetry of location and appearance. This augmented tree relation graph is simpler than a general graph model because the weak edges can only enforce certain constraints. The proposed model greatly simplifies the optimization problem while at the same time has enough expressive power to enforce a correct body configuration.

2.3. Multi-Trellis Transformation

The pose search problem is to find the optimal assignment of parts in the model graph to part candidates in the image. In this section, we transform the model graph into a multi-trellis structure based on how the optimization problem is formulated. Fig. 3 shows the multiple trellises for the human body part model illustrated in Fig. 2. We build a trellis for each linear structure emanating from the root part. The four trellises in Fig. 3 correspond to the arms and legs. Each column in a trellis corresponds to a single part in the original model. Since each linear structure starts at the root part, the root part appears in the first column of each trellis. The round nodes in the trellises correspond to the body part candidates provided by the part detectors. The square nodes correspond to the occlusion state for the part in that column. For each trellis, we also insert source nodes and sink nodes. These nodes do not correspond to body part states, and are simply anchors for the shortest path computations.

Directed edges are introduced between part candidate nodes corresponding to body parts connected by strong edges in the model. These solid intra-trellis edges are weighted according to the cost of the corresponding pairwise part configuration (see §2.4). There are two types of inter-trellis constraint arcs. The dotted *coexistence* arcs between edges emanating from the source nodes simply enforce that the first edge in each path passes through the same root part candidate (so that the arms and legs are all connected to the same torso). The dashed inter-trellis *exclusion* arcs are generated from weak model edges; they translate the exclusion constraints between model parts into constraints among trellis nodes and arcs (see §2.5). These arcs operate by permitting at most one of the connected edges or nodes to be used in the paths through the trellises. Each possible body part configuration therefore corresponds to a directed path linking a sequence of root and body part nodes from source to sink in each trellis; a good configuration additionally complies with the inter-trellis arc and node constraints.

The exclusion constraint arcs are used for various purposes. Part candidates that overlap in space produce an arc between trellis nodes, so that only one may accept a body part. Part candidates that do not satisfy positional symmetry constraints (e.g. hip joint positions relative to the torso) produce an arc between trellis edges so that only appropriately symmetric configurations are admitted. Part candidates that differ significantly in color produce an arc between trellis nodes so that only configurations with symmetric appearance are admitted. Paths through the trellises that honor these exclusion arcs therefore correspond to feasible whole body configurations. Because the solid edges are weighted according to their local configural fitness, and the constraint arcs enforce non-local configural constraints, finding the shortest "exclusive" paths through the trellises corresponds to finding the optimal body configuration.

2.4. Intra-Trellis Edges (Body Plan)

Denote the node in trellis k corresponding to candidate m of part i as $v_{k,i,m}$. Let r denote the root part, so that nodes corresponding to root part candidates are denoted as $v_{k,r,m}$. The arc from node $v_{k,i,m}$ to node $v_{k,j,n}$ is denoted

as $(v_{k,i,m}, v_{k,j,n})$. Define the weight on arc $(v_{k,i,m}, v_{k,j,n})$ as:

$$w_{k,i,m,j,n} = \begin{cases} \alpha_1 + \alpha_2, \text{ if } v_{k,j,n} \text{ is sink or occlusion node} \\ c_{j,n} + \alpha_2, \text{ if } v_{k,i,m} \text{ is source or occlusion node} \\ c_{j,n} + \lambda d(\mathbf{p}_{i,m}, \mathbf{p}_{j,n}) \\ \quad + \mu(1 - \cos(\theta_{i,m} - \theta_{j,n})), \text{ otherwise} \end{cases}$$

where $c_{j,n}$ is the cost of assigning candidate n to part j; $d(\mathbf{p}_{i,m}, \mathbf{p}_{j,n})$ is the distance between the end point of part i's mth candidate and the starting point of part j's nth candidate; and $1 - \cos(\theta_{i,m} - \theta_{j,n})$ is the orientation difference between the two part candidates. λ and μ are constants. The lower the edge weight, the better the local configuration of the two body part candidates. A local configuration is good if the parts connect and line up. To reduce complexity, we also remove edges between nodes if the distances of the corresponding part candidates are above threshold.

