Digital Tapestry

Carsten Rother [1] Sanjiv Kumar [2] Vladimir Kolmogorov [1] Andrew Blake [1]

[1] *Microsoft Research, Cambridge, UK* [2] *Carnegie Mellon University, Pittsburgh, USA*

Abstract

This paper addresses the novel problem of automatically synthesizing an output image from a large collection of different input images. The synthesized image, called a digital tapestry*, can be viewed as a visual summary or a virtual 'thumbnail' of all the images in the input collection. The problem of creating the tapestry is cast as a multi-class labeling problem such that each region in the tapestry is constructed from input image blocks that are salient and such that neighboring blocks satisfy spatial compatibility. This is formulated using a Markov Random Field and optimized via the graph cut based expansion move algorithm. The standard expansion move algorithm can only handle energies with* metric *terms, while our energy contains non-metric (soft and hard) constraints. Therefore we propose two novel contributions. First, we extend the expansion move algorithm for energy functions with non-metric* hard *constraints. Secondly, we modify it for functions with "almost" metric* soft *terms, and show that it gives good results in practice. The proposed framework was tested on several consumer photograph collections, and the results are presented.*

1 Introduction

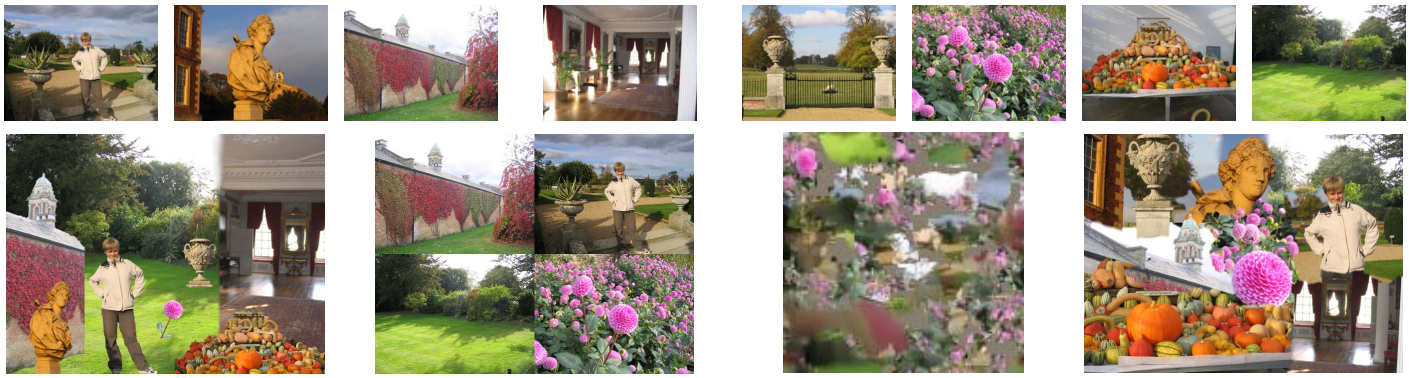
Consider the collection of consumer photographs shown in fig. 1(top). This paper addresses the problem of automatically summarizing such a collection in a single photomontage, termed a *digital tapestry*. Fig. 1(a) shows a manually generated tapestry using commercial image editing software [1]. The digital tapestry will be useful for two main purposes: to remind the user of the photo collection, a 'thumbnail' of the image collection; to act as an image retrieval system - e.g. by selecting a part of the tapestry, all images that have similar regions can be retrieved from the collection.

In principle, one could generate a naive version of a tapestry by first selecting a subset of images from the input collection based on some global image properties, e.g. color, and then creating a mosaic using this subset. Fig.1(b) shows an example mosaic tapestry using a subset of 4 images. The advantage of using entire images is that the shape and appearance of the tapestry regions are preserved. However, in addition to not being visually appealing, the main drawback of such a tapestry is that it comprises information from a very small number of images, where several regions are potentially uninformative (*grass* in this case). In a tapestry one would like to include as many salient regions from different images in the collection as possible. Another possible choice is to synthesize a tapestry through texture synthesis. However, the traditional texture synthesis techniques, both parametric [16] as well as non-parametric [5] address the problem of synthesizing a large texture image given a small sample. Clearly, these will be insufficient to generate a rich structure in tapestry because of the variety of objects and textures contained in different input images.

Jojic et al. [7] recently proposed a generative framework to obtain a condensed version of the input image called an epitome. An epitome contains the essence of the shape and appearance of the original image. Potentially, the epitome framework could be extended to create a photomontage from different images. However, as pointed out in [7], the epitome model for several input images works well only if these images contain *similar* objects (e.g., different images of the same scene). On the contrary, the collection from which a tapestry is to be created will usually have many different images. Fig. 1(c) shows the epitome for the mosaic images in fig. 1(b)[1]. The structure of the regions is not preserved in the epitome. It should be noted that an epitome can generate the original image again using a smooth map which is also learned in the framework. However, in the tapestry framework, the aim is to obtain a salient representative image from the collection of input images, free of the need to generate input images from the tapestry.

In the domain of user-assisted techniques, recently a system called digital photomontage has been proposed [2] which combines parts of a set of photographs into a single composite picture. The input set of images are assumed to be of the same scene and roughly registered, e.g. several images from the same camera viewpoint. In our case, the images from a collection will usually contain very different scenes and registration is infeasible. Kwatra et al. [11] have described a framework to combine parts from two different images. But the problem of which image parts to select and where to place them is left to the user. In this paper we present a framework to create a tapestry fully automatically from a large number of input images. As we will see, the basic concept for our framework is related to [11].

[1] http://research.microsoft.com/ jojic/software.htm. To remove artificial seams, the epitome was initialized with the top, left image in fig. 1.



