Particle Filter with Unordered Observations for Solving Image Jigsaw Puzzles

Anonymous CVPR submission

Paper ID ****

Abstract

We deal with an image jigsaw puzzle problem, which is defined as reconstructing an image from a set of square and non-overlapping image patches. It is known that this is a NP-complete problem, and it is also challenging for humans, since in the considered framework the original image is not given. Recently a graphical model has been proposed to solve this problem, where each patch location is a node and each image patch is a label. The target label probability function is then maximized using loopy belief propagation. In this paper we use the same target function and exactly the same pairwise potentials. Our main contribution is a novel inference approach in the sampling framework of Particle Filter (PF). Usually in the PF framework it is assumed that the observations arrive sequentially, e.g., the observations are naturally ordered by their time stamps in the tracking scenario. Based on this assumption, the posterior density over the corresponding hidden states is estimated. In the jigsaw puzzle problem all observations (puzzle pieces) are given at once without any particular order. In this paper we relax the assumption of having ordered observations and extend the PF framework to simultaneously estimate the posterior density while exploring different orders of observations. This significantly broadens the scope of applications of the PF inference. Our experimental results demonstrate that the proposed inference framework significantly outperforms loopy belief propagation in solving the image jigsaw puzzle problem. In particular, the extended PF inference triples the accuracy of the label assignment of the loopy belief propagation.

1. Introduction and Problem Formulation

As shown in [4] the jigsaw puzzle problem is NP-complete if the pairwise affinity among jigsaw pieces is unreliable. Following [2], we focus on reconstructing the original image from square and non-overlapping patches. This type of puzzles does not contain the shape information of individual pieces, which is quite important to determine the pairwise affinities among them. This makes the problem more challenging, since it is more difficult to evaluate pairwise affinities among puzzles. Thus, only the image content information of the puzzle pieces is available. This is different from most of the previous approaches [10, 7, 13, 17], where the shape of the puzzle pieces is utilized.

Now we briefly review the PF inference. We begin with a classical tracking example. A robot is moving around and taking images at discrete time intervals. The images form a sequence of observations $Z = (z_1, \ldots, z_m)$, where z_t is an image taken at time t. With each observation z_t there is associated a hidden state x_t. In our example the value of x_t is the robot pose (its 2D position plus orientation). The goal of PF inferences, is to derive the most likely sequence of the hidden states, i.e., to find a state vector $x_{1:m} = (x_1, \ldots, x_m)$ that maximizes the posterior $p(x_{1:m}|Z)$. We observe that here the observations are ordered following their time stamps. In PF inference, this order is utilized to sequentially infer the values of states x_t for $t = 1, \ldots, m$. Now imagine that the robot's clock broke and the time stamps are random. Thus, we are given a set of observations $Z = \{z_1, \ldots, z_m\}$, they are indexed but their index is irrelevant. Of course, we can still associate state x_t with observation z_t, but the set of observations is not ordered, and consequently, there state vectors x_t are not ordered. Thus, we deal with unordered observations. This is exactly the scenario of the image jigsaw puzzle problem, see Fig. 1. We are given m square puzzle pieces described by a set of m observations $Z = \{z_1, \ldots, z_m\}$. Each observation z_t describes part of the original image on piece t and is given by a vector of features, which are the color values of the pixels on piece t in our experimental results. The puzzle pieces are numbered with index t, but their numbering is random, e.g., the number in Fig. 1(b). The value of the state x_t of puzzle piece t is a location of an empty square in the square grid, e.g., the value of x_t is the index of an empty square in the square gird shown in Fig. 1(c). Our goal is to determine the state vector $x_{1:m}$ that maximizes the posterior probability $p(x_{1:m}|Z)$. Since the original image is not provided, this probability is determined based on pairwise appearance consistency of the local puzzle images, i.e., the posterior distribution is a function of

Figure 1. The goal is to build the original image (a) given the jigsaw puzzle pieces (b). The original image is not known, thus, it needs to be estimated given the observations shown in (b). The empty squares in (c) form possible locations for the puzzle pieces in (b).

how well adjacent pieces fit together once they are placed on the grid. In other words, a vector of grid locations $x_{1:m}$ maximizes $p(x_{1:m}|Z)$ if the puzzle pieces placed at these locations form the most consistent image. We observe that the posterior distribution $p(x_{1:m}|Z)$ usually is very complicated and has many local maxima. This is particulary the case when the local image information of the puzzle pieces is not very descriptive.

Our main contribution is a new PF inference framework that works in this scenario. In the proposed framework we extend PF to handle the situations where we have unordered set of observations that are given simultaneously. One of our key ideas is the fact that it is possible to extend the importance sampling from the proposal distribution so that the particle resampling automatically determines most informative orders of observations. This allows us to simultaneously consider different orders of observation in the process of estimating the posterior distribution. If our goal is maximizing the posterior distribution, we can view the order of observations that led to the global maximum as a most informative. Of course, this order does not have to be unique, since it is possible that other orders of observations lead to the same global maximum.

In our experimental results, we compare the solutions of (**??**) obtained by the proposed inference framework to the solutions of the loopy believe propagation used in [2] under identical settings on the dataset provided in [2]. In particular, we use exactly the same dissimilarity-based compatibility D as defined above. The proposed PF inference significantly outperforms the loopy believe propagation in all evaluation measures. The main of these measures is the accuracy of the label assignment, where the difference is most significant. The accuracy of the loopy believe propagation is 24% while the accuracy of the proposed PF inference is 69%.