Any body part except for the root part may also be occluded, corresponding to the first two cases in the equation. α_1 is a constant chosen to be between typical good and bad candidate assignment costs; α_2 is another constant to compensate for typical distance and angle costs.

2.5. Inter-Trellis Arcs (Coexistence and Exclusion)

Inter-trellis arcs express constraints on the multi-paths through the trellises. When an inter-trellis arc connects a pair of trellis edges or a pair of trellis nodes, the paths must pass through both (coexistence) or at most one (exclusion) of the pair. This mechanism is very flexible, and can be used to enforce a variety of configural rules.

2.5.1 Common root candidate constraint

The first constraint is that each linear body part structure must use the same root body part candidate. Therefore, if we select a candidate m for the root part r in trellis k (so that we are using arc $(s_k, v_{k,r,m})$, then we must use that same root candidate m in every trellis. In other words, arc $(s_k, v_{k,r,m})$ must share the same status with the corresponding arc $(s_{\bar{k}}, v_{\bar{k},r,m})$ in each other trellis ($\bar{k} \neq k$).

2.5.2 Spatial position exclusion constraint

This constraint prohibits different body parts from occupying the same spatial location. This is accomplished by having each part exclude all other parts that conflict with it spatially. Fig. 4 illustrates the manner in which we determine spatial conflict. If all of the distances d_1, d_2, and l are under threshold, candidate n of part j conflicts with candidate m of part i; they will then be excluded from appearing in the same whole body configuration. For node $v_{k,i,m}$, define its conflict set as $\mathcal{A}(v_{k,i,m})$. If part i selects candidate m (and so $v_{k,i,m}$ is on the shortest path in its trellis), no node in $\mathcal{A}(v_{k,i,m})$ can be on any other path.

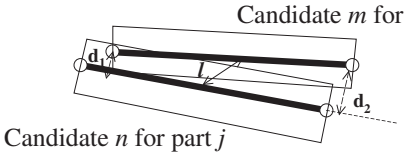

Figure 4. Two parts conflict if they they overlap sufficiently. d_1 and d_2 are the orthogonal distances between one parts' endpoints and the other part; l is the distance between the part centers. If all of these three distances are under threshold, the parts conflict. In the multi-trellis graph, this constraint corresponds to an exclusion between two nodes in different trellises.

2.5.3 Symmetry of appearance exclusion constraint

Just as parts may exclude one another on the basis of spatial position, they may also exclude one another on the basis of appearance. For example, to constrain the two upper arms to have similar appearance, we may add exclusion arcs between any two upper arm part candidates (in different trellises) that have sufficiently divergent appearance. Similarly of appearance is given by the SSD between parts in RGB when sub-sampled to 5×2 pixels. Symmetry of appearance is enforced between all symmetric body parts. The conflict set $\mathcal{A}$ for each trellis node is augmented accordingly.

2.5.4 Symmetry of position exclusion constraint

The trick of exclusion can also be used to express symmetry conditions between triples of body parts. For example, the two hip joints should be placed symmetrically about the lower end of the torso, and likewise for the shoulder joints. Figure 5 illustrates the situation. Consider candidate l for torso part r and candidate m for left upper arm part i. Let n and q be two candidates for right upper arm part j. In this example, candidate q would be excluded from coexisting with parts l and i because its starting point does not fall inside the gray circle denoting the target area for symmetric alignment with part i about part l. This is encoded by adding an exclusion arc between the two trellis edges $(v_{k,r,l}, v_{k,i,m})$ and $(v_{h,r,l}, v_{h,j,q})$, where k and h correspond to the two upper limbs. Denote the conflict edge set of edge $(v_{k,r,l}, v_{k,i,m})$ as $\mathcal{B}(v_{k,r,l}, v_{k,i,m})$.