(a) Manual Tapestry (b) Mosaic of representative Images (c) Epitome (d) Digital Tapestry

Figure 1. Tapestry comparison. Top: eight different images from a consumer photo collection. Bottom: Four different ways to visually summarize such a collection: (a) Manually created tapestry using commercial image editing software [1]; (b) mosaic of representative images; (c) shape and appearance preserving epitome [7]; (d) digital tapestry created with the proposed method.

First, we describe the properties of what we call an ideal tapestry. An ideal tapestry should contain visually informative regions from as many different input images as possible. These regions should be placed in the tapestry realistically. The redundancy in appearance between different regions should be minimal and the main texture types from different images should be represented. However, the tapestry does not have to resemble a real image. For this, we propose a framework which answers three essential questions: which regions of the input images should be selected, where to place them in the tapestry, and finally how to remove any residual visual artifacts. The main contribution of this paper is to present a fully automatic framework which addresses these questions in a principled manner. Our technique does not require a-priori scene understanding or detection of generic objects. We also demonstrate how high-level knowledge, such as face detection can be incorporated in the framework which improves the tapestry. One example tapestry created automatically by our framework is given in fig. 1(d). Compared to fig. 1(b), redundant information is removed, so that all input images are represented. The difference between fig. 1(a) and (d) clearly shows the lack of high-level knowledge in our framework.

In the proposed framework, we formulate the tapestry problem as a multi-class labeling problem over a Markov Random Field (MRF), such that each region in the tapestry is constructed from salient input image blocks and neighboring blocks satisfy spatial compatibility. The formulation is presented in detail in sec. 2. In sec. 3 we describe how the MRF energy is minimized using the expansion move algorithm of Boykov et. al. [3]. The standard expansion move algorithm can only handle energies with *metric* terms, however, our energy contains hard and soft constraints which are *non-metric*. This leads to two novel contributions. Firstly, we extend the expansion move algorithm for energy functions with non-metric hard constraints. Secondly, we modify it for functions with "almost" metric soft terms. Sec. 4 describes some more experiments on consumer photo collections, and finally we conclude in sec. 5.

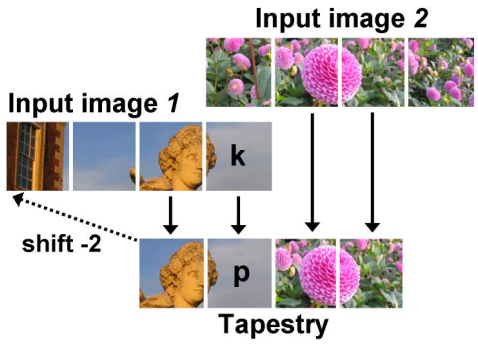
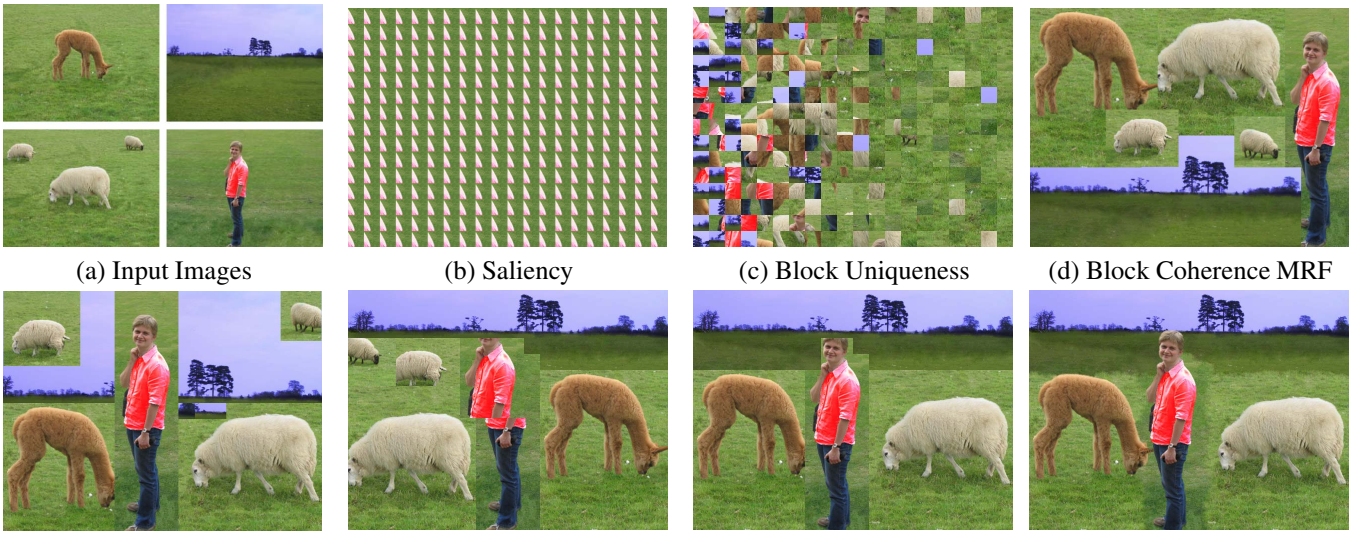


Figure 2. Tapestry Labeling. Input blocks are matched from the input images to the tapestry. For example, image $i = 1$ has shift $s = -2$ with respect to the tapestry. Matching an input block from this image to the tapestry means that the tapestry block p has label $f_p = (i, s) = (1, -2)$. Furthermore, given the tapestry block p at position $x_p = 2$ in the tapestry, and its label f_p we can derived uniquely the input image block $k \in \mathcal{K}$ in image $i = 1$ at position $x_p - s = 4$. Note that for simplicity this is a $1D$ illustration.