The classical PF framework has been developed for sequential state estimation like tracking [9, 14] or robot localization [15, 5]. There, the observations arrive sequentially and are indexed by their time stamps, as our tracking example illustrates. It is possible to apply the classical PF framework as stochastic optimization to solve this problem by utilizing a fix order of observations. However, by doing so, we would have selected an arbitrary order, and the puzzle construction may fail because of the selected order. If we would have selected a different order, the labeling task could have been successful. Moreover, the observations are given simultaneously at the same time. Consequently, there is no reason to favor any particular order without utilizing this fact.

The rest of the paper is organized as follows. Section 7 describes several closely related approaches. The details of the proposed framework are explained in Section **??**. Section 6 shows and evaluates the experimental results not only the dataset from [2], but also an extended dataset.

2. Basic Facts about Particle Filters

In this section we summarize basic facts about Particle Filters (PFs). They will be utilized in the following section when we introduce the proposed framework. Given is a sequence of observations $Z = \{z_1, \ldots, z_m\}$, i.e., the observations are ordered. Our goal is to maximize the posterior distribution $p(x_{1:m} \mid Z)$, i.e., to find the values $\widehat{x}_t$ of states x_t such that

$$\widehat{x}_{1:m} = \underset{x_{1:m}}{\operatorname{argmax}}\, p(x_{1:m} \mid Z), \tag{1}$$

where $x_{1:m} = (x_1, \ldots, x_m) \in \mathcal{X}^m$ is a state space vector. We observe that each state x_t corresponds to observation z_t for $t = 1, \ldots, m$.

This goal can be achieved by approximating the posterior distribution with a finite number of samples in the framework of Bayesian Importance Sampling (BIS). Since it is usually difficult to draw samples from the pdf $p(x_{1:m}|Z)$, we will draw samples $x^{(i)}_{1:m} \sim q(x_{1:m}|Z)$ for $i = 1, \ldots, N$ from a proposal pdf q, form which samples are easily generated. Then approximation to the density p is given by

$$p(x_{1:m}|Z) \approx \sum_{i=1}^{N} w^{(i)} \delta(x_{1:m} - x^{(i)}_{1:m}), \tag{2}$$

where δ is the Dirac delta function and

$$w^{(i)} = \frac{p(x_{1:m}^{(i)}|Z)}{q(x_{1:m}^{(i)}|Z)} \tag{3}$$

are normalized weights (so that they sum to one). The state vector $x_{1:m}^{(i)}$ with the largest weight $w^{(i)}$ is then taken as the solution of (1).

Since it is still computationally intractable to draw samples from q due to high dimensionality of $x_{1:m}$, Sequential Importance Sampling (SIS) is usually utilized. In the classical PF approaches, samples are generated recursively following the order of dimensions in state vector $x_{1:m} = (x_1, \ldots, x_m)$:

$$x_t^{(i)} \sim q_t(x|x_{1:t-1}, Z) = q_t(x|x_{1:t-1}, z_{1:t}) \tag{4}$$

for $t = 1, \ldots m$, and the particles are built sequentially $x_{1:t}^{(i)} = (x_{1:t-1}^{(i)}, x_t^{(i)})$ for $i = 1, \ldots, N$. The subscript t in q_t indicates from which dimension of the state vector the samples are generated. We use this notation to stress that q_t is a probability density function (pdf) from which we sample. (For $t = 1$, $x_{1:t-1} = x_{1:0}$ denotes an empty vector, thus, we sample $x_1^{(i)} \sim q_1(x|Z)$.) Since q factorizes as

$$q(x_{1:m}|Z) = q_1(x_1|Z) \prod_{t=2}^{m} q_t(x_t|x_{1:t-1}, Z), \tag{5}$$

we obtain that $x_{1:m}^{(i)} \sim q(x_{1:m}|Z)$. In over words, by sampling recursively $x_t^{(i)}$ from each dimension t according to (4) and appending the samples to a vector $x_{1:m}^{(i)}$ we obtain a sample from $q(x_{1:m}|Z)$

Since at a given iteration we have a partial sample vector $x_{1:t}^{(i)}$ for $t < m$, we also need an evaluation procedure of this partial sample vector. For this we observe that the weights can be recursively updated according to [16]:

$$w(x_{1:t}^{(i)}) = \frac{p(z_t|x_{1:t}^{(i)}, z_{1:t-1}) p(x_t^{(i)}|x_{1:t-1}^{(i)})}{q_t(x_t^{(i)}|x_{1:t-1}^{(i)}, z_{1:t})} \, w(x_{1:t-1}^{(i)}). \tag{6}$$

The key observation is that when $t = m$, the weight $w(x_{1:m}^{(i)})$ of particle (i) is equal to $w^{(i)}$ (defined in (3)). Consequently, we have arrived at the SIS theorem:

Theorem 1. By sampling particles according to (4) and weighting them according (6), we obtain a set of weighted samples form $p(x_{1:m}|Z)$ once $t = m$. Hence we can approximate $p(x_{1:m}|Z)$ with any precision if the number of particles N is sufficiently large. Thus, we can write

$$p(x_{1:m}|Z) \approx \sum_{i=1}^{N} w(x_{1:t}^{(i)}) \delta(x_{1:m} - x_{1:m}^{(i)}). \tag{7}$$