2.6. Integer Program and Solution Method

Searching for the shortest paths that comply with the aforementioned coexistence and exclusion constraints is a hard combinatorial search problem which has as many as $N_r(N_1 + 1)(N_2 + 1)(N_3 + 1)...(N_m + 1)$ possible configurations, where N_r is the number of root parts and N_n, $n = 1..m$, is the number of candidates for part n. In this section, we formulate an efficient solution to this problem.

The multiple path optimization problem can be formulated as an integer linear program. We define binary variables $\xi_{k,i,m,j,n}$: If part i chooses candidate m and part

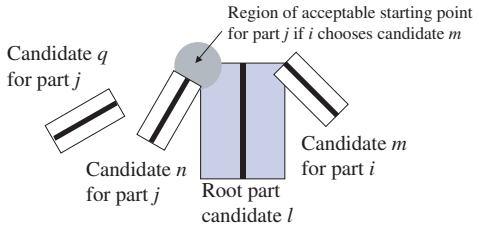

Figure 5. Two parts conflict if their starting points are not placed symmetrically with respect to a third part. For example, we may wish to place the shoulders symmetrically about the top of the torso. In this example, candidates m and q conflict when l is the root part candidate, but candidates m and n do not. In the multi-trellis graph, this constraint corresponds to an exclusion arc between two edges in different trellises.

j chooses candidate n in trellis k, then $\xi_{k,i,m,j,n}$ equals 1; otherwise it equals 0. In other words, $\xi_{k,i,m,j,n}$ indicates whether the arc $(v_{k,i,m}, v_{k,j,n})$ is used in a path. Let $\xi_{k,s,0,r,n}$ denote the arc between the source node of trellis k and the nth candidate of root part r. Let $w_{k,i,m,j,n}$ be the arc weight, as defined in §2.4. Let V be the set of trellis nodes not including the source and sink nodes, O the set of occlusion nodes, E the set of strong trellis edges, and L_i the candidate state label set for part i. The integer linear program can be written as:

$$\min \left\{ \sum_{(v_{k,i,m}, v_{k,j,n}) \in E} w_{k,i,m,j,n} \cdot \xi_{k,i,m,j,n} \right\}$$

subject to *common root node constraints* to ensure that each trellis uses the same single root candidate:

$$\xi_{0,s,0,r,n} = \xi_{k,s,0,r,n}, \ \forall (v_{k,s,0}, v_{k,r,n}) \in E.$$

subject to *spatial position exclusion constraints* to ensure that parts don't overlap and *symmetry of appearance exclusion constraints* to enforce consistent appearance:

$$\kappa_{k,j,n} + \kappa_{h,i,m} \leq 1, \ \forall v_{k,j,n} \in V \backslash O, \forall v_{h,i,m} \in \mathcal{A}(v_{k,j,n})$$
$$\text{where } \kappa_{k,j,n} = \sum_{m \in L_i} \xi_{k,i,m,j,n}$$

(binary variable $\kappa_{k,j,n}$ indicates whether part j uses candidate n); subject to *symmetry of position exclusion constraints* to ensure that configurations are biomechanically plausible:

$$\xi_{k,r,m,i,n} + \xi_{h,r,m,j,q} \leq 1,$$
$$\forall (v_{k,r,m}, v_{k,i,n}) \in E \text{ and } v_{k,i,n} \notin O$$
$$\forall (v_{h,r,m}, v_{h,j,q}) \in \mathcal{B}(v_{k,r,m}, v_{k,i,n})$$

and subject to *flow constraints* to ensure there is a single unit flow path through each trellis:

$$\sum_{m \in L_r} \xi_{0,s,0,r,m} = 1.$$
$$\sum_{m \in L_i} \xi_{k,i,m,j,n} = \sum_{q \in L_p} \xi_{k,j,n,p,q}, \ \forall v_{k,j,n} \in V$$