2 Problem formulation

After introducing the notation, we explain in sec. 2.1 the basic constraints which are necessary to obtain tapestries without any prior scene knowledge. These basic constraints give us a problem which we call *matching with smoothness*, which is discussed in sec. 2.2. To improve the quality of the tapestry we introduce in sec. 2.3 additional constraints based on prior scene understanding.



(a) Input Images (b) Saliency (c) Block Uniqueness (d) Block Coherence MRF

(e) Location Constraint (f) Appearance dependent MRF (g) Image-Shift uniqueness and face detection (h) Block artifact removal

Figure 3. Digital Tapestry. (a) Input images. (b) Without block uniqueness constraint the most salient block is duplicated over the whole tapestry. (c) With block uniqueness the tapestry contains the most salient blocks (column-wise starting top,left). (d) A constant pairwise constraint (MRF with Potts model) has the positive effect that regions of input blocks are coherent. (e) Given appearance-based clustering, certain textures, e.g. sky, can only occur at specific locations, e.g. on the top. (f) Transitions between neighboring blocks are realistic, e.g. the sheep are now placed on a grass background. (g) Each image can only appear with one unique shift in the tapestry, e.g. the upper part of the body and the legs in (f) are now correctly placed. Also face detection [15] prevents faces from being chopped off or excluded. (h) Visual artifact removal using pixel-wise graph cut texture [11] and feathering [14].

Assume that the input images and the tapestry image are divided into equally sized blocks, here 32×32 pixels. The tapestry image is created by matching a subset of input image blocks to the tapestry[2]. We will view this problem as a labeling problem of the tapestry blocks, see fig. 2.

Let $\mathcal{I}$ be the set of input images and i an image. $\mathcal{K}$ is the set of all input image blocks where k is a block. Furthermore, let $p, q \in \mathcal{P}$ be blocks in the tapestry. Let us define a label space $\mathcal{L} = \mathcal{I} \times \mathcal{S}$, where $\mathcal{S}$ is the set of all possible 2D "block-shifts" of an input image with respect to the tapestry image[3]. The labeling of tapestry block p is defined as $f_p = (i, \mathbf{s})$ with $\mathbf{s} \in \mathcal{S}$. Given the tapestry block p, at position $\mathbf{x}_p$ in the tapestry, and its label $f_p = (i, \mathbf{s})$ we can derive uniquely the input image block as $b(p, f_p) = k \in \mathcal{K}$, at position $\mathbf{x}_p - \mathbf{s}$ in image i (see fig. 2). The function $b(p, f_p)$ is the unique (backward) mapping from the pair $\langle p, f_p \rangle$ onto the set of all input blocks $\mathcal{K}$. The synthesis problem is to find a mapping (configuration) $f : \mathcal{P} \to \mathcal{L}$, which assigns uniquely a label to each tapestry block. We define an energy function $E(f)$ for each configuration f,

$$E(f) = \sum_p D_p(f_p) + \sum_{p,q \in \mathcal{N}_V} V_{p,q}(f_p, f_q) + \sum_{p,q \in \mathcal{N}_H} H_{p,q}(f_p, f_q). \quad (1)$$

It consists of three terms: the data term $D_p(\cdot) \in \mathbb{R}$ imposes unary constraints, the pairwise soft constraint $V_{p,q}(\cdot, \cdot) \in \mathbb{R}$ encodes smoothness between neighboring blocks p, q in the tapestry, and $H_{p,q}(\cdot, \cdot) \in \{0, \infty\}$ encodes hard constraints, which prohibits certain configurations.

2.1 Basic Constraints

The goal of a tapestry is to remind a user of the summarized photo collection (typically their personal collection). When we consider individual image blocks, we believe that some blocks are more informative (salient) than others to prompt the user's memory. To confirm this conjecture, we ideally have to learn "saliency" from a psychological experiment. Until then we assume that blocks with high contrast are salient. A high contrast block is more likely to contain shape information. For instance, a block of uniform sky is less salient than its neighboring block containing the horizon (fig. 3(a)). Obviously, the uniform sky block is also needed to explain the neighboring horizon block. We will deal with this spatial constraint of neighboring blocks later. The most salient blocks of the input images are shown in fig. 3(c) (column-wise starting top, left).

To determine the contrast blocks we first smooth and down-sample the image, so that an image block is now of size 2×2 pixels. The contrast is then computed as the sum of the gradient magnitudes within the down-sampled image

[2] This is similar to [11] where a block is replaced by a pixel.

[3] For simplicity we assume that all input images are of the same size.

block. This defines the data term as

$$D_p(f_p) = -\text{Saliency}(b(p, f_p)) \ . \quad (2)$$

In order to encourage that each image contributes to the tapestry, the saliency value of all blocks of one image are normalized to one. Additionally, we use the heuristic assumption that the image center is more informative about the image's content than the border, details are omitted.

Optimizing the energy E in eqn. (1) with the saliency constraint as the only term has the effect that the most salient block is duplicated over the whole tapestry (fig. 3(b)). To avoid this, we have to constrain the energy so that any two tapestry blocks are from different input blocks, i.e. $b(p, f_p) \neq b(q, f_q)$ for all blocks p, q. In terms of energy, we can write this **block uniqueness** constraint as a hard constraint of the form

$$H_{p,q}(f_p, f_q) = \infty \ \text{ if } \ b(p, f_p) = b(q, f_q) \ . \quad (3)$$

This hard constraint gives us a *matching* problem: we want to find a mapping between image blocks and tapestry blocks such that each tapestry block has one match.