Weight equation (6) can be simplified by making a common assumption that $q_t(x_t^{(i)}|x_{1:t-1}^{(i)}, z_{1:t}) = p(x_t^{(i)}|x_{1:t-1}^{(i)})$, i.e., we take as the proposal distribution the conditional pdf of the state at time t conditioned on the current state vector $x_{1:t-1}^{(i)}$. This assumption simplifies the recursive weight update formula to

$$w(x_{1:t}^{(i)}) = w(x_{1:t-1}^{(i)}) p(z_t|x_{1:t}^{(i)}, z_{1:t-1}), \tag{8}$$

and implies that the samples are generated from

$$x_t^{(i)} \sim p_t(x|x_{1:t-1}^{(i)}). \tag{9}$$

Analog to (4) we added the index t to p in (9) to indicate the dimension of the state space from which the samples are generated.

Now we summarize the derived **standard PF algorithm**. For every time step $t = 1, \ldots, m$ and for every particle $i = 1, \ldots, N$ execute the following three steps:
1) **Importance sampling / proposal:** Sample followers of particle (i) according to (9) (a special case of (4)) and set $x_{1:t}^{(i)} = (x_{1:t-1}^{(i)}, x_t^{(i)})$.
2) **Importance weighting / evaluation:** An importance weight is assigned to each particle $x_{1:t}^{(i)}$ according to (8) (a special case of (6)).
3) **Resampling:** Sample with replacement N new particles form the current set of N particles

$$\{x_{1:t}^{(i)} | i = 1, \ldots, N\}$$

according to their weights. We obtain a set of new particles $x_{1:t}^{(i)}$ for $i = 1, \ldots, N$, and renormalize their weights to sum to one. This procedure is a variant of Sampling Importance Resampling (SIR) [16]. It is an important part of any PF algorithm, since resampling prevents weight degeneration of particles.

3. Unordered Observations

In the proposed approach, the order of the observations is not predetermined, in particular, we do not follow the order of indices of the observations in Z. Our key idea is to extend the PF framework to examine all possible orders of observations and to follow the most informative orders. This way we are able to utilize the the most informative observations first. Intuitively, it makes sense, for example, if the first puzzle piece has a local image very similar to many other puzzle pieces and the second puzzle pieces has a very distinctive local image that matches only a few other pieces, then our approach will first process the second puzzle piece, since it is more informative.

We stress that the standard SIS in Eq. 9 and particle evaluation in Eq. 8 utilize the sequential order of the states $x_{1:m} = (x_1, \ldots, x_m)$. In many applications, this order is

determined naturally by the time stamp of the observations, e.g., a single robot is collecting laser measurements at consecutive time points, in which case x_t denotes the robot pose at time t. The goal of our work is to extend SIS to applications in which there is no natural order of observations.

The key idea of the proposed approach is not to utilize the fix order of the states, but instead explore the most likely orders of the states $(x_{i_1}, \ldots, x_{i_m})$ so that the corresponding sequence of observations $Z = (z_{i_1}, \ldots, z_{i_m})$ is most informative. To achieve this we modify the first step of the PF algorithm so that the importance sampling is performed for every dimension not yet represented by the current particle.

In order to formally define the proposed sampling rule, w need to explicitly represent different orders of states with a permutation $\sigma : \{1, \ldots, m\} \rightarrow \{1, \ldots, m\}$ and use the shorthand notation $\sigma(1:t)$ to denote $(\sigma(1), \sigma(2), \ldots, \sigma(t))$ for $t \leq m$. For each particle (i) we may have a different permeation $\sigma^{(i)}$, but we will drop the index (i) from $\sigma^{(i)}$ in expressions like $x^{(i)}_{\sigma(1:t)}$, since the state vector already carries the particle index (i). With reference to our toy example, Fig. 1(c) shows the configuration at time $t = 2$, where puzzle pieces numbered 3 and 1 in Fig. 1(b) are placed at locations 1 and 2, correspondingly. Hence $\sigma^{(i)}(1:2) = (3, 1)$ and $x^{(i)}_{\sigma(1:2)} = (1, 2)$. We stress that now the sequence of states $x_{\sigma(1:t-1)}$ visited before time t is not a sequence of consecutive numbers $(1, \ldots, t-1)$ but any subsequence $(i_1, \ldots, i_{t-1})$ formed by $t-1$ different numbers in $\{1, \ldots, m\}$.

We are ready now to precise formulate the proposed importance sampling. At each iteration $t \leq m$, for each particle (i) and for each $s \in \overline{\sigma^{(i)}(1:t-1)}$, we sample

$$x^{(i)}_s \sim p_s(x | x^{(i)}_{\sigma(1:t-1)}), \tag{10}$$

where $\overline{\sigma^{(i)}(1:t-1)} = (1, \ldots, m) \setminus \sigma^{(i)}(1:t-1)$, i.e., the indices in $1:m$ that are not present in $\sigma^{(i)}(1:t-1)$ for $t \leq m$, and the subscript s at the posterior pdf p_s indicates that we sample from values for state s. We generate at least one sample for each state $s \in \overline{\sigma^{(i)}(1:t-1)}$. This means that the single particle $x^{(i)}_{\sigma(1:t-1)}$ is multiplied and extended to several follower particles $x^{(i)}_{\sigma(1:t-1),s}$. Consequently, at iteration $t < m$ particle (i) has $m - t + 1$ followers. Each follower is a sample from a different dimension of the state vector (i.e., represents a location of a different puzzle piece). Going back to our toy puzzle example, we recall that the current state vector of particle (i) in Fig. 1(c) at time $t = 2$ is $x^{(i)}_{\sigma(1:2)} = (1, 2)$, where $\sigma^{(i)}(1:2) = (3, 1)$. For sampling at time $t = 3$, we have $\overline{\sigma^{(i)}(1:t-1)} = (2, 4, 5, 6)$. Consequently, we sample four followers of particle (i) in (10), one for each state $s = 2, 4, 5, 6$, where $x^{(i)}_2$ is the sampled location of puzzle piece 2, $x^{(i)}_4$ is the sampled location of puzzle piece 4, and so on.