Directly solving the integer linear program is not efficient, so we propose a 2-stage relaxation scheme. At the first stage, we solve a mixed integer linear program where ξ variables linking source nodes to root parts remain integer, and all other variables are relaxed to non-negative real values. We adopt a branch and bound approach to solve this first stage program. The solution to the program with all variables relaxed yields an initial root part candidate that corresponds to the largest value source-root edge variable. We then solve two linear programs corresponding to that variable set to 0 and 1. If the 0-program solution is worse than the 1-program solution, then we are sure that the optimum must be the solution of the 1-program. Otherwise, we set the variable to 0 in a new linear program to find the next root part candidate, and solve two more linear programs for the two possible values of that variable. This is repeated until one 1-branch has the lowest optimal value. In practice, the optimal solution is found in just a few iterations.

The first stage solution does not guarantee integer values for non-root edges; the integer program is still too large to solve directly, and rounding does not work well. However, the first stage optimization does set many variables to zero: If we discard all such variables, the residual integer program is small enough to be solved directly using a similar branch and bound scheme to that described above.

3. Results

Fig. 6 illustrates the proposed method.[1] Panels (b)-(f) show the body part detection results. Using the proposed solution method, the linear program has 23,148 variables and 7,393 constraints and converges in less than 2 seconds. The first relaxation solution is shown in panel (g), and the final solution is shown in panel (h). In this example, the 2-stage relaxation finds the global optimum. Panel (i) shows the result of using dynamic programming (DP) to optimize a tree model consisting of just the strong edges in our graph model. Since DP does not permit coupling between body parts not adjacent in the tree, it always detects both arms and both legs at the same locations; in this example, it also detects the legs incorrectly. With spatial exclusion constraints, the proposed method prefers to find two unoccluded legs when the image evidence supports that hypothesis. Sequential DP is the typical manner in which to "fix" DP so that it may find both arms and both legs; however, when the first stage of DP makes a mistake as in this example, sequential

[1] Please note that the figures in this paper are best viewed in color.

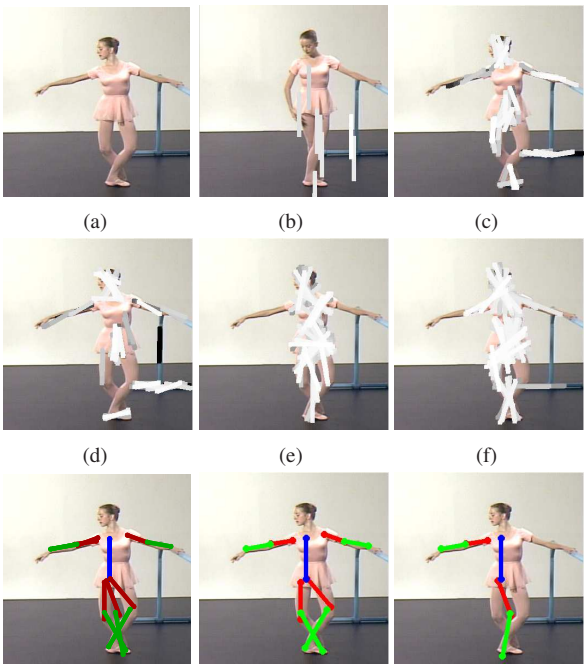

Figure 6. Body part detection example and comparison with dynamic programming (DP). Panel (a) shows the input image. Panels (b-f) show detections of (b) torso, (c) upper arm, (d) lower arm, (e) upper leg, and (f) lower leg. Panel (g) shows the configuration after the first relaxation, and (h) shows the final result of the proposed method. Panel (i) shows the DP result, which does not include any exclusion constraints; see §3 for discussion. In all panels, the rectangular parts are depicted using their main symmetry axis; line intensity indicates the part match score (b-f) or edge variable value (g), where darker denotes smaller.

DP cannot improve the result. Fig. 7 shows another comparison with DP where the proposed method models occlusion more effectively, and where sequential DP would not help.