A tapestry which contains just the most salient blocks is not very informative, fig. 3(c). In order to capture larger, salient image regions, we want to "grow", in the tapestry, blocks which are from the same input image. The most simple spatial constraint on the tapestry blocks is to introduce a MRF with a Potts model. We add to our energy the soft **block coherence** constraint

$$V_{p,q}(f_p, f_q) = \lambda_1 \ \text{ if } f_p \neq f_q \ , \quad (4)$$

where V has an 8-neighborhood system $\mathcal{N}_V$ and the weight λ_1 decides how spatial coherent the tapestry is. Note that neighboring tapestry blocks p, q with identical labels $f_p = f_q$ are always neighboring blocks in the input image. Fig. 3(d) shows that these three basic constraints are sufficient to obtain a reasonably looking tapestry. To improve the result, additional constraints based on intermediate and high-level knowledge are necessary, as we will discuss in sec. 2.3

The coherence constraint transforms the matching problem, introduced above, to the problem of *matching with smoothness*, as we denote it. How this problem can be addressed is discussed in the next section.

2.2 Matching with smoothness

The block uniqueness constraint and block coherence constraint gives us a problem which we call *matching with smoothness*. Minimizing the corresponding energy function E is NP-hard, therefore, we have to resort to approximation techniques such as graph cut based expansion move algorithm or loopy belief propagation. It might seem that they are not practical since we have a fully connected MRF model. However, we will show in sec. 3.4 that our problem can be tackled efficiently by the expansion move algorithm [3]. In particular, in each step of the algorithm the number of edges needed for enforcing the block uniqueness constraint is at most linear in the number of tapestry blocks.

The expansion move algorithm was also used for the matching with smoothness problem in the context of stereo [8, 9]. Unlike our problem, stereo is symmetric: occlusions are allowed in both left and right images, while we allow "occlusions" (non-matched blocks) for input blocks but not tapestry blocks. Algorithms in [8, 9] could be adapted to our problem by setting appropriate occlusion penalties to infinity. However, the graphs constructed during α-expansion steps would contain approximately twice as many nodes as in our formulation.

2.3 Additional Constraints

To further improve the quality of the tapestry, we introduce additional constraints based on prior scene understanding.

Localization constraint. One of the requirements while creating a tapestry is where to place the salient regions from different images in the tapestry. We formulate this by introducing a hidden variable, h, which can be seen as a cluster variable representing the appearance clusters of the input image blocks. As appearance of a block we currently use its dominant block color. Thus, the distribution over locations, x_p, in the tapestry for a given block $k = b(p, f_p)$ can be expressed as: $P(x_p|k) = \sum_h P(x_p|h)P(h|k)$, where $P(x_p|h)$ encodes the preference of certain cluster of blocks to appear at particular locations (e.g. sky tends to be on the top - see fig. 3(e)). The second term, $P(h|k)$, indicates the cluster membership of the given block k. This term can be obtained using EM for mixture of Gaussian (MoG) clustering. Assuming the independence of the image blocks, the first term can be written as,

$$P(x_p|h) \propto \prod_{k' \in \mathcal{K}} \delta(x_{k'} - x_p) p(k'|h),$$

where $\delta(t) = 1$ if $t = 0$ and 0 otherwise, and $p(k'|h)$ is the cluster likelihood, obtained directly from MoG. We add to the data term, define in eqn. (2), the location constraint as

$$D_p(f_p) \rightarrow D_p(f_p) - \lambda_2 \ \log \ P(x_p|b(p, f_p)) \ . \quad (5)$$

Appearance dependent MRF. We defined our ideal tapestry with the property that image regions should be placed realisticly. Fig. 3(e) shows that many block transitions are violating this constraint, e.g. the sheep have grass as background, however, are placed in the sky. Similar to Kwatra et al. [11] we introduce an appearance based block transition. We replace the pairwise constraint in eqn. (4) by

$$V_{p,q}(f_p, f_q) = \lambda_1 + \lambda_3 \min \Big(\big\| C(b(p,f_p)) - C(b(p,f_q)) \big\|_2, \big\| C(b(q,f_q) - C(b(q,f_p) \big\|_2 \Big) \ \text{ if } \ f_p \neq f_q \ ,$$

where $C(k)$ is the appearance of block k, which is currently its dominant color. To be robust to salient blocks, which

might contain two different textures, we use the $\min(\cdot)$ function. Note that V might be non-metric, which can not be handled by the standard expansion move algorithm. Therefore, we extend the algorithm in sec. 3.3.

Image-Shift uniqueness. One possibly undesirable phenomena is that two blocks, assosciated with one object in some input image, both appear in the tapestry but mutually misregistered, i.e. with a different image-shift, such as the upper part of the body and the legs in fig. 3(f). Therefore, we introduce the hard constraint that every image can only be present in the tapestry with one unique shift. This has the negative side effect that less salient parts, like the two smaller sheep in fig. 3(f) are discarded, fig. 3(g).

Let us introduce the label $f_i \in \mathcal{S}$ which represents the shift of image i. We replace the block uniqueness constraint (eqn. (3)) by the image-shift uniqueness constraint

$$H_{p,i}(f_p, f_i) = \infty \text{ if } f_p = (i, \mathbf{s}) \text{ and } f_i \neq \mathbf{s} \ . \qquad (6)$$

An alternative option for avoiding two differently shifted parts, in the tapestry, of one salient region is described next, which however involves object detection.

Face (object) detection. Assume an object detection system identifies that blocks in an image belong to the same object, such as a face in fig. 3(g). Obviously these blocks should appear as a connected region in the tapestry. Therefore, we add the hard constraints $H_{p,q}(f_p, f_q) = \infty$ if $f_p \neq f_q$ and $b(p, f_p), b(q, f_q)$ are two blocks of the same object. We currently use the face detection system of [15]. Since faces are important to prompt the user's memory, we would like to include the face of *each* person present in the photo collection. The data term of a "face block" $b(p, f_p)$ is adapted to $D_p(f_p) = -const$, where $const > max(D(\cdot) + 8\ max(V(\cdot, \cdot))$. To reduce the risk of duplicating faces of the *same* person, we adapt only those "face blocks" which appear in the image with the most faces, typically a group photograph. The automatic clustering of faces [6] is a challenging task which we plan to address.

