

```

entry:
  %0 = bitcast %struct.djpeg_dest_struct* %dinfo to %struct.bmp_dest_struct*
  %mem = getelementptr inbounds %struct.jpeg_decompress_struct,
  ... %struct.jpeg_decompress_struct* %cinfo, i64 0, i32 1
  %1 = load %struct.jpeg_memory_mgr*, %struct.jpeg_memory_mgr** %mem, align 8,
  ... !tbaa !2
  %access_virt_sarray = getelementptr inbounds %struct.jpeg_memory_mgr,
  ... %struct.jpeg_memory_mgr* %1, i64 0, i32 7
  %2 = load i8** (%struct.jpeg_common_struct*, %struct.jvirt_sarray_control*,
  ... i32, i32, i32)*, i8** (%struct.jpeg_common_struct*,
  ... %struct.jvirt_sarray_control*, i32, i32, i32)** %access_virt_sarray, align 8,
  ... !tbaa !10
  %3 = bitcast %struct.jpeg_decompress_struct* %cinfo to
  ... %struct.jpeg_common_struct*
  %whole_image = getelementptr inbounds %struct.djpeg_dest_struct,
  ... %struct.djpeg_dest_struct* %dinfo, i64 1, i32 1
  %4 = bitcast void (%struct.jpeg_decompress_struct*,
  ... %struct.djpeg_dest_struct*, i32)** %whole_image to
  ... %struct.jvirt_sarray_control**
  %5 = load %struct.jvirt_sarray_control*, %struct.jvirt_sarray_control** %4,
  ... align 8, !tbaa !13
  %cur_output_row = getelementptr inbounds %struct.bmp_dest_struct,
  ... %struct.bmp_dest_struct* %0, i64 0, i32 6
  %6 = load i32, i32* %cur_output_row, align 4, !tbaa !16
  %call = tail call i8** %2(%struct.jpeg_common_struct* %3,
  ... %struct.jvirt_sarray_control* %5, i32 %6, i32 1, i32 1) #6
  %7 = load i32, i32* %cur_output_row, align 4, !tbaa !16
  %inc = add i32 %7, 1
  store i32 %inc, i32* %cur_output_row, align 4, !tbaa !16
  %8 = load i8*, i8** %call, align 8, !tbaa !17
  %output_width = getelementptr inbounds %struct.jpeg_decompress_struct,
  ... %struct.jpeg_decompress_struct* %cinfo, i64 0, i32 26
  %9 = load i32, i32* %output_width, align 8, !tbaa !18
  %cmp33 = icmp eq i32 %9, 0
  br i1 %cmp33, label %for.end, label %for.body.preheader

```

```

for.body.preheader:
  %buffer = getelementptr inbounds %struct.djpeg_dest_struct,
  ... %struct.djpeg_dest_struct* %dinfo, i64 0, i32 4
  %10 = load i8**, i8*** %buffer, align 8, !tbaa !19
  %11 = load i8*, i8** %10, align 8, !tbaa !17
  %12 = add i32 %9, -1
  %13 = zext i32 %12 to i64
  %14 = mul nuw nsw i64 %13, 3
  %15 = add nuw nsw i64 %14, 3
  %xtraiter = and i32 %9, 1
  %lcmp.mod = icmp eq i32 %xtraiter, 0
  br i1 %lcmp.mod, label %for.body.preheader.split, label %for.body.prol

```

```

for.body.prol:
  %incdec.ptr.prol = getelementptr inbounds i8, i8* %11, i64 1
  %16 = load i8, i8* %11, align 1, !tbaa !20
  %arrayidx3.prol = getelementptr inbounds i8, i8* %8, i64 2
  store i8 %16, i8* %arrayidx3.prol, align 1, !tbaa !20
  %incdec.ptr4.prol = getelementptr inbounds i8, i8* %11, i64 2
  %17 = load i8, i8* %incdec.ptr.prol, align 1, !tbaa !20
  %arrayidx5.prol = getelementptr inbounds i8, i8* %8, i64 1
  store i8 %17, i8* %arrayidx5.prol, align 1, !tbaa !20
  %incdec.ptr6.prol = getelementptr inbounds i8, i8* %11, i64 3
  %18 = load i8, i8* %incdec.ptr4.prol, align 1, !tbaa !20
  store i8 %18, i8* %8, align 1, !tbaa !20
  %add.ptr.prol = getelementptr inbounds i8, i8* %8, i64 3
  %dec.prol = add i32 %9, -1
  br label %for.body.preheader.split