3.1. Evaluation and Discussion

We evaluate the proposed method on three challenging image sequences: *Ballet*, *Man*, and *Fitness*. Frames from these sequences are depicted in Figs. 10, 11, and 12. All three sequences suffer from low contrast, complex body part motions, and frequent occlusion between body parts. The *Man* sequence challenges the part model with baggy clothing; the *Fitness* sequence is the most difficult, adding interlacing effects and increased background clutter.

The bars labeled *Proposed* in Fig. 8 show the error rates for pose detection in these sequences for the proposed algorithm. The error for a frame is simply the number of missed or incorrect visible parts; we do not require that occluded parts be detected. The bar segments in the figure show the proportion of frames with n errors, so the first (zero) segment gives the overall success rate.

Although the error rates for the proposed algorithm are

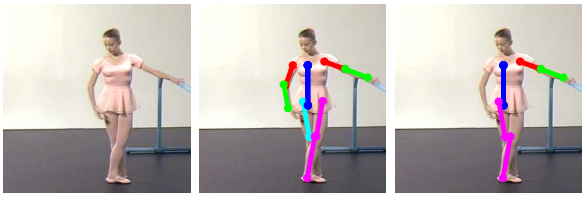

Figure 7. A second example and comparison with DP. Panel (a) shows the input image, panel (b) the result of the proposed method, and panel (c) the DP result. The proposed method correctly identifies a complete left leg and an occluded lower right leg, while DP greedily constructs an incorrect leg.

low, one would like to know whether a significant number of errors is introduced by the proposed 2-stage relaxation method, or if the errors are intrinsic to limitations of the model or formulation. The proposed algorithm is an approximation to an integer program. The bars labeled *Int. Prog.* in Fig. 8 show the error rates for the full integer program using the same model. We see that the increase in error rate due to the relaxation is minimal. In addition, if we compare the solutions found by the two methods numerically, we find that the proposed method achieves the global optimum in 92% of the frames in the *Ballet* and *Man* sequences, and in 56% of the frames in the *Fitness* sequence.

Although the proposed two-stage relaxation method achieves error rates comparable to the full integer program, one might ask if a simpler relaxation method would work equally well. The bars labeled *Round* in Fig. 8 show the error rates for a simple rounding scheme. Again using the same parts and model as in the proposed method, in this case we solve the fully relaxed linear program, and simply select the maximum weight candidate for each part. The performance of this rounding algorithm is dismal, indicating that a good optimization method is, indeed, critical for these sorts of pose models.

The proposed two-stage relaxation method is able to achieve both low error and low computational cost. Figs. 9 and 10 show why this is the case. Fig. 9 shows the detection result after the first stage on the *Ballet* sequence. This first stage solves a somewhat relaxed mixed integer program that is used to nail down the root (torso) part in a location where plausible arms and legs may be found. In this step, only the torso receives a permanent label whereas other body parts receive labels with fractional confidence. Only non-zero confidence part candidates are drawn in the figure, showing that most of the part candidates (ξ variables) vanish in the first stage. The second stage integer program is therefore much reduced in size, which is the origin of the good computational performance of the proposed method. The final detection results for these same frames are shown in Fig. 10.