Class uniqueness. In a typical photo collection very similar images frequently appear, e.g. same shot with different lightning. Since the tapestry can only represent a fraction of all images, depending on tapestry size and λ_1, different tapestry regions with very similar appearance have to be avoided. To address this issue, we cluster "similar" images, based on global image properties, such as histogram of colors [13]. We introduce the constraint that the tapestry must contain only *one* sample image per cluster.

Let us define a cluster variable $c \in \mathcal{C}$. This variable has a label $f_c \in \mathcal{I}$ which denotes that the cluster c, if present in the tapestry, is represented by image f_c. We add to our energy the class uniqueness constraint

$$H_{p,c}(f_p, f_c) = \infty \text{ if } f_p = (i, \mathbf{s}),\ b(p, f_p) = c \text{ and } f_c \neq i. \qquad (7)$$

This constraint is used for all examples in sec. 4. We plan to extend this constraint from redundant image detection to redundant region detecting, which however needs a reliable and un-supervised texture clustering as addressed in [4].

Removing Visual Artifacts. In fig. 3(g) we see that using blocks instead of pixels introduced two noticeable artifacts. Firstly, true object boundaries may be missing, like the hairline of the person. We remove this artifact by using a variation of [11] where the seam prefers to follow existing boundaries. Secondly, the boundaries between the blocks introduce an artificial seam, e.g. transition from dark to bright grass. We adjust the colors at the seam using feathering within a ribbon around the seam [14]. Since we do not want to blur true objects boundaries, the size of the ribbon depends on the strength of the existing boundary. The final tapestry is shown in fig. 3(h).

3 Energy minimization via Graph Cuts

In this section we discuss the optimization framework for the energy introduced above. As discussed in sec. 2.2, we believe that expansion move algorithm of Boykov et al. [3] is the most suitable technique for our problem. Indeed, we show in sec. 4 that iterated conditional modes (ICM) performs worse.

The energy contains non-metric (soft and hard) constraints, which can not be handled by the standard expansion move algorithm. Therefore, in sec. 3.2 we extend the expansion move algorithm for functions with non-metric *hard* terms. Sec. 3.3 introduces a modify version for general functions with non-metric *soft* terms.

3.1 The Expansion Move Algorithm

The basic idea of the expansion move algorithm is to reduce the problem of minimizing function E (eqn. (1)) with *multiple* labels to a sequence of *binary* minimization problems. These subproblems are called *alpha expansions*. They can be described as follows, see [3, 10] for details.

Suppose that we have a current configuration (set of labels) f and a fixed label $\alpha \in \mathcal{L}$. In the α-expansion operation each pixel (or block in our case) $p \in \mathcal{P}$ makes a binary decision: it can either keep its old label or switch to label α. Therefore, we introduce a binary vector $\mathbf{x} \in \{0, 1\}^{\mathcal{P}}$ which defines the auxiliary configuration $f[\mathbf{x}]$ as follows:

$$\forall p \in \mathcal{P} \quad \text{it is} \quad f[\mathbf{x}]_p = \begin{cases} f_p & \text{if } x_p = 0 \\ \alpha & \text{if } x_p = 1 \end{cases} \ .$$

This auxiliary configuration $f[\mathbf{x}]$ transforms the energy E with *multiple* labels into an energy function of *binary* variables $\mathcal{E}(\mathbf{x}) = E(f[\mathbf{x}])$[4]. It can be written in the form similar to eqn. (1):

$$\mathcal{E}(\mathbf{x}) = \sum_p \mathcal{E}_p(x_p) + \sum_{p,q \in \mathcal{N}} \mathcal{E}_{p,q}(x_p, x_q) \ , \qquad (8)$$

[4]Note that the notation $E, \mathcal{E}$ is used for energies with multiple and binary labels respectively.

where $\mathcal{N} = \mathcal{N}_V \cup \mathcal{N}_H$. Individual terms are defined by the terms of function E. For example, for $p, q \in \mathcal{N}_V$ we have

$$\begin{array}{|c|c|} \hline \mathcal{E}_{p,q}(0,0) & \mathcal{E}_{p,q}(0,1) \\ \hline \mathcal{E}_{p,q}(1,0) & \mathcal{E}_{p,q}(1,1) \\ \hline \end{array} = \begin{array}{|c|c|} \hline V_{p,q}(f_p, f_q) & V_{p,q}(f_p, \alpha) \\ \hline V_{p,q}(\alpha, f_q) & V_{p,q}(\alpha, \alpha) \\ \hline \end{array} .$$

Under certain conditions, described below, a global minimum of $\mathcal{E}$ can be computed efficiently using graph cuts.

The expansion move algorithm starts with an initial configuration f^0. It computes optimal α-expansion moves for labels α in some order, accepting the moves only if they decrease the energy. The algorithm is guaranteed to converge. Its output is a *strong* local minimum characterized by the property that *no α-expansion can decrease the energy E*.

3.2 Class of Energies with Strong Local Minima

In this section we discuss for which multi-label functions E we can obtain a strong local minimum. As described above, this can be done if for any α-expansion a global minimum of function $\mathcal{E}$ in eqn. (8) can be computed efficiently.