In contrast, in the standard application of rule (9), at each iteration t particle (i) has one follower. Even when sometimes each particle (i) has many followers, all followers are samples form the same state, since there is a unique state and only one state at time t, which means that we only determine possible locations of say puzzle piece 2 and do not consider locations of puzzle piece 4 for particle (i), since a strict order of the state dimensions is followed in the classical setting.

We do not make any Markov assumption in (10), i.e., the new state $x^{(i)}_s$ is dependent on all previous states $x^{(i)}_{\sigma(1:t-1)}$ for each particle (i).

4. Particle Filter for Unordered Observations

We outline in this section the proposed **PF for unordered observations (PFUO)** algorithm. The key change is in the importance sampling step in comparison to the standard PF algorithm presented in Section 2. The other two steps are adjusted to accommodate this key change. For every time step $t = 1, \ldots, m$ and for every particle $i = 1, \ldots, N$ execute the following three steps:

1) **Importance sampling / proposal:** Sample followers $x^{(i)}_s$ of particle (i) from each dimension $s \in \overline{\sigma^{(i)}(1:t-1)}$ according to (10), which we repeat here for completeness,

$$x^{(i)}_s \sim p_s(x | x^{(i)}_{\sigma(1:t-1)}), \tag{11}$$

and set $x^{(i,s)}_{\sigma(1:t)} = (x^{(i)}_{\sigma(1:t-1)}, x^{(i)}_s)$ and $\sigma^{(i,s)}(t) = s$, which means that $\sigma^{(i,s)}(1:t) = (\sigma(1:t-1), s)$. As stated before, we drop the superscript (i,s) in $x^{(i,s)}_{\sigma(1:t)}$, since it is already present as the particle index.

2) **Importance weighting/evaluation:** An individual importance weight is assigned to each follower particle $x^{(i,s)}_{\sigma(1:t)}$ according to

$$w(x^{(i,s)}_{\sigma(1:t)}) = w(x^{(i)}_{\sigma(1:t-1)}) p(z_s | x^{(i,s)}_{\sigma(1:t)}, z_{\sigma^{(i)}(1:t-1)}), \tag{12}$$

3) **Resampling:** Sample with replacement N new particles form the current set of $N \times (m - t + 1)$ particles

$$\{x^{(i,s)}_{\sigma(1:t)} \mid i = 1, \ldots, N,\ s \in \overline{\sigma^{(i)}(1:t-1)}\}. \tag{13}$$

according to the weights. We obtain a set of new particles $x^{(i)}_{\sigma(1:t)}$ for $i = 1, \ldots, N$, i.e., we assign new indices $i = 1, \ldots, N$ to the new particles. We also renormalize their weights to sum to one. This is a variant of the standard Sampling Importance Resampling (SIR) step [16] as in the classical PF framework, but the set of particles that is resampled is different.

We observe that the particle weight evaluation in (12) is analog to (8) in that the conditional probability of observation z_s is a function of two corresponding sequences of observations and states plus the state x_s. The only difference is that the sequences may not be in the original index order but in the order determined by the permutation $\sigma^{(i)}$ applied to $1:t-1$.

When $t = m$ all state dimensions are present in the state vector $x^{(i)}_{\sigma(1:m)}$ for each particle (i). Hence we can reorder the sequence of state dimensions $\sigma^{(i)}(1:m)$ to form the original order $1:m$, i.e., we apply the inverse of $\sigma^{(i)}$ to $\sigma^{(i)}(1:m)$. We denote the so obtained state vector with $x^{(i)}_{1:m}$, it is composed form the same state values as $x^{(i)}_{\sigma(1:m)}$ but sorted according to the original state indices $1:m$. We set $w(x^{(i)}_{1:m}) = w(x^{(i)}_{\sigma(1:m)})$, i.e., the particle weight remains invariant to the sorting. The following theorem guarantees that PFUO computes an approximation to (2).

Theorem 2. For $t = m$ the particles $x^{(i)}_{1:m}$ with $i = 1,\ldots,N$ provide an approximation to the posterior distribution $p(x_{1:m}|Z)$ for sufficiently large N, i.e.,

$$p(x_{1:m}|Z) \approx \sum_{i=1}^{N} w(x^{(i)}_{1:m})\delta(x_{1:m} - x^{(i)}_{1:m}). \qquad (14)$$

Proof of Theorem 2. Let us fix a single particle index (i). We have that

$$w(x^{(i)}_{1:m}) = w(x^{(i)}_{\sigma(1:m)}) = w^{(i)} = \frac{p(x^{(i)}_{\sigma(1:m)}|Z)}{q(x^{(i)}_{\sigma(1:m)}|Z)}. \qquad (15)$$

To see this, let us take the state dimension order $\sigma^{(i)}(1:m)$ as the initial order of state dimensions. It will then not change during the computation of particle (i), i.e., the recursive weight update formula simply follows the standard weight update of PF. Consequently, $x^{(i)}_{\sigma^{(i)}(1:m)}$ is a weighted sample from $p(x^{(i)}_{\sigma^{(i)}(1:m)}|Z)$.