We also evaluate the performance gained by expanding

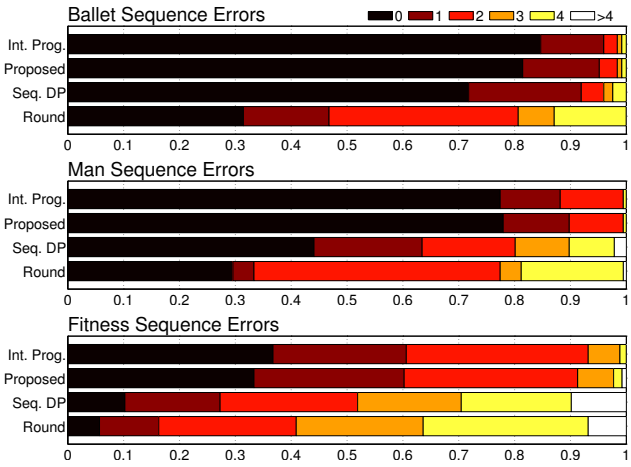

Figure 8. Detection error rates for three image sequences using four different optimization methods: integer programming (*Int. Prog.*), the proposed method of this paper (*Proposed*), sequential dynamic programming (*Seq. DP*), and direct rounding (*Round*). The bars show the fraction of frames with N errors; each missed or incorrect part is counted as one error. The proposed method achieves error rates comparable to the full integer program at about 10% the computational expense. In addition, the proposed method significantly outperforms sequential dynamic programming, which in turn outperforms direct rounding. The various exclusion constraints that the proposed method introduces into the pose model are the critical difference between it and sequential DP. Figs. 10, 11, and 12 show estimated poses for individual frames from these sequences. See §3.1 for discussion.

the model with non-tree edges, which enables us to encode additional non-local pose constraints. The bars labeled *Seq. DP* in Fig. 8 show the error rates for sequential dynamic programming. By simply removing all weak edges from our model, what remains is a tree model that is amenable to fast optimization via dynamic programming. Since a single application of DP will not detect more than one leg or arm, we remove parts that intersect the detected arms and legs, and run DP a second time. This is the sequential DP algorithm. The figure shows a precipitous drop in performance between the proposed model and sequential DP. The essential difference between the two models is that our model includes various exclusion and symmetry constraints between parts that the tree-based DP model cannot encode. These constraints not only permit occlusion reasoning, but also constrain the search to physically plausible poses.

4. Conclusion

We propose an efficient body pose estimation method based on a non-tree graph model and relaxed convex programming. This model can be used to explicitly express occlusion constraints and body part symmetry constraints that cannot be expressed in a tree-structured body part relation model. Experiments show that the proposed scheme can

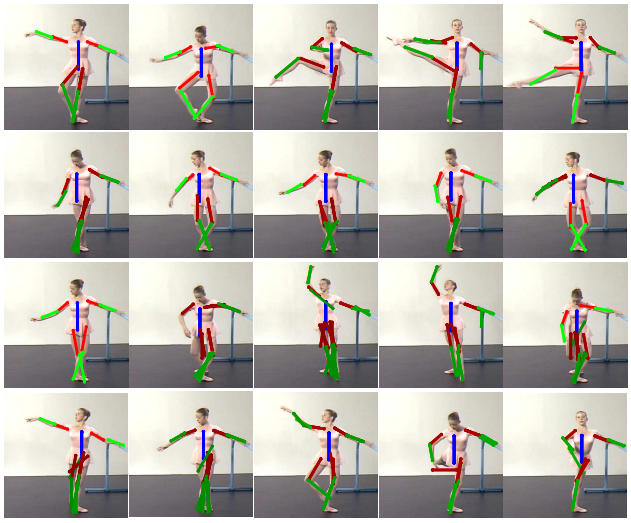

Figure 9. Estimated pose from selected frames of the *Ballet* sequence after the first relaxation stage of the proposed method.

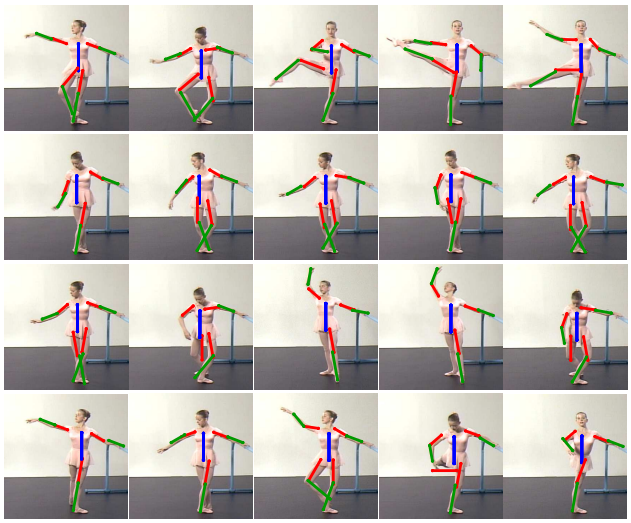

Figure 10. Estimated pose from selected frames of the 124 frame *Ballet* sequence using the proposed two-stage relaxation method.