In previous work researchers considered the case when the hard constraint term H in eqn. (8) is not present. Boykov et al. [3] gave a graph construction which is applicable when V is a *metric*. This condition was generalized in [10], where they showed that $\mathcal{E}$ can be minimized efficiently if it is *regular*, i.e. each term $\mathcal{E}_{p,q}$ satisfies the following inequality:

$$\mathcal{E}_{p,q}(0,0) + \mathcal{E}_{p,q}(1,1) \leq \mathcal{E}_{p,q}(0,1) + \mathcal{E}_{p,q}(1,0) \; . \qquad (9)$$

Clearly, $\mathcal{E}$ will be regular for any α-expansion if the term V satisfies $V(\beta,\gamma) + V(\alpha,\alpha) \leq V(\beta,\alpha) + V(\alpha,\gamma)$ for all labels α, β, γ. We refer to such a V as *expansion-regular*.

We now extend the class of energies for which we are guaranteed to converge to a strong local minimum. Specifically, we allow a rather general hard constraint term H, which does *not* need to be expansion-regular.

Theorem 1 *Suppose that each term $V_{p,q}$ in eqn. (1) is expansion-regular and each term $H_{p,q}(\cdot,\cdot) \in \{0, \infty\}$ has zero diagonal: $H_{p,q}(\alpha,\alpha) = 0$ for any label α. Then for any α-expansion, function $\mathcal{E}$ in eqn. (8) will be regular assuming that the initial configuration satisfies the hard constraints, i.e. $E(f^0) < \infty$.*

The proof is given in Appendix A. Note that [9] uses a hard (visibility) constraint in their formulation of the stereo problem, which is a particular example of the class defined in the theorem.

3.3 Modifying Expansion Move Algorithm for General Energies

Unfortunately, in some applications such as [11, 2] and ours not all terms $V_{p,q}$ are expansion-regular. However, the number of such terms is relatively small, therefore the expansion move framework still seems desirable. This framework is used in [11, 2] with excellent results; however the authors do not discuss what they do when a term is not expansion-regular. We now show how to modify the algorithm to treat such terms correctly.

Suppose that during the α-expansion step we obtain a function $\mathcal{E}$ where some of the terms $\mathcal{E}_{p,q}$ are not regular. Similar to [10] we could construct a graph for such a term. However, it would contain edges with negative weights, so a maxflow algorithm cannot be applied.

We propose to "truncate" non-regular terms $\mathcal{E}_{p,q}$, i.e. replace them with regular terms $\widehat{\mathcal{E}}_{p,q}$ (as defined below), and then minimize the new function $\widehat{\mathcal{E}}$. This is justified by the following theorem whose proof is given in Appendix A.

Theorem 2 *Suppose that functions $\widehat{\mathcal{E}}, \mathcal{E}$ (eqn. (8)) satisfy the following conditions: Unary terms $\widehat{\mathcal{E}}_p$ and $\mathcal{E}_p$ are the same, and for any $p, q \in \mathcal{N}$ we have $\widehat{\mathcal{E}}_{p,q}(0,0) \leq \mathcal{E}_{p,q}(0,0)$ and $\widehat{\mathcal{E}}_{p,q}(x_p, x_q) \geq \mathcal{E}_{p,q}(x_p, x_q)$ for $(x_p, x_q) \neq (0,0)$. If $\mathbf{x}^*$ minimizes function $\widehat{\mathcal{E}}$ then $\mathcal{E}(\mathbf{x}^*) \leq \mathcal{E}(\mathbf{0})$.*

Note that $\mathcal{E}(\mathbf{0})$ is the energy of the current configuration, and $\mathcal{E}(\mathbf{x}^*)$ is the energy of the proposed α-expansion move. Therefore, the theorem states that the energy does not increase. It means that the expansion move algorithm with truncation is a valid energy minimization technique for arbitrary functions, which is guaranteed to converge. However, each step is no longer guaranteed to find an *optimal* α-expansion move, and the output does not necessarily has the property of a strong local minimum.

Although the algorithm with truncation can be applied to any energy function, it is likely to give good results only for some applications. Intuitively, the method seems suitable in situations when most of the terms V are expansion-regular.

Let us summarize the truncation procedure. If term $\mathcal{E}_{p,q}$ does not satisfy inequality (9) then we perform one of the following three operations: we either decrease $\mathcal{E}_{p,q}(0,0)$ or increase $\mathcal{E}_{p,q}(0,1)$ or $\mathcal{E}_{p,q}(1,0)$ until we obtain an equality. Note that the fourth term should not be modified: to make $\mathcal{E}_{p,q}$ regular we would need to decrease $\mathcal{E}_{p,q}(1,1)$, but then it may happen that $\mathcal{E}(\mathbf{x}^*) > \mathcal{E}(\mathbf{0})$.

Note that a different truncation procedure for *semi-metric* terms was given in [3]. They propose to replace non-metric terms V with *Potts* terms. This approach has certain approximation bound guarantees[5]. However, very little information about the structure of V is used.

3.4 Applying Expansion Move Algorithm to Our Energy Function

It can be seen that the block uniquness constraint (sec. 2.1) belongs to the class of hard constraints defined in theorem 1. Moreover, it can be implemented very efficiently despite the fact that the neighborhood system is

[5] In fact, for functions with semi-metric terms without hard constraints we proved that our technique yields the same approximation bound as given in [3], assuming that truncation does not change $\mathcal{E}_{p,q}(0,0)$.

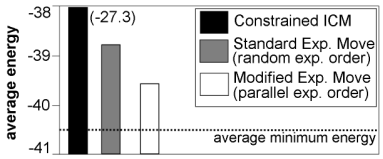


Figure 4. Comparison of optimization techniques. Average performance (50 runs) over 8 photo collections with each about 40 images. The energy and initialisation is the same for all algorithms. As to be expected, the expansion move algorithm outperforms ICM. For our application, the standard expansion move algorithm (with random expansion order) performed worse than our modified version with parallel expansion order (sec. 4).

the complete graph. Indeed, for each α-expansion it yields term $\mathcal{E}_{p,q}$ for blocks $p, q \in \mathcal{P}$ only if $b(p, f_p) = b(q, \alpha)$ or $b(p, \alpha) = b(q, f_q)$. Is is not difficult to see that the number of such terms is at most $|\mathcal{P}|$. Therefore, the number of edges that we need to add to the graph constructed for minimizing $\mathcal{E}$ is at most linear in the number of tapestry blocks.