Now $p(x^{(i)}_{\sigma^{(i)}(1:m)}|Z)$ is the joint distribution of m random variables, and as such the order of the random variables is not relevant. (This follows from the fact that a joint probability is defined as the probability of the intersection of the sets representing events corresponding to the value assignments of the random variables, and set intersection is independent of the order of sets.) Consequently, we have

$$p(x^{(i)}_{\sigma^{(i)}(1:m)}|Z) = p(x^{(i)}_{1:m}|Z). \qquad (16)$$

The equation (16) is true for every particle (i), although we may have different permutations $\sigma^{(i)}(1:m)$ for different particles. Consequently, $x^{(i)}_{1:m}$ is a weighted sample from $p(x_{1:m}|Z)$ for every $i = 1,\ldots,N$, which proves the theorem.

We stress again that the particular order of dimensions $\sigma^{(i)}(1:m)$ is extremely important for the proposed PFUO. However, once we have this state vector, we can simply present the state vector in the original order of dimensions.

The Sampling Importance Resampling (SIR) replaces current particles with N new particles with the weights normalized to sum to one, which provides an approximation to the same target pdf. This proves the theorem.

The fact that we can consider more than one follower of each particle and reduce the number of followers by resampling is known in the PF literature and is referred to as prior boosting [8, 1]. It is used to capture multi-modal likelihood regions. We stress that the resampling in our framework plays an additional and a very crucial role. It selects the the most informative orders of states. Since the weights of $w(x^{(i,s)}_{\sigma(1:t)})$ are determined by the corresponding order of observations $z_{\sigma^{(i)}(1:t-1)}$, and the resampling uses the weights to selects new particles $x^{(i)}_{\sigma(1:t)}$, the resampling determines the order of state dimensions. Consequently, the order of state dimensions is heavily determined by their corresponding observations, and this order may be different for each particle (i). This is in strong contrast to the classical PF, where observations are considered only in one order Z.

In order to utilize the derived PF algorithm to solve the jigsaw puzzle problem, we need to define the proposal pdf (11) and the conditional pdf of a new observation (12). Both are defined in the next section.

5. Implementation Details

Given is an rectangular grid of m empty squares $G = \{g_1,\ldots,g_m\}$, as illustrated in Fig. 1(c) for $m = 6$, where each g_i is the index of a square in the grid. In order to solve an image jigsaw puzzle with the puzzle pieces $P = \{1,\ldots,m\}$, we need to assign locations on S to the puzzle pieces in P. The observation associated with each puzzle piece is the local image depicted on it, i.e., z_i is a $K \times K \times 3$ matrix of pixel color values for $i \in P$ and $Z = \{z_1,\ldots,z_m\}$ is the set of observations.

A state value assignment at time $t \leq m$ of our PF algorithm is given by a state vector $x_{\sigma(1:t)} = (x_{\sigma(1)},\ldots,x_{\sigma(t)})$, where each $\sigma(i) \in P$ is an index of a puzzle piece for $j = 1,\ldots,t$ and $x_{\sigma(j)} \in S$ meaning that puzzle piece $\sigma(i)$ is placed on the grid square with index $x_{\sigma(i)}$. The vector of corresponding observations $z_{\sigma(1:t)} = (z_{\sigma(1)},\ldots,z_{\sigma(t)})$ represents the local images of the puzzle pieces. In this section we drop the particle index (i), since all definitions apply to every particle.

We define now an affinity matrix A representing the compatibility of the local images on the puzzle pieces. It is a 3D matrix with the third dimension being an adjacency type, since two puzzle pieces can be adjacent in four differ-

ent ways: left/right, right/left, top/bottom, and bottom/top, which we denote with LR, RL, TB, and BT. Hence the dimension of A are $m \times m \times 4$.

In order to be able to compare our experimental results to the results in [2] we define A following the definitions in [2]. They first define the dissimilarity-based compatibility D. D measures dissimilarity between puzzle piece images z_j, z_i by summing the squared LAB color difference along the boundary, e.g., the left/right (LR) dissimilarity is defined as

$$D(j,i,LR) = \sum_{k=1}^{K}\sum_{c=1}^{3}(z_j(k,u,c) - z_i(k,v,c))^2, \quad (17)$$

where u indexes the last column of z_j and v indexes the first column of z_i. ($D(x_j, x_i)$ is not a distance, since it is not symmetric, i.e., $D(x_j, x_i) \neq D(x_i, x_j)$.) Finally, the weight of the LR connection is given by

$$A(j,i,LR) = \exp(-\frac{D(j,i,LR)}{2\sigma^2}), \quad (18)$$

where σ is adaptively set as the difference between the smallest and the second smallest D values between puzzle piece i and all other pieces in P, see [2] for more details.