efficiently detect the body parts of articulated objects with low error rates. Especially when clutter and occlusion are present, this new method significantly outperforms sequential dynamic programming at single-frame pose estimation. The proposed optimization scheme is generic, and could be used with more sophisticated part detectors to further improve pose estimation for articulated objects.

References

[1] G. Mori and J. Malik. "Estimating human body configurations using shape context matching", ECCV'02, 666-680.

[2] R. Rosales and S. Sclaroff. "Inferring body pose without tracking body parts", CVPR 2000, 2:721-727.

[3] L. Sigal and M.J. Black. "Measure locally, reason globally: occlusion-sensitive articulated pose estimation", CVPR

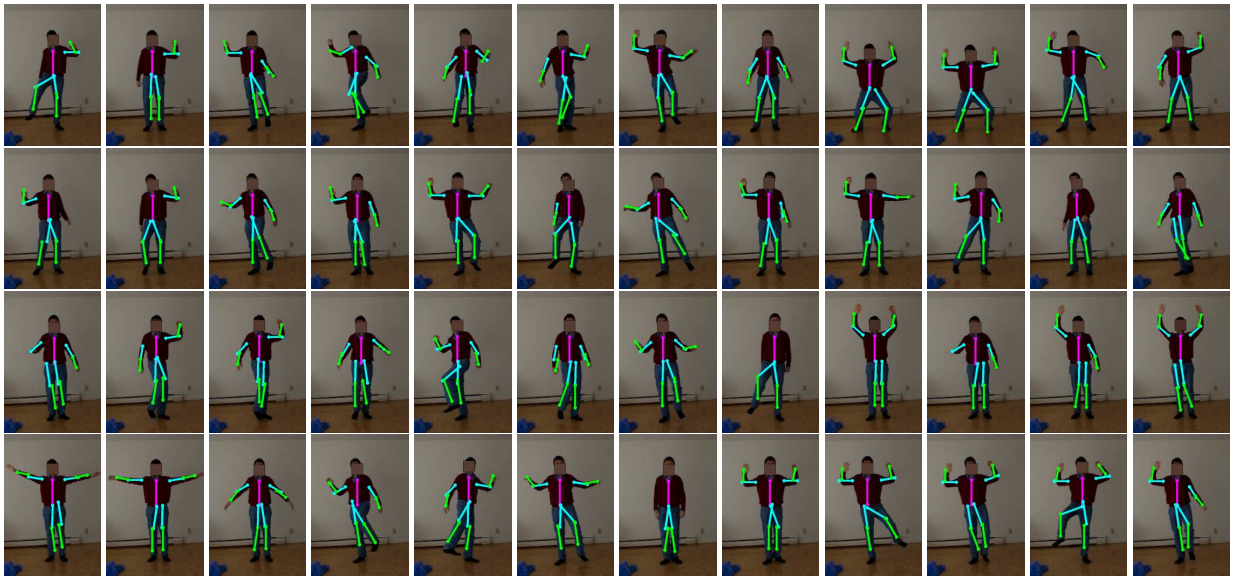

Figure 11. Estimated pose from selected frames of the 186 frame *Man* sequence using the proposed method.