In sec. 3 we assumed that all variables f_p have the same range $\mathcal{L}$. It is straightforward to extend the framework to handle image-shift and cluster variables f_i and f_c (sec. 2.3) whose ranges are $\mathcal{S}$ and $\mathcal{I}$, respectively. For these variables we need to define the meaning of an α-expansion move. Consider, for example, the image-shift variable. If $\alpha = (i, s)$ then we set

$$f[\mathbf{x}]_i = \begin{cases} f_i & \text{if } x_i = 0 \\ s & \text{if } x_i = 1 \end{cases} ,$$

variables f_j for $j \in \mathcal{I} - \{i\}$ do not change during this expansion. It can be seen that theorem 1 still holds assuming that terms $H_{p,i}$ satisfy $H_{p,i}((i, s), s) = 0$ for all $\alpha = (i, s)$.

4 Experiments

We have tested digital tapestry on 8 different photo collections which contained on average 40 images. Various results are shown in fig. 1, 3, 5 and 6. They were achieved with a standard set of parameters (except where explicitly mentioned). In this paper, the tapestry size is the same as the image size, due to limited space. The supplementary material shows larger tapestries which contain up to 32 (out of 40) input images. We initialise the expansion move algorithm with the single image that is most salient (or a mosaic of images for larger tapestries). This initialisation satisfies all hard constraints and has in general a lower energy than a random collection of the most salient blocks. In fig. 5 we illustrate how the coherence strength influencies the tapestry.

A performance evaluation of different optimisation techniques is shown in fig. 4. We have encountered that for our applications the final result heavily depends on the order of expansion moves (see fig. 5(b) and fig. 1(d)). The standard expansion move algorithm [3] performs the move in no particular order, e.g. random. We introduce an expansion order scheme, denoted as "parallel expansion", which lead to an improved performance. Assume we can afford a fixed number of expansion moves, larger than the number of all labels. The rough block layout is often decided in the first expansion moves. Therefore, we run K (here $K = 5$) parallel processes (for a subset of labels in a random order) and determine after R iterations which process gives the lowest energy. This process is then run again over all labels. R is choosen so that the maximum number of moves is not exceeded.

5 Conclusions and Future Work

The main contribution of this paper is a framework for creating a visual summary - *a tapestry* - fully automatically from a large number of different input images. The tapestry can be used to remind the user of the photo collection. To verify this point, we plan to conduct a user study where tapestries are compared to mosaics of representative images. We also plan to learn the saliency measurement from a psychological experiment.

The creation of the tapestry is phrased as a multi-label energy minimization problem, and solved using expansion move algorithm. The energy contains non-metric (soft and hard) constraints, which can not be handled by the standard expansion move algorithm. Therefore, we extended the algorithm to incorporate these constraints, which is a further contribution. In future work, we plan to improve the quality of the tapestry by exploiting further high-level knowledge, such as reliable un-supervised texture clustering, face clustering, and automatic image scale detection.

A Proofs of Theorems

Theorem 1. Since $E(f^0) < \infty$ it is $H(f_p, f_q) = 0$ for all pixels p, q. The energy $\mathcal{E}$ for an α-expansion move is regular since $V_{p,q}$ is expansion-regular and $H(\alpha, f_q) + H(f_p, \alpha) \geq H(f_p, f_q) + H(\alpha, \alpha) = 0$ is valid for all p, q. After α-expansion the energy of new labeling f^1 is still finite since $E(f^1) \leq E(f^0) < \infty$. Therefore, we can apply the same argument (i.e. use induction).

Theorem 2. Without loss of generality we can assume that unary terms $\mathcal{E}_p$ are not present since they can be viewed as pairwise terms. Let us denote $\mathcal{N}_0 = \{p, q \in \mathcal{N} \mid (x_p^*, x_q^*) = (0, 0)\}$, $\mathcal{N}_1 = \mathcal{N} \backslash \mathcal{N}_0$ and $C = \sum_{p,q \in \mathcal{N}_0} \mathcal{E}_{p,q}(0, 0)$. We can show that

$$\begin{aligned} \mathcal{E}(\mathbf{x}^*) - C &= \sum_{p,q \in \mathcal{N}_1} \mathcal{E}_{p,q}(x_p^*, x_q^*) \leq \sum_{p,q \in \mathcal{N}_1} \widehat{\mathcal{E}}_{p,q}(x_p^*, x_q^*) \\ &= \widehat{\mathcal{E}}(\mathbf{x}^*) - \sum_{p,q \in \mathcal{N}_0} \widehat{\mathcal{E}}_{p,q}(0, 0) \leq \widehat{\mathcal{E}}(\mathbf{0}) - \sum_{p,q \in \mathcal{N}_0} \widehat{\mathcal{E}}_{p,q}(0, 0) \\ &= \sum_{p,q \in \mathcal{N}_1} \widehat{\mathcal{E}}_{p,q}(0, 0) \leq \sum_{p,q \in \mathcal{N}_1} \mathcal{E}_{p,q}(0, 0) . \end{aligned}$$

(a) Weak coherence (b) Standard coherence (c) Strong coherence (d) Very strong coherence

Figure 5. Setting parameters. Four alternative tapestries for the input data set of fig. 1(top). The strength of the block coherence constraint does steer the number of different labels (number of represented image parts) which appear in the tapestry (22 labels in (a), 9 in (b), 3 in (c), and 1 in (d)). (a) A tapestry with weak coherence looks "overloaded". (d) If the coherence of the MRF is very strong the expansion move algorithm returns the initial configuration - the most salient image. Note, two random variations of a tapestry, same parameters, in (b) and fig. 1(d) have nearly the same energies ($E = -40.32$ and -40.41). It indicates that the energy function might contain many different local minima.