Given is a vector $x_{\sigma(1:t)}$ of state values (i.e., grid locations) at time t. Let $\mathcal{C}(\sigma(j)) \subset \{LR, RL, TB, BT\}$ denotes a set of possible connections of puzzle piece $\sigma(j)$ for $j = 1, \ldots, t$, e.g., if the left and top sides of $\sigma(j)$ are adjacent to the other puzzle pieces $\sigma(1:t)$ and the right and bottom sides are adjacent to empty grid squares, then $\mathcal{C}(\sigma(j)) \subset \{RL, BT\}$. We can extend this definition to the whole state vector by

$$\mathcal{C}(\sigma(1:t)) = \bigcup\{\mathcal{C}(\sigma(j)) \mid j = 1, \ldots, t\}.$$

The proposal distribution $p_s(\cdot|x_{\sigma(1:t-1)}) : G \rightarrow \mathbb{R}$ is assigns a probability of placing puzzle piece s to each grid square x. $p_s(x|x_{\sigma(1:t-1)}) = 0$ if x is occupied or is not adjacent to any square in $\sigma(1:t-1)$. Now say x is adjacent and is to the right of grid square $x_{\sigma(j)}$ for some $j = 1, \ldots, t$. Then

$$p_s(x|x_{\sigma(1:t-1)}) \propto A(s, \sigma(j), RL). \quad (19)$$

Intuitively, this probability is proportional to the LR similarity between puzzle pieces s and $\sigma(j)$.

Now let x_s be a sample from (11) at time t, and as above x_s is adjacent and is to the right of grid square $x_{\sigma(j)}$ for some $j = 1, \ldots, t$. Then

$$p(z_s|x_{\sigma(1:t)}, z_{\sigma(1:t-1)}) \propto A(s, \sigma(j), RL), \quad (20)$$

where we recall that $\sigma(t) = s$ and $\sigma(1:t) = (\sigma(1:t-1), \sigma(t))$. The definitions for other adjacency types LR, TB, BT are analogous.

To summarize the proposal distribution is a function of how well puzzle piece s fits to the already placed pieces and assigns the probability of placing s to all grid squares, while in the evaluation we already know the grid location of puzzle piece s as well as its adjacent square. We then use this information to compute the evaluation probability according to A. Hence, both the proposal and evaluation of a given particle are functions of how well adjacent pieces fit together following the order in which the pieces has been added. In order to compare the proposed PF inference to the loopy believe propagation for solving the image jigsaw puzzle, we use exactly the same functions to define the probabilities as in [2]. The comparison is presented in the next section.

For a given image jigsaw puzzle with m pieces, the time complexity of the proposed inference framework is $O(m^2N)$, where N is the number of particles.

6. Experimental Results

We compare the image jigsaw puzzle solutions obtained by the proposed PF inference framework to the solutions of the loopy believe propagation used in [2] under identical settings. We ran the software released by the authors of [2] to obtain their results. The results are compared on the dataset provided in [2], which we call MIT Dataset. It is composed of 20 images. In addition, we also consider an extended dataset composed of 40 images, i.e., we added 20 images. As we will see below the results of both methods on the original and extended datasets are comparable.

The experimental results in [2] are conducted in two different settings: with and without any prior on the target image layout. In [3] the prior of the image layout is given by a low resolution version of the original image. [2] weakens this assumption to a statistics of the possible image layout. We focus on the results without any prior of the image layout. Consequently, we focus on a harder problem, since we only use the pairwise relation between the image patches. It is given by pair-wise compatibilities of located puzzle pieces as defined in Section 5.

In the probabilistic framework in [2], each grid location is assigned a puzzle piece. In our PF framework, it is more natural to assign a grid location to a puzzle piece. The solutions of both methods are equivalent, since a final puzzle solution is a set of m pairs composed of (puzzle piece, grid location), where m is the number of pairs. We call such pairs the solution pairs.

Three types of evaluation methods are used in [2]. Each focuses on different aspects of the quality of the obtained puzzle solutions. The most natural and strict one is **Direct Comparison**. It simply computes the percentage of correctly placed puzzle pieces, i.e., for a puzzle with m pieces, Direct Comparison is the number of correct solution pairs divided by the total number of solution pairs m.

The second measure is **Cluster Comparison**. It is less strict than Direct Comparison. It tolerates an assignment error as long as the puzzle piece is assigned to a location that belongs to a similar puzzle piece. The puzzle pieces are first clusters into clusters of similar pieces and only the cluster labels are used in the count of the number of correct solution pairs.

Moreover, due to lack of prior knowledge of target image, the reconstructed image may shift compared to the ground truth image. Therefore, the third measure called **Neighbor Comparison** is used to evaluate the label consistency of adjacent puzzle pieces independent of their grid location, i.e., the location of two adjacent puzzle pieces is considered correct if two puzzle pieces are left-right neighbors in the ground truth image and they are also left-right neighbors in he inferred image and similarly for the other three possible adjacency relations. Neighbor Comparison is the fraction of correct adjacent puzzle pieces. This measurement do not penalize the accuracy as long as the adjacent patches in original image are adjacent in the reconstructed image.

The results on the MIT Dataset are shown in 1 and on the extended dataset in Table 2. We stress that the proposed PF inference framework outperforms the loopy believe propagation in all three performance measures. Moreover, the reconstruction accuracy of the original images by our algorithm is three times better according to the most natural measure of Direct Comparison.

In order to demonstrate that the considered image jigsaw puzzle problem is also very challenging to humans, we show some example results in Fig. 2. There we show the original images, but we stress again in our experimental evaluation, the original images are not provided. Fig. 2 also demonstrates that the reconstructed images obtained by the proposed algorithm compare very favorably to the results of [2].