```

```

for.body.preheader.split:
  %col.036.unr = phi i32 [ %9, %for.body.preheader ], [ %dec.prol,
  ... %for.body.prol ]
  %outptr.035.unr = phi i8* [ %8, %for.body.preheader ], [ %add.ptr.prol,
  ... %for.body.prol ]
  %inptr.034.unr = phi i8* [ %11, %for.body.preheader ], [ %incdec.ptr6.prol,
  ... %for.body.prol ]
  %19 = icmp eq i32 %12, 0
  br i1 %19, label %for.end.loopexit, label %for.body.preheader.split.split

```

```

for.body.preheader.split.split:
  br label %for.body

```

```

for.body:
  %col.036 = phi i32 [ %col.036.unr, %for.body.preheader.split.split ], [
  ... %dec.1, %for.body ]
  %outptr.035 = phi i8* [ %outptr.035.unr, %for.body.preheader.split.split ],
  ... [ %add.ptr.1, %for.body ]
  %inptr.034 = phi i8* [ %inptr.034.unr, %for.body.preheader.split.split ], [
  ... %incdec.ptr6.1, %for.body ]
  %incdec.ptr = getelementptr inbounds i8, i8* %inptr.034, i64 1
  %20 = load i8, i8* %inptr.034, align 1, !tbaa !20
  %arrayidx3 = getelementptr inbounds i8, i8* %outptr.035, i64 2
  store i8 %20, i8* %arrayidx3, align 1, !tbaa !20
  %incdec.ptr4 = getelementptr inbounds i8, i8* %inptr.034, i64 2
  %21 = load i8, i8* %incdec.ptr, align 1, !tbaa !20
  %arrayidx5 = getelementptr inbounds i8, i8* %outptr.035, i64 1
  store i8 %21, i8* %arrayidx5, align 1, !tbaa !20
  %incdec.ptr6 = getelementptr inbounds i8, i8* %inptr.034, i64 3
  %22 = load i8, i8* %incdec.ptr4, align 1, !tbaa !20
  store i8 %22, i8* %outptr.035, align 1, !tbaa !20
  %add.ptr = getelementptr inbounds i8, i8* %outptr.035, i64 3
  %incdec.ptr.1 = getelementptr inbounds i8, i8* %inptr.034, i64 4
  %23 = load i8, i8* %incdec.ptr6, align 1, !tbaa !20
  %arrayidx3.1 = getelementptr inbounds i8, i8* %outptr.035, i64 5
  store i8 %23, i8* %arrayidx3.1, align 1, !tbaa !20
  %incdec.ptr4.1 = getelementptr inbounds i8, i8* %inptr.034, i64 5
  %24 = load i8, i8* %incdec.ptr.1, align 1, !tbaa !20
  %arrayidx5.1 = getelementptr inbounds i8, i8* %outptr.035, i64 4
  store i8 %24, i8* %arrayidx5.1, align 1, !tbaa !20
  %incdec.ptr6.1 = getelementptr inbounds i8, i8* %inptr.034, i64 6
  %25 = load i8, i8* %incdec.ptr4.1, align 1, !tbaa !20
  store i8 %25, i8* %add.ptr, align 1, !tbaa !20
  %add.ptr.1 = getelementptr inbounds i8, i8* %outptr.035, i64 6
  %dec.1 = add i32 %col.036, -2
  %cmp.1 = icmp eq i32 %dec.1, 0
  br i1 %cmp.1, label %for.end.loopexit.unr-icssa, label %for.body

```

```

for.end.loopexit.unr-icssa:
  br label %for.end.loopexit

```

```

for.end.loopexit:
  %scevgep = getelementptr i8, i8* %8, i64 %15
  br label %for.end

```

```

for.end:
  %outptr.0.lcssa = phi i8* [ %8, %entry ], [ %scevgep, %for.end.loopexit ]
  %pad_bytes = getelementptr inbounds %struct.djpeg_dest_struct,
  ... %struct.djpeg_dest_struct* %dinfo, i64 1, i32 3
  %26 = bitcast %struct._IO_FILE** %pad_bytes to i32*
  %27 = load i32, i32* %26, align 8, !tbaa !21
  %cmp930 = icmp sgt i32 %27, 0
  br i1 %cmp930, label %while.body.preheader, label %while.end

```

```

while.body.preheader:
  %28 = add i32 %27, -1
  %29 = zext i32 %28 to i64
  %30 = add nuw nsw i64 %29, 1
  call void @llvm.memset.p0i8.i64(i8* %outptr.0.lcssa, i8 0, i64 %30, i32 1,
  ... i1 false)
  br label %while.end

```

```

while.end:
  ret void

```

CFG for 'put_pixel_rows' function