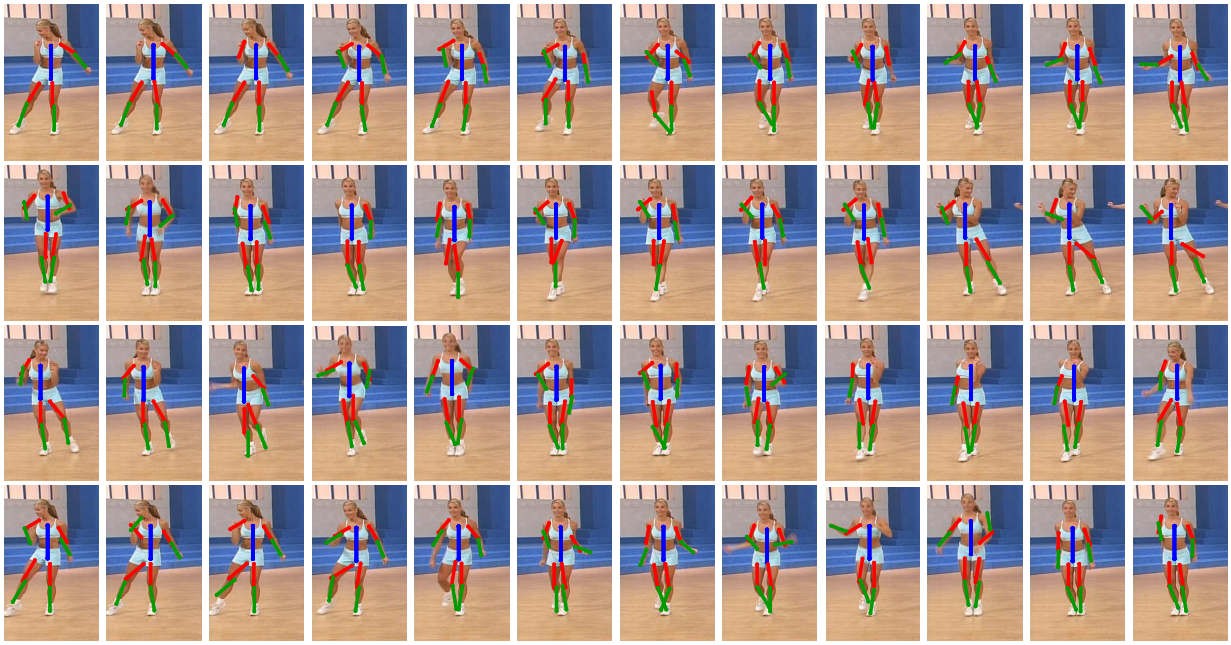

Figure 12. Estimated pose from selected frames of the 264 frame *Fitness* sequence using the proposed method.

2006.

[4] D. Ramanan, D.A. Forsyth, and A. Zisserman. "Strike a pose: tracking people by finding stylized poses", CVPR 2005.

[5] P.F. Felzenszwalb and D.P. Huttenlocher. "Pictorial structures for object recognition", IJCV 61(1), Jan. 2005.

[6] R. Ronfard, C. Schmid, and B. Triggs. "Learning to parse pictures of people", ECCV 2002, 4:700-714.

[7] E.Sudderth, M. Mandel, W.T. Freeman, and A. Willsky. "Distributed occlusion reasoning for tracking with nonparametric belief propagation", NIPS 2004.

[8] S. Ioffe and D.A. Forsyth. "Probabilistic methods for finding people", IJCV 43(1):45-68, June 2001.

[9] M.W. Lee and I. Cohen. "Human upper body pose estimation in static images", ECCV 2004.

[10] C. Sminchisescu and B. Triggs. "Kinematic jump processes for monocular 3D human tracking", CVPR 2003, 1:69-76.

[11] X.F. Ren, A.C. Berg and J. Malik. "Recovering human body configurations using pairwise constraints between parts", ICCV 2005, 1:824-831.

[12] Y. Weiss and W.T. Freeman. "On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs", IEEE Trans. Info. Theory, 47(2):723-735, 2001.

[13] A. Gupta, A. Mittal and L.S. Davis. "Constraint integration for efficient multiview pose estimation with self-occlusions", IEEE TPAMI, 30(3), 2008, pp. 493-506.