Both methods are initialized with one anchor patch, i.e., with one correct (puzzle piece, grid location) pair. We always assign a correct image patch to the top left corner of the image. In all our experiments we divide each test image into 108 square patches resulting in $m = 108$ puzzle pieces. Of course, the more more patches, the smaller are the images on the puzzle pieces, thus, larger number of patches makes the problem harder. Since the performance of [2] on 108 puzzle pieces is already very low, in particular, as measured by Direct Comparison, a larger number of pieces does not seem to lead to any meaningful results.

In order to demonstrate the dynamic of the proposed PF inference, we show reconstructed images of the best particle at different times (iterations) in Fig. 3. The puzzle pieces of the best particles at different iterations are different. This illustrates the facts that different dimensions of the state vector are considered by different particles and the orders of considered dimensions are different.

	[2]	Our algorithm
Direct Comparison	0.2366	0.6921
Neighbor Comparison	0.6628	0.8620
Cluster Comparison	0.4657	0.7810

Table 1. Experimental results on MIT Dataset.

	[2]	Our algorithm
Direct Comparison	0.2137	0.7097
Neighbor Comparison	0.6458	0.8770
Cluster Comparison	0.4500	0.8018

Table 2. Experimental results on the extended dataset.

7. Related Work

The first work on Jigsaw Puzzle Problem is [6]. Since shape is an important clue for accurate pairwise relation, many methods [10, 7, 13, 17] focus on matching distinct shapes among jigsaw pieces to solve the problem. The pairwise relations among jigsaw pieces are measured by the fitness of shapes. It is straightforward to consider both the shape and image content [11, 12, 18]. Most methods solve the problem with a greedy algorithm and report results on just one or few images. Recently, Cho et. al [2] propose to use loopy belief propagation to solve the problem and evaluate on a test dataset.

describe PF approaches

8. Conclusion

We introduce a novel inference framework for solving labeling or assignment problems expressed as dynamic state estimation. Our key contribution is an extension of the PF framework to work with unordered observations. Intuitively, this means that the weighted particles explore different orders of label assignment hypothesis. We prove that so obtained weighted particles represent samples form the target, posterior distribution of label assignments. We evaluate the labeling performance on a problem of image jigsaw puzzles, which is very challenging even for humans. As the experimental results demonstrate, the proposed PF inference significantly outperforms the loopy belief propagation.

References

[1] J. Carpenter, P. Clifford, and P. Fearnhead. Building robust simulation-based filters for evolving data sets, 1999. 5

[2] T. S. Cho, S. Avidan, and W. T. Freeman. A probabilistic image jigsaw puzzle solver. In *CVPR*, 2010. 1, 2, 6, 7, 8

[3] T. S. Cho, M. Butman, S. Avidan, and W. T. Freeman. The patch transform and its applications to image editing, booktitle=. 6

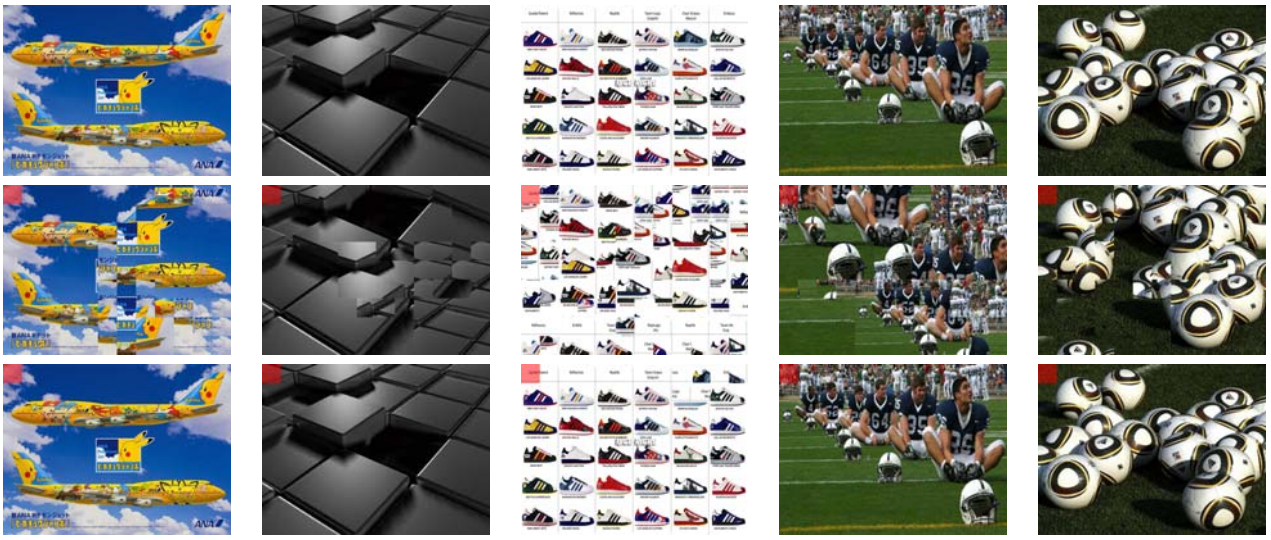
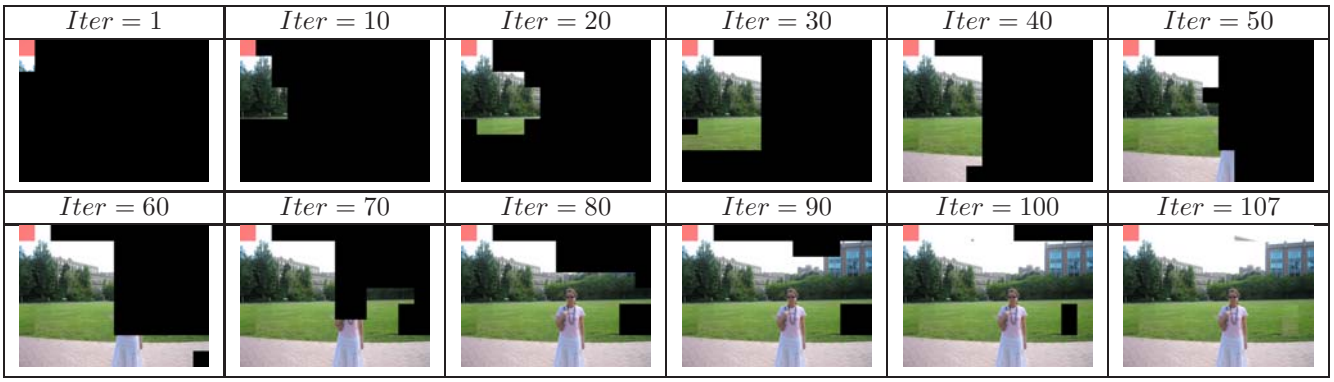