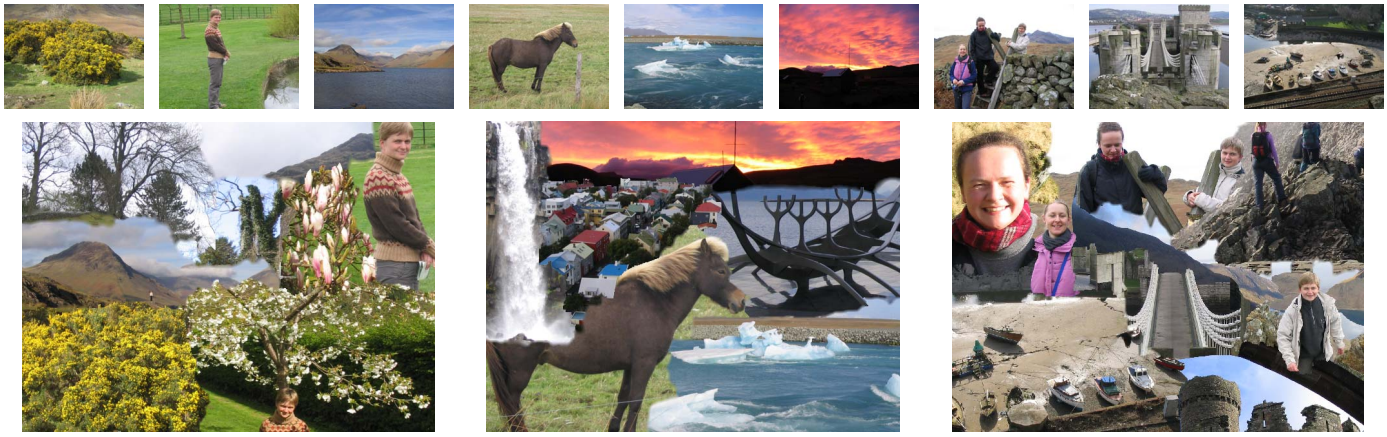


Figure 6. Further results. Tapestry for three different consumer photo collections (top: sample input images, bottom: tapestry). In the last tapestry it is apparent that automatic face clustering is an important issue.

Therefore, $\mathcal{E}(\mathbf{x}^*) \leq C + \sum_{p,q \in \mathcal{N}_1} \mathcal{E}_{p,q}(0,0) = \mathcal{E}(\mathbf{0})$.

References

[1] Adobe Systems Incorp. 2003. Adobe Photoshop User Guide.

[2] Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., Cohen, M. 2004. Interactive Digital Photomontage. ACM Transactions on Graphics vol. 23(3), SIGGRAPH 2004, p. 294-295.

[3] Boykov, Y., Veksler, O., Zabih, R. 2001. Fast Approximate Energy Minimization via Graph Cuts. PAMI, Vol. 23(11).

[4] Carson, C., Belongie, S., Greenspan, H., and Malik, J. 2002. Blobworld: Image segmentation using Expectation-Maximization and its application to image querying. PAMI Vol. 24(8): 1026-1038.

[5] Efros, A. and Freeman, W. 2001. Image Quilting for Texture Synthesis and Transfer. ACM Transactions on Graphics vol. 20(3), SIGGRAPH 2001, p. 341-346.

[6] Fitzgibbon, A. W. and Zisserman, A. 2002. On Affine Invariant Clustering and Automatic Cast Listing in Movies. ECCV, Vol. 3, Copenhagen, Denmark, pp. 304-320.

[7] Jojic, N., Frey, B., Kannan, A. 2003. Epitomic Analysis of appearance and shape. ICCV, Nice, France, pp. 34-41.

[8] Kolmogorov, V. and Zabih, R. 2001. Computing Visual Correspondence with Occlusions using Graph Cuts. ICCV, Vancouver, Canada, pp. 508-515.

[9] Kolmogorov, V. and Zabih, R. 2002. Multi-camera Scene Reconstruction via Graph Cuts. ECCV, Vol. 3, pp. 82-96.

[10] Kolmogorov, V. and Zabih, R. 2004. What Energy Functions can be Minimized via Graph Cuts? PAMI, Vol 26(2).

[11] Kwatra, V., Schödl, A., Irfan, E., Turk, G. and Bobick, A. 2003. Graphcut Textures: Image and Video Synthesis Using Graph Cuts. ACM Transactions on Graphics vol. 22(3), SIGGRAPH 2003, p. 277-286.

[12] Pérez, P., Gangnet, M. and Blake, A. 2003. Poisson image editing. ACM Transactions on Graphics vol. 22(3), SIGGRAPH 2003, p. 313-318.

[13] Swain, M.J. and Ballard, D.H. 1991. Color Indexing IJCV, vol. 7(11):11-32.

[14] Uyttendaele, M., Eden, A. and Szeliski, R. 2001. Eliminating ghosting and exposure artifacts in image mosaics. CVPR, vol. 2, 509-519.

[15] Viola, P. and Jones, M. 2001. Rapid object detection using a boosted cascade of simple features. CVPR, Vol 1.

[16] S. C. Zhu, Y. N. Wu and D.B. Mumford, 1998. FRAME: Filters, Random field And Maximum Entropy: Towards a Unified Theory for Texture Modeling. IJCV, 27(2) pp.1-20.