Figure 2. The images in the first row are the original images. The jigsaw puzzle solution of Cho et. al [2] are shown in the second row. The solutions of our algorithm are in the third row.



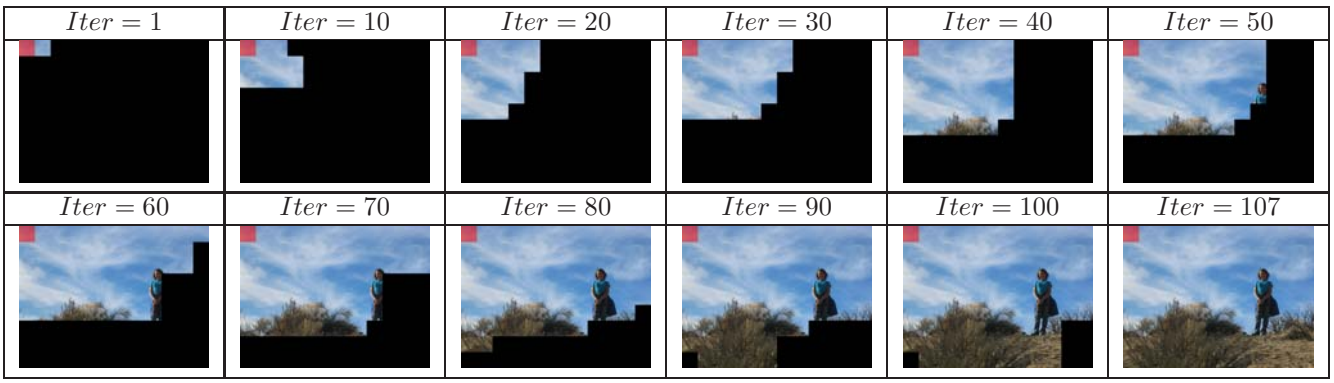


Figure 3. The reconstructed images of the best particle at different iterations.

[4] E. D. Demaine and M. L. Demaine. Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. *Graphs and Combinatorics*, 2007. 1

[5] D. Fox, S. Thrun, F. Dellaert, and W. Burgard. Particle filters for mobile robot localization. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer Verlag, New York, 2000. 2

[6] H. Freeman and L. Garder. Apictorial jigsaw puzzles: the computer solution of a problem in pattern recognition. *IEEE TEC*, 13:118–127, 1964. 7

[7] D. Goldberg, C. Malon, and M. Bern. A global approach to automatic solution of jigsaw puzzles. In *Symposium on Computational Geometry*, 2002. 1, 7

[8] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *Radar and Signal Processing, IEE Proceedings of*, volume 140, pages 107–113, April 1993. 5

[9] Z. Khan, T. Balch, and F. Dellaert. An mcmc-based particle filter for tracking multiple interacting targets. In *ECCV*, 2004. 2

[10] W. Kong and B. B. Kimia. On solving 2d and 3d puzzles using curve matching. In *CVPR*, 2001. 1, 7

[11] M. Makridis and N. Papamarkos. A new technique for solving a jigsaw puzzle. In *ICIP*, 2006. 7

[12] T. R. Nielsen, P. Drewsen, and K. Hansen. Solving jigsaw puzzles using image features. *PRL*, 2008. 7

[13] G. Radack and N. Badler. Jigsaw puzzle matching using a boundary-centered polar encoding. In *CGIP*, 1982. 1, 7

[14] K. Smith, D. Gatica-Perez, and J.-M. Odobez. Using particles to track varying numbers of interacting people. In *CVPR*, 2005. 2

[15] S. Thrun. Particle filters in robotics. In *Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)*, 2002. 2

[16] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press Cambridge, 2005. 3, 4

[17] H. Wolfson, E. Schonberg, A. Kalvin, and Y. Lamdam. Solving jigsaw puzzles by computer. *Annals of Operations Research*, 1988. 1, 7

[18] F. H. Yao and G. F. Shao. A shape and image merging technique to solve jigsaw puzzles. *PRL*, 2003